SIEMENS

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Manual

Edition November 2002

Documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the "Remarks" column.

Status code in the "Remarks" column:

A .... New documentation.
B .... Unrevised reprint with new Order No.
C .... Revised edition with new status.

Edition	Order No.	Remarks
04.98	6AT1880-0AA00-1BA0	**A**
10.99	6AT1880-0AA00-1BC0	**C**
11.2002	6AT1880-0AA00-1BE0	**C**

SIMOVERT MASTERDRIVES MC, SIMATIC and SINUMERIK are registered trademarks of Siemens AG. Other names in this publication might be trademarks whose use by a third party for his own purposes may violate the rights of the registered holder.

We have checked that the contents of this document correspond to the hardware and software described. Nonetheless, differences might exist and therefore we cannot guarantee that they are completely identical. The information contained in this document is, however, reviewed regularly and any necessary changes will be included in the next edition. We welcome suggestions for improvement.

Subject to change without prior notice

Order No.: 6AT1880-0AA00-1BE0
Printed in the Federal Republic of Germany

Siemens-Aktiengesellschaft

SIEMENS

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Preface

Edition 11.2002

1 Overview of Motion Control

Contents

In this chapter you will find an overview of Motion Control.

1.1 Why Motion Control?

For All Trades and All Technologies

Whether you manufacture packaging machines or textile machines, printing machines, woodworking machines or presses, whether your application primarily involves positioning, coupled motion or user-specific technology functions - Motion Control gives you more performance through the perfect synchronization of all movement sequences.

Intelligent Software Instead of a Centralized Drive

The line shafts, cam disks, operating cams, draw-bars and couplings used to implement the large variety of functions on conventional centralized drives are no longer compatible with today's need for rapid retooling. These purely mechanical components are quick to reach their limits when increased cycle times, improved accuracy or flexible retooling is required. With intelligent software, however, these requirements are no longer a problem.

New Flexibility

Accurate coordination is required in applications where a large number of individual drives performs a large number of movements to a high degree of precision. This is where Motion Control comes into its own. You can group several drives together to implement technological functions for maximum flexibility in your machine configuration. Whether you use a distributed or centralized configuration is entirely up to you, because Motion Control provides a complete range of functions for the modular SIMATIC M7 computers and the highly dynamic and compact MASTERDRIVES MC servo converters.

An Integrated Solution

Whatever your drive concept: the new software solves all your Motion Control tasks on an integrated automation platform. PROFIBUS-DP ensures integrated communication for configuring, parameter assignment, programming, and operator control and monitoring. And the handling of centralized and distributed topologies is identical. Whether you store the Motion Control functions on the computer or integrate them in the converter, the interfaces and engineering tools are the same. Incidentally, they are the same tools you use for your SIMATIC programmable logic controllers (PLCs) – uniting the worlds of PLCs and drives.

Easy to Program

Motion Control unites the world of drives with the world of SIMATIC. This gives you access to user-friendly engineering tools that require absolutely no programming knowledge. Positioning and synchronization functions are standardized with Motion Control.

All the Advantages of Digital Technology

Motion Control gives you all the advantages of innovative technology: installation, troubleshooting and diagnostics are child's play, and you can retool your equipment easily at the press of a button without the need for mechanical modifications. The obvious result is that you always have your costs under control.

1.2 Centralized or Distributed - The Choice is Yours

Centralized Solution on the M7 Computer with SIMATIC Motion Control

If you need large volumes of sequence data (NC programs, tables, etc) or non-axis-specific NC programs, the centralized solution with SIMATIC Motion Control is the right choice for you. This way, you integrate all Motion Control functions simply in an M7 computer.

The converter then handles conventional tasks such as current, speed and position control.

SIMOLINK or Profibus-DP for Motion Control is then used to exchange all the relevant data between the computer and the drive.

SIMATIC Motion Control includes all of the important functions for a motion control solution.

Distributed Solution on the Actual MASTERDRIVES MC

You choose the kind of flexibility you need. If, for example, you attach great importance to modular machines and equally modular control and automation solutions, then a distributed solution is the obvious choice.

It's all the same whether you need individual independent drives or want to achieve perfect synchronization between groups of drives. You simply use the integrated Motion Control functionality of the MASTERDRIVES MC drives.

Motion Control "resides" on the actual MASTERDRIVES MC drive. This totally new approach offloads your control system considerably, and you don't even need any extra hardware.

Logic and arithmetic blocks are available here in addition to the technological functions.

1.3 The Highest Performance and User-Friendly Handling of Network, Links and Tools

The Backbone of Your Automation Solution: PROFIBUS-DP

The communication platform for Motion Control is PROFIBUS-DP. A transmission rate of 12 Mbit/s ensures perfect communication between all components. And the shared communication system used for configuring, parameter assignment, programming, diagnostics and operation ensures smooth data exchange on the bus network.

Perfect Drive Synchronization: SIMOLINK

SIMOLINK is an ideal digital link for drives in applications requiring time-critical signal transmission, e.g. for absolute synchronization of multiple drives or cycle-synchronous transfer of setpoints and actual values. This link is implemented as a fiber-optic ring network and synchronizes up to 200 drives. The ring transfers a 32-bit value in only 6.3 µs. SIMOLINK makes perfect synchronization easy. Motion Control can be connected via SIMOLINK to electrically operated shafts and gearboxes which operate at lightning speed with absolutely no fluctuation.

Easy-to-Use High-Performance Tools

It is remarkably easy to engineer and operate this type of Motion Control system or automation solution. This is because the SIMATIC software requires no special knowledge. The same user interface is available for visualization and diagnostics on the PLC, computer and MASTERDRIVES MC drives. This user interface is graphical and is called Windows®.

All that you need is a SIMATIC programming device with integrated STEP 7 software plus GMC-CONTROL (this tool can be found on the GMC Basic CD). This allows you to configure, start up and diagnose your system. You also have the option of starting up a distributed solution with SIMOVIS/DriveMonitor running directly under Windows.

It has never been possible to automate complex machines this easily or rapidly.

1.4 Powerful Standards: Our Standard Product Base

Concept

The Motion Control concept from Siemens offers you a complete open and closed-loop control and drive solution for your machine, based on a standardized system platform. And, because the name of this platform is SIMATIC, all the components are equally easy to handle, use the same software tools, have unified communication interfaces and user interfaces, and are frequently even based on the same hardware modules.

This added value provides a major payback in terms of significantly lower engineering costs, reduced inventory costs and the ability to respond more rapidly to customer needs.

Perfect Integration of Computing and Control: SIMATIC M7

A SIMATIC M7-FM is a high-performance computer with an Intel processor. It handles all Motion Control tasks as an integral component of a SIMATIC S7 PLC.

Intelligent and Compact Drive Technology: MASTERDRIVES MC

They are small, highly intelligent and compact; they are supplied complete with the full Motion Control functionality; and they set a completely new standard in servo drives. The modular drive system contains DC and AC modules. It fits perfectly into any drive and installation concept: The smallest converter measures only 4.5 x 26 x 36 cm and is thus easy to install in the direct vicinity of the machine. MASTERDRIVES MC is suitable for all types of servo motor (including asynchronous motors), all encoder systems, all applications and all industries. 300 % overload gives you a comfortable safety cushion, and digital technology provides the highest level of performance and precision.

Operator Control and Monitoring on a Unified System Platform: SIMATIC HMI

You also benefit from the powerful standards used in SIMATIC operator panels, text displays and visualization systems. These units feature high legibility, have the same simple user interface as all the other components, and are configured with the same tools. Because the database and communication systems are totally integrated, you never have to enter the same data twice when configuring your human-machine interface.

2 How to Use the Manual

Target Group

This manual is intended for:

- Project design engineers for creating the plant configuration
- Installation engineers for plant installation and commissioning
- Service engineers for on-site troubleshooting

Knowledge Required

Particular reference is made if special knowledge is required for any part of this manual. The general safety guidelines, VDE regulations and national regulations apply in all other cases, in addition to the information provided in this manual.

Orientation Aids

The indexes within the individual tabs and the general index in the Appendix are provided as orientation aids. A topic overview is provided at the beginning of each tab. Each main chapter also has a table of contents.

References within the Manual

References to pages or chapters within the manual, e.g. in the general index, are organized as follows:

- 2/3.1 → Tab 2 Chapter 3.1
- 3/5-5 → Tab 3 Page 5-5
- 4-10 → Page 4-10 within the same tab

Structure of the Manual

The manual is organized into different parts. The following parts may be relevant for your purposes, depending on the basis on which you use the technology, i.e. whether you use a centralized or distributed solution:

♦ Part 1: Technology Functions
   - Function Description – Positioning and Synchronization
   - Programming Guide – Writing NC Programs

  This part of the manual is always relevant for your purposes and describes the functions of the technology. The Programming Guide provides you with all the information you need to write an NC program for automatic mode. Part 1 of the manual is also intended as a supplement to the MASTERDRIVES MC Compendium.

♦ Part 2: SIMATIC S7 Communication Functions
   - GMC-BASIC Standard Software
   - Task Description

  Whatever the approach of your solution, the application interface is always implemented centrally in the SIMATIC S7-CPU. This interface is described in the "GMC-BASIC Standard Software" tab. Tasks are used to activate the different functions. These are described centrally in the "Task Description" tab.

♦ Part 3: User Interfaces

  The Motion Control HMI package is not contained on the enclosed "Motion Control Configuring Package" CD and must be ordered separately. However, a description of the HMI package "GMC-OP-OAM Standard Software" is contained in the manual.

♦ Part 4: Appendices

  In the Appendix you will find a description of all the error messages including their cause, effect, occurrence and remedy. An explanation of various terms can be found in the glossary. The general index helps you to find information quickly, e.g. on specific functions or machine data. The appendices also include a list of abbreviations.

Changes Made Since the Last Version

This manual has been changed at the following points as compared to version 10.1999:

- ♦ Notes have been added in the description of various functions, clarifying which software versions are needed to implement the functions.
- ♦ M7-FM designation replaced by SIMATIC Motion Control

Chapter	Change
Preface: Definitions and safety notices	
SIMOVIS has been replaced generally by SIMOVIS/DriveMonitor	
Description of functions – positioning and synchronized operation	
1.2	Machine data list
	New parameter (no. 6)
	Modified upper input limit MD56 New meaning for MD57 New machine data MD66 and MD67
1.3	Table of function parameters modified
1.6	Linear/rotary axis: Changes to notes
1.9	Acceleration and deceleration: New Important note
	Description of MD22
1.13	Modified heading, introduction and description of acceleration breakpoint
1.19	Description of "Encoder monitoring message" function
1.21	Changes to MD56, MD57, MD59
	Description of MD66 and MD67
1.22	Changes to individual points
2.1	New control signal S_DSP
2.2	Modified designation of control signals for MASTERDRIVES MC
2.6	Supplemented control signal S_DSP
2.9	Changed meaning for technology operation on SIMATIC Motion Control
3.1.1	Modified default setting for velocity stage 2
3.3	Completely new description
10	Chapter completely revised
12.4	Chapter completely revised
13	New chapter
Programming Guide – creation of traversing programs	
5.2	Chapter completely revised

GMC-BASIC standard software	
3.2.3.1	Supplements to SIMOLINK module
5.2	Important note supplemented
7.2	Modified evaluation of diagnostic alarm in the S7-CPU
8.8	Modified tables
8.10	Modified values
Task description	
2.5	New task
5.9	New task
6.1	Additional identifiers 19 to 37
9.9	Modified task
10	Chapter completely revised
11.1	Supplementary axis enabling information
13	List revised
GMC-OP-OAM standard software	
6.2.3	New sections 6.2.3 to 6.2.5
6.4	Modified tables
8	Chapter completely revised
Appendix A: Error messages	
The following error messages have been modified or added: 137; 139; 149; 157 to 159; 207; 209; 230; 244; 245 to 249; 2032; 2040; 2110; 2125; 2126; 2137; 2158; 2159; 2233; 2234 to 2236; 2250; 4444; 4207	

3 Function List

Technology Functions in Conjunction with Firmware Versions

The table below lists newly added functions (German edition: 12.2000) with the relevant firmware version (MC column for MASTERDRIVES MC or M7 column for SIMATIC MOTION CONTROL) and the version in which the function becomes available. For the solution involving MASTERDRIVES Motion Control, the table also specifies whether the function needs to be enabled via F01 (F01 column) and/or whether it is supported via the task interface (ASS column). However, the list is not guaranteed to be complete.

Function		**Firmware version**		
	ASS	MC	F01	**M7**
Machine data **/ Function parameters**				
MD1 to MD50	X	V1.2	X	V1.0
MD51 to MD70 (MD for IM178)	X	-	-	V1.0
MD8: Extended homing function	X	V1.4	X	-
MD22: Rounding time constant	X	-	-	V1.00.48
MD24: Extension M fct. act.-val.-dependent	X	V1.32	X	-
MD45: Extension reversal cam for homing	X	V1.4	X	-
MD48: Variant 6, encoder monitoring signal	X	V1.51	X	-
Function parameters: FP1 to FP10	X	V1.4	X	V1.00.48
FP6: Extension limit-value monitoring encoder switchover	X	V1.41	X	-
Tasks				
Introduction function parameters input / output	X	V1.4	X	V1.00.48
Extension software version with generation date output	X	V1.4	X	V1.0
Actual-value output: ID 18: No. of free interpolation points	X	V1.4	X	-
Actual-value output: IDs 19, 20 and 21: Actual values for encoder switchover	X	V1.41	X	-
Actual-value output: IDs 30 to 37: Actual values to IM178	X	-	-	V1.0
Extension to max. 8 traversing tables with max. 400 interpolation points M7: 8 tables with 1022 interpolation points each	X	V1.4	X	V1.0
Extended traversing table mode Change in slave axis scaling (E)	X	V1.4	X	V1.0
Extended traversing table mode Table change absolute/relative (F)	X	V1.5	X	V1.00.48

Function		Firmware version		
	ASS	MC	F01	**M7**
Introduction of window traversing table Actual-value control	X	V1.51	X	V1.00.48
Extended position correction: Max. correction velocity	X	V1.4	X	V1.00.48
Extended position correction: Acceleration	X	V1.5	X	V1.00.48
Extension to actual synchronous operation values: Position correction differential value Current offset, residual offset distance and actual offset value Slave axis status	X	V1.3 V1.5	X	V1.00.48
Introduction of master value correction	X	V1.4	X	-
Introduction of master value correction V branch	X	V1.5	X	-
Introduction of real master	X	V1.4	X	-
Introduction of synchronous operation parameters, offset angle setting	X	V1.3	X	V1.00.48
Introduction of catch-up	X	V1.31	X	V1.00.48
Catch-up: Extended synchronous operation parameters, ramp-down deceleration, ramp-up acceleration, rounding mode	X	V1.4	X	V1.00.48
Extended master value synchronization: Correction mode	X	V1.5	X	V1.00.48
Introduction of roll feed with encoder switchover	X	V1.41	X	-
Write data to EEPROM or RAM input/output	X	V1.4	X	V1.0
Axis machining, enabling	X	-	-	V1.0
Extended axis machining, enabling catch-up, offset angle setting	X	-	-	V1.00.48
Cam controller	X	-	-	V1.0
MD51 to MD70 (MD for IM178)	X	-	-	V1.0
Extended I/Os	X	-	-	V1.0

Function	ASS	Firmware version MC	F01	M7
Control and checkback signals				
Control signals ADD_CTRL_SYNC_MODE		V1.31	X	V1.00.48
Checkback signals ADD_STAT_SYNC_MODE		V1.31	X	V1.00.48
CU_TR added to control signals (accept trigger stop position)		V1.4	X	V1.00.48
EN_RF added to control signals (on-the-fly setting of home position)		V1.4	X	-
Mode synchronous operation: Catch-up added to Select Operation control signals		V1.4	X	V1.00.48
S_DSP added to control signals (set offset setting value)		V1.5	X	V1.00.48
Positioning functions				
On-the-fly setting of home position in all modes	X	V1.4	X	-
Home position approach with BERO only	X	V1.4	X	-
Home position approach with zero mark only	X	V1.4	X	-
Home position approach with reversal cam	X	V1.4	X	-
Automatic mode: Axis-global automatic programs, allowing several axes to be integrated in one automatic program with the smoothing, coupling and 2D linear interpolation functions.	X	-	-	V1.0
Rounding in setup mode	-	V1.5	X	-
Roll feed with encoder switchover, positioning with machine encoder	X	V1.41	X	-
Roll feed with encoder switchover, positioning with motor encoder	X	V1.51	X	-
Valid only for axes **other than** roll feed axes in MDI and setup modes: MD22: Rounding time constant (setpoint rounding)	X	-	-	V1.00.48
Valid only for axes **other than** roll feed axes in MDI and setup modes: Extended definition of traversing curve with MD21, MD29, MD30, MD31 and MD32 (jerk limitation)	X	-	-	V1.00.48

Function	ASS	Firmware version MC	F01	M7
Synchronization functions				
Master value synchronization	X	V1.3	-	V1.00.48
Catch-up	X	V1.31	X	V1.00.48
Offset angle setting	X	V1.3	X	V1.00.48
Inner / outer window for position correction	X	V1.4	X	V1.00.48
Real master	X	V1.4	-	-
Master value correction	X	V1.4	X	-

4 Definitions of Notices and Symbols

Qualified Personnel

For the purpose of this documentation and the product warning labels, a "Qualified person" is someone who is familiar with the installation, mounting, start-up, operation and maintenance of the product. He or she must have the following qualifications:

- Trained or authorized to energize, de-energize, ground and tag circuits and equipment in accordance with established safety procedures.
- Trained or authorized in the proper care and use of protective equipment in accordance with established safety procedures.
- Trained in rendering first aid.

Proper Usage

The equipment may only be used for the applications described in this manual. The equipment may only be installed, commissioned and operated in accordance with the instructions contained in this manual. Third-party equipment and components may only be used on condition that the equipment is recommended or approved by Siemens.

Warning Symbols

Important information is highlighted in this manual by warning symbols.

Various pictograms are used in this publication to distinguish between different danger levels. The pictograms and their meaning are described below.

DANGER

indicates an **imminently** hazardous situation which, if not avoided, will result in death, serious injury and considerable damage to property.

WARNING

indicates a **potentially** hazardous situation which, if not avoided, could result in death, serious injury and considerable damage to property.

CAUTION

used with the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

CAUTION

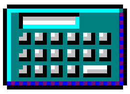
used without safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in property damage.

NOTICE

NOTICE used without the safety alert symbol indicates a potential situation which, if not avoided, may result in an undesireable result or state.

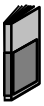
NOTE

For the purpose of this documentation, "Note" indicates important information about the product or about the respective part of the documentation which is essential to highlight.



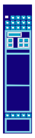
EQUATION

indicates an equation, e.g. for the calculation of a machine data.

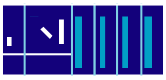


Further Information

indicates a reference to another part of the documentation.



The procedure described here is relevant only if the technology is installed on the MASTERDRIVES MC.



The procedure described here is relevant only if the technology is installed on the M7-FM.

WARNING

Hazardous voltages are present in this electrical equipment during operation.

Non-observance of the warnings can thus result in severe personal injury or property damage.

Only qualified personnel should work on or around the equipment

This personnel must be thoroughly familiar with all warning and maintenance procedures contained in this documentation.

The successful and safe operation of this equipment is dependent on correct transport, proper storage and installation as well as careful operation and maintenance.

NOTE

This documentation does not purport to cover all details on all types of the product, nor to provide for every possible contingency to be met in connection with installation, operation or maintenance.

Should further information be desired or should particular problems arise which are not covered sufficiently for the purchaser's purposes, the matter should be referred to the local SIEMENS sales office.

The contents of this documentation shall not become part of or modify any prior or existing agreement, commitment or relationship. The sales contract contains the entire obligation of SIEMENS AG. The warranty contained in the contract between the parties is the sole warranty of SIEMENS AG. Any statements contained herein do not create new warranties or modify the existing warranty.

SIEMENS

System Solutions

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Function Description 11.2002

Positioning and Synchronization

SIEMENS

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Part 1: Technology Functions

Function Description -
Positioning and Synchronization

Edition 11.2002

Documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the "Remarks" column.

Status code in the "Remarks" column:

A .... New documentation.
B .... Unrevised reprint with new Order No.
C .... Revised edition with new status.

Edition	Order No.	Remarks
04.98	6AT1880-0AA00-1BA0	**A**
10.99	6AT1880-0AA00-1BC0	**C**
11.2002	6AT1880-0AA00-1BE0	**C**

SIMATIC and SINUMERIK are registered trademarks of Siemens AG. Other names in this publication might be trademarks whose use by a third party for his own purposes may violate the rights of the registered holder.

We have checked that the contents of this document correspond to the hardware and software described. Nonetheless, differences might exist and therefore we cannot guarantee that they are completely identical. The information contained in this document is, however, reviewed regularly and any necessary changes will be included in the next edition. We welcome suggestions for improvement.

Subject to change without prior notice

Order No.: 6AT1880-0AA00-1BE0
Printed in the Federal Republic of Germany

Siemens-Aktiengesellschaft

Contents

1 Machine Data and Parameters of the Technology

Contents

In this section you will find all the information about the machine data of the technology.

1.1 General Information

Overview

The machine data of the technology determine the basic response of the technological functions:

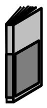
- Definition of the traversing curve
- Setting of monitoring parameters
- Connection of digital inputs and outputs

Machine data are determined and entered at the time of installation and, although they normally never need to be modified, they can be edited at any time.

When you use the technology for the first time, all the machine data are already initialized with default settings, allowing the axis to be traversed in all operating modes.

Your task is to optimize the necessary part of the machine data, in accordance with the mechanical conditions in your plant and/or your technical requirements.

Whether you use the technology on SIMATIC Motion Control or on the actual MASTERDRIVES MC, the position controller, speed controller and current controller are installed on the MASTERDRIVES MC (they are part of the basic functionality of the MASTERDRIVES MC). For this reason, you will need to start up the controllers on the MASTERDRIVES MC before you start working with the technology.



Further Information

The basic functionality of MASTERDRIVES MC is described in the manuals entitled "Compendium" and "Operator's Guide".

Definitions

Each machine data is described in accordance with the table below:

Abbreviation	Meaning
MD No.	Number of the machine data.
I	The machine data is required for incremental position measuring systems.
A	The machine data is required for absolute position measuring systems.
W	The machine data is required for roll feed systems.
Name	Name of the machine data. The MASTERDRIVES MC parameter in which the machine data is stored is specified in parentheses if you use the technology on the MASTERDRIVES MC.
Lower/Upper Input Limit	Smallest or largest possible input value.
Unit	Unit of the machine data with reference to the position of the decimal point.
Default	Value of the machine data at the time of supply.

Units

All units derived in any way from the position measuring system are specified with the abbreviation LU.

Example:

You have a position measuring system which resolves 5000 increments in a traversing distance of 10 mm. In order to normalize the actual position value to 1 µm (so that the actual position value is displayed in µm), you need 10000 increments in a traversing distance of 10 mm. You must multiply the increments by 2 in order to achieve this. This multiplication factor is the actual value weighting factor AVWF. You must specify the AVWF when you start up the position controller.

You have thus defined the following settings:

- The AVWF for the position controller is 2
- The actual position value is normalized to 1 µm, that is the smallest length unit selected is **LU** = **1 µm**.

Once the length unit LU is known, LU can be substituted by the appropriate length in the units of all the data.

Examples for normalization to:	1µm	10^{-3} degrees
LU	µm	10^{-3} degrees
1000*LU	mm	degrees
1000*LU/min	mm/min	degrees/min

1.2 List of Machine Data

The table below shows a complete list of the machine data. The individual machine data are described in the sections following these machine data list.

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
1				Encoder type / axis type (U501.01)	0: Axis does not exist 1: Axis with incremental position encoder 2: Axis with absolute position encoder 3: Roll feed 4: IM178 with incremental position encoder 5: IM178 with absolute position encoder		-	1
2				Axis assignment (U501.02)	1: X axis 7: XA axis 2: Y axis 8: YA axis 3: Z axis 9: ZA axis 4: A axis 10: AA axis 5: B axis 11: BA axis 6: C axis 12: CA axis 13: XB axis 14: YB axis etc. until max. 32 axis		-	1,2,3,4
3				Reference point – coordinate (U501.03)	-999 999.999	999 999.999	1000*LU	0
4				Reference point – offset (U501.04)	-999 999.999	999 999.999	1000*LU	0
5				Reference point – approach direction (U501.05)	1: Reference point right of Bero 2: Reference point left of Bero 3: Set reference point		-	1
6				Reference point – reducing velocity (U501.06)	1	19 999 999	1000*LU/min	500
7				Reference point – approach velocity (U501.07)	1	19 999 999	1000*LU/min	5000
8				Reference point approach	0: with Bero and zero mark 1: with Bero only 2: with zero mark only		-	-
9				Reserved	-	-	-	-

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
10				Position encoder – adjustment (U501.10)	-999 999.999	999 999.999	1000*LU	0
11				Linear/rotary axis (U501.11)	Rotary = 0.001 Linear = 0	999 999.999	1000*LU	0
12				Software limit switches – negative (U501.12)	-999 999.999	999 999.999	1000*LU	-999 999.999
13				Software limit switches – positive (U501.13)	-999 999.999	999 999.999	1000*LU	999 999.999
14				Following error monitoring – at standstill (U501.14)	0.001	100.000	1000*LU	0.100
15				Following error monitoring – in motion (U501.15)	0.001	999 999.999	1000*LU	20.000
16				In position – timer monitoring (U501.16)	0.010	99.999	s	0.500
17				In position – exact stop window (U501.17)	0.001	99.999	1000*LU	0.100
18				Acceleration (U501.18)	1	99 999	1000*LU/s²	1000
19				Deceleration valid for setup, MDI, automatic, single block and slave modes (U501.19)	1	99 999	1000*LU/s²	1000
20				Deceleration for collision (U501.20)	1	99 999	1000*LU/s²	1000
21				Jerk limiting – positive (U501.21)	0 = inactive 1	999 999	1000*LU/s³	0
22				Rounding time constant	0	1000	ms	0
23				Traversing velocity – maximum (U501.23)	1	19 999 999	1000*LU/min	12 288

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
24				M functions – output type (U501.24)	1: During positioning, time-driven 2: During positioning, acknowledge-driven 3: Before positioning, time-driven 4: Before positioning, acknowledge-driven 5: After positioning, time-driven 6: After positioning, acknowledge-driven 7: Actual value-dependent, time-driven 8: Current -value-dependent, acknowledge-driven 9: Expanded, actual value dependent, time-driven 10: Expanded, actual value dependent, acknowledge-driven		-	1
25				M functions – output time (U501.25)	0.004	99.999	s	0.500
26				Time override (U501.26)	0: Time override active 1: Time override inactive		-	1
27				Corner rounding window 1 (SIMATIC Motion Control only) (U501.27)	0.001 0 = inactive	999 999.999	1000*LU	0
28				Corner rounding window 2 (SIMATIC Motion Control only) (U501.28)	0.001 0 = inactive	999 999.999	1000*LU	0
29				Acceleration breakpoint – velocity (U501.29)	1 0 = inactive	1 500 000	1000*LU/min	0
30				Deceleration breakpoint – velocity (U501.30)	1 0 = inactive	1 500 000	1000*LU/min	0
31				Acceleration breakpoint – acceleration (U501.31)	1 0 = inactive	99 999	1000*LU/s²	0
32				Deceleration breakpoint – deceleration (U501.32)	1 0 = inactive	99 999	1000*LU/s²	0
33				Constant travel time (U501.33)	0.001 0 = inactive	99.999	s	0

MD No.	I	A	W	Name	Input Limit Lower	Upper	Unit	Default
34				Pre-position reached – lead time (U501.34)	0.001 0 = inactive	99.999	s	0
35				Pre-position reached – output time (U501.35)	0.001 0 = inactive	99.999	s	0
36				Acceleration overshoot (U501.36)	0	100	%	0
37				Response after abort (U501.37)	0: Standard response 1: Approach last target position with no evaluation of direction of movement 2: Approach last target position with evaluation of direction of movement		-	0
38				Backlash compensation (U501.38)	0	9.999	1000*LU	0
39				Backlash compensation – preferred position (U501.39)	1: Preferred position positive (no backlash compensation is calculated during first positive traversing movement) 2: Preferred position negative (no backlash compensation is calculated during first negative traversing movement)		-	1
40				Backlash compensation – velocity limitation (U501.40)	1 0 = inactive	999	1000*LU/min	999
41				Acceleration time, operating mode "reference point approach / control" (U501.41)	0.001 0 = inactive	99.999	s	1.000
42				Deceleration time, operating mode "reference point approach / control" and synchronization (U501.42)	0.001 0 = inactive	99.999	s	1.000
43				Deceleration time during errors (U501.43)	0.001 0 = inactive	99.999	s	0
44				External NC block change – setting (U501.44)	0: Warning at end of NC block 1: No warning at end of NC block		-	0

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
45				Digital inputs – function 1 (U501.45)	0: No function 1: Start ORed 2: Start ANDed 3: Set actual value on-the-fly 4: External NC block change 5: In-process measurement 6: Collision 7: Bero for reference point approach 8: Reverse cam for reference point approach 9: Read-in enable, externally programmable		6-digit input	0
46				Digital inputs – function 2 (U501.46)	0: No function 1: Disable actual value 2: External read-in enable 3: External read-in enable ANDed 4: Set floating reference point 5: No function 6: No function 7: No function 8: Trigger signal inserter/ejector OR connected 9: Synchronize traversing table OR connected		6-digit input	0
47				Digital outputs – function 1 (U501.47)	0: No function 1: Destination reached, axis stationary [DRS] 2: Axis travels forwards [FWD] 3: Axis travels backwards [BWD] 4: M change from M97 5: M change from M98 6: Start enable [ST_EN]		6-digit input	0

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
48				Digital outputs – function 2 (U501.48)	0: No function 1: Constant travel 2: Acceleration 3: Deceleration 4: Acceleration or deceleration 5: Pre-position reached 6: Encoder evaluation message		6-digit input	0
49				Feedforward control – speed (U501.49)	0	150	%	0
50				Feedforward control – acceleration (U501.50)	1 0 = inactive	99 999	1000*LU/s²	0

If you are using the technology in conjunction with SIMATIC Motion Control and the IM178 interface module, the MD51 to MD70 is available as machine data for this application (MD1 =4 or 5).

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
51				Actual value evaluation mode	0: Evaluation using AVWF pre-point / post-point		-	0
52				Actual value evaluation factor pre-point	0	999	-	1
53				Actual value evaluation factor post-point	0	99 999 999	-	0
54				Reserved	-	-	-	1
55				Reserved	-	-	-	1
56				SSI tolerance window Sensor monitoring	1 0 = inactive	99 999 999	increments	0
57				Kv factor (positioning)	0.100	20.000	1000*LU/ min/LU	1.000
58				Drift compensation	-9999	+9999	LU	0
59				Lower limit for dyn. following error monitoring	1000	999 999	LU	1000
60				System adjustment Setpoint output	0: no inversion – positive speed → positive output voltage 1: inversion of output		-	0
61				Output voltage at v-max	0,5	10.000	Volt	10.000
62				Setpoint output on simulation	0: setpoint output off (0 Volt) 1: setpoint output on		-	0
63				Parameterization of output Q0	0: output Q0 not used 1..6: output O1..O6		-	0
64				Parameterization of output Q1	0: output Q1 not used 1..6: output O1..O6		-	0
65				Remove hardware monitoring (IM178)	Bit 0: sensor signal lead error Bit 1: sensor error Bit 2: zero mark error Bit 3: SSI sensor validity error		-	0
66				Kv factor (synchronization)	0.100	20.000	1000*LU/ min/LU	1.000
67				DAC limitation factor	1 0= inactive	100	-	0
68 to 70				Reserved				

Transfer/ Activation

You can transfer machine data to the axis at any time, since the transfer does **not** activate the data. To activate the machine data, you must execute the function "activate machine data".

Input/output and/or activation of the machine data is performed with the standard user interfaces available, or in STEP 7 using the "machine data input/output" task or "activation of machine data" (on the MASTERDRIVES MC you can activate the machine data with parameter U502).

Further Information

You will find a description of the "input/output machine data" task and "activation of machine data" in the "Task Description" in the chapter entitled "Machine Data Tasks".

1.3 Function Parameter

FP-No.	FW version		Name	Input Limit		Unit	Default
	MC	M7		Lower	Upper		
1	-	-	Reserved (U504.01)	-		-	0
2	V1.4	V1.0 0.48	Set floating reference point, position correction Inner window (window 1) (U504.02)	-2 147 483 648 0: deactivated	2 147 483 647	LU	0
3	V1.4	V1.0 0.48	Set floating reference point, position correction Outer window (window 2) (U504.03)	-2 147 483 648 0: deactivated	2 147 483 647	LU	0
4	V1.4	-	Correction mode, set floating reference point (U504.04)	0: Shortest path correction 1: positive correction only 2: negative correction only		-	0
5	-	-	Reserved (U504.05)	-		-	-
6	V1.4 1	-	Limit monitoring - encoder changeover (U504.06)	1	999 999	LU	2000
7	-	-	Reserved (U504.07)	-		-	-
8	-	-	Reserved (U504.08)	-		-	-
9	-	-	Reserved (U504.09)	-		-	-
10	-	-	Reserved (U504.10)	-		-	-

Transfer/ Activation

The function parameters are valid immediately after transfer. Therefore they do not need to be activated in the same way as the machine data.

Further Information

You will find a description of the "input/output function parameters" in the "Task Description" in the chapter entitled "Function Parameter Tasks".

1.4 General Machine Data

You only need to enter these machine data if the default settings differ from your configuration.

Encoder Type/ Axis Type

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
1				Encoder type / axis type (U501.01)	0: Axis does not exist 1: Axis with incremental position encoder 2: Axis with absolute position encoder 3: Roll feed 4: IM178 with incremental position encoder 5: IM178 with absolute position encoder		-	1

Encoder Type

The position encoder type is derived from the position measuring system used on the MASTERDRIVES MC and specifies how the technology is to weight an actual position value which has been transmitted. With SIMATIC Motion Control, the technology is also informed whether an IM178 is used for path logging.

- If an incremental position measuring system is used (resolver, encoder, etc.), you must enter a value of 1, 3 or 4 in machine data 1.
- If an absolute position measuring system is used (SSI, Endat, etc.) you must enter a value of 2 or 5 in machine data 1.

CAUTION

If you are using an IM178 in conjunction with an absolute value detector, the actual value can jump if the detector range is exceeded. If this happens, the axis is stopped with a following error.

Axis Type

There are 2 axis types:

- Axis type "axis" can be used with both incremental and absolute position encoders. This type of axis is used if an absolute reference to the position measuring system is required throughout the entire period of operation. This is also the only type of axis to support functions such as interpolation (SIMATIC Motion Control only), synchronization and complex NC programs. If axis type "axis" is to be used, you must enter a value of 1 or 2 in machine data 1.
- Axis type "roll feed" can only be used in association with incremental position encoders, because there is no absolute reference to the position measuring system. All movements are performed relatively, and the actual position value of the axis is automatically reset to zero before each positioning operation. The performance capability of "roll feed" is primarily concerned with providing a high dynamic response. Additional machine data are available in order to optimize the traversing curve further for this purpose. If axis type "roll feed" is to be used, you must enter a value of 3 in machine data 1.

NOTE

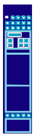
If the axis type is set as "drum feed", then synchronization mode cannot be selected.

You can deactivate the axis by defining machine data 1 = 0. This is only practical if you are using the technology with several axes on SIMATIC Motion Control and you want to take one axis out of commission temporarily (e.g. in order to reduce the load on the processor).

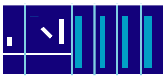
The deactivated axis continues to be capable of communication (i.e. data exchange via tasks remains possible), however none of the technological functions can be operated. In this state, the technological control signals are inactive and the technological checkback signals are defined as 0.

Axis Assignment

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
2				Axis assignment (U501.02)	1: X axis 2: Y axis 3: Z axis 4: A axis 5: B axis 6: C axis 13: XB axis 14: YB axis etc. up to max. 32 axes	7: XA axis 8: YA axis 9: ZA axis 10: AA axis 11: BA axis 12: CA axis	-	1,2,3,4

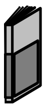


With the MASTERDRIVES Motion Control solution (technology option F01), only one axis is ever installed on a MASTERDRIVES MC. You can assign a name to this axis with the axis assignment function. With MASTERDRIVES MC, it is only possible to allocate axes 1 to 6.



With the SIMATIC Motion Control solution several axes can be managed by one control. A definite axis name must be assigned to each of these SIMATIC Motion Control axes. It is not permitted to assign the same axis name to several axes on one control.

The axis names are needed exclusively for NC program execution, since different axes in an NC program can be addressed using their axis names.



Further Information

You can find further information in the Programming Guide in Chapter 1.2 "Traverse block".

1.5 Machine Data for Reference Point Approach

You only need to enter these machine data if the position encoder type / axis type = 1 or 4 (MD1).

Further Information

The reference point approach function is described in the section entitled "Reference Point Approach Mode".

Reference Point Coordinate

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
3				Reference point – coordinate (U501.03)	-999 999.999	999 999.999	1000*LU	0

The value in the reference point coordinate can be used to match the reference point of the axis to the coordinate system of the machine. When the reference point is reached, the axis sets the actual position value to the value entered in machine data 3.

Reference Point Offset

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
4				Reference point – offset (U501.04)	-999 999.999	999 999.999	1000*LU	0

In order to prevent the axis from coming to a standstill at the synchronization point after reference point approach/set reference point, but to displace it by a defined amount, you can enter the necessary value in machine data 4. Once the reference point offset has been traversed, the reference point is set to the reference point coordinate.

Refernce point direction of approach

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
5				Reference point – approach direction (U501.05)	1: Reference point to right of Bero 2: Reference point to left of Bero 3: Set reference point		-	1

A setting of 1 or 2 in machine data 5 defines the direction in which the axis is to depart from the reference point Bero during reference point approach, in order to use the next zero pulse of the measuring system as a synchronization pulse:

- ♦ Reference point to the right of the Bero means the axis departs in the direction of increasing actual values.
- ♦ Reference point to the left of the Bero means the axis departs in the direction of decreasing actual values.

NOTE

The Bero required for reference point approach must be controlled by a digital input. See machine data 45 for more information.

CAUTION

The setting defined here with reference to the position of the reference point **must** match the setting of MASTERDRIVES parameter P183 (reference point detection mode).

A setting of 3 in machine data 5 initiates a "set reference point" operation. At the current position of the axis, the reference point is set to the reference point coordinate or the reference point offset is traversed and the reference point is subsequently set to the reference point coordinate.

"Set reference point" is used if you want to synchronize the measuring system without using a Bero and zero pulse.

Reference Point Reducing Velocity

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
6				Reference point – reducing velocity (U501.06)	1	19 999 999	1000*LU/min	500

Machine data 6 defines the velocity at which the axis departs from the reference point cam in order to search for the zero pulse.

NOTE

The reference point reducing velocity cannot be modified by the velocity override [OVERRIDE].

Reference Point Approach Velocity

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
7				Reference point – approach velocity (U501.07)	1	19 999 999	1000*LU/min	5000

Machine data 7 defines the velocity at which the reference point cam is to be approached.

Reference Point Approach

NOTE

Machine data 8 is only available from MASTERDRIVES MC firmware version ≥ V1.4.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
8				Reference point approach	0: with Bero and zero mark 1: with Bero only 2: with zero mark only		-	-

Machine data 8 can be used to define other types of reference point approaches, and the reference point approach used until this point, "with Bero and zero mark" is parameterized as MD8 = 0.

NOTE

Reference point approach with Bero only: The Bero required for reference point approach must be controlled by one of the digital inputs 4 or 5, as these are the only one which trigger an interrupt. See machine data number 45.

1.6 Machine Data for Actual Value Weighting

The basic actual value weighting is evaluated with reference to MASTERDRIVES MC parameter P169 (actual value weighting factor – before decimal point) and P170 (actual value weighting factor – after decimal point), if the motor sensor evaluation module in slot C is used.

Encoder Alignment

You only need to enter these machine data with encoder type/axis type = 2 (MD1) if you require encoder alignment.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
10				Encoder – alignment (U501.10)	-999 999.999	999 999.999	1000*LU	0

Machine data 10 can be used to correct (shift) the absolute actual encoder value of the axis. Absolute position measuring systems only indicate positive actual values. If the actual value system is to include a negative area on the axis, the positive actual value of the position encoder must be shifted.

Example:
The position encoder indicates actual values in the range from 0 to 100 000 mm but the actual value display is to indicate a range from -50 000 to +50 000. This can be achieved by entering –50 000 in machine data 10.

EQUATION

MD10 = actual value display - actual position value

NOTE

Notwithstanding the encoder alignment capability, you must make sure with linear axes that the zone transition on the absolute position encoder is located outside the possible traversing range.

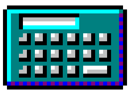
Linear/Rotary Axis You only need to enter these machine data if you use a rotary axis.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
11				Linear/rotary axis (U501.11)	Rotary axis = 0.001 Linear axis = 0	999 999.999	1000*LU	0

If the axis is operated as a linear axis, a value of 0 must be entered in machine data 11. If the axis is operated as a rotary axis, the normalized distance for one revolution of the rotary axis must be entered in machine data 11. When the value in machine data 11 is crossed, the actual position value starts again from zero.

Examples:

- If the measuring system is normalized to 0.001 degrees, a value of 360.000 [degrees] must be entered in machine data 11.
- If the rotary axis has a scope of 300 mm and the actual value is normalized to 1 µm, a value of 300.000 [mm] must be entered in machine data 11.



EQUATION

Linear axis: **MD 11** = 0

Rotary axis: **MD 11** = Distance for one revolution of the rotary axis in accordance with the normalization

CAUTION

When setting the rotary axis length and the traverse speed maximum, the following **must** be taken into consideration:

$$\frac{\text{vmax [1000} * \text{LU/min]} * \text{TA [ms]}}{60} < \frac{\text{Rotary axis length [LU]}}{2}$$

Failure to observe this relationship leads to uncontrolled movement!

NOTE

It is only possible to use absolute encoders with continuously rotating axes under the following conditions:

- One revolution of the rotary axis equals exactly one revolution of the position encoder (with single-turn encoders).
- One revolution of the rotary axis equals exactly one zone revolution of the position encoder (with multi-turn encoders).

1.7 Machine Data for Software Limit Switch Monitoring

Software Limit Switches – Negative and Positive

You only need to enter these machine data if you are not satisfied with the default setting, however for safety reasons it is advisable to ensure that the software limit switches are correctly set with reference to the mechanical conditions of your system.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
12				Software limit switches – negative (U501.12)	-999 999.999	999 999.999	1000*LU	-999 999.999
13				Software limit switches – positive (U501.13)	-999 999.999	999 999.999	1000*LU	999 999.999

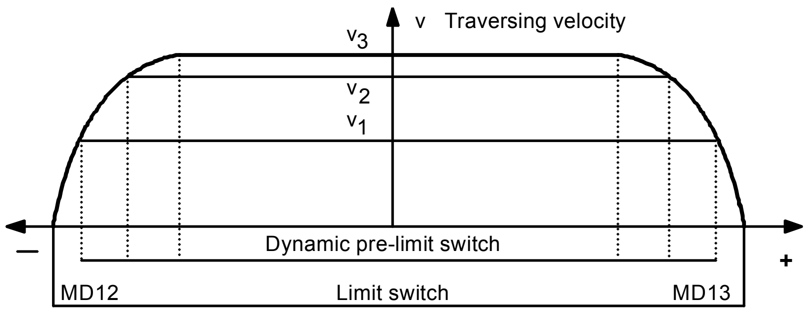
Two machine data are available for limiting the traversing area. These define the positive and negative limit positions.

In systems with absolute encoders, the software limit switch monitoring is active immediately after you switch on the machine. With incremental encoders, the measuring system first has to be synchronized by means of "reference point approach" or "set reference point".

The approach of a software limit switch is basically the same as a positioning operation, i.e. the software limit switch is approached, to the nearest increment, via the defined deceleration ramp. The software limit switch approach is also indicated by a warning number [WARN_NO] and by the "overtravel" checkback signal [OTR].

In order to move away from a software limit switch which has been approached, the user must first acknowledge (clear) the warning. You can then only move away from the software limit switch in the "setup" or "MDI" modes.

Software limit switch as a function of traversing velocity:



CAUTION

It is possible to cross the software limit switches in "control" mode and, if incremental encoders are used, when synchronization (reference point approach) has not been performed. The software limit switches are not a substitute for the hardware limit switches used for emergency stop responses.

If you do not require the software limit switches, you should initialize machine data 12 and 13 with values which are outside the possible traversing area (this is the default setting). For rotary axes, this means that the negative software limit switch must be < 0 and the positive software limit switch > machine data 11 (linear/rotary axis).

1.8 Machine Data for Position Control Monitoring

You only need to enter these machine data if you are not satisfied with the default settings, however for safety reasons it is advisable to set the following error monitoring parameters in accordance with the mechanical conditions.

Following Error Monitoring – at Standstill

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
14				Following error monitoring – at standstill (U501.14)	0.001	100.000	1000*LU	0.100

The current position must be maintained when the axis is at a standstill. This task is handled by the position control system, however it is still possible for external forces to move the axis out of position. The position control system immediately attempts to compensate for such a deviation. If this is not possible because of the magnitude of the mechanical forces, and if the "following error monitoring- at standstill" tolerance window is consequently exceeded, the position controller is deactivated and the speed controller is activated with a speed setpoint of 0. A warning number [WARN_NO] is also output.

EQUATION

MD14 >= 5 * actual following error at standstill

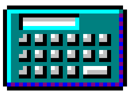
Following Error Monitoring – in Motion

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
15				Following error monitoring – in motion (U501.15)	0.001	999 999.999	1000*LU	20.000

The following error must also be verified while the axis is traversing. The following error which occurs during travel depends on both the traversing velocity and on the gain of the position control loop. If the "following error monitoring - in motion" tolerance window is exceeded while the axis is traversing, the position controller is deactivated and the axis is stopped with speed control mode being initiated for the duration of the deceleration time during error (machine data 43). A warning number [WARN_NO] is also output.

The following data must be available in order to calculate the necessary value:

- Gain factor (Kv) setting on the position controller: MASTERDRIVES MC parameter P204
- Traversing velocity – maximum (machine data 23)



EQUATION

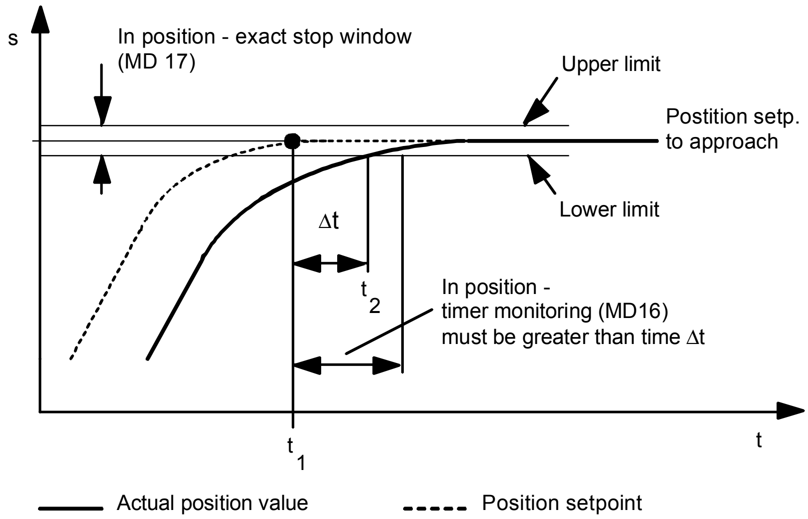
MD15 >= 1.5 * MD23 / Kv factor (P204)

Example for normalization = 1µm:
P204 = 1,7 [mm/min/µm]
MD23 = 45 000 [mm/min]

MD15 >= 1,5 * 45 000 / 1,7 ~ 40.000 [mm]

In Position - Timer Monitoring and Exact Stop Window

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
16				In position – timer monitoring (U501.16)	0.010	99.999	s	0.500
17				In position – exact stop window (U501.17)	0.001	99.999	1000*LU	0.100



Two position monitoring routines are activated when the position is approached:

1. If no more partial setpoints are generated in the interpolator, the position setpoint has reached the target position to be approached. The "in position – timer monitoring" (machine data 16) is started at this point.
2. If the difference (following error) between the position setpoint and the actual position value falls below the "in position – exact stop window", the "in position – timer monitoring" is stopped and the checkback signal "destination reached, axis stationary" [DRS] indicates that the correct target position has been reached.

Error situation:
The actual position value does not reach the "in position – exact stop window". When the time in "in position – timer monitoring" expires, the positioning operation is aborted, the position controller is deactivated and the speed controller is activated with a setpoint of 0. A warning [WARN] is also output by the axis.

EQUATION

A reference value of 500 to 2000 ms results for "position reached – timer monitoring", depending on the velocity and mass inertia.

MD16 ~ 0.5 to 2.0 s

To prevent the "following error standstill" warning from occurring on every positioning operation, MD17 must be < MD14 (following error monitoring – at standstill).

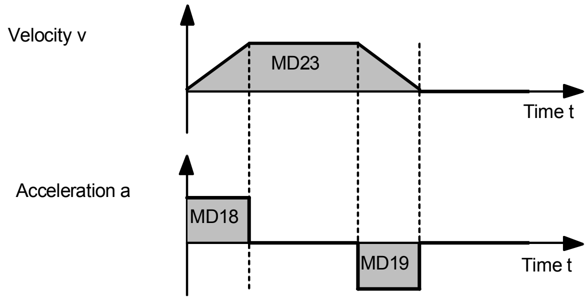
1.9 Machine Data for Defining the Traversing Curve

Acceleration and Deceleration

You only need to enter these machine data if you are not satisfied with the default setting.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
18				Acceleration (U501.18)	1	99 999	1000*LU/s²	1000
19				Deceleration valid for setup, MDI, automatic, single block and slave modes (U501.19)	1	99 999	1000*LU/s²	1000

The mass inertia of the mechanical components counteract sudden changes in state. The machine data for acceleration and deceleration match the transient response of the reference variable to the transient response of the control loop.



EQUATION for µm normalization:

MD18 or MD19 = MD23 [mm/min] / 60 * t [s]

The gradient of the acceleration and deceleration ramps must be selected such that a high current is used to accelerate and decelerate, but the current never reaches the maximum permissible level.

NOTICE

The ratio between acceleration MD18 and deceleration MD19 must be less than/equal to 1:10. Otherwise the "activate machine data" task will end with error message 2040 and the ratio is then internally limited to 1:10.

NOTICE

The position control loop must never reach the current limit, because this "disconnection of the control loop" would cause an overshoot.

Example:
To accelerate to the maximum traversing velocity (MD23 = 30 000 mm/min) in 10 s:

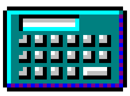
Calculation:
MD19 = MD23 / 60 * t = 30 000 / 60 * 10 = 50 [mm/s²]

Deceleration for Collision

You only need to change these machine data if you want to use the collision function in automatic mode.

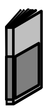
MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
20				Deceleration for collision (U501.20)	1	99 999	1000*LU/s²	1000

In certain applications, it is necessary to stop an axis or axes as rapidly as possible in position control mode as a result of an external condition. The collision function can be selected in NC programs and is triggered when the programmable digital input is canceled (the digital input is programmed with the collision function, MD45). While the machine data "acceleration" (MD18) and "deceleration" (MD19) can be mitigated by functions such as "time override" (MD26), "acceleration override" (see the chapter on control and checkback signals) or interpolation (see the Programming Guide), "deceleration for collision" is executed without modification.



EQUATION

MD20 >= MD19
(if MD19 < MD20, the deceleration value is MD19)



Further Information

The function is described in the Programming Guide in Chapter 3.9 "Collision Monitoring".

Rounding Time Constant

When configuring a rounding time constant the position setpoint is influenced. The traversing curve is rounded at the beginning and at the end, i.e. the axis is accelerated more slowly and also decelerated more slowly. As a result, the positioning process is extended.

Rounding can, for example, be used to protect the mechanical system or also to prevent oscillating motions of the axis.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
22				Rounding time constant	0	1 000	ms	0

NOTE

The "rounding time constant" machine data MD22 is only valid for SIMATIC Motion Control.

Parameter U505 [817] is provided for this in MASTERDRIVES MC.

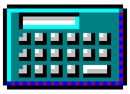
Traversing Velocity – Maximum

When the position controller is installed, MASTERDRIVES MC parameter P205 must be configured with the rated velocity v_{Rated}. The value of v_{Rated} must be defined as the "traversing velocity– maximum" in order to ensure that functions such as feedforward control operate correctly.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
23				Traversing velocity – maximum (U501.23)	1	19 999 999	1000*LU/min	12 288

The "traversing velocity– maximum" defines the upper limit for all velocity inputs:

- If you attempt to enter velocity values (e.g.: NC blocks) greater than MD23, they are automatically limited to the magnitude of MD23.
- If a velocity override [OVERRIDE] results in a velocity greater than MD23, the velocity is limited to the magnitude of MD23.
- If a path velocity is defined during interpolation such that the velocity component of the axis exceeds MD23, the path velocity is reduced until the velocity component of the axis matches MD23.



EQUATION for µm normalization:

MD23 = P205 (v_{Rated}) [mm/min]

NOTE

When setting the rotary axis length and the traverse speed maximum, the following **must** be taken into consideration:

$$\frac{\text{vmax}\,[1000 * \text{LU/min}] * \text{TA}\,[\text{ms}]}{60} < \frac{\text{Rotary axis length}\,[\text{LU}]}{2}$$

Failure to observe this relationship leads to uncontrolled movement!

1.10 Machine Data for the M Functions

You only need to change these machine data if you use automatic and/or single-block mode and you are not satisfied with the default settings.

M functions are block elements of an NC program. Their main purpose is to control machine functions. During execution of the NC program, the M numbers programmed in the NC blocks are output as numbers in the checkback signals of the axis for further processing by the user program.

NOTE The actual value-dependent M function output is only available from MASTERDRIVES MC firmware version ≥V1.32.

Further Information

The handling of M functions in control and checkback signals is described in the chapter entitled "Control and Checkback Signals".

M-Functions – Output Type

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
24				M functions – output type (U501.24)	1: During positioning, time-driven 2: During positioning, acknowledge-driven 3: Before positioning, time-driven 4: Before positioning, acknowledge-driven 5: After positioning, time-driven 6: After positioning, acknowledge-driven 7: Actual value-dependent, time-driven 8: Actual value-dependent, acknowledge-driven		-	1

Two types of function are defined with "M functions – output type":

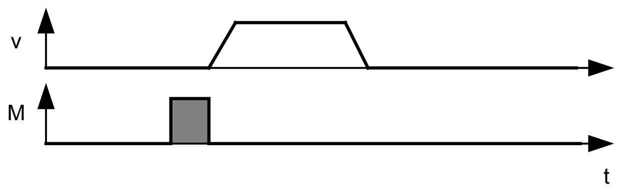
- M function acknowledgement with the option of a time-driven or acknowledge-driven response
- The output time of the M functions with the option of output before, during or after positioning or depending on the actual value.

Acknowledgement feature:

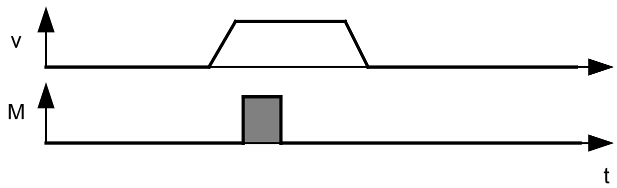
Time-driven M functions are output for a defined length of time (MD25). Acknowledge-driven M functions are output until they are acknowledged by the user with the control signal "acknowledge M function" [ACK_M].

Output time:

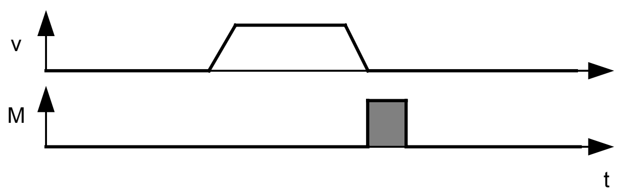
- Output of M function before positioning



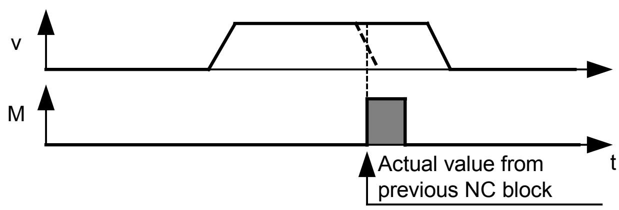
- Output of M function during positioning



- Output of M function after positioning



- Actual value-dependent M function output



M Functions – Output Time

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
25				M functions – output time (U501.25)	0.004	99.999	s	0.500

"M functions – output time" is used to define the M function output time for the acknowledgement feature.

1.11 Machine Data for the Time Override

You only need to change these machine data if you are not satisfied with the default setting.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
26				Time override (U501.26)	0: Time override active 1: Time override inactive		-	1

The "time override" machine data can be used to activate or deactivate the time override.

- ♦ When the time override is inactive, the acceleration and deceleration ramp is not affected when you change the velocity with the override (control signal).
  Example:
  If the override control signal is changed from 100 % to 50 %, only the velocity is halved; the positioning time takes correspondingly longer.
- ♦ When the time override is active, a change in the velocity caused by the override (control signal) causes the acceleration and deceleration ramp to be modified such that the positioning time is altered in proportion to the override setting.
  Example:
  If the override control signal is changed from 100 % to 50 %, the velocity is halved, the acceleration and deceleration are quartered and the positioning time is thus exactly doubled.

Further Information

You will find a detailed description of the effect of the time override in Chapter 2.4.9 "Override".

1.12 Machine Data for the Chaining Function (SIMATIC Motion Control only)

You only need to change these machine data if you use automatic mode and are not satisfied with the default settings.

In NC programs, it is possible to define the movement sequences of several axes using the "chaining" function. At transitions of movement from one axis to another, there is an overlap in the movement at the corner rounding point (before one axis has decelerated completely, the other is already accelerating). This results in high-speed sequences of movement in which the programmed target positions can be approached, with a greater or lesser level of accuracy, through the corner rounding behavior. The machine data below can be used to define the level of accuracy with which these programmed target positions are approached.

Either of the two corner rounding windows (MD27 and MD28) can be selected in the NC program, in order to switch between 2 corner rounding responses.

Further Information

You will find a detailed description of the corner rounding windows and their conditions in NC programs in the Programming Guide in Chapter 3.10.3 "Corner rounding windows 1 and 2".

Corner Rounding Window 1

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
27				Corner rounding window 1 (SIMATIC Motion Control only) (U501.27)	0.001 0 = inactive	999 999.999	1000*LU	0

If you define corner rounding window 1 = 0, corner rounding window 1 is inactive. There is no impact on the corner rounding response.

The value entered in corner rounding window 1 determines the proximity of the actual position value on the axis to the programmed target position at which the corner rounding (acceleration of the following axis) takes place.

Corner Rounding Window 2

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
28				Corner rounding window 2 (SIMATIC Motion Control only) (U501.28)	0.001 0 = inactive	999 999.999	1000*LU	0

If you define corner rounding window 2 = 0, corner rounding window 2 is inactive. There is no impact on the corner rounding response.

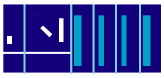
The value entered in corner rounding window 2 determines the proximity of the actual position value on the axis to the programmed target position at which the corner rounding (acceleration of the following axis) takes place.

1.13 Machine Data for Extended Definition of the Traversing Curve for Roll Feed or for Non Roll Feed Axes for the MDI and Setup Modes

These machine data only have to be changed if you use the roll feed axis type (MD1) or if you require a traversing curve with breakpoints for non roll feed axes in the MDI and setup modes.

NOTES

Machine data for extended definition of the traversing curve are only available on MASTERDRIVES MC for roll feed axes.



On SIMATIC Motion Control machine data are also available for the extended definition of the traversing curve for **non** roll feed axes for the MDI and setup modes from firmware version ≥ V1.00.48 onwards.

The following description refers exclusively to roll feed. It is also accordingly applicable for the non roll feed axes on SIMATIC Motion Control.

Roll feeds are highly dynamic drives. Their purpose is to transport the material as rapidly as possible with the greatest possible care. The contact between the roll feed motor and the material consists only of the friction of the drive rollers.

In applications where lesser demands are made on the roll feed, it is usually sufficient to operate without breakpoints, i.e. using only the settings for acceleration (MD18) and deceleration (MD19).

Every mass counteracts a change in movement with its natural mass inertia. The material transported by a roll feed is no exception. During acceleration, it is important to avoid roller slippage and material tear. During deceleration, it is necessary to achieve a high positioning accuracy without compressing the material. In order to meet these requirements, breakpoints are available in the acceleration and deceleration phase for optimizing the traversing curve in line with the mechanical conditions.

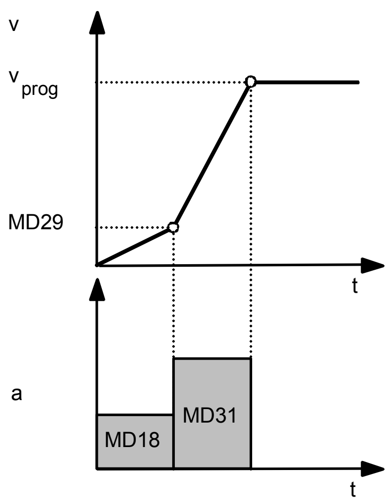
Acceleration Breakpoint – Velocity and Acceleration

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
29				Acceleration breakpoint – velocity (U501.29)	1 0 = inactive	1 500 000	1000*LU/min	0
31				Acceleration breakpoint – acceleration (U501.31)	1 0 = inactive	99 999	1000*LU/s²	0

NOTICE

The acceleration breakpoint – acceleration MD31 and deceleration breakpoint – deceleration MD32 ratio must be less than or equal to 1:10. The "activate machine data" task is otherwise terminated with error message 2040 and the ratio is then internally limited to 1:10

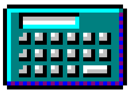
If a value of 0 is defined for the acceleration breakpoints (MD29 and MD31), there is no acceleration breakpoint. In this case, only the machine data "acceleration" (MD18) is active during the acceleration phase.



"Acceleration breakpoint – velocity" defines the start point of the break. The normal acceleration (MD18) applies up to the break. "Acceleration breakpoint – acceleration" defines the value of the acceleration (MD31) after the break.

Special situations:

- If the programmed velocity v_{prog} is less than the breakpoint velocity (MD29), there is no break during acceleration.
- The same naturally applies if the programmed velocity cannot be reached because the positioning path is too short or a reduced override is used.



EQUATION

0 < MD29 < MD23 (traversing velocity – maximum)

MD31 > MD18 (acceleration)

Setting the breakpoint for the acceleration phase:

- The movement normally starts with a flat acceleration ramp (MD18), in order to prevent roller slippage or material tear.
- After the initial phase, once the velocity (MD29) has been reached, the system switches to the steep acceleration ramp (MD31) and accelerates to the programmed velocity v_{prog}.

This subdivision of the acceleration phase of the traversing curve is the normal solution for rapid transport with low wear on the material. However, it is possible to use any other combination by changing the values in the machine data.

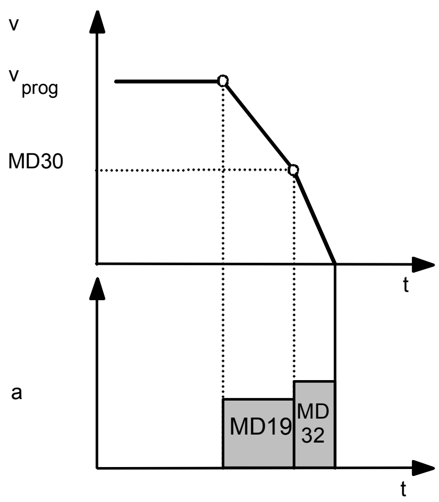
NOTE

The time override affects machine data MD18 and MD31 but not the breakpoint MD29.

Deceleration Breakpoint – Velocity and Deceleration

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
30				Deceleration breakpoint – velocity (U501.30)	1 0 = inactive	1 500 000	1000*LU/min	0
32				Deceleration breakpoint – deceleration (U501.32)	1 0 = inactive	99 999	1000*LU/s²	0

If a value of 0 is defined for the deceleration breakpoints (MD30 and MD32), there is no deceleration breakpoint. In this case, only the machine data "deceleration" (MD19) is active during the deceleration phase.



"Deceleration breakpoint – velocity" defines the start point of the break. The normal deceleration (MD19) applies up to the break. "Deceleration breakpoint – deceleration" defines the value of the deceleration (MD32) after the break.

Special situations:

- If the programmed velocity vprog is less than the breakpoint velocity (MD30), there is no break during deceleration.
- The same naturally applies if the programmed velocity cannot be reached because the positioning path is too short or a reduced override is used.

EQUATION

0 < MD30 < MD23 (traversing velocity – maximum)

MD32 > MD19 (deceleration)

Setting the breakpoint for the deceleration phase:

- The deceleration process normally starts with a flat ramp (MD29), in order to prevent compression of the material.
- When the velocity falls below MD30, the deceleration (MD32) is increased, in order to reach the target position rapidly.

This subdivision of the deceleration phase of the traversing curve is the normal solution for rapid transport with low wear on the material. However, it is possible to use any other combination by changing the values in the machine data.

NOTE

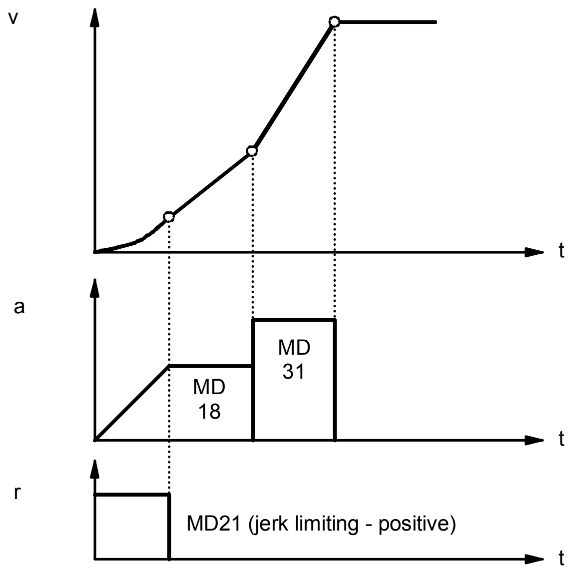
The time override affects machine data MD19 and MD32 but not the breakpoint MD30.

Jerk Limiting – Positive

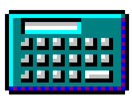
You only need to change these machine data if you use axis type (MD1) roll feed and require jerk limiting.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
21				Jerk limiting – positive (U501.21)	0 = inactive 1	999 999	1000*LU/s³	0

In order to increase the resolution of the traversing curve even more when moving the axis into position, you can use the jerk limiting to implement the actual acceleration as a ramp. Because the acceleration (MD18) is not effective immediately, but increases continuously as a ramp, a particularly smooth start is produced.



If a value of 0 is defined for "jerk limiting – positive", no jerk limiting is active. A small value causes a very high degree of rounding in the start-up phase and thus a large jerk limiting effect. The larger the value, the smaller the degree of rounding and thus the smaller the jerk limiting effect.



EQUATION for normalization to 1 µm:

MD21 [mm/s³] = MD18 [mm/s²] / t [s]

Special situation when using the acceleration breakpoint:

If the value of "jerk limiting – positive" is so small that the acceleration breakpoint (MD29, MD31) is reached before the acceleration value (MD18), the breakpoint is shifted upwards until the acceleration value (MD18) has been reached. The system then switches to the acceleration after the breakpoint (MD30).

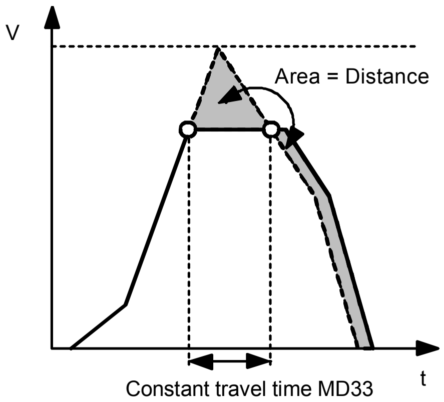
NOTE

Jerk limiting is only effective in positioning modes, not in setup mode. The time override has no impact on the jerk limiting value.

Constant Travel Time

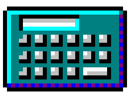
MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
33				Constant travel time (U501.33)	0.001 0 = inactive	99.999	s	0

With very short feed lengths and high traversing velocities, the roll feed axis does not reach the programmed end velocity vprog. This situation is known as delta mode. The sudden immediate transition from acceleration to deceleration places very high demands on the mechanical equipment and the motor.



The definition of a constant travel time transforms the delta-shaped traversing curve into a trapezium. The constant travel time defines the length of time the axis traverses at constant velocity.

Capping the delta tip produces a lower end velocity. Since the distance to travel remains the same, the positioning time is increased.



EQUATION

MD33 ~ 2 to 200 ms

1.14 Special Machine Data for the Roll Feed Version

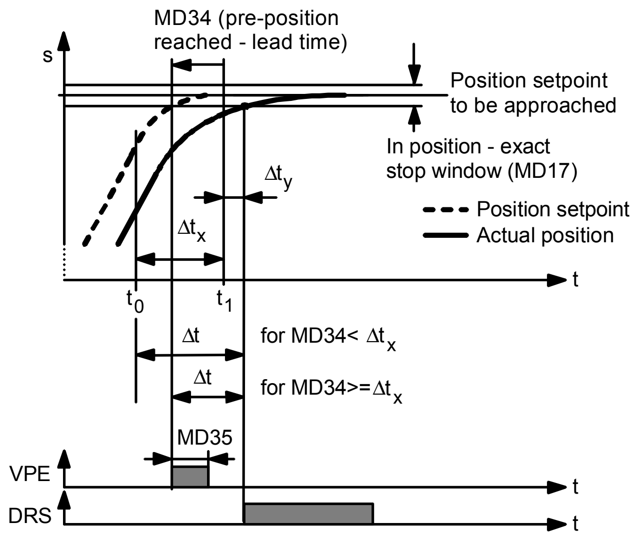
You only need to change these machine data if you use the axis type (MD1) roll feed and are not satisfied with the default settings.

Pre-position Reached – Lead Time and Output Time

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
34				Pre-position reached – lead time (U501.34)	0.001 0 = inactive	99.999	s	0
35				Pre-position reached – output time (U501.35)	0.001 0 = inactive	99.999	s	0

In normal situations, the checkback signal "destination reached, axis stationary [DRS]" is used as the condition for activating the next machining process. For high-speed movement sequences, however, it is necessary to initiate the machining process (e.g. actuation of shears) in advance, in order to achieve high cycle counts.

To use the advance initiation, you define the output time of the "pre-position reached" signal (programmable digital output MD48) by entering the setting in the "pre-position reached - lead time" machine data.



t_0 The deceleration start point is the earliest possible time of actuation of the "pre-position reached" signal (VPE).

t_1 The interpolator setpoint has reached the end position to be approached (digital 0). This is the latest possible time of actuation of the "pre-position reached" signal (VPE).

NOTE

The time Δt_y depends on the position controller gain factor and the magnitude of the "in position – exact stop window" (MD17). This yields a the time between "pre-position reached" and "destination reached, axis stationary [DRS]".

EQUATION

$\Delta t = MD34 + \Delta t_y$ for $MD34 < \Delta t_x$

$\Delta t = \Delta t_x + \Delta t_y$ for $MD34 >= \Delta t_x$

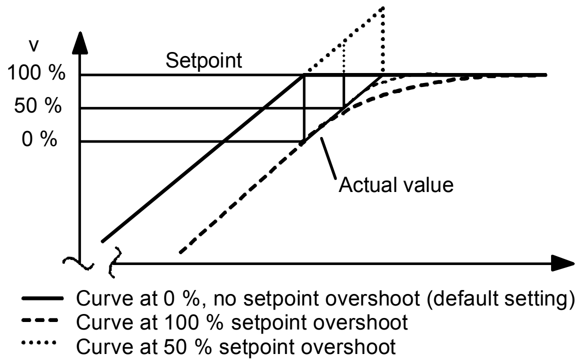
With "pre-position reached - lead time", the time of output of the "pre-position reached" signal with reference to t_1 can now be brought forward. If a value $\geq \Delta t_x$ is entered for "pre-position reached – lead time", the "pre-position reached" digital output is actuated at t_0.

With "pre-position reached - output time" (MD35), the duration of the "pre-position reached" digital output can be defined. "Pre-position reached" is canceled by the time of the next positioning movement at the latest.

Acceleration Overshoot

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
36				Acceleration overshoot (U501.36)	0	100	%	0

The acceleration response of the roll feed can also be controlled using an acceleration setpoint overshoot. This enables the actual velocity to be reached faster.



No acceleration overshoot, MD36 = 0 %:

As soon as the velocity setpoint has reached the programmed velocity, the approach of the actual velocity to the setpoint velocity starts to flatten out because of the proportional-action response of the position controller. The actual velocity reaches the velocity setpoint correspondingly late.

Maximum acceleration overshoot, MD36 = 100 %:

The velocity setpoint continues to increase after reaching the programmed velocity until the actual velocity has reached the programmed velocity. The excessive velocity setpoint is now reduced abruptly to avoid an overshoot in the actual velocity.

Any value between 0 % and 100 % is possible, however an actual value overshoot can arise with values approaching 100 %, depending on the control quality of the roll feed motor. The optimum value must be determined by trial and error.

NOTE

An acceleration overshoot is only output in MDI, automatic and single-block modes if the programmed velocity is reached, i.e. if a transition to the constant velocity takes place.

Response after Abort

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
37				Response after abort (U501.37)	0: Standard response 1: Approach last target position with no evaluation of direction of movement 2: Approach last target position with evaluation of direction of movement		-	0

The basic operating response of the roll feed after a positioning abort can be defined with "response after abort".

In the default setting (MD37 = 0), the positioning operation always starts again from the beginning after a restart following a positioning abort.

The setting "approach last target position with no evaluation of direction of movement" (MD37 = 1) causes the positioning operation to be continued after a restart following a positioning abort. The roll feed does not check whether the direction of movement has changed.

The setting "approach last target position with evaluation of direction of movement" (MD37 = 2) causes the positioning operation to be continued after a restart following a positioning abort. The direction of movement is evaluated, and if the roll feed detects a change in direction (e.g. because the motor crossed the target position during coastdown), the operation is aborted with an appropriate warning number.

1.15 Machine Data for Backlash Compensation

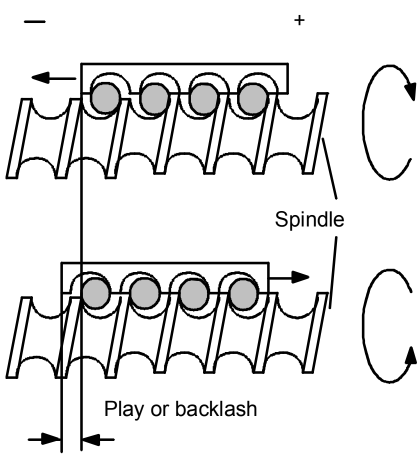
You only need to change these machine data if backlash compensation is required.

Backlash Compensation

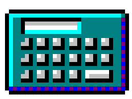
MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
38				Backlash compensation (U501.38)	0	9.999	1000*LU	0

The "backlash compensation" machine data is used to compensate for mechanical backlash. When the direction of movement is reversed on systems with an indirect measuring system (position encoder on the motor), the axis traverses through the distance of the mechanical backlash before moving in the opposite direction. Positioning errors are a consequence.

Examples for determining the backlash:



Move the axis in setup mode through a distance in the positive direction which must be greater than the backlash. Then attach a dial gauge to the mechanical equipment so that the deflection takes place in the negative traversing direction. Now, make a note of the present actual position value of the axis. In MDI mode, define a position of -1 µm in incremental dimensions (G91) and any velocity value. Now, start the axis and move it through 1 µm in the negative direction until an axis movement can be detected on the dial gauge. The magnitude of the backlash is determined by calculating the difference between the actual value now shown and the actual value you noted before.



EQUATION for µm normalization:

MD38 [mm] = old actual value [mm] – new actual value [mm]

Backlash compensation is not required (MD38 = 0):

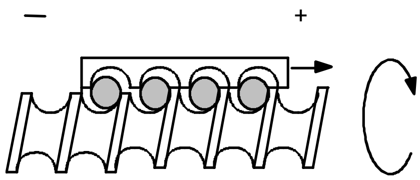
- With direct position measuring systems
- Where the backlash < accuracy requirements

Backlash Compensation – Preferred Position

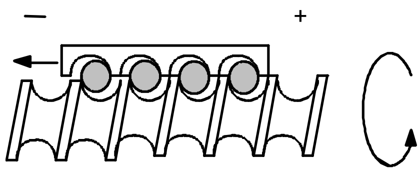
MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
39				Backlash compensation – preferred position (U501.39)	1: Preferred position positive (no backlash compensation is calculated during first positive traversing movement) 2: Preferred position negative (no backlash compensation is calculated during first negative traversing movement)		-	1

Where the axes use incremental encoders, a preferred position for the backlash is reached automatically when the control is switched on and the reference point has been approached. With absolute encoders, however, it is necessary to define the preferred position with "backlash compensation – preferred position".

Preferred position positive MD39 = 1:
No backlash is calculated during the first positive traversing movement after the control is switched on.



Preferred position negative MD39 = 2:
No backlash is calculated during the first negative traversing movement after the control is switched on.



Backlash compensation (MD38) is performed on every further reversal in the direction of movement.

NOTE

If there is no preferred position after the control is switched on, positioning errors cannot be avoided. Backlash compensation is not possible. You must enter 0 in MD38.

Backlash Compensation – Velocity Limitation

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
40				Backlash compensation – velocity limitation (U501.40)	1 0 = inactive	999	1000*LU/min	999

If you define a value of 0 for "backlash compensation – velocity limitation", the backlash is performed using an automatically adapted velocity:

V_{ulk} = MD18*15/62

This is generally totally adequate, however if you need to compensate for very large backlashes, it may be necessary to reduce the velocity.

The value in "backlash compensation – velocity limitation" defines the velocity used to traverse the backlash.

1.16 Machine Data for Speed-Controlled Operation

You only need to change these machine data if you are not satisfied with the default settings.

Speed control is used instead of position control when operating the axis in "reference point approach" and "control" modes. The acceleration and deceleration ramps for the speed controller are set using machine data 41 and 42.

Acceleration Time

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
41				Acceleration time, operating mode "reference point approach / control" (U501.41)	0.001 0 = inactive	99.999	s	1.000

If you define a value of 0 for "acceleration time, operating mode "reference point approach / control"", the speed controller accelerates abruptly to the speed setting. If you define a value greater than 0 for "acceleration time, operating mode "control"", the specified time value defines the time taken to accelerate to the rated speed.

Deceleration Time

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
42				Deceleration time, operating mode "reference point approach / control" (U501.42)	0.001 0 = inactive	99.999	s	1.000

If you define a value of 0 for "deceleration time, operating mode "reference point approach / control"", the speed controller decelerates abruptly to zero speed. If you define a value greater than 0 for "deceleration time, operating mode "control"", the specified time value defines the time taken to decelerate from the rated speed to zero.

1.17 Machine Data for Deceleration Time During Errors

You only need to change these machine data if you are not satisfied with the default setting.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
43				Deceleration time during errors (U501.43)	0.001 0 = inactive	99.999	s	0

The position controller shuts down abruptly in the event of certain technology warnings (e.g. following error monitoring) and if the "follow-up mode" [FUM] control signal is activated. This situation would cause the axis to stop equally abruptly.

To protect the mechanical equipment in these situations, you can define a deceleration ramp for the speed controller with "deceleration time during errors".

- ♦ If you define a value of 0 for "deceleration time during errors", the axis stops abruptly under any of the above conditions.
- ♦ If you define a value greater than 0 for "deceleration time during errors", the specified time value defines the time taken to decelerate from the rated speed to zero.

1.18 Machine Data for External Block Change

You only need to change these machine data if you use the "external block change" function and are not satisfied with the default setting.

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
44				External block change – setting (U501.44)	0: Warning at end of NC block 1: No warning at end of NC block		-	0

The "external block change" function can be activated in NC programs, and is triggered by a programmable digital input (MD45).

"External block change – setting" can be used to define the response if the digital input is not actuated.

- "External block change – setting" = 0: NC program execution is aborted at the end of the NC block with the "external block change" function, and a warning [WARN] is generated.
- "External block change – setting" = 1: The program automatically jumps to the next NC block at the end of the block. NC program execution is not aborted.

Further Information

The "external block change" function is described in Chapter 3.10.7 in the Programming Guide.

1.19 Machine Data for Configuration of Digital Inputs and Outputs

You only need to change these machine data if you want to use digital inputs and outputs.

In various applications, it is necessary to route actuation or checkback signals directly to digital inputs or outputs, instead of across a bus system, in order to achieve the fastest possible response times.

The "high-speed inputs/outputs" are programmed by entering a numeric value in the corresponding machine data. The numeric value is created by joining the function number(s) to the number of the digital input or output.

The number of the digital input or output is represented by a power of ten, as shown below, where n represents the function number:

Input/Output	Power of Ten
I1 / O1	n
I2 / O2	n0
I3 / O3	n00
I4 / O4	n 000
I5 / O5	n0 000
I6 / O6	n00 000

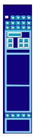
Example:

The axis is to be equipped with the functions "start ORed" and "set actual value on-the-fly". The external signal is to be connected to digital input I5 for the "start ORed" function (n = 1) and to digital input I4 for the "set actual value on-the-fly" function (n = 3). In addition, the Bero (n = 7) for reference point approach is to be connected to digital input I6:

Input	Power of Ten	*	Factor	=	Subtotal
I1	1	*	0	=	0
I2	10	*	0	=	0
I3	100	*	0	=	0
I4	1 000	*	3	=	3000
I5	10 000	*	1	=	10000
I6	100 000	*	7	=	700000
Numeric value (subtotal tally)					713000

A value of 713000 must be entered in machine data 45.

NOTE There must be no duplicate assignments when assigning the digital inputs and outputs with MD45 to MD48.



Assignment of digital inputs/outputs on the MASTERDRIVES MC:

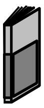
Six possible connections are located on connector X101 (control terminal). Four of these can be used as either a digital input or a digital output, and two can be used as a digital input only. The following standard configuration is implemented for the technology functions by means of a standard parameter set:

Terminal Name	Used for
X101 / 3	A1
X101 / 4	A2
X101 / 5	A3
X101 / 6	E4
X101 / 7	E5
X101 / 8	E6

NOTE

Please note the following when determining the values for the "digital inputs" and "digital outputs" machine data:

- In the standard configuration, outputs O1 to O3 and inputs I4 to I6 are used
- If a reference point cam (Bero) is required, it must be connected to I6 (standard configuration). This configuration is valid for reference point approach "with Bero and zero mark" and reference point approach "with zero mark only".
  Generally, the following is applicable: the same input must be parameterized as the input for the reference point cam as is parameterized in MASTERDRIVES as the rough impulse for the position logging for the motor sensor (P178).
- One of the interrupt-capable inputs I4 or I5 must be used for the following functions:
  - "In process measurement"
  - "External block change"
  - "Set actual value on-the-fly"
  - "Set floating reference point"
  - "Position correction via registration mark"
  - Bero for reference point approach, when the reference point approach is parameterized as "with Bero only" (MD8 = 1)



Further Information

The note above referring to the assignment of digital inputs/outputs applies to the standard circuit. If you choose a different circuit, you thereby define a different assignment of digital inputs/outputs. Please follow the instructions in the documentation "MASTERDRIVES MC Compendium".

CAUTION

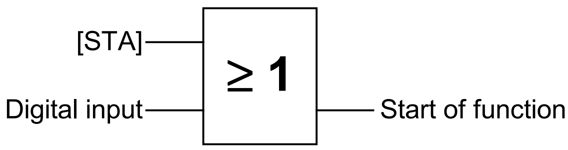
The digital inputs I4 and I5 can be parameterized to trigger interrupts (P647.1 and P648.1).

This means that the functions "in process measurement", "set actual value on-the-fly", "external block change", "set floating reference point " and "Bero for reference point approach" (reference point approach "with Bero only") must not be used in combination!

Digital Inputs – Function 1

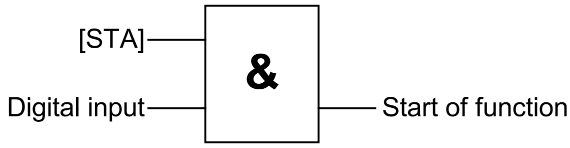
MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
45				Digital inputs – function 1 (U501.45)	0: No function 1: Start ORed 2: Start ANDed 3: Set actual value on-the-fly 4: External block change 5: In process measurement 6: Collision 7: Bero for reference point approach 8: Reference point cam for reference point approach 9: Read-in enable, externally programmable		6-digit input	0

"Start ORed" function (1):



The control signals of the axis include the "start" signal [STA], which triggers a positioning operation or function in MDI, automatic and single-block modes. The digital input with the "start ORed" function is ORed with the "start" [STA] control signal. The [STA] signal and the digital input can start a function independently of each other. To stop the function, you must cancel both [STA] and the digital input.

"Start ANDed" function (2):



[STA] and the digital input can only start a function together. To stop the function, you only need to cancel one of the signals.

NOTE

Only one of the digital inputs with the "start ANDed" or "start ORed" function can be used on the same axis.

"Set actual value on-the-fly" function (3):

The "set actual value on-the-fly" function can be used in NC programs to reset the actual value system temporarily by actuating the digital input. To do this, you need to use one of the interrupt capable inputs, I4 or I5.

Further Information

The "set actual value on-the-fly " function is described in Chapter 3.10.8 in the Programming Guide.

"External block change" function (4):

The "external block change" function can be used in NC programs to trigger an NC block change by actuating the digital input. To do this, you need to use one of the interrupt capable inputs, I4 or I5.

Further Information

The "external block change" function is described in Chapter 3.10.7 in the Programming Guide.

"In process measurement" function (5):

The "in process measurement" function is used to save the current actual value on actuation of the high-speed input. The user can read out the value for further processing. To do this, you need to use one of the interrupt capable inputs, I4 or I5.

"Collision" function (6):

The "collision" function can be used in NC programs to stop the traversing movement of an axis or axes rapidly when the collision monitoring is active and the digital input is deactivated. The deceleration for collision is defined in MD20.

Further Information

The "collision monitoring" function is described in the Programming Guide in Chapter 3.9.

"Bero for reference point approach" function (7):

The "Bero for reference point approach" function is used to connect the reference point cam required for synchronization of the measuring system to a digital input.

"Reference cam point for reference point approach" function (8):

The function "reference cam point for reference point approach " stores the reverse cams required for the synchronization of the measurement system to a digital input.

Further Information

The "reference cam point for reference point approach" function is only available from MASTERDRIVES MC firmware version >=V1.4x.

Reference point approach is documented in Chapter "Reference point approach".

"Read-in enable, externally programmable" function (9):

The "read-in enable, externally programmable" function can be used in NC programs to trigger the interruption and continuation of NC program execution with a digital input.

Further Information

The "read-in enable, externally programmable" function is described in the Programming Guide in Chapter 3.10.9.

Digital Inputs – Function 2

The expansion of the digital inputs can only be used in conjunction with axis type (MD1) roll feed on MASTERDRIVES MC.

In order to also be able to freely configure the digital inputs of MASTERDRIVES MC on the SIMATIC Motion Control, the "trigger signal inserter/ejector" and "synchronize traverse table" functions are available.

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
46				Digital inputs – function 2 (U501.46)	0: No function 1: Disable actual value 2: External read-in enable 3: External read-in enable ANDed 4: Set floating reference point 5: No function 6: No function 7: No function 8: Trigger signal inserter / ejector OR connected 9: Synchronize traverse table OR connected		Eingabe 6-stellig	0

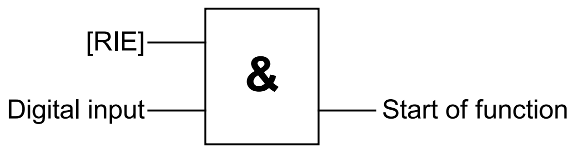
"Disable actual value" function (1):

The "disable actual value" function is used to suppress actual value measurement during the actuation of a digital input, in order to deactivate the position control system during "ventilation" of the feed rollers.

"External read-in enable" function (2):

The "external read-in enable" function is used to trigger positioning of the roll feed through the actuation of the high-speed input. The external read-in enable operates in the same way as the read-in enable control signal [RIE].

"External read-in enable ANDed" function (3):



The only difference from "external read-in enable" is that the read-in enable control signal [RIE] is ANDed with the digital input. [RIE] and the digital input can only start a function together. The function can be stopped by canceling one of the signals.

"Set floating reference point" function (4):

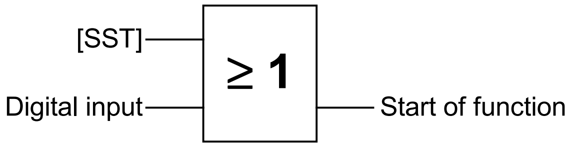
The "set floating reference point" function means that, in any of the different modes, the actual value system can always be reset by controlling the digital input. To do this, you need to use one of the interrupt capable inputs, I4 or I5.

Further Information

The "set floating reference point" function is only available from MASTERDRIVES MC firmware version >=V1.4x.

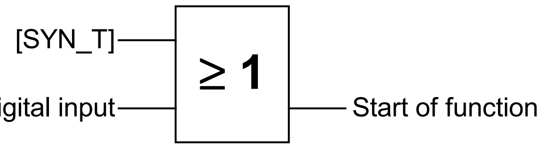
The exact description of the function is documented in the respective chapters of the operating mode description under a separate point "Set floating reference point".

"Trigger signal for inserter / ejector mode OR connected" function (8):



The MASTERDRIVES MC digital input which displays the trigger signal for insertion mode or ejection mode is OR connected with the trigger signal for insertion / ejection mode [SST]. The digital input and the control signal can start the function with equal priority.

"Synchronize traverse table OR connected" function (9):



The MASTERDRIVES MC digital input which displays the external signal for the synchronization of the traverse table is OR connected with the synchronize traverse table signal [SYN_T]. The digital input and the control signal can allocate the current position of the master axis to the beginning of the traverse table with equal priority.

Digital Outputs – Function 1

Further Information

The function of the checkback signals is described in detail in the chapter entitled "Control and Checkback Signals".

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
47				Digital outputs – function 1 (U501.47)	0: No function 1: Destination reached, axis stationary [DRS] 2: Axis travels forwards [FWD] 3: Axis travels backwards [BWD] 4: M change from M97 5: M change from M98 6: Start enable [ST_EN]		6-digit input	0

"Destination reached, axis stationary" function (1):
The digital output is actuated when the axis has reached the programmed target position. The signal state matches the checkback signal "destination reached, axis stationary" [DRS].

"Axis travels forwards" function (2):
The digital output is actuated when the axis moves forwards. The signal state matches the checkback signal "axis travels forwards" [FWD].

"Axis travels backwards" function (3):
The digital output is actuated when the axis moves backwards. The signal state matches the checkback signal "axis travels backwards" [BWD].

"M change from M97" function (4):
The digital output is actuated when M function number 97 occurs during NC program execution. The signal state matches the checkback signal "M change signal" [STR_M] with M97.

"M change from M98" function (5):
The digital output is actuated when M function number 98 occurs during NC program execution. The signal state matches the checkback signal "M change signal" [STR_M] with M98.

"Start enable" function (6):
The digital output is actuated when the axis is ready to start. The signal state matches the checkback signal "start enable" [ST_EN].

Digital Outputs – Function 2

The digital output extension is only used in association with axis type (MD1 = 3) roll feed.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
48				Digital outputs – function 2 (U501.48)	0: No function 1: Constant travel 2: Acceleration 3: Deceleration 4: Acceleration or deceleration 5: Pre-position reached 6: encoder monitoring signal		6-digit input	0

"Constant travel" function (1):
The digital output is actuated while the roll feed is in the constant travel phase.

"Acceleration" function (2):
The digital output is actuated while the roll feed is in the acceleration phase.

"Deceleration" function (3):
The digital output is actuated while the roll feed is in the deceleration phase.

"Acceleration or deceleration" function (4):
The digital output is actuated while the roll feed is in the acceleration or deceleration phase.

"Pre-position reached" function (5):
The digital output is actuated when the roll feed has reached the pre-position (see also MD34 and MD35).

"Encoder monitoring signal" (6):
The digital output is actuated if the difference between the actual value of encoder 1 minus the actual value of encoder 2 is greater than the function parameter FP6.

1.20 Machine Data for Feedforward Control

Feedforward Control – Speed

You only need to change these machine data if you require feedforward speed control.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
49				Feedforward control – speed (U501.49)	0	150	%	0

Feedforward Control – Acceleration

You only need to change these machine data if you require feedforward acceleration control.

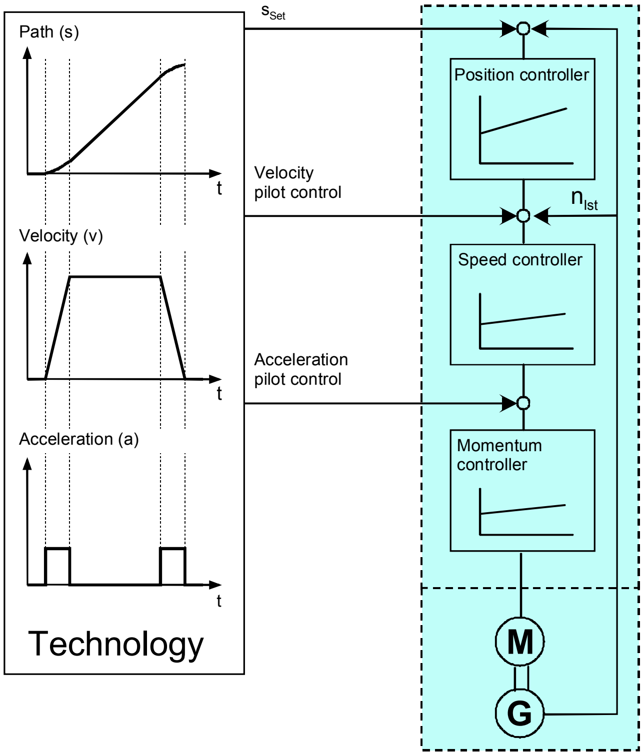
MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
50				Feedforward acceleration control (U501.50)	1 0 = inactive	99 999	1000*LU/s²	0

These machine data define the intensity of the feedforward acceleration control.

- 0 means there is no feedforward acceleration control and/or deactivated feedforward acceleration control
- 1 to 150: percentage value of feedforward acceleration control
- 1 to 99 999: 1000*LU/s² of feedforward acceleration control

It is generally necessary to compensate for the following error in particular in association with synchronization functions.

How it works:



The technology makes the position setpoint available to the position controller. The speed setpoint for the speed controller is generated by calculating the difference between the position setpoint and the actual position value (position difference = following error). Without feedforward control, the actual position value would thus always lag behind the position setpoint by an amount equal to the following error. Different following errors would be produced as a result of different velocities.

In order to prevent this from occurring, the technology generates a feedforward control value which is transmitted directly to the speed controller, bypassing the position controller.

The same relationship is also applicable to the formation of the momentary setpoint by the speed controller. Only if there is a deviation from the rules in the speed controller is a momentary setpoint calculated using the parameters Kp and Tn.

The address behavior, and thus the whole control dynamics, can be improved using the acceleration pre-control.

Example:

50 % of the rated velocity (rated speed) is required for traversing, where the rated velocity v = 10000 mm/min and the position controller gain KV = 1 mm/min/µm.

Without feedforward control, the position controller must actuate the speed controller at 50 %. To achieve this, the position controller requires a position difference of 0.5 * (v/KV) = 5000 µm.

If 100 % feedforward control is used, the technology already actuates the speed controller at 50 %. The speed setpoint from the position controller is now zero. This means that the position difference is also zero and traversing is performed without a following error. In this situation, the task of the position controller is now to compensate for load fluctuations and to ensure that the positions are approached accurately.

1.21 Machine Data for IM178

Actual Value Weighting

These machine data is used to evaluate the actual position value.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
51				Actual value weighting mode	0: Evaluation using AVWF pre-point / post-point		-	0
52				Actual value weighting factor Pre-point	0	999	-	1
53				Actual value weighting factor Post-point	0	99 999 999	-	0

The setting 0 in machine data number 51 means that the pre-point/post-point mode is used to evaluate the actual position value.

Using the actual value weighting factor, the actual position value measured can be converted to another reference system. The current actual value evaluation factor is given by the total of MD52/53.

NOTE

If the pre-point/post-point setting is selected for the AVWF, the actual value weighting factor is accurate to 8 decimal places.

Tolerance Window for SSI Sensor Monitoring

These machine data serve to monitor the serial position encoder.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
56				Tolerance window for SSI sensor monitoring	1 0 = inactive	99 999 999	Increments	0

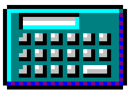
The data from the serial position encoder is checked for validity on a cyclical basis. With each read action, the current actual value change is compared with the permissible actual value change (tolerance window for SSI monitoring). If the actual value change is not within the allowable range (position encoder error, data connection problems), the faulty actual value is ignored, and an actual value appropriate to the set speed is taken. If four cycles in a row produce unacceptable actual values for the position encoder, the axis is brought to a halt and a warning is given (warning number 233).

Kv Factor (Positioning)

These machine data serve to set the position control circuit amplification for the positioning modes.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
57				Kv factor (positioning)	0.100	20.000	1000*LU/ min/LU	1.000

The position control circuit amplification is known as the Kv factor.



EQUATION

Kv factor = velocity v / following error s

The Kv factor must be high to ensure good dynamics in the position control circuit. However, if the Kv factor is too high, it leads to instability, overshooting and even to excessive machine strain. The maximum allowable Kv factor depends on the design and / or the speed of the drives (rise time, acceleration and braking power) and the quality of the machine.

In practice, experience shows that values between 0.8 and 1.5 are most appropriate for serial machines.

Drift Compensation

The last stage in setting up the drive is to carry out drift compensation for the speed controller. However, closing the position control circuit can result in renewed drift. This can be compensated using MD58, so MD58 is effectively for fine tuning.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
58				Drift compensation	-9999	+9999	LU	0

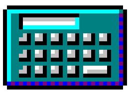
The position control circuit must be closed for the level of drift to be determined (e.g. set-up mode; axis not moving, control enable switched on). Then the following error can be read out using the command "read out status data".

For a well balanced axis, the following error should oscillate about zero. However, if a one-sided position is determined, e.g. following error +15µm, the drift has to be compensated by -15µm. If you check the following error again, you should find that it now oscillates about zero.

Lower Limit for Dynamic Following Error Monitoring

While an axis is in motion, the following error is monitored on a dynamic basis, i.e. the axis uses the Kv factor set and the speed of travel to determine the allowable following error at that point.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
59				Lower limit for dynamic following error monitoring	1000	999 999	LU	1000



EQUATION

MD59 = 1,5 * v_{set} / v_{max} * max. following error

Where max. following error = max. traverse speed / Kv factor

NOTE

The lower limit for the dynamic following error monitoring is only applicable for positioning, and not for synchronization, control or reference point approach.

The lower limit for the dynamic adjustment is defined as a set parameter, MD59.

The calculated following error limit can be read using task HA3, identifier 9.

System Adjustment Setpoint Output

You have to set the direction in which your axis moves, i.e. which direction should be seen as positive and which as negative. The next stage is to adjust the polarity of the setpoint voltage and the counting direction for the position encoder to the direction of motion using MD60.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
60				System adjustment setpoint output	0: no inversion, positive speed → positive output voltage 1: inversion of output		-	0

Depending on the observations you have made, e.g. in setup mode and jogging mode [J_FWD], you get the following inputs for system adjustment (MD60):

- ♦ positive direction of movement MD60 = 0
- ♦ negative direction of movement MD60 = 1

Output Voltage at Maximum Velocity

Although the digital analog converter of the IM178 can output voltage of up to ±10 V, the drive controller connected should be balanced at a lower rated voltage so that the maximum traverse speed can be achieved at the output voltage parameterized for MD61.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
61				Output voltage at maximum velocity	0.5	10.000	Volt	10.000

It is worth balancing the drive controller at somewhere between 8 V and 9.5 V in order to ensure that there is enough reserve in the additional voltage up to 10 V.

Setpoint Output during Simulation

You can use machine data number MD62 to select whether to output a setpoint during simulation or not.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
62				Setpoint output during simulation	0: Setpoint output off (0 Volt) 1: Setpoint output on		-	0

Parameterization of Onboard Outputs

Six parameterizable outputs are available for each axis for controlling the technology functions. Two of these outputs can be allocated to the free outputs Q0 and Q1 on the IM178 by parameterizing the machine data number MD63 and MD64.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
63				Parameterization of output Q0	0: Output Q0 not used 1..6: Output O1..O6		-	0
64				Parameterization of output Q1	0: Output Q1 not used 1..6: Output O1..O6		-	0

Disable Hardware Monitoring

If MD 65 = 0, then all monitoring systems are active. The masking of the error messages is bit-oriented.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
65				Disable hardware monitoring (IM178)	Bit 0: Sensor signal cable error Bit 1: Signal error Bit 2: Zero mark Bit 3: SSI sensor validity error		-	0

Mit Bit 0 = 1 disables the sensor signal cable error (F230),
Bit 1 = 1 disables the signal error (F231),
Bit 2 = 1 disables the zero mark error (F232),
Bit 3 = 1 disables the SSI sensor validity error (F233).

Example:
Bit 3 needs to be set to disable the SSI sensor validity error.

Kv Factor (Synchronization)

This machine data is used for setting the position control circuit gain for the synchronization mode.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
66				Kv factor (synchronization)	0.100	20.000	1000*LU/ min/LU	1.000

The position control circuit gain is designated the kv factor.

EQUATION

Kv factor = velocity v / following error s

Further Information

See also the description of machine data MD57.

DAC Limitation Factor

The DAC limitation factor is used for adapting the position controller to the various parameters of the control circuit. In addition to the internal resolution of the measurement system the following quantities are decisive for calculating this factor:

- Maximum traverse speed
- Kv factor positioning
- Output voltage at v-max

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
67				DAC limitation factor	1 0 = inactive	100	-	0

The technology calculates the traversing curve of the axis for a traverse movement and makes this value available as a setpoint. By means of the actual value of the position encoder the axis is informed of the current position of the mechanical system.

Every deviation between setpoint and actual value is processed immediately as a following error. After multiplying with the DAC factor, this deviation is made available to the digital-to-analog converter as a DAC value. The resultant voltage causes the drive controller to compensate the difference between setpoint and actual value and the control circuit will then close again. The DAC factor is used to adapt the position controller to the various parameters of the control circuit and should not be greater than 10.

EQUATION

$$\text{DAC factor} = \frac{30440 * MD57 * MD61}{10000 * MD23}$$

The following applies:

MD 23: Maximum traverse speed [mm/min]

MD 57: Kv factor [mm/min/µm]

MD 61: Output voltage at v-max [mV]

NOTE

The DAC limitation factor is only valid for the positioning modes.

1.22 Function Parameters

1.22.1 Parameters for Setting Floating Reference Point

Inner and Outer Window

FP-No.	FW-Stand		Name	Input Limit		Unit	Default
	MC	M7		Lower	Upper		
2	V1.4	V1.00.48	Set floating reference point, position correction inner window (window 1) (U504.02)	-2 147 483 648 0: deactivated	2 147 483 647	LU	0
3	V1.4	V1.00.48	Set floating reference point, position correction outer window (window 2) (U504.03)	-2 147 483 648 0: deactivated	2 147 483 647	LU	0

Two windows can be parameterized to ignore discrepancies which are too large or to avoid unnecessary intervention.

The inner window (window 1) defines the minimum discrepancy which has to occur before a correction has to be made. Discrepancies which are lower than the amount set are ignored and no corrections are made.

The outer window (window 2) defines the maximum allowable discrepancy for which a correction is made automatically. If the discrepancy is greater than the amount set, no correction is made. At the same time, the error is reported by means of the output "Error – reference point outside window2" (B829). The output remains until the value is next logged.

Parameterizing the value as 0 switches the window evaluation system off.

The following conditions should be taken into consideration for the window evaluation system:

- ♦ The windows are only evaluated if the axis is referenced (Signal ARFD B361 = 1). If the axis is not referenced, every discrepancy is evaluated.

NOTE

If using absolute value detectors, the ARFD signal is always "1".

- ♦ If the inner window is parameterized such that it is larger than the outer window, the values will never be corrected. The error output will be set as a function of the size of the discrepancy.

Correction Mode, Set Floating Reference Point

FD No.	FW Version		Name	Input Limit		Unit	Default
	MC	M7		Lower	Upper		
4	V1.4	-	Correction mode, set floating reference point (U504.04)	0: correction to shortest path 1: positive correction only 2: negative correction only		-	0

The calculation of the correction value depends on the axis type:

- For a linear axis, the correction value is always the absolute difference between the reference point coordinate and the measured value.
- For a rotary axis, the calculation is controlled by the parameter U504.04 (FP4).

If the value = 0, the correction value is always the shortest path. If the value = 1, the correction is always positive, and if the value = 2, it is always negative.

1.22.2 Parameters for Limit Monitoring – Encoder Changeover

Limit Monitoring – Encoder Changeover

FD No.	FW Version		Name	Input Limit		Unit	Default
	MC	M7		Lower	Upper		
6	V1.41	-	Limit monitoring – encoder changeover (U504.06)	1	999 999	LU	2000

If the encoder changeover function is used, the difference between the actual value of encoder 1 and the actual value of encoder 2 is compared to the function parameter FP6. If the difference is greater than parameterized in function parameter FP6, alarm 139 is issued.

EQUATION

Function parameter FP6 ≥ actual value of encoder 1 – actual value of encoder 2

1.22.3 Parameters for Jerk limiting Rounding Time Constant

Name	Input Limit		Unit	Default
	Lower	Upper		
Jerk limiting – rounding time constant (U505)	0	1000	ms	0

This parameter sets the rounding time constant for positioning. With increasing rounding, the speed curve becomes more even, which protects the mechanical system.

NOTE Input and output is via MCT parameters.

2 Control and Checkback Signals

Contents

In this chapter you will find a description of the control and checkback signals.

Related Terms

Control bytes, control bits, checkback bytes, checkback bits, commands, feedback

Overview

Control signals are used to issue commands to the axis. Checkback signals provide the user with information on the current machining status. The interface for the control and checkback signals is the GMC_DB_CMD.

This chapter describes the individual control and checkback signals with general reference to all modes. The individual modes and their associated control and checkback signals are described in the following chapters.

2.1 Overview of Control Signals

Definition

Control signals are signals which are transmitted cyclically from the application interface (S7) to the technology. The control signals are used to issue commands to the technology. The control signal references on MASTERDRIVES MC are shown in square brackets in the section with the abbreviations.

Interface

The overview below shows the interface for the control signals.

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx	RES	RES	RES	RES	RES	LB	RES	RES	BIN	IN_1
DBBx+1	ACK_F	RES	RES	RES	ENC	OFF3	OFF2	OFF1	BIN	IN_2
DBBx+2	MODE_IN				J_FWD	F_S	J_BWD	BLSK	BIN	IN_3
DBBx+3	OVERRIDE								DEC	IN_4
DBBx+4	PROG_NO ODER MDI_NO								DEC	IN_5
DBBx+5	SIST	RST	FUM	ACK_M	CRD	STA	RIE	TGL_I	BIN	IN_6
DBBx+6	R_VM	S_VM	EN_RF	SSC	OPERATION		FUNCTION		BIN	IN_7
DBBx+7	ST_VM	TABLE_NO			SYN_T	SST	ST_S	SET_T	BIN	IN_8
DBBx+8	CU_DR	CU_EN	CU_SP	SYNC	DI_RN	DI_RP	DI_JN	DI_JP	BIN	IN_9_0
DBBx+9	RES	RES	RES	RES	RES	RES	S_DSP	CU_TR	BIN	IN_9_1
DBWx+10	OPTIONAL VALUE 1 INPUT									IN_9_2
DBDx+12	OPTIONAL VALUE 2 INPUT									IN_10
DBDx+16	OPTIONAL VALUE 3 INPUT									IN_11

The DBB address x is calculated as follows:
x = (axis number-1) * 80 + 26

Meaning of the Control Signal Abbreviations

Control signals to the basic functionality of MASTERDRIVES MC:

LB	**L**ife **b**it [control word 1 bit 10]
ACK_F	**Ack**nowledgement **f**ault [control word 1 bit 7]
ENC	**En**able **C**ontroller [control word 1 Bit 3]
OFF3	OFF3 not activated [control word 1 bit 2]
OFF2	OFF2 not activated [control word 1 bit 1]
OFF1	OFF1 not activated [control word 1 bit 0]

Control signals for the positioning and synchronization technology:

MODE_IN	Mode in [U710.29 to U710.32]
J_FWD	**J**og **f**or**w**ar**d**s [U710.28]
F_S	**F**ast/**s**low [U710.27]
J_BWD	**J**og **b**ack**w**ar**d**s [U710.26]
BLSK	**Bl**ock **sk**ip [U710.25]
OVER	**Over**ride [U710.17 to U710.24]
PROG_NO	**Prog**ram number [U710.09 to U710.16]
MDI_NO	**MDI** number [U710.09 to U710.16]
SIST	**Si**ngle **st**ep [U710.08]
RST	**R**e**s**e**t** (technology) [U710.07]
FUM	**F**ollow **u**p **m**ode [U710.06]
ACK_M	**Ack**nowledge **M** function [U710.05]
CRD	**C**ancel **r**emaining **d**istance [U710.04]
STA	**Sta**rt [U710.03]
RIE	**R**ead **i**n **e**nable [U710.02]
TGL_I	**T**og**gl**e bit **i**n [U710.01]
R_VM	**R**eset **v**irtual **m**aster [U684.01]
S_VM	**S**et start value **v**irtual **m**aster [U684.03]
EN_RF	**En**able **ref**ering (enable setting of free reference point) [U675.02], available with MCT version V1.4 or higher
SSC	**S**tart/**s**top **c**ycle continious [U612.01]
OPERATION	Select **operation** [U602]
FUNCTION	Select **function** [U603]
ST_VM	**St**art **v**irtual **m**aster [U684.02]
TABLE_NO	NC table number [U650]
SYN_T	**Syn**chronize **t**able [U621]
SST	**S**tart/**s**top cycle **t**rigger [U612.01]
ST_S	**St**art **s**lave (under development)
SET_T	**S**et **t**able [U619]

Meaning of synchronization control signals (ADD_CTRL_SYNC_MODE, MCT ≥ V1.31 and SIMATIC MOTION CONTROL ≥ V1.00.48 only):

DI_JP	**Di**splacement **j**og **p**ositive [U696.01]
DI_JN	**Di**splacement **j**og **n**egative [U696.02]
DI_RP	**Di**splacement **r**elativ **p**ositive [U694.01]
DI_RN	**Di**splacement **r**elativ **n**egative [U694.02]
SYNC	Start **sync**hronisation (synchronize conductivity) [U676]
CU_SP	**C**atch **u**p **S**to**p** [U625.01]
CU_EN	**C**atch **u**p **en**able positioning [U625.02]
CU_DR	**C**atch **u**p **d**isable **r**amp (enable / disable acceleration / deceleration ramp) [U625.03]

Meaning of synchronization control signals (ADD_CTRL_SYNC_MODE, MCT ≥ V1.4x and SIMATIC MOTION CONTROL ≥ V1.00.48 only):

CU_TR	**C**atch **u**p **Tr**igger stand still position
S_DSP	**S**et **d**isplacement **s**et**p**oint [U672]

Optional extension of control signals:

OPTIONAL VALUE 1 to 3 INPUT

2.2 Overview of Checkback Signals

Definition

Checkback signals are signals which are transmitted cyclically from the technology to the application interface (S7). The technology uses the checkback signals to indicate its current machining status. The checkback signal references on MASTERDRIVES MC are shown in square brackets in the section with the abbreviations.

Interface

The overview below shows the interface for the checkback signals.

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy	RES	RES	RES	RES	OTM	OTC	OLC	S MAX	BIN	OUT_1
DBBy+1	RES	WA RN	OFF3	OFF2	FAU LT	IOP	RDY	RTS	BIN	OUT_2
DBBy+2	FAULT_NO								DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4
DBBy+4	STR_ M	ARFD	FUR_ VM	OTR	FUT	BWD	FWD	DRS	BIN	OUT_5
DBBy+5	M_NO_1								DEC	OUT_6
DBBy+6	MODE_OUT				FUR	ST_ EN	T_R	TGL _O	BIN	OUT_7
DBBy+7	M_NO_2								DEC	OUT_8
DBBy+8	CU_ TE	CU_ VR	CU_ PR	SYNC	DI_A	POS_ A	CL_A	VM_ RA	BIN	OUT_9_0
DBBy+9	RESERVIERT									OUT_9_1
DBWy+10	OPTIONAL VALUE 1 OUTPUT									OUT_9_2
DBDy+12	OPTIONAL VALUE 2 OUTPUT									OUT_10
DBDy+16	OPTIONAL VALUE 3 OUTPUT									OUT_11

The DBB address y is calculated as follows:
y = (axis number-1) * 80 + 54

Meaning of the Checkback Signal Abbreviations

Checkback signals from the basic functionality of MASTERDRIVES MC:

OTM	**O**ver**t**emperature **m**otor [status word 2 bit 26]
OTC	**O**ver**t**emperature **c**onverter [status word 2 bit 24]
OLC	**O**ver**l**oad **c**onverter [status word 2 bit 22]
SMAX	**Max**imum rotational **s**peed [status word 2 bit 18]
WARN	Warning [status word 1 bit 7]
OFF3	OFF3 not active [status word 1 bit 5]
OFF2	OFF2 not active [status word 1 bit 4]
FAULT	Fault [status word 1 bit 3]
IOP	**I**n **op**eration [status word 1 bit 2]
RDY	**R**ea**dy** [status word 1 bit 1]
RTS	**R**eady **t**o **s**tart [status word 1 bit 0]
FAULT_NO	(Fault number)
WARN_NO	(Warning number)

Checkback signals from the positioning and synchronization technology:

STR_M	**Str**obe signal for **M** functions [B362]
ARFD	**A**xis **ref**erence**d** [B361]
FUR_VM	**Fu**nction **r**unning **v**irtual **m**aster [B360]
OTR	**O**ver**tr**avel [B359]
FUT	**Fu**nction **t**erminated [B358]
BWD	Axis moves **b**ack**w**ar**d**s [B357]
FWD	Axis moves **f**or**w**ar**d**s [B356]
DRS	**D**estination **r**eached, axis **s**tationary [B355]
M_NO_1	**M** function number **1** [n540.34]
MODE_OUT	Mode out [n540.15]
FUR	**Fu**nction **r**unning [B354]
ST_EN	**St**art **en**able [B353]
T_R	Dwell **t**ime **r**unning [B352]
TGL_O	**T**og**gl**e bit **o**ut [B351]
M_NO_2	**M** function number **2** [n540.35]

Meaning of synchronization checkback signals (ADD_STAT_SYNC_MODE, MCT ≥ V1.31 and SIMATIC MOTION CONTROL ≥ V1.00.48 only):

VM_RA	**V**irtual **m**aster **r**eference speed **a**chieved [B817]
CL_A	**Cl**utch **a**ctive (engaging/disengaging) [B803]
POS_A	**Pos**ition correction **a**ctive [B800]
DI_A	**Di**splacement correction **a**ctive [B810]
SYNC	Axis **sync**hronized [B811]
CU_PR	**C**atch **u**p **p**osition **r**eached [B821]
CU_VR	**C**atch **u**p **v**elocity **r**eached [B822]
CU_TE	**C**atch **u**p **te**rminated [B820]

Optional extension of checkback signals:

OPTIONAL VALUE 1 to 3 OUTPUT

2.3 Description of the Control and Checkback Signals for the MASTERDRIVES MC

Interface

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx+1	ACK_F	RES	RES	RES	ENC	OFF3	OFF2	OFF1	BIN	IN_2

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy+1	RES	WARN	OFF3	OFF2	FAULT	IOP	RDY	RTS	BIN	OUT_2

Meaning of the Control Signals

The control signals for the speed controller and current controller support different shut-down variants. The operating condition of the control signals must be selected before the axis can operate.

[OFF1] = 1: Operating condition

[OFF1] = 0: Pulse disable.
If [OFF1] is cancelled while the axis is moving, the axis is decelerated via the ramp-function generator of the setpoint preparation circuit.

[OFF2] = 1: Operating condition

[OFF2] = 0: Pulse disable
If [OFF2] is cancelled while the axis is moving, the pulse disable is activated immediately. If the motor does not brake, it coasts down.

[OFF3] = 1: Operating condition

[OFF3] = 0: Rapid stop, pulse disable
If [OFF3] is cancelled while the axis is moving, the motor is decelerated at the current limit level (rapid stop). The pulse disable is activated when the axis comes to a standstill.

[ENC] = 1: Operating condition, controllers are enabled.

[ENC] = 0: Pulse disable
If [ENC] is cancelled while the axis is moving, the pulse disable is activated immediately. If the motor does not brake, it coasts down.

Meaning of the Checkback Signals

The checkback signals of the speed and current controller indicate the current operating status.

[RTS] = 0: Drive is not ready to start, pulse disable is active.
Faults [FAULT] and [FAULT_NO] are active.

[RTS] = 1: Drive is ready to start, pulse disable is active.

[RDY] = 0: Operating status, if [IOP] = 1 then controller enable, else pulse disable.

[RDY] = 1: Ready and pulse disable.
Ready is indicated when all enables ([OFF1], [OFF2] and [OFF3]) are available but the controller enable is still absent [ENC].

[IOP] = 1: Drive is in operation, controller is enabled; operating status.

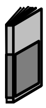
[IOP] = 0: Drive is not in operation, pulse disable.

[OFF2] = 0: [OFF2] is active, pulse disable.
Checkback signal [OFF2] represents the signal state of control signal [OFF2].

[OFF2] = 1: [OFF2] is not active; operating status.

[OFF3] = 0: [OFF3] is active, pulse disable.
Checkback signal [OFF3] represents the signal state of control signal [OFF3].

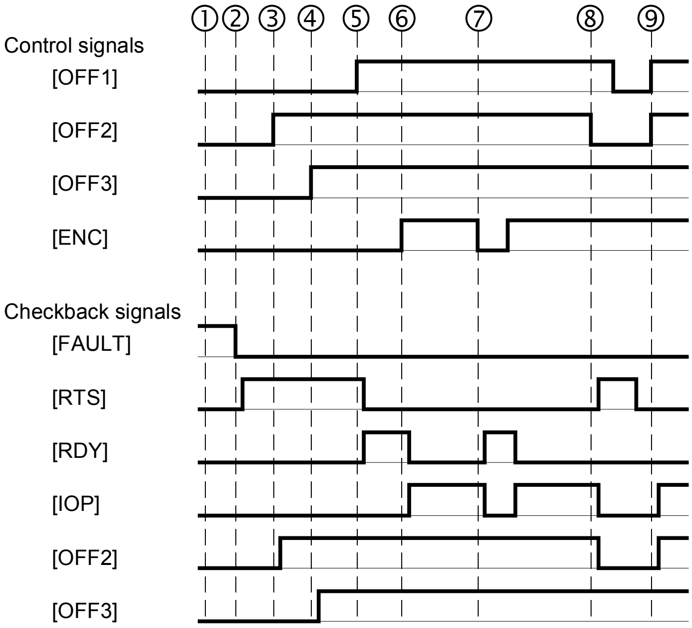
[OFF3] = 1: [OFF3] is not active; operating status.



Further Information

You will find further information about the control and checkback signals in the MASTERDRIVES MC Compendium.

Relationship between Control and Checkback Signals



① No ready to start ([RTS] = 0) because [FAULT] = 1.

② [FAULT] = 0 cleared, ready to start [RTS] = 1.

③ On control signal [OFF2] = 1 checkback signal [OFF2] =1.

④ On control signal [OFF3] = 1 checkback signal [OFF3] = 1.

⑤ On control signal [OFF1] = 1 ready to start is cancelled [RTS]= 0 and ready is activated [RDY] = 1.

⑥ If the controller enable [ENC] = 1 is now activated, the ready signal is cancelled ([RDY] = 0) and the in operation signal is activated ([IOP] = 1). The axes cannot move until this status is active. All control signals except for [OFF1] can be activated simultaneously or in any sequence. [OFF1] must be the last signal to be activated in order to achieve the [IOP] status.

⑦ If [ENC] is cancelled (= 0) in the [IOP] = 1 status, [IOP] is inactive (= 0) for this period, and [RDY] = 1.

⑧ If [OFF1] = 0, [OFF2] = 0 or [OFF3] = 0 in the [IOP] = 1 state, [IOP] is cancelled (= 0) and [RTS] = 1.

⑨ To attain the [IOP] = 1 state again, [OFF1] must be activated (0 → 1).

2.4 Description of the Control and Checkback Signals for the Technological Functions of the Axis

2.4.1 Life Bit [LB]

Interface

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx	RES	RES	RES	RES	RES	LB	RES	RES	BIN	IN_1

Meaning

The life bit [LB] **must always** be set to "1".

NOTE

The life bit is only required in association with the Profibus interface to MASTERDRIVES MC. If the life bit [LB] = 0, the control signals are not transmitted to the MASTERDRIVES MC.

Background information:
If several stations transmit the control signals to the same MASTERDRIVES MC, the stations can use the life bit as an interlock.

2.4.2 Mode Selection [MODE_IN] / Mode Checkback [MODE_OUT]]

Interface

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx+2	MODE_IN				J_FWD	F_S	J_BWD	BLSK	BIN	IN_3

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy+6	MODE_OUT				FUR	ST_EN	T_R	TGL_O	BIN	OUT_7

Meaning

The axis can be operated in different modes. Different functions are available according to the mode selected. The user selects the mode with [MODE_IN] and the axis indicates the mode [MODE_OUT].

If an invalid mode is selected, the axis returns [MODE_OUT] = 0 ([MODE_IN] > 7 except for [MODE_IN] = 10, or [MODE_IN] 2 with serial encoders or roll feed). The mode is only activated if the selected mode [MODE_IN] matches the returned mode [MODE_OUT].

The mode checkback signal [MODE_OUT] is not output during a mode change until the movement in the old mode has been completed. If a mode change is initiated during an active traversing movement, the axis activates the deceleration ramp. The mode checkback signal ([MODE_IN] = [MODE_OUT]) does not switch until the axis has reached a standstill.

Available Operating Modes

The following modes can be selected:

- ♦ 0001 Setup mode [MODE] 1
- ♦ 0010 Reference point approach mode [MODE] 2
- ♦ 0011 MDI mode [MODE] 3
- ♦ 0100 Control mode [MODE] 4
- ♦ 0101 Automatic mode [MODE] 5
- ♦ 0110 Automatic single-block mode [MODE] 6
- ♦ 1010 Slave mode [MODE] 10 (with SIMATIC Motion Control only)
- ♦ 1011 Synchronization mode [MODE] 11

All other modes in MODE_IN generate mode 0000 in MODE_OUT.

2.4.3 Jog Forwards [J_FWD] / Jog Backwards [J_BWD]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+2	MODE_IN				J_FWD	F_S	J_BWD	BLSK	BIN	IN_3

Meaning

The meaning of control signals [J_FWD] and [J_BWD] depends on the mode.

Setup Mode

The signals define the direction in which the axis traverses in position control mode.

[J_FWD] = 1 moves the axis in the direction of increasing actual values.

[J_BWD] = 1 moves the axis in the direction of decreasing actual values.

Reference Point Approach Mode

The signals define the direction of movement in which the axis has to traverse in position control mode, in order to reach the reducing cam.

[J_FWD] = 1 moves the axis in the direction of increasing actual values.

[J_BWD] = 1 moves the axis in the direction of decreasing actual values.

Special situation: With "set reference point", the activation of [J_FWD] or [J_BWD] sets the reference point.

If [J_FWD] and [J_BWD] are activated simultaneously, no axis movement takes place or the axis movement in progress is stopped.

Control Mode

The signals define the direction of movement in which the axis traverses in speed control mode.

[J_FWD] = 1 moves the motor in the positive direction of rotation.

[J_BWD] = 1 moves the motor in the negative direction of rotation.

If [J_FWD] and [J_BWD] are activated simultaneously, the motor rotation is suppressed or stopped.

MDI Mode

The signals select the direction of rotation for a rotary axis with absolute dimension programming (G90).

[J_FWD] moves the rotary axis in the positive direction
(0 → 180 → 360).

[J_BWD] moves the rotary axis in the negative direction
(360 → 180 → 0).

[J_FWD] and [J_BWD] are only selection variables in this context, since the movement in MDI mode is initiated by the start [STA] control signal.

NOTE

If you position a rotary axis with absolute dimensions, e.g. from 40 degrees to 200 degrees, there are two possible movements: either from 40 degrees via 100 degrees to 200 degrees, or from 40 degrees via 0 degrees to 240 degrees. If you do not select a direction of movement, the axis is positioned across the shortest path. [J_FWD] and [J_BWD] are used to define an explicit direction of axis movement. If [J_FWD] and [J_BWD] are activated simultaneously, no direction of movement is selected.

NOTE

MASTERDRIVES MC parameter P595 can be used to invert the direction of rotation of the motor and thus the direction of movement of the axis (P595 = 0: clockwise, default setting; P595 = 1: anticlockwise).

2.4.4 Selection of Fast / Slow [F_S]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+2	MODE_IN				J_FWD	F_S	J_BWD	BLSK	BIN	IN_3

Meaning

The [F_S] control signal switches between two settable velocities/speeds for the setup and control modes.

Setup Mode

You can switch at will between level 1 and level 2. The active velocity is reached in position control mode via the acceleration or deceleration ramp.

[F_S] = 0 → Selects velocity level 1

[F_S] = 1 → Selects velocity level 2

Control Mode

You can switch at will between 10 % and 100 %. The current speed is reached in speed control mode via "acceleration time, operating mode control" or "deceleration time, operating mode control" (MD42).

The following is applicable for the MASTERDRIVES MC:
The speed setting for control mode can be changed using the parameters P2511 Index 1 and P2511 Index 2. The factory settings are 10 % and 100 %.

[F_S] = 0 → [P2511 Index 1] % or 10 % of the rated speed by default

[F_S] = 1 → [P2511 Index 2] % or 100 % of the rated speed by default

NOTE

The level selected with [F_S] is still affected by the override setting [OVERRIDE].

2.4.5 Block Skip [BLSK]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+2	MODE_IN				J_FWD	F_S	J_BWD	BLSK	BIN	IN_3

Meaning

Any block can be defined as a skippable block when programming an NC program.

If control signal [BLSK] = 0 when the program is running, all blocks of the NC program are executed. If control signal [BLSK] = 1, all skippable blocks are skipped without being executed.

2.4.6 Axis Referenced [ARFD]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+4	STR_M	ARFD	FUR_VM	OTR	FUT	BWD	FWD	DRS	BIN	OUT_5

Meaning

Checkback signal [ARFD] informs the user that the measuring system has been synchronized. With absolute encoders, [ARFD] is activated statically, while, with incremental encoders, the axis first has to be referenced in "reference point approach" mode.

NOTE

[ARFD] has no meaning with the roll feed function and is always displayed with "0".

WARNING

Since the axis can be operated without the [ARFD] signal in all functions, you must ensure that [ARFD] is used as an interlock in your application. If the axis is not referenced, no monitoring of the software limit switches takes place. The "reset axis" control signal [RST] clears the [ARFD] checkback with incremental encoders.

2.4.7 Display and Acknowledgement of Faults and Warnings

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+1	RES	WARN	OFF3	OFF2	FAULT	IOP	RDY	RTS	BIN	OUT_2
DBBy+2	FAULT_NO								DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+1	ACK_F	RES	RES	RES	ENC	OFF3	OFF2	OFF1	BIN	IN_2

Meaning

There are two different types of error:

- Faults
- Warnings

Faults

Faults are only generated by the basic functionality of the MASTERDRIVES MC and trigger an immediate interruption in the power supply to the motor (pulse disable).

If a fault occurs, checkback signal [FAULT] = 1 is activated and the fault number is entered in [FAULT_NO].

Before a fault can be cleared, control signal [ACK_F] must be activated. The fault acknowledgement is only triggered by a rising edge.

Further Information

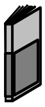
The meaning and cause of faults are described in the documentation entitled MASTERDRIVES MC "Compendium".

Warnings

Warnings are generated both by the basic functionality of MASTERDRIVES MC and by the technology (centrally on SIMATIC Motion Control or decentrally on the MASTERDRIVES MC). Warnings are handled in different ways: some warnings have no influence on the movement of the axes, other warnings bring the drive to a controlled standstill (in speed or position control mode). The power supply to the motor is never interrupted.

When a warning occurs, checkback signal [WARN] = 1 is activated and the warning number is entered in [WARN_NO]. Since several warnings can be active at a time, [WARN_NO] always indicates the last warning which occurred.

Warning numbers 1 to 128 are generated by the basic functionality of the MASTERDRIVES MC. These warnings have no effect on the movement of the axis. They are also cleared automatically once the cause of the fault has been remedied.

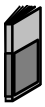


Further Information

The meaning and cause of warning numbers 1 to 128 are described in the documentation entitled MASTERDRIVES MC "Compendium".

Warning numbers 129 to 255 are generated by the technology. Moving axes are brought to a controlled standstill depending on the type of error. It is possible to read out up to eight currently active warnings with the "output GMC technology warnings" task.

To clear a technology warning, control signal [ACK_F] must be activated. It is only the rising edge of this signal which initiates fault acknowledgement. Technology warnings can also be cleared with the "acknowledge GMC technology warnings" task and the "reset technology" [RST] control signal.



Further Information

The meaning and cause of technology warnings 129 to 255 are described in the Appendix "Error Messages".

2.4.8 Reset Technology [RST]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+5	SIST	RST	FUM	ACK_M	CRD	STA	RIE	TGL_I	BIN	IN_6

Meaning

The [RST] control signal resets the technological function of the axis. The current machining status, the following error, the actual position value, queued technology warnings and, with incremental encoders, the "axis is referenced" [ARFD] checkback signal are all cleared.

Setpoint output and the controller enable for the drive controller are interrupted immediately. With incremental encoders, it is necessary to synchronize the measuring system again in "reference point approach" mode.

If machine data 1 (axis type / encoder type) is changed, the technology also requests a technology reset ([WARN]).

Encoder faults must be cleared with [RST]. A status change of the simulation must be activated with [RST].

NOTE

With MASTERDRIVES MC, switching the electronic power supply on or off (mains off/on) also generates an RST signal.

2.4.9 Override [OVERRIDE]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+3	OVERRIDE								DEC	IN_4

Meaning

The override modifies the defined traversing velocity by a percentage factor. The percentage value of the [OVERRIDE] specifies at what percent of the defined traversing velocity a movement is to take place. The [OVERRIDE] is active in all operating modes except for synchronization and slave modes and can be varied in a range from 0 to 255 %.

In "control" mode, the [OVERRIDE] specifies the speed in percent, where 100 % is 10 % or 100 % of the rated speed, depending on control signal [F_S].

In the positioning modes (MDI, automatic and single-block), the [OVERRIDE] can be used purely as a velocity override or as a time override.

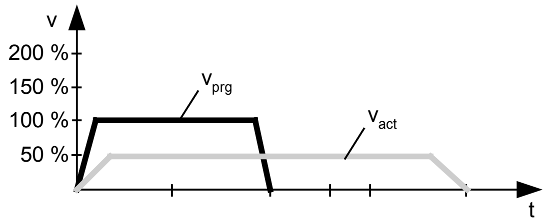
NOTE

The time override must be enabled with machine data 26. An enabled time override is only possible in the range from 0 % to 100 % [OVERRIDE].

Time Override

The [OVERRIDE] from 0 % to 100 % is a time override. This means that both the velocity and the acceleration and deceleration are modified so that the time is in direct relation to the override value.

Example: The override value is halved from 100 % to 50 %:



- Velocity v is halved
- Acceleration and deceleration are quartered
- Positioning time is doubled

$$v_{act} = \frac{v_{prog} * Override}{100}$$

$$a_{act} = \frac{a * Override^2}{100^2}$$

$$t_{act} = \frac{t * 100}{Override}$$

The calculation of the override as a time override is subject to the following additional condition:

In cases where a traversing movement comprises several NC blocks with block change on-the-fly (the axis does not stop between blocks), only the velocity is affected when the override value is changed. The additional impact on the acceleration and deceleration is not included in the calculation until the axis has stopped (e.g. if the direction is reversed).

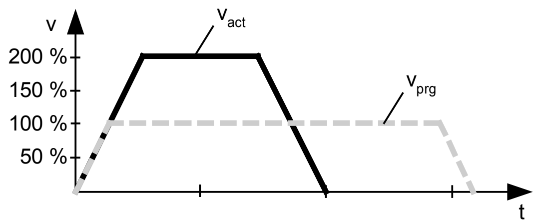
NOTE

An [OVERRIDE] of 0 % stops the axis (axes). Checkback signals [FWD] and [BWD] are still output, however.

Velocity Override

The [OVERRIDE] is a pure velocity override, i.e. the percentage value refers to the velocity.

Example: The override value is doubled from 100 % to 200 %



- Velocity v is doubled with the same acceleration and deceleration ramps
- The positioning time is not halved

2.4.10 Program Number [PROG_NO] or MDI Number [MDI_NO]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+4	PROG_NO ODER MDI_NO								DEC	IN_5

MDI Mode

Ten MDI blocks can be stored in the memory of the technology for MDI mode.

The MDI block number, [MDI_NO] = 1 to 10, must be specified before the start [STA] of an MDI block.

Automatic Mode

Up to 200 NC programs can be stored in the memory of the technology for "automatic" mode.

The program number must be specified with [PROG_NO] = 1 to 200 before the program is activated by the read-in enable [RIE] control signal.

2.4.11 Read-In Enable [RIE]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+5	SIST	RST	FUM	ACK_M	CRD	STA	RIE	TGL_I	BIN	IN_6

Meaning

The read-in enable control signal [RIE] is dependent on the axis type (machine data 1).

NOTE

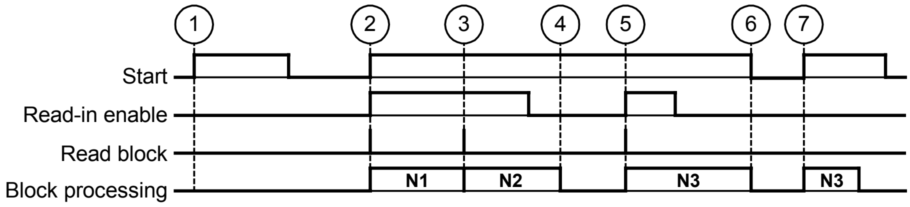
Machine data 48 (digital outputs – function 2) can be used to configure an "external read-in enable" as a digital input. The digital input assumes the function [RIE] or is ANDed with [RIE], depending on the configuration.

Axis Type 1 or 2

If axis type 1 or 2 (axis with incremental or absolute position encoder) was entered, the read-in enable [RIE] is only effective in automatic mode.

When the read-in enable is activated ([RIE] = 1), decoding of the NC program (advance decoding of multiple blocks) is started. Program execution does not yet take place. The program does not start running and the axes do not move until the start control signal [STA] is enabled.

Example:



① If start is activated without the read-in enable, the program does not start running.

② If the read-in enable is available, the first block is read in and executed after the start signal.

③ If the read-in enable remains active, the next NC block is automatically read in and executed.

④ If the read-in enable is cancelled during execution of an NC block, processing continues up to the end of the current block but the next block is not read in.

⑤ When the read-in enable is reactivated, the next block is read in and executed.

⑥ Canceling the start signal interrupts execution of the NC block.

⑦ To continue a block which has already been read in, it is sufficient to activate the start signal; it is not necessary to activate the read-in enable.

Cancellation of the read-in enable always interrupts decoding of the NC program.

Axis Type 3

With the roll feed version (axis type 3), the read-in enable [RIE] is effective in MDI, automatic and single-block modes.

Unlike axis types 1 and 2, the start signal [STA] only prepares a movement (all necessary internal technology data are prepared). The movement itself is triggered by the rising edge of the read-in enable [RIE].

- MDI mode:
  When the start signal is active, each positive edge of the read-in enable signal [RIE] triggers a new execution of the MDI block.
- Automatic mode:
  When the start signal is active, each positive edge of the read-in enable signal [RIE] triggers the execution of a block (or multiple NC blocks with special programming). Machining stops at the end of the program. The start signal must be enabled again in order to start the NC program again.
- Single-block mode:
  When the start signal is active, each positive edge of the read-in enable signal [RIE] executes the block again.

2.4.12 Start [STA]

Interface

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx+5	SIST	RST	FUM	ACK_M	CRD	STA	RIE	TGL_I	BIN	IN_6

Meaning

The start control signal [STA] depends on the axis type (machine data 1).

Axis Type 1 or 2

If axis type 1 or 2 (axis with incremental or absolute encoder) was entered, a function or subfunctions is triggered in MDI, automatic and single-block modes when the start control signal [STA] = 1. Canceling the start signal ([STA] = 0) stops the function.

- MDI mode:
  Start [STA] = 1 triggers execution of the MDI block. The positioning movement can be interrupted by stopping with [STA] = 0. A new [STA] = 1 continues the positioning operation.
- Automatic mode:
  Start [STA] = 1 triggers program execution. The program can be interrupted at any time by stopping with [STA] = 0.
- Single-block mode:
  Start [STA] = 1 triggers execution of the block. The block can be interrupted at any time by stopping with [STA] = 0.
- Synchronization mode:
  Start [STA] = 1 activates the synchronization functions; the slave axis moves according to the selected synchronization function. If the start signal is cancelled ([STA] = 0), the synchronization functions are deactivated. A moving axis is brought to a standstill in speed control mode (see MD42). At a standstill, the axis is position-controlled.

Axis Type 3

With the roll feed version (axis type 3), activating the start signal ([STA] = 1) in MDI, automatic and single-block modes prepares a function or subfunctions. The actual axis movement is triggered by the read-in enable control signal [RIE]. Canceling the start signal ([STA] = 0) interrupts the movement. The start signal must be reactivated ([STA] = 1) in order to continue the interrupted function.

The start enable [ST_EN] is required for the start operation, otherwise [STA] is ignored.

2.4.13 Selection of Cancel Remaining Distance [CRD]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+5	SIST	RST	FUM	ACK_M	CRD	STA	RIE	TGL_I	BIN	IN_6

Meaning

The "cancel remaining distance" control signal [CRD] is effective in MDI, automatic and single-block modes.

If a positioning operation is interrupted by canceling the start signal [STA], you can select with "cancel remaining distance" [CRD] whether the axis moves through the remaining distance or whether the remaining distance is cancelled.

NOTE

The remaining distance is defined as the distance remaining between the current position setpoint and the setpoint of the position to be approached.

Effect of Selecting Cancel Remaining Distance on Positioning Abort

If the positioning movement was aborted, the following options are available for the start signal [STA]:

- MDI mode:

  [CRD] = 0: The positioning movement is continued across the remaining distance after the start signal.

  [CRD] = 1: The remaining distance is cancelled after the start signal. If a new MDI block has already been loaded under the same MDI number, this block is executed immediately. If a new MDI block has not yet been loaded, the selected MDI block is executed from the beginning.

  Selecting another MDI number [MDI_NO] cancels the remaining distance when the start signal [STA] is activated, irrespective of the state of the "cancel remaining distance" control signal [CRD].

- Automatic mode:

  [CRD] = 0: Execution of the interrupted block continues on "start".

  [CRD] = 1: Execution resumes immediately with the next block on "start".

  Selecting a new program number [PROG_NO] cancels the remaining distance when the start signal is activated [STA], irrespective of the state of the "cancel remaining distance" control signal [CRD].

- Single-block mode:

  [CRD] = 0: Execution of the interrupted block continues on "start".

  [CRD] = 1: The remaining distance is cancelled on "start". If a new NC block has already been loaded, it is executed immediately. If a new block has not yet been loaded, the active block is executed from the beginning.

Canceling the Remaining Distance

The remaining distance is also cancelled as a result of the following conditions:

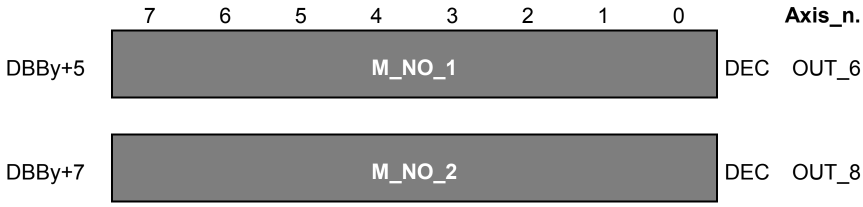
- ♦ Reset control signal [RST]
- ♦ Mode change [MODE_IN]
- ♦ Warning occurrence [WARN] and [WARN_NO]

NOTE

Control signal [CRD] is a selection bit, i.e. the [CRD] signal does not actually cancel the remaining distance. Only when the start signal [STA] is activated does the axis check the status of [CRD] before initiating the response described above.

2.4.14 M Function Number [M_NO_1], [M_NO_2]

Interface



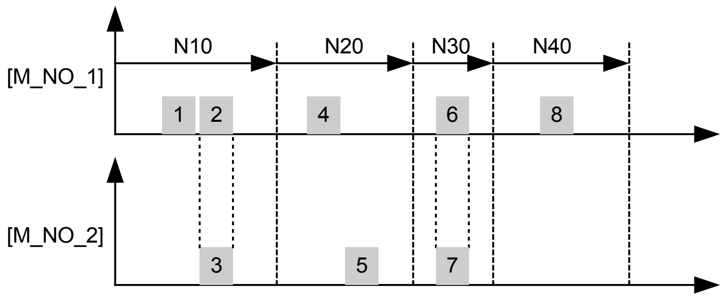
Meaning

The M functions allow you to define the actuation of your machine functions at the program development stage. The M numbers are output during program execution in the sequence programmed in [M_NO_1] and/or [M_NO_2].

Programming the M Function Number

Up to 3 M functions (M numbers) can be programmed in one NC block. The M functions are output as follows:

Programming:			
N10	M1 = 1	M2 = 2	M3 = 3
N20	M1 = 4	M3 = 5	
N30	M2 = 6	M3 = 7	
N40	M1 = 8		



N10: M1=1 is output first, then M2=2 and M3=3 are output simultaneously.
N20: M1=4 is output first, then M3=5.
N30: M2=6 and M3=7 are output simultaneously.
N40: M1=8 is output.

2.4.15 Strobe Signal for M Functions [STR_M]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+4	STR_M	ARFD	FUR_VM	OTR	FUT	BWD	FWD	DRS	BIN	OUT_5

Meaning

As soon as the axis detects an NC block with M functions during program execution, the M function output is declared by enabling the M function strobe signal [STR_M] = 1, i.e. the M function number can be read from [M_NO_1] or [M_NO_2].

The strobe signal for M functions [STR_M] = 1 remains active until:

- The defined time has expired in the case of time-driven M functions.
- The user has acknowledged the M function [ACK_M] = 1 in the case of acknowledge-driven M functions.

2.4.16 Acknowledge M Function [ACK_M]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+5	SIST	RST	FUM	ACK_M	CRD	STA	RIE	TGL_I	BIN	IN_6

Meaning

The "acknowledge M function" control signal [ACK_M] is only relevant if "acknowledge-driven" was specified as the output type for the M function.

The M function strobe signal [STR_M] = 1 is output together with the M function. The M function remains active until it has been acknowledged by the user [ACK_M] = 1. The next M functions are not output until the acknowledgement signal has been cancelled [ACK_M] = 0. When all M functions have been output and the acknowledgement signal has been reset ([ACK_M] = 0), the conditions have been met for further execution of the program.

2.4.17 Start Enable [ST_EN]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+6	MODE_OUT				FUR	**ST_EN**	T_R	TGL_O	BIN	OUT_7

Meaning

The start enable informs the user that the axis is ready to accept commands.

The start enable [ST_EN] is not output until:

- The mode command [MODE_IN] and mode checkback [MODE_OUT] signals match (precondition: a valid mode has been selected).
- [WARN] and [FAULT] are not active.
- [RST] has not been activated.
- The axis is not currently executing a function (i.e. it has not yet been started)

If the axis is operated as a slave axis [MODE] = 10, no start enable is output, because the operation is performed via the master axis.

The start enable is cancelled when a function is activated. The start enable is the precondition for all user operations. If you attempt to start a function without the start enable, there is no response from the technology.

2.4.18 Function Running [FUR]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+6	MODE_OUT				**FUR**	ST_EN	T_R	TGL_O	BIN	OUT_7

Meaning

The "function running" signal [FUR] indicates the status of the active function. "Function running" is active for the entire period of the function, even if the user has initiated a "stop".

[FUR] is cancelled as a result of:

- The occurrence of a warning [WARN] or fault [FAULT]
- The "reset" [RST] control signal
- A mode change [MODE_IN]

If the axis is operated as a slave axis ([MODE_IN] = Slave), "function running" [FUR] is always "0", because this axis responds passively, i.e. it is controlled by a master axis.

The output of [FUR] depends on the mode.

Setup Mode

During the active movement via "jog forwards" [J_FWD] or "jog backwards" [J_BWD] until the axis stops.

Reference Point Approach Mode

During the approach movement until the reference point is reached and the measuring system is synchronized, including traversing of the reference point offset.

MDI Mode

During the positioning operation until the position is reached [DRS].

NOTE

With the roll feed axis type, "function running" is output after the first start either continuously or until the loop counter ("roll feed loop count MDI" task) has reached the value "0".

Control Mode

During speed-controlled rotation of the motor with "jog forwards" [J_FWD] or "jog backwards" [J_BWD] until the motor has come to a final standstill.

Automatic Mode

After activation of the program by the read-in enable [RIE], throughout the entire period of execution up to the end of the program.

NOTE

If the NC program is programmed as an infinite loop, "function running" [FUR] is permanently active.

Single-Block Mode

During the entire period of NC block execution.

Synchronization Mode

After activation of the synchronization functions with the start signal [STA] until cancellation of [STA] or the occurrence of a fault [FAULT] or warning [WARN].

2.4.19 Dwell Time Running [T_R]

Interface

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy+6	MODE_OUT				FUR	ST_EN	T_R	TGL_O	BIN	OUT_7

Meaning

The "dwell time running" checkback signal [T_R] is only active in "automatic" and "single-block" modes. When an NC block with dwell is executed, [T_R] is output for the programmed duration.

If the axis is operated as a slave axis ([MODE_IN] = Slave), "dwell time running" [T_R] is always "0", because this axis responds passively, i.e. it is controlled by a master axis.

2.4.20 Follow-Up Mode [FUM]

Interface

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx+5	SIST	RST	FUM	ACK_M	CRD	STA	RIE	TGL_I	BIN	IN_6

Meaning

The [FUM] control signal is independent of the selected mode. When the signal is activated ([FUM] = 1), the position controller is deactivated and switched to follow-up mode. Follow-up mode means that the position setpoint follows the actual position value. The speed controller is thus actuated with a speed setpoint of 0.

If [FUM] = 1 is activated during an active position-controlled axis movement, the axis is brought to a standstill via a deceleration ramp in speed control mode (machine data 43 - deceleration time during errors).

If [FUM] = 1 is activated during program execution, the current program data (machining status of the NC program) are deleted. This means that the position controller is activated again when the signal is cancelled ([FUM] = 0) and the NC program has to be started again. Machining resumes at the start of the program.

Certain warnings (e.g. following error) cause the technology to switch automatically to follow-up mode, since the integrity of the position control system cannot be maintained in these circumstances.

NOTE

If machine data 43 (deceleration time during errors) = 0, the axis is stopped abruptly when "follow-up mode" is activated [FUM = 1].

2.4.21 Single-Step [SIST]

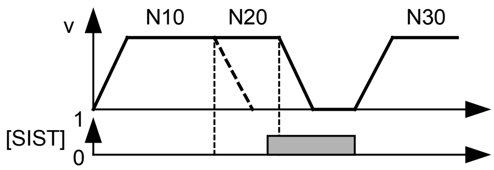
Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+5	SIST	RST	FUM	ACK_M	CRD	STA	RIE	TGL_I	BIN	IN_6

Meaning

Program execution normally takes place automatically, i.e. the technology automatically executes the NC blocks of a program in succession.

The single-step control signal [SIST] is only active in "automatic mode". When the single-step control signal is active ([SST] = 1), only one block is executed after each start signal [STA]. When the single-step signal is cancelled again ([SIST] = 0), continuous program execution resumes after the next start signal [STA].



The [SIST] control signal is evaluated at the deceleration start point.

NOTE

The [SIST] control signal has no effect with the roll feed version, because programs are always executed in single-step mode.

2.4.22 Destination Reached, Axis Stationary [DRS]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+4	STR_M	ARFD	FUR_VM	OTR	FUT	BWD	FWD	DRS	BIN	OUT_5

Meaning

When the specified target position is reached, the "destination reached, axis stationary" signal is activated ([DRS] = 1) and remains active until the next axis movement. A precondition for output of the [DRS] checkback signal is that a specified tolerance (machine data 16 – in position, exact stop window) is reached within a specified time (machine data 17 – in position, timer monitoring). If this is not the case, a warning [WARN] is generated and the positioning operation is aborted.

In setup mode, checkback signal [DRS] is not activated, however an existing [DRS] is retained when "setup" is selected from another operating mode. The [DRS] checkback signal is not cancelled until a traversing movement is performed in the setup mode. [DRS] is also cancelled as a result of:

- Axis reset [RST]
- Warnings [WARN] which deactivate the position controller (e.g. following error fault)

"Destination reached, axis stationary" [DRS] is a dynamic checkback signal, i.e. if the axis is pushed out of position, thereby leaving the exact stop window, [DRS] is cancelled. The [DRS] checkback signal appears again when the axis moves back into the exact stop window in response to the position controller.

Activation of the [DRS] checkback signal is mode-dependent and only occurs under certain conditions.

Reference Point Approach Mode

The reference point has been reached completely (including the reference point offset).

MDI Mode

When the MDI block has been traversed up to the end of the block.

Automatic Mode

When a block has been traversed up to the end of the block and the axis dwells in the position as a result of a condition (e.g. stop at end of block), and at the end of the program.

Single-Block Mode

When the block has been traversed up to the end of the block.

2.4.23 Axis Moves Forwards [FWD], Axis Moves Backwards [BWD]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+4	STR_M	ARFD	FUR_VM	OTR	FUT	**BWD**	**FWD**	DRS	BIN	OUT_5

Meaning

As soon as an active traversing movement is queued, the checkback signal "axis moves forwards" ([FWD] = 1) or "axis moves backwards" ([BWD] = 1) is output, according to the direction of movement. [FWD] and [BWD] can only be active alternately.

[FWD] = 1 means that the axis moves towards increasing actual values.

[BWD] = 1 means that the axis moves towards decreasing actual values.

[FWD] or [BWD] is activated at the beginning of the acceleration phase and remains active until the axis comes to a standstill. The [FWD] and [BWD] checkback signals are even active with an override of 0 %.

In "control" mode, [FWD] or [BWD] is activated during active speed control.

2.4.24 Function Terminated [FUT]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+4	STR_M	ARFD	FUR_VM	OTR	**FUT**	BWD	FWD	DRS	BIN	OUT_5

Meaning

In MDI, automatic and single-block modes, a function is activated with the start control signal ([STA] = 1). Setting [STA] = 0 means stop. The "function terminated" [FUT] checkback signal is provided in order to facilitate start/stop coordination.

[FUT] = 1 indicates that the function (subfunction) triggered with start [STA] has terminated and that the user can cancel the start signal [STA] again. "Function terminated" is reset ([FUT] = 0) when the start signal [STA] is cancelled.

[FUT] = 1 is also enabled if the function is aborted as a result of an error and the start control signal [STA] = 1. In this case, the error checkback signals [WARN] and [FAULT] must be used to perform diagnostics and remedy the fault before the function can be started again.

Various situations require a stop/start edge for continuation of the function. This is indicated by [FUT] = 1.

MDI Mode

After execution of an MDI block for the following MDI block.

Automatic Mode

- After execution of an NC program for a new program start.
- After a programmed stop (M00) at the end of the block for the execution of the subsequent blocks.
- After any block if the single-step control signal is active ([SIST] = 1).

Single-Block Mode

After execution of the block for the next block.

Synchronization Mode

If an abort occurs ([WARN], [FAULT], [RST]) when start [STA] is active.

NOTE

The "function terminated" control signal [FUT] = 1 is also activated if [STA] = 1 and [ST_EN] = 0.

2.4.25 Overtravel [OTR]

Interface

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy+4	STR_M	ARFD	FUR_VM	OTR	FUT	BWD	FWD	DRS	BIN	OUT_5

Meaning

Machine data 12 (software limit switch negative) and machine data 13 (software limit switch positive) are available for limiting the traversing area. These machine data define the negative and positive limit positions. The software limit switches are not active until the axis has been referenced.

While the measuring system is not synchronized until the reference point is available in the case of axis type 1 (axis with incremental encoder); the measuring system is always synchronized with axis type 2 (axis with absolute encoder).

The "overtravel" checkback signal is output ([OTR] = 1) when the actual value of the position encoder is positioned exactly on top of or behind the negative or positive limit switch. If the axis traverses up to the software limit switch in position control mode, a warning [WARN] is output. This warning indicates whether the negative or positive software limit switch was approached.

NOTE

Once a software limit switch has been approached, you can only depart from that position in setup or MDI mode.

If you do not require the software limit switches, machine data 12 and 13 must be initialized with values which lie outside the possible traversing area.

DANGER

It is possible to cross the software limit switches in "control" mode. The software limit switches do not substitute the hardware limit switches for emergency stop reactions.

2.4.26 Select Function [FUNCTION]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+6	R_VM	S_VM	EN_RF	SSC	OPERATION		FUNCTION		BIN	IN_7

Meaning

For synchronization mode, the synchronization method for the slave axis is selected with [FUNCTION].

The following three functions can be selected:

- [FUNCTION] = 00 → 1:1
  The slave axis is coupled 1:1 with the master axis. The result is velocity synchronization.
- [FUNCTION] = 01 → Gearbox
  The slave axis is coupled with the master axis via an electronic gearbox. The gear factor can be used to step the transmission ratio up or down.
- [FUNCTION] = 10 → Table
  The slave axis is coupled with the master axis via an NC table. The table contains the pattern of movement of the slave axis with reference to the master axis.

2.4.27 Select Operation [OPERATION]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+6	R_VM	S_VM	EN_RF	SSC	OPERATION		FUNCTION		BIN	IN_7

Meaning

For synchronization mode, the extended selection of slave operation is performed with [OPERATION].

The following operations are available:

- [OPERATION] = 00 → Continuous operation
  The slave axis tracks the master axis in accordance with the function selected in [FUNCTION].
- [OPERATION] = 01 → Start cycle
  The slave axis is stationary. When the start condition (coupling position) is reached, the slave axis tracks the master axis for the specified time in accordance with the function selected in [FUNCTION].
- [OPERATION] = 10 → Stop cycle
  The slave axis tracks the master axis in accordance with the function selected in [FUNCTION]. When the stop condition (coupling position) is reached, the slave axis remains stationary for the specified time.
- [OPERATION] = 11 → Catch-up (MCT ≥ V1.4 and SIMATIC MOTION CONTROL >= V1.00.48 only)
  The slave axis tracks the master axis in accordance with the function selected in [FUNCTION]. When the stop condition (coupling position) is reached, the slave axis remains stationary for the specified time.

2.4.28 Start/Stop Cycle Continuous [SSC]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+6	R_VM	S_VM	EN_RF	SSC	OPERATION		FUNCTION		BIN	IN_7

Meaning

The "start/stop cycle continuous" control signal [SSC] specifies the following additional condition for the start/stop cycle:

- SSC = 0: Start/stop cycle by trigger signal
  The trigger for the start/stop cycle is a trigger signal [SST]. Only when [SST] has generated a 0 → 1 edge does a start/stop cycle take place when the next coupling position is crossed.
- SSC = 1: Start/stop cycle continuous
  A start/stop cycle is triggered each time the coupling position is crossed.

2.4.29 Start/Stop Cycle Trigger [SST], Clutch active [CL_A]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+7	ST_VM	TABLE_NO			SYN_T	SST	ST_S	SET_T	BIN	IN_8

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+8	CU_TE	CU_VR	CU_PR	SYNC	DI_A	POS_A	CL_A	VM_RA	BIN	OUT_9_0

Meaning

The "start/stop cycle trigger" control signal [SST] sets the trigger on the rising edge. The slave axis is then coupled or decoupled from the master axis the next time the coupling position is crossed.

Precondition:
The control signal [SSC] must have the status 0 (start/stop cycle continuous).

Advanced checkback signal (with MCT >=V1.31 and SIMATIC MOTION CONTROL >= V1.00.48 or higher):
[CL_A] = 1 means that the start / stop cycle is currently active. The signal is given from the point at which the axes cease to be in synchronization to the point at which they are in synchronization again.

2.4.30 NC Table Number [TABLE_NO]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+7	ST_ VM	**TABLE_NO**			SYN_ T	SST	ST_S	SET_ T	BIN	IN_8

Meaning

The table number required for the "table" synchronization function is selected in [TABLE_NO].

If the technology is operated in conjunction with a MASTERDRIVES MC (MCT), up to 2 NC tables can be selected.

If the technology is operated in conjunction with an M7-FM, up to 8 NC tables can be selected.

[TABLE_NO] = 000: Table number 1
[TABLE_NO] = 001; Table number 2
.....
[TABLE_NO] = 111: Table number 8

2.4.31 Set NC Table to Master Axis Set Value [SET_T]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+7	ST_ VM	TABLE_NO			SYN_ T	SST	ST_S	**SET_ T**	BIN	IN_8

Meaning

The "set table" control signal [SET_T] assigns the current position of the master axis to a specified table position (master axis: set value).

Further Information

You will find further information in the chapter entitled "Synchronization Functions" and in the "Task Description" in Chapter "NC Table Tasks" in Section "Input/Output Table Parameters".

2.4.32 Synchronize Table [SYN_T]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+7	ST_ VM	TABLE_NO			**SYN_ T**	SST	ST_S	SET_ T	BIN	IN_8

Meaning

The "synchronize table" control signal [SYN_T] assigns the current position of the master axis to the start of the NC table.

Further Information

You will find further information in the chapter entitled "Synchronization Functions" and in the "Task Description" in Chapter "NC Tables" in Section "Input/Output Table Parameters".

2.4.33 Position Slave Axis at NC Table Position [ST_S] i.V.

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+7	ST_ VM	TABLE_NO			SYN_ T	SST	**ST_S**	SET_ T	BIN	IN_8

Meaning

This function is under development.

2.4.34 Enable Setting of Floating Reference Point

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+6	R_VM	S_VM	**EN_ RF**	SSC	OPERATION		FUNCTION		BIN	IN_7

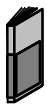
NOTE

The control signal "enable setting of floating reference point" [EN_RF] is available only with MCT and firmware version ≥ V1.4x or higher.

Meaning

You can dynamically enable or disable the "floating reference point setting" function using control signal [EN_RF]. It allows you to reference an axis in any operating mode.

[EN_RF] 0 signal = setting of floating reference point is disabled
1 signal = setting of floating reference point is enabled



Further Information

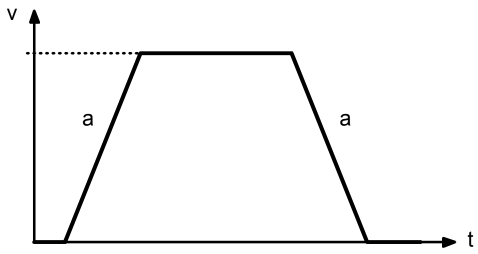
A more detailed description of the "set floating reference point" function cab be found in the following chapters: "Setup mode", "MDI mode", "Control mode", "Automatic mode" and "Synchronization mode".

2.5 Description of the Control and Checkback Signals of the Virtual Master

The position generator integrated in each axis is known as a virtual master. The virtual master is not tied to any specific operating mode and is generally used as a source of control values for synchronized axes.

The virtual master operates on the basis of the following parameters:

- ♦ Velocity setpoint v:
  The sign of the velocity setpoint determines the direction in which the position value is counted.
- ♦ Acceleration/deceleration a:
  Defines the ramps for the acceleration and deceleration.



- ♦ Set position:
  Determines the start position
- ♦ Linear/rotary axis:
  Determines the response of the position value

Linear/rotary axis = 0:	The position setpoint counts up to infinity; it is necessary to prevent the number range from being exceeded → linear axis.
Linear/rotary axis > 0:	The position value counts up to the linear/rotary axis value – 1 and starts again from zero → rotary axis. The virtual master should normally be operated as a rotary axis in order to prevent a number range overflow.

Further Information

The parameters for the virtual master can be input or output using the command "Input / output synchronization parameters". Further information can be found in the "Command description" section.

NOTE

If any of the parameters set speed, acceleration / delay or linear / rotary axis are changed while the virtual master is running, the amended parameters immediately affect the behavior of the virtual master..

2.5.1 Set Start Value Virtual Master [S_VM]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+6	R_VM	**S_VM**	EN_ RF	SSC	OPERATION		FUNCTION		BIN	IN_7

Meaning

The position value of the virtual master is set to the set position of the virtual master on the rising edge of control signal [S_VM].

NOTE

If control signal [S_VM] is activated while the virtual master is running, the current position value is overwritten by the set position of the virtual master on the rising edge of [S_VM]. The virtual master starts counting again from the set position.

This jump in the path setpoint has a similarly abrupt effect on the synchronized coupled-motion axes.

2.5.2 Reset Virtual Master [R_VM]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+6	**R_VM**	S_VM	EN_ RF	SSC	OPERATION		FUNCTION		BIN	IN_7

Meaning

The position value of the virtual master is set to the set position of the virtual master on the rising edge of control signal [S_VM].

NOTE

If control signal [R_VM] is activated while the virtual master is running, the current position value is overwritten by the set position of the virtual master on the rising edge of [R_VM]. As long as [R_VM] is activated, the position value remains at zero.

This jump in the path setpoint has a similarly abrupt effect on the synchronized coupled-motion axes.

2.5.3 Start Virtual Master [ST_VM]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+7	ST_VM	TABLE_NO			SYN_T	SST	ST_S	SET_T	BIN	IN_8

Meaning

The virtual master can be started or stopped with control signal [ST_VM].

[ST_VM] = 1: The virtual master is started.
If the acceleration/deceleration parameter = 0, there is a jump from standstill to the velocity setpoint.
If the acceleration/deceleration parameter <> 0, the velocity setpoint is reached via the acceleration ramp.

[ST_VM] = 0: The virtual master is stopped.
If the acceleration/deceleration parameter = 0, there is a jump from the velocity setpoint to standstill.
If the acceleration/deceleration parameter <> 0, standstill is reached via the deceleration ramp.

2.5.4 Function Running Virtual Master [FUR_VM], Set Speed Reached [VM_RA]

Interface

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+4	STR_M	ARFD	FUR_VM	OTR	FUT	BWD	FWD	DRS	BIN	OUT_5

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+8	CU_TE	CU_VR	CU_PR	SYNC	DI_A	POS_A	CL_A	VM_RA	BIN	OUT_9_0

Meaning

The [FUR_VM] checkback signal indicates the status of the virtual master.

[FUR_VM] = 0: The virtual master is stationary or has a velocity of 0.

[FUR_VM] = 1: The virtual master is running.

[VM_RA] = 1: The virtual master has reached its velocity setpoint (MCT≥V1.31 and SIMATIC Motion Control ≥V1.00.48 or higher)

2.6 Description of the Control and Checkback Signals for the Technological Functions of Master Value Correction and Offset Angle Setting

Interface

The control and checkback signals for master value correction and offset angle setting are stored in the first byte of the optional range.

	7	6	5	4	3	2	1	0		Axis_n.
DBBx+8	CU_ DR	CU_ EN	CU_ SP	SYNC	DI_ RN	DI_ RP	DI_JN	DI_JP	BIN	IN_9_0
DBBx+9	RES	RES	RES	RES	RES	RES	S_ DSP	CU_ TR	BIN	IN_9_1

	7	6	5	4	3	2	1	0		Axis_n.
DBBy+8	CU_ TE	CU_ VR	CU_ PR	SYNC	DI_A	POS_ A	CL_A	VM_ RA	BIN	OUT_9_0

NOTE

The control signals in byte DBBx+8 [ADD_CTRL_SYNC_MODE] and the checkback signals in byte DBBy+8 [ADD_STAT_SYNC_MODE] are available only with MCT and firmware version ≥ V1.31 or higher and SIMATIC MOTION CONTROL ≥ V1.00.48.

Control bit S_DSP in byte DBBx+9 is available only with MCT and firmware version ≥ V1.4x or higher and SIMATIC MOTION CONTROL ≥ V1.00.48.

Meaning

Continual adjustment of the offset angle in jogging mode by controlling one of the inputs, offset angle jog + [DI_JP] = 1 or offset angle jog – [DI_JN] = 1. An adjustment is made for as long as the appropriate bit has the signal status "1".

[DI_JP] = 1 effects an adjustment in a positive direction
[DI_JN] = 1 effects an adjustment in a negative direction

The relative offset angle set is implemented when both control signals for the start are given. With each positive transition on the start inputs, the current offset angle is altered by the appropriate amount.

[DI_RP] = 0 → 1 starts offset angle adjustment in a positive direction

[DI_RN] = 0 → 1 starts offset angle adjustment in a negative direction

The synchronization control signal [SYNC] = 1 activates calculation of the master value correction and implements the corrective movement on the slave axis. The master value synchronization starts with every positive edge of the control signal.

[SYNC] = 0 → 1	starts master value synchronization
[S_DSP] = 0 → 1	a rising edge sets the offset to the parameterized value.

The following checkback signals are available:

[POS_A] = 1	means that position correction is active
[DI_A] = 1	means that offset angle adjustment is active
[SYNC] = 1	means that the axis is synchronized, and balancing movements are complete

NOTE

More information can be found in the description of function, in the chapters "Synchronization mode" and "Synchronization functions".

2.7 Description of Control and Checkback Signals for the Technological Catch-Up Function

Interface

The control and checkback signals for the catch-up function are saved in the first and second byte of the optional range.

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx+8	**CU_ DR**	**CU_ EN**	**CU_ SP**	SYNC	DI_ RN	DI_ RP	DI_JN	DI_JP	BIN	IN_9_0
DBBx+9	RES	RES	RES	RES	RES	RES	**S_ DSP**	**CU_ TR**	BIN	IN_9_1

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy+8	**CU_ TE**	**CU_ VR**	**CU_ PR**	SYNC	DI_A	POS_ A	CL_A	VM_ RA	BIN	OUT_9_0

NOTE

The control signals in byte DBBx+8 [ADD_CTRL_SYNC_MODE] and the checkback signals in byte DBBy+8 [ADD_STAT_SYNC_MODE] are available only with MCT and firmware version ≥ V1.31 or higher.

The control signal in byte DBBx+9 is available only with MCT and firmware version ≥ V1.4x or higher and SIMATIC MOTION CONTROL >= V1.00.48.

Meaning

The catch-up is activated using the control input [CU_SP] "Catch-up / stop".

[CU_SP] = 1 means activate catch-up
[CU_SP] = 0 means stop

After the activation of the control signal "enable positioning" the axis continues moving at the set speed until the parameterized target position can be reached with the return delay and without changing direction.

[CU_EN] = 1 means enable positioning

In the "Isolated operation" mode, the acceleration / deceleration ramp and the curve are switched off. The axis follows every change in set speed without delay.

[CU_DR] =1 means enable acceleration / deceleration ramp

Using the catch-up, you can move the axis into a defined home position. The set home position is triggered by a rising edge of the control signal CU_TR.

[CU_TR] rising edge 0 → 1 means adopt home position

The following checkback signals are available:

[CU_PR] = 1	means that the axis has stopped in the home position. I.e. the catch-up has reached the position.
[CU_VR] = 1	means that the "catch-up set speed" has been reached.
[CU_TE] = 1	means that the axis has reached the synchronous speed. The catch-up function is complete.

NOTE

More information can be found in the description of function, in the chapters "Synchronization mode" and "Synchronization functions".

2.8 Warning Checkback Signals of Speed and Current Controller

Interface

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy	RES	RES	RES	RES	**OTM**	**OTC**	**OLC**	**S MAX**	BIN	OUT_1

Meaning

[SMAX] = 1 Rated motor speed reached or exceeded
[SMAX] = 0: Rated motor speed not reached

[OLC] = 1: Converter overload warning
[OLC] = 0: No converter overload warning

[OTC] = 1: Converter overtemperature warning
[OTC] = 0: No converter overtemperature warning

[OTM] = 1: Motor overtemperature warning
[OTM] = 0: No motor overtemperature warning

2.9 Optional Extension of Control and Checkback Signals [OPTIONAL VALUE 1-3 INPUT], [OPTIONAL VALUE 1-3 OUTPUT]

Interface

Optional extension of control signals:

	7 6 5 4 3 2 1 0	**Axis_n.**
DBWx+10	OPTIONAL VALUE 1 INPUT	IN_9_2
DBDx+12	OPTIONAL VALUE 2 INPUT	IN_10
DBDx+16	OPTIONAL VALUE 3 INPUT	IN_11

Optional extension of checkback signals:

	7 6 5 4 3 2 1 0	**Axis_n.**
DBWy+10	OPTIONAL VALUE 1 OUTPUT	OUT_9_2
DBDy+12	OPTIONAL VALUE 2 OUTPUT	OUT_10
DBDy+16	OPTIONAL VALUE 3 OUTPUT	OUT_11

NOTE

The optional range is reduced by two bytes from MASTERDRIVES MC FW version >V1.3. This means that the data range "OPTIONAL VALUE 1 INPUT" and "OPTIONAL VALUE 1 OUTPUT" (four bytes in size up to FW version V1.2x) is now two bytes in size.

See also chapters 2.1 and 2.2.

Meaning

The control and checkback signals present the fastest interface between the S7-CPU and the technology and are exchanged cyclically (S7 cycle).

Operation of Technology on the MASTERDRIVES MC (MCT)

The optional extension of control signals allows you to control all of the values which are available as parameters on the MASTERDRIVES MC directly and quickly across this channel.

[OPTIONAL VALUES 1 to 3 INPUT] represent Profibus PZD words 6 to 10 on the MASTERDRIVES MC. These words must be associated with the appropriate parameters.

Examples:

- ♦ MDI block comprising G function, position and velocity
- ♦ Gear factor comprising nominator and denominator

The optional extension of checkback signals allows you to transfer all of the values available as binectors or connectors on the MASTERDRIVES MC across this channel for display or further processing on the S7 CPU.

[OPTIONAL VALUES 1 to 3 OUTPUT] represent Profibus PZD words 6 to 10 on the MASTERDRIVES MC. These words must be associated with the appropriate parameters.

Examples:

- ♦ Actual position value
- ♦ Position setpoint

Further Information

For a description of interconnection using BICO technology, please refer to the MASTERDRIVES MC "Compendium".

Operation of Technology on the M7-FM (MCB)

The optional additional control signals mean you can transfer the MDI traversing set 0 in MDI mode.

In both cases, the transferred values are only adopted when the toggle bit [TGL_I] is inverted.

Further Information

A more detailed description of the allocation of the optional range can be found in the chapters "Manual Data Input Mode" and "Synchronization Mode".

---

3 Setup Mode

Contents

In this chapter you will find all the information about setup mode.

Related Terms

Setup mode, JOG mode, JOG, manual mode

Overview

In setup mode, the axis is moved via position control using the "jog forwards" and "jog backwards" keys.

The "fast/slow" speed selection can be used to choose between two separate velocities. It is also possible to use an override for additional velocity control.

The position approached using the direction keys can be saved as a position setpoint for NC programs (teach-in).

Software limit switches are used to prevent the axis from crossing the mechanical limit. If a software limit switch is reached in setup mode, the software limit switch can only then be left by moving in the opposite direction.

Further Information

A description of the software limit switch can be found in chapter 1.6 "Machine data for software limit switch monitoring".

3.1 Function Parameters

3.1.1 Setup Velocities Level 1 and Level 2

Value Range

You can switch between two velocity levels in setup mode. Default values are assigned for the two velocity levels and these can be changed if necessary.

Name	Input Limit		Unit	Default
	Lower	Upper		
Velocity - level 1 (U510.01)	0.01	MD23	1000*LU/min	1000.00
Velocity - level 2 (U510.02)	0.01	MD23	1000*LU/min	10000.00

Selection of the Velocity Level

The velocity level is selected with the "fast/slow" control signal [F_S], where

- [F_S] = 0 activates velocity – level 1 and
- [F_S] = 1 activates velocity – level 2.

The velocities are input/output using the standard user interface available or in STEP 7 with the "input/output setup velocities" task.

Further Information

You will find a description of the "input/output setup velocities" task in the "Task Description" in the chapter entitled "Operating Data Tasks".

3.1.2 Control and Checkback Signals

General Information

The control signals are used to operate the axis in setup mode. The checkback signals indicate the processing status of the axis.

Control Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to setup mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBx						LB=1			BIN	IN_1
DBBx+1	ACK_F				ENC	OFF3	OFF2	OFF1	BIN	IN_2
DBBx+2	MODE_IN = 0001				J_FWD	F_S	J_BWD		BIN	IN_3
DBBx+3	OVERRIDE								DEC	IN_4
DBBx+4									DEC	IN_5
DBBx+5		RST	FUM						BIN	IN_6
DBBx+6	R_VM	S_VM	EN_RF						BIN	IN_7
DBBx+7	ST_VM								BIN	IN_8

DBB address x is calculated as follows:
x = (axis number -1) * 80 + 26

Checkback Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to setup mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBy					OTM	OTC	OLC	SMAX	BIN	OUT_1
DBBy+1		WARN	OFF3	OFF2	FAULT	IOP	RDY	RTS	BIN	OUT_2
DBBy+2	FAULT_NO								DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4
DBBy+4		ARFD	FUR_VM	OTR		BWD	FWD	DRS	BIN	OUT_5
DBBy+5									DEC	OUT_6
DBBy+6	MODE_OUT = 0001				FUR	ST_EN			BIN	OUT_7
DBBy+7									DEC	OUT_8

DBB address y is calculated as follows:
y = (axis number -1) * 80 + 54

3.2 Function Description

3.2.1 Handling by the User

General Information

The following control signals should always be activated:

- ♦ Controller enable [ENC] = 1
- ♦ OFF1 to OFF3 [OFF1..OFF3] = 1

Pulse Diagram

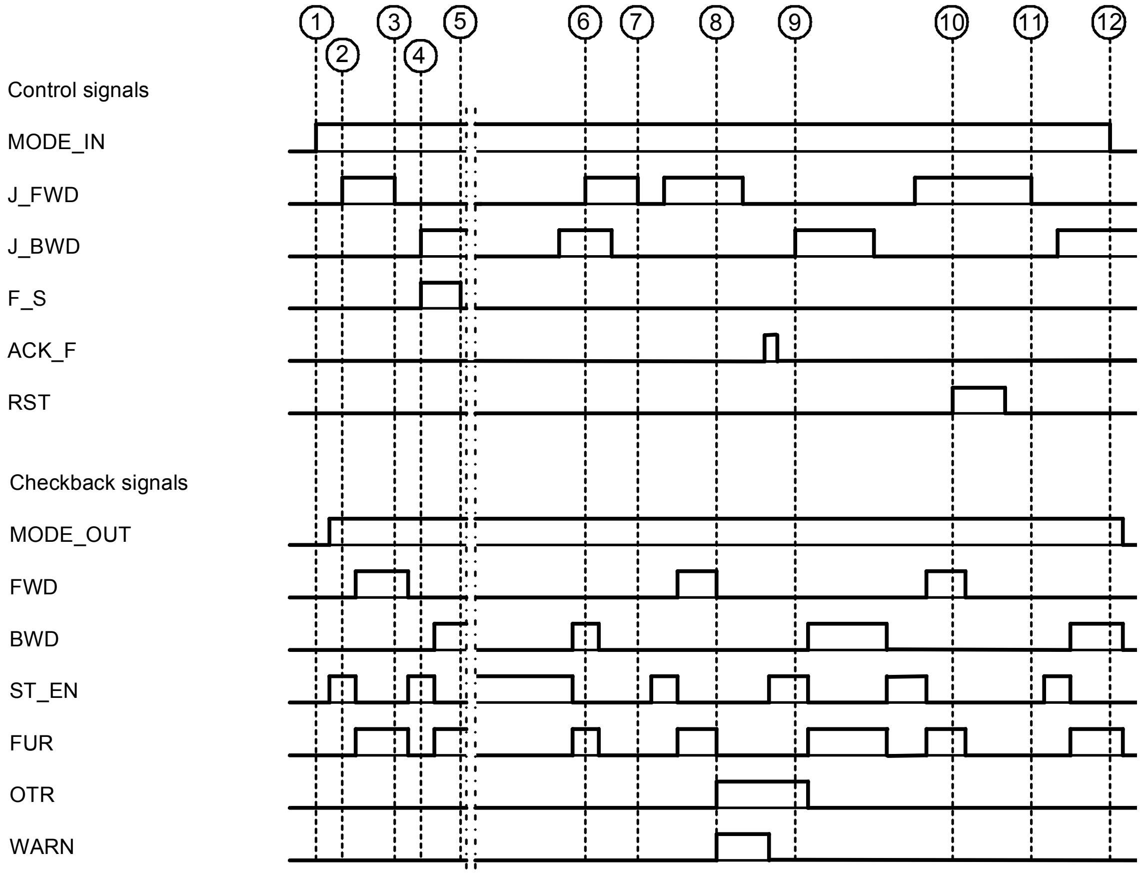


Fig. 3-1 Pulse Diagram for Setup Mode

Pulse Diagram Sequence

(1) The user selects setup mode ([MODE_IN] = 0001). The axis returns setup mode ([MODE_OUT] = 0001) and the start enable ([ST_EN] = 1).

(2) If the start enable is active ([ST_EN] = 1), jog forwards is activated ([J_FWD] = 1). The axis cancels the start enable ([ST_EN] = 0) and activates the signals function running ([FUR] = 1) and axis moves forwards ([FWD] = 1).

(3) Jog forwards is canceled ([J_FWD] = 0). When the axis has come to a standstill via the deceleration ramp, the function running and axis moves forwards signals are canceled ([FUR] = 0 and [FWD] = 0) and the start enable is activated again ([ST_EN] = 1).

(4) Jog forwards is activated ([J_BWD] = 1) together with fast/slow ([F_S] = 1). The axis traverses at the velocity level and returns the signals function running ([FUR] = 1) and axis moves backwards ([BWD] = 1). The start enable is canceled ([ST_EN] = 0).

(5) Switching between fast/slow [F_S] initiates a dynamic transition (acceleration or deceleration ramp) between velocity levels 1 and 2.

Pulse Diagram Sequence, Special Situations

(6) While the axis is moving with jog backwards ([J_BWD] = 1), jog forwards is activated ([J_FWD] = 1). The axis is stopped because of the ambiguous direction command. Axis moves backwards and function running are canceled ([BWD] = 0 and [FUR] = 0). The start enable is not activated ([ST_EN] = 0).

(7) The start enable is activated ([ST_EN] = 1) only when both jog directions have been canceled ([J_FWD] = 0 and [J_BWD] = 0).

(8) The axis moves. During the traversing movement, the positive software limit switch is approached ([OTR] = 1). Axis moves backwards and function running are canceled ([BWD] = 0 and [FUR] = 0). A warning is activated ([WARN] = 1) and the corresponding warning number [WARN_NO] is entered.

(9) When the error has been acknowledged ([ACK_F] = 0->1), the axis is moved away from the software limit switch with jog backwards ([J_BWD] = 1). Axis moves backwards and function running are activated ([BWD] = 1 and [FUR] = 1); the overtravel signal and the start enable are canceled ([OTR] = 0 and [ST_EN] = 0).

(10) During the traversing movement, the reset axis signal is activated ([RST] = 1). The axis stops abruptly; axis moves forwards and function running are canceled ([FWD] = 0 and [FUR] = 0). The axis must be referenced again on systems with incremental encoders.

(11) Only when jog forwards and reset axis are canceled ([J_FWD] = 0 and [RST] = 0) is the start enable activated again ([ST_EN] =1).

(12) Setup mode is deselected during the traversing movement ([MODE_IN] 0001 -> 000x). The axis is stopped via the deceleration ramp. Axis moves backwards and function running are canceled ([BWD] = 0 and [FUR] = 0).

3.2.2 Teach-In

General Information

The teach-in function can be used to save a position which has been approached with control signals "jog forwards" [J_FWD] and/or "jog backwards" [J_BWD] as a position setpoint for NC blocks.

The teach-in function is triggered with the "input teach-in" task.

Further Information

You will find a description of the "input teach-in" task in the "Task Description" in the chapter entitled "Operating Data Tasks".

Procedure

The desired position is approached with control signals [J_FWD] and/or [J_BWD]. The current position is stored in an NC block as the position setpoint with the "input teach-in" task. The desired NC program and NC block number are transmitted with the "input teach-in" task.

3.2.3 Set Floating Reference Point

NOTE

The "floating reference point setting" function is available only with MASTERDRIVES MC firmware version >=V1.4x or higher.

General Information

Setting the floating reference point allows an axis to be referenced in any mode including setup. The reference point is logged by means of a BERO (positive edge) without evaluating the sensor zero mark.

Setting the floating reference point has no influence on the course of the axis. As described in 3.2.1, the axis should be controlled by the user program.

The set position value and the actual position value are corrected on the basis of the discrepancy between the setpoint (reference point coordinate) and the measured value logged by the BERO. The function can also be influenced by an enable signal ([EN_RF]) and by the definition of an inner and an outer window.

Parameters

The set floating reference point function is configured by means of the following parameters:

- Reference point – coordinate (MD3)
- Digital inputs – function 2 (MD46)
- Set floating reference point, inner window (FP2)
- Set floating reference point, outer window (FP3)
- Set correction value for floating reference point (FP4)

Further Information

A description of the parameters can be found in the function description in the chapter on “Machine data and technology parameters".

Pulse Diagram

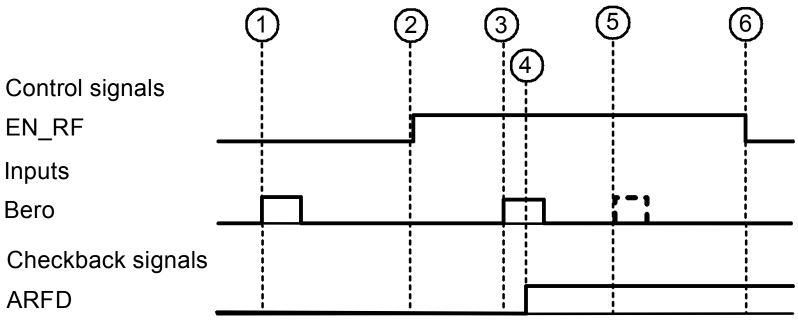


Fig. 3-2 Pulse diagram for setting floating reference point

Pulse Diagram Sequence

Conditions: The user has selected setup mode, the axis is moving and the axis is not referenced.

(1) The axis runs over the BERO and enable is not set. The axis is not referenced.

(2) The user sets the enable ([EN_RF] = 1).

(3) The next rising edge on the BERO triggers the evaluation. As the axis has not yet been referenced ([ARFD] = 0), the reference point is always set.

(4) The checkback signal that the axis is referenced is set ([ARFD] = 1).

(5) As long as the enable is active, the axis is referenced again with every rising edge of the BERO. Setting the reference point depends on the evaluation of the discrepancy between the measured value and the reference point coordinate on the basis of the windows set. Repeated referencing has no influence on the ARFD signal.

(6) The user resets the enable. The BERO is no longer evaluated.

3.3 Setting the Rounding in Setup Mode

NOTE

The option of applying rounding in setup mode is available only with MASTERDRIVES MC firmware version ≥ V1.5 or higher.

3.3.1 Activate Rounding

General Information

In earlier versions, rounding is active in modes MDI, Automatic and Single Block only if you have set the rounding time constant to a value other than zero.

You need to enable rounding in MDI mode explicitly by a configuration setting in the MASTERDRIVES MC.

Parameters

Name	Input Limit		Unit	Default
	Lower	Upper		
Rounding time constant (U505)	0	1 000	ms	0
Enable rounding in setup mode (U512)	0 = Rounding is not enabled (compatible mode) 1 = Rounding is enabled		-	0

In the MASTERDRIVES MC factory setting, the enabling signal for rounding [U512] is preset to fixed binector 0 which means that rounding is not enabled.

NOTE

Scriptfile 9_Scriptfile_MCT_rounding_time_setup_mode.ssc is provided for configuring the rounding function.

As a SIMOVIS/DriveMonitor Scriptfile, this interconnection is included on the SIMATIC Motion Control CD, Configuring Package GMC Basic.

4 Reference Point Approach Mode

Contents

In this chapter you will find all the information about reference point approach mode.

Related Terms

Referencing; Calibration; Synchronization; Setting a floating reference point

NOTE

Setting a floating reference point allows an axis to be referenced in setup, MDI, control, automatic and synchronization mode. If you want to use this function, you can find the appropriate information in the function description in the chapter for the mode you wish to use.

NOTE

Setting the floating reference point is also carried out in the "Reference point approach" mode according to its function. As far as the selected reference point approach is concerned, it does not influence either the process or the result!

Overview

In systems with incremental encoders, there is no relationship between the measuring system (incremental encoder) and the mechanical position of the axis when the control is switched on. It is necessary to approach a defined reference point every time the system is switched on.

The following ways of finding the reference point are available:

- reference point approach with proximity switch (Bero) and zero mark
- set reference point:
  The reference point depends on the mechanical position of the axis on which it is at the point that the reference point is set.

In V1.4 and above of MASTERDRIVES MC, the following additional reference point approach options are also available:

- reference point approach with proximity switch only
- reference point approach with zero mark only
- advanced functionality for the "with proximity switch and zero mark" and "with proximity switch only" reference point approach options using a reverse cam.

In most cases, the reference point approach is used for synchronizing the measurement system, as it has incremental accuracy.

Therefore, setting the reference point is only used if neither a rough pulse (proximity switch) nor a fine impulse is available, or if you wish to synchronize the axis in different places because of the application.

CAUTION

The software limit switches are not effective in reference point approach mode.

NOTE

Reference point approach mode is not available for the roll feed axis type, because roll feeds perform relative movements and do not require an absolute position reference.

4.1 Function Parameters

4.1.1 Reference Point Proximity Switch (Bero)

Reference Point Proximity Switch Reverse Cam

A reference point proximity switch is required for reference point approach. This serves as a warning for the zero pulse (next zero pulse after leaving the reference point proximity switch). The electrical contact for the proximity switch must take the form of a closing contact. The reverse cam helps to find the reference point more quickly.

Information on parameterizing machine data 45 can be found in the chapters on the appropriate reference point approach.

MD No.	I	A	W	Name	Input Limit Upper	Input Limit Lower	Unit	Default
45				Digital inputs – Function 1 (U501.45)	0: No function 1: Start ORed 2: Start ANDed 3: Set actual value on-the-fly 4: External block change 5: Inprocess measurement 6: Collision **7: Proximity switch for reference point approach** **8: Reverse cam for reference point approach** 9: External read-in enable program-dependent		6 digit input	0

4.1.2 Reference Point – Coordinate

Reference Point – Coordinate

The value in the reference point coordinate can be used to match the reference point of the axis to the coordinate system of the machine. When the reference point is reached, the axis sets the actual position value to the value entered in machine data 3.

MD No.	I	A	W	Name	Input Limit Upper	Input Limit Lower	Unit	Default
3				Reference point – coordinate (U501.03)	-999 999.999	999 999.999	1000*LU	0

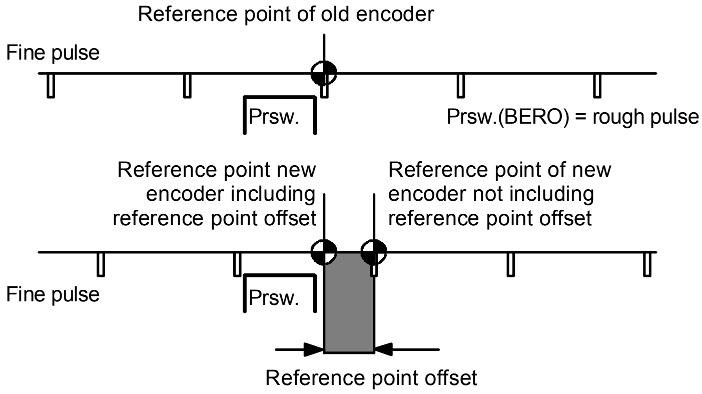
4.1.3 Reference Point – Offset

Reference Point – Offset

In order to prevent the axis from coming to a standstill at the synchronization point after reference point approach/set reference point, but to displace it by a defined amount, you can enter the necessary value in machine data 4. Once the reference point offset has been traversed, the reference point is set to the reference point coordinate.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Upper	Lower		
4				Reference point – offset (U501.04)	-999 999.999	999 999.999	1000*LU	0

Example



After replacement of the position encoder and reference point approach, a position deviation of +147 µm is measured with a dial gauge.

The original position of the reference point is established again by entering a reference point offset of -147 µm.

4.1.4 Reference Point – Approach Direction

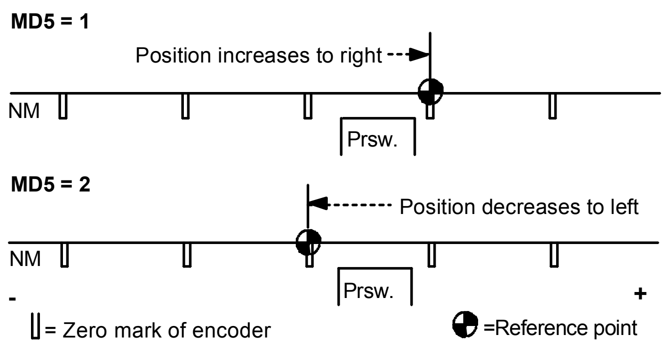
Reference Point – Approach Direction

A setting of 1 or 2 in machine data 5 defines the direction in which the axis is to depart from the reference point proximity switch during reference point approach, in order to use the next zero pulse of the measuring system as a synchronization pulse.

MD No.	I	A	W	Name	Input Limit Upper	Input Limit Lower	Unit	Default
5				Reference point – approach direction (U501.05)	1: Reference point to right of proximity switch 2: Reference point to left of proximity switch 3: Set reference point		-	1

With a rotary position encoder (resolver), the zero pulse is the rotary encoder zero; with an incremental encoder it is the zero mark which repeats once per revolution.

- ♦ Reference point to the right of the proximity switch means the axis departs in the direction of increasing actual values.
- ♦ Reference point to the left of the proximity switch means the axis departs in the direction of decreasing actual values.



As shown in the diagram, there are two possible reference points represented by the two zero marks of the incremental encoder to the left (negative) and right (positive) of the proximity switch.

NOTICE

The proximity switch required for reference point approach must be controlled by a digital input (see machine data 45).

The setting defined here with reference to the position of the reference point **must** match the setting of MASTERDRIVES parameter P183 (reference point detection mode).

A setting of 3 in machine data 5 initiates a "set reference point" operation. At the current position of the axis, the reference point is set to the reference point coordinate or the reference point offset is traversed and the reference point is subsequently set to the reference point coordinate.

4.1.5 Reference Point – Reducing Velocity

Reference Point – Reducing Velocity

Machine data 6 defines the velocity at which the axis departs from the reference point cam in order to search for the zero pulse.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Upper	Lower		
6				Reference point – reducing velocity (U501.06)	1	1 500 000	1000*LU/min	500

NOTE

The reference point reducing velocity cannot be modified by the velocity override [OVERRIDE].

4.1.6 Reference Point – Approach Velocity

Reference Point – Approach Velocity

Machine data 7 defines the velocity at which the reference point cam is to be approached.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Upper	Lower		
7				Reference point – approach velocity (U501.07)	1	1 500 000	1000*LU/min	5000

4.1.7 Reference Point Approach Type

Reference Point Approach

Machine data 8 sets the type of reference point approach.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Upper	Lower		
8				Reference point approach	0: with proximity switch and zero mark 1: with proximity switch only 2: with zero mark only		-	-

4.1.8 Axis Referenced

When the reference point has been successfully approached or set, the axis indicates this status by activating the "axis referenced" [ARFD] checkback signal. From this point on, there is an absolute position reference to the mechanical system. The software limit switches are now also active. These prevent a linear axis from reaching its mechanical limit.

The following conditions cancel the "axis referenced" checkback signal:

- The "reset axis" [RST] control signal
- Switching off the MASTERDRIVES MC

4.1.9 Speed Control or Position Control

Control Status

Reference point approach is performed in speed control and position control mode. From the beginning of the reference point approach until the zero pulse of the position measuring system is reached, the axis moves in speed control mode. The subsequent positioning operation required to reach the reference point or traverse through the reference point offset is performed in position control mode.

This means that the acceleration or deceleration ramps during speed control are derived from machine data 41 (acceleration time, operating mode "reference point approach / control") and 42 (deceleration time, operating mode "reference point approach / control"). During position control, the ramps are derived from machine data 18 (acceleration) and 19 (deceleration).

4.1.10 Control and Checkback Signals

General Information

The control signals are used to operate the axis in reference point approach mode. The checkback signals indicate the current status of the axis.

Control Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to reference point approach mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBx						LB=1			BIN	IN_1
DBBx+1	ACK_F				ENC	OFF3	OFF2	OFF1	BIN	IN_2
DBBx+2	MODE_IN = 0010				J_FWD		J_BWD		BIN	IN_3
DBBx+3	OVERRIDE								DEC	IN_4
DBBx+4									DEC	IN_5
DBBx+5		RST	FUM						BIN	IN_6
DBBx+6	R_VM	S_VM							BIN	IN_7
DBBx+7	ST_VM								BIN	IN_8

DBB address x is calculated as follows:

x = (axis number -1) * 80 + 26

Checkback Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to reference point approach mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBy					OTM	OTC	OLC	S MAX	BIN	OUT_1
DBBy+1		WARN	OFF3	OFF2	FAULT	IOP	RDY	RTS	BIN	OUT_2
DBBy+2	FAULT_NO								DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4
DBBy+4		ARFD	FUR_VM	OTR	FUT	BWD	FWD	DRS	BIN	OUT_5
DBBy+5									DEC	OUT_6
DBBy+6	MODE_OUT = 0010				FUR	ST_EN			BIN	OUT_7
DBBy+7									DEC	OUT_8

DBB address y is calculated as follows:

y = (axis number -1) * 80 + 54

4.2 Function Description

General Information

The following control signals should always be activated:

- ♦ Controller enable [ENC] = 1
- ♦ OFF1 to OFF3 [OFF1..OFF3] = 1

Practical Tips

- ♦ Reference point velocities
  Bearing in mind that you operate the axis manually in reference point approach mode, you should select velocities that do not have an adverse impact on your response time in the event of an error (you may need to vary the override command). In all other operating modes, the software limit switches ensure hazard-free operation (exception: control mode).
- ♦ Reference point
  You should choose a reference point that can be approached easily and rapidly when the control is switched on.
- ♦ Proximity switch (reference point approach “with proximity switch and zero mark”)
  - You should choose a mechanical position for the proximity switch where its switching signal edge (on departure from the proximity switch) is located between two zero pulses.
    Reason:
    If the switching edge is near the zero pulse, the switching edge of the next zero pulse can be masked sometimes and evaluated other times in the event of different response times. This can cause the axis to be displaced sporadically by one encoder revolution after reference point approach.
  - To ensure a unique approach direction to the proximity switch after switching on the control, the rail that actuates the proximity switch should project beyond the mechanically possible traversing area. When the control is switched on, the axis can thus only be located on the proximity switch or on one specific side of the proximity switch.

4.2.1 Reference Point Approach with Proximity Switch and Zero Mark

Function Description

During the reference point approach, the axis moves via a reference point proximity switch (rough pulse) to the zero pulse (fine pulse) of the incremental position encoder or to the zero transition of the resolver signal. The synchronization pulse sets the measurement system to a defined coordinate, which provides the absolute position reference to the mechanical system.

Reference Point Approach

You can use machine data 8 to set other types of reference point approach – the reference point approach used so far ‘with proximity switch and zero mark” is parameterized by setting MD8 to zero.

Reference Point Proximity Switch

A reference point proximity switch is required for reference point approach. This serves as a warning for the zero pulse (next zero pulse after leaving the reference point proximity switch). The electrical contact for the proximity switch must take the form of a closing contact. In the standard settings for MASTERDRIVES MC, the reference point proximity switch should be connected to socket X101 / 8. This corresponds to input I6.

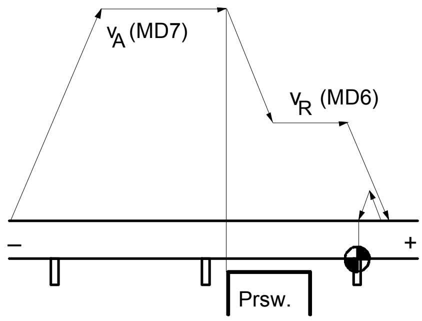
Digital Inputs – Function 1

The value 7xxxxx **must** be entered as machine data 45 in order to communicate the connection point of the proximity switch to the technology.

Initial Status

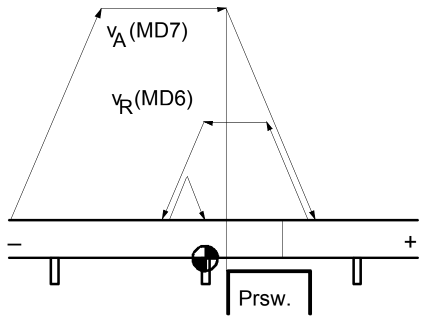
When you switch on the MASTERDRIVES MC, the mechanical position of the axis is not known to the control system. The initial status for reference point approach can be any one of the following.

Axis to Left of Proximity Switch and MD5 = 1



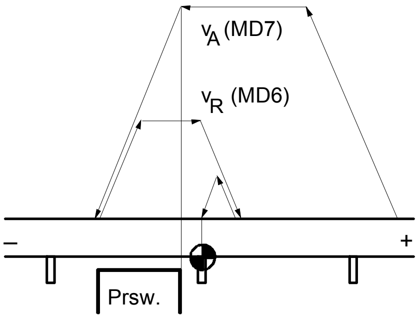
You have to activate the control signal “jog forwards” " [J_FWD]. The axis moves in speed control mode at the “Reference point – approach velocity” V_A [MD7] in the direction of the proximity switch. When the proximity switch trips, the axis decelerates to the "Reference point – reducing velocity" V_R [MD6]. Once the axis leaves the proximity switch in a positive direction, the system searches for the next zero pulse from the position encoder. When the zero pulse is detected the axis is brought to a halt in speed control mode. The resulting deceleration path is canceled by retracting the axis and positioning it on the zero pulse. The axis activates the checkback signals “axis referenced” [ARFD], "destination reached, axis stationary" [DRS] and "function terminated" [FUT].

Axis to Left of Proximity Switch and MD5 = 2



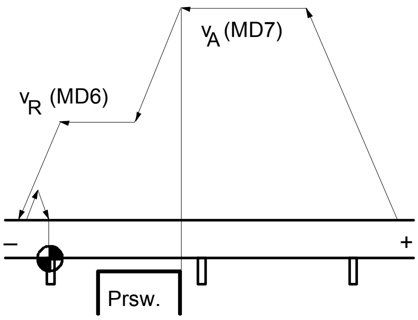
You must activate the "jog forwards" [J_FWD] control signal. The axis traverses in speed control mode at the "reference point – approach velocity" V_A [MD7] in the direction of the proximity switch. When the proximity switch trips, the axis is brought to a standstill, reverses and accelerates to the "reference point – reducing velocity" V_R [MD6]. Once the axis leaves the proximity switch in the negative direction, the search begins for the next zero pulse of the position encoder. When the zero pulse is detected, the axis is brought to a standstill in speed control mode. The resulting deceleration path is canceled by retracting the axis and positioning it on the zero pulse. The axis activates the checkback signals "axis referenced" [ARFD], "destination reached, axis stationary" [DRS] and "function terminated" [FUT].

Axis to Right of Proximity Switch and MD5 = 1



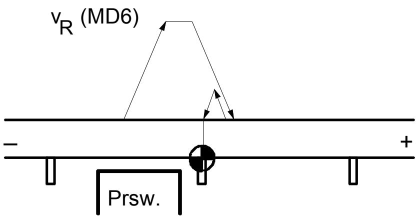
You must activate the "jog backwards" [J_BWD] control signal. The axis traverses in speed control mode at the "reference point – approach velocity" V_A [MD7] in the direction of the proximity switch. When the proximity switch trips, the axis is brought to a standstill, reverses and accelerates to the "reference point – reducing velocity" V_R [MD6]. Once the axis leaves the proximity switch in the positive direction, the search begins for the next zero pulse of the position encoder. When the zero pulse is detected, the axis is brought to a standstill in speed control mode. The resulting deceleration path is canceled by retracting the axis and positioning it on the zero pulse. The axis activates the checkback signals "axis referenced" [ARFD], "destination reached, axis stationary" [DRS] and "function terminated" [FUT].

Axis to Right of Proximity Switch and MD5 = 2



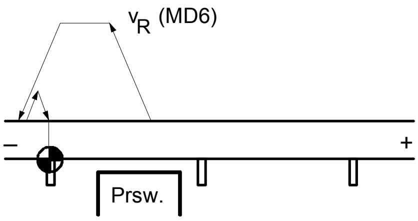
You must activate the "jog backwards" [J_BWD] control signal. The axis traverses in speed control mode at the "reference point – approach velocity" V_A [MD7] in the direction of the proximity switch. When the proximity switch trips, the axis decelerates to the "reference point – reducing velocity" V_R [MD6]. Once the axis leaves the proximity switch in the negative direction, the search begins for the next zero pulse of the position encoder. When the zero pulse is detected, the axis is brought to a standstill in speed control mode. The resulting deceleration path is canceled by retracting the axis and positioning it on the zero pulse. The axis activates the checkback signals "axis referenced" [ARFD], "destination reached, axis stationary" [DRS] and "function terminated" [FUT].

Axis Already On Proximity Switch and MD5 = 1



Since the axis is already on the proximity switch, the direction to the reference point is known. It is therefore irrelevant whether you activate the "jog forwards" [J_FWD] or "jog backwards" [J_BWD] control signal. The axis accelerates in speed control mode to the "reference point reducing velocity" V_R and leaves the proximity switch in the positive direction. When the zero pulse is detected, the axis is brought to a standstill in speed control mode. The resulting deceleration path is canceled by retracting the axis and positioning it on the zero pulse. The axis activates the checkback signals "axis referenced" [ARFD], "destination reached, axis stationary" [DRS] and "function terminated" [FUT].

Axis Already On Proximity Switch and MD5 = 2



Since the axis is already on the proximity switch, the direction to the reference point is known. It is therefore irrelevant whether you activate the "jog forwards" [J_FWD] or "jog backwards" [J_BWD] control signal. The axis accelerates in speed control mode to the "reference point reducing velocity" V_R and leaves the proximity switch in the negative direction. When the zero pulse is detected, the axis is brought to a standstill in speed control mode. The resulting deceleration path is canceled by retracting the axis and positioning it on the zero pulse. The axis activates the checkback signals "axis referenced" [ARFD], "destination reached, axis stationary" [DRS] and "function terminated" [FUT].

Reference Point Coordinate

When the reference point is reached, the position setpoint and the actual position value are set to the "reference point coordinate" (MD3).

Reference Point Offset

If a "reference point offset" [MD4] has been entered, the axis traverses this distance on the next zero pulse of the encoder. The movement takes place as follows, depending on the sign of the reference point offset:

- ♦ If the axis is to traverse through the reference point offset in the direction in which it left the proximity switch, it decelerates to a standstill in speed control mode when the zero mark is detected. It then traverses in position control mode through the offset (positioning movement) in the same direction at velocity V_R.
- ♦ If the axis is to traverse through the reference point offset in the opposite direction to departure from the proximity switch, it decelerates to a standstill in speed control mode when the zero mark is detected. It then traverses in position control mode through the offset (positioning movement) in the opposite direction at velocity V_R.

The position setpoint and the actual position value are then set to the "reference point coordinate" and the checkback signals "axis referenced" [ARFD], "destination reached, axis stationary" [DRS] and "function terminated" [FUT] are activated.

NOTE

If the "jog forwards" [J_FWD] or "jog backwards" [J_BWD] control signals are canceled during the reference point approach, the reference point approach is interrupted:

- ♦ If the axis is still positioned in front of or on top of the proximity switch after the interruption, the reference point approach can be continued.
- ♦ If the proximity switch has already been crossed after the interruption, the reference point approach must be restarted in the opposite direction (the axis is now located on the other side of the proximity switch).

Since the axis has to reduce its velocity from V_A to V_R via the deceleration setting (MD42) when the proximity switch is reached, and it is not permitted to cross the proximity switch, the proximity switch must be sufficiently long.

The velocity override [OVER] only affects the "reference point – approach velocity".

WARNING

Since no software limit switch monitoring is active before the measuring system has been synchronized, you must ensure that the reference point approach is not started in the wrong direction.

4.2.2 Reference Point Approach with Proximity Switch Only

NOTE

The reference point approach option “with proximity switch only” is only available in MASTERDRIVES MC from firmware version >= V1.4.

Function Description

The reference point is set with the falling edge (1→0) of the reference point proximity switch. All other parameterizations – reference point coordinate MD3, reference point shift MD4, reference point approach direction MD5, reference point reducing velocity MD6 and reference point approach velocity MD7 – remain valid.

Reference Point Approach

For reference point approach “with proximity switch only”, you have to set machine data 8 to “1”.

Reference Point Proximity Switch

A reference point proximity switch is required for reference point approach. The electrical contact for the proximity switch must take the form of a closing contact. The reference point proximity switch must be connected to socket X101 / 6 or X101 / 7 on MASTERDRIVES MC. This corresponds to inputs E4 or E5.

Digital Inputs – Function 1

In this case, the value x7xxxx or xx7xxx **must** be entered as machine data 45 in order to communicated the proximity switch connection point to the technology.

NOTE

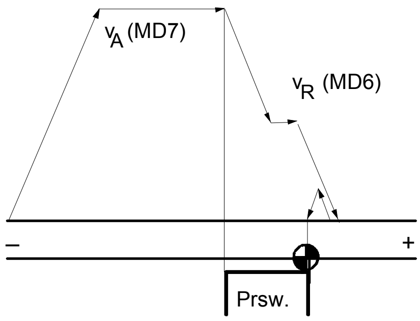
The proximity switch required for the reference point approach must be set to digital input 4 or 5, as these are the only ones which can trigger an interrupt. See also Chapter 1, machine data 45.

In addition, you do need to change one setting in MASTERDRIVES MC:

Parameter 647.1 or 648.1 = 4
    With the 1 -> 0 edge, input 4 (647.1) or
    or input 5 (648.1) triggers an interrupt.

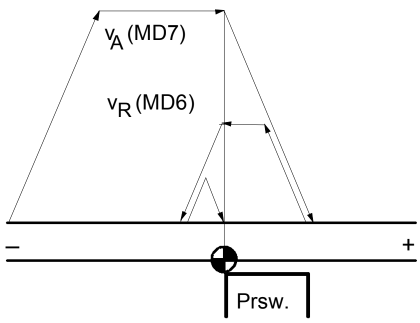
Parameter 178 = 0x16 or 0x18
    Rough pulse parameterization:
    Input 4 (0x16 = 16 Hex) or
    Input 5 (0x18 = 18 Hex).

Axis to Left of Proximity Switch, MD5 = 1 and MD8 = 1



You have to activate the control signal “jog forwards” " [J_FWD]. The axis moves in speed control mode at the “reference point – approach velocity" V_A [MD7] in the direction of the proximity switch. When the proximity switch trips, the axis decelerates to the "reference point – reducing velocity" V_R [MD6]. Once the axis leaves the proximity switch in a positive direction, or, to be more precise, with the falling edge of the proximity switch input, the measured value memory is read and the axis is brought to a halt in speed control mode. The resulting deceleration path is cancelled by means of a position-controlled retraction movement (positioning) to the position of the transition of the proximity switch input (difference between measured value memory and rest position). The axis activates the checkback signals “axis referenced” [ARFD], "destination reached, axis stationary" [DRS] and "function terminated" [FUT].

Axis to Left of Proximity Switch, MD5 = 2 and MD8 = 1



You must activate the "jog forwards" [J_FWD] control signal. The axis traverses in speed control mode at the "reference point – approach velocity" V_A [MD7] in the direction of the proximity switch. When the proximity switch trips, the axis is brought to a standstill, reverses and accelerates to the "reference point – reducing velocity" V_R [MD6]. Once the axis leaves the proximity switch in the negative direction, or, to be more precise, with the falling edge of the proximity switch input, the measured value memory is read and the axis is brought to a halt in speed control mode. The resulting deceleration path is cancelled by means of a position-controlled retraction movement (positioning) to the position of the transition of the proximity switch input (difference between measured value memory and rest position). The axis activates the checkback signals “axis referenced” [ARFD], "destination reached, axis stationary" [DRS] and "function terminated" [FUT].

4.2.3 Reference Point Approach with Zero Mark Only

NOTE

The reference point approach option “with zero mark only” is only available in MASTERDRIVES MC from firmware version >= V1.4.

Function Description

Once reference point approach “with zero mark only” mode has been started, a proximity switch signal is reproduced internally for four cycles of the controller. At the end of this time, the reference point is set when the first zero mark is detected. All other parameterizations – reference point coordinate MD3, reference point shift MD4, reference point approach direction MD5 and reference point reducing velocity MD6 – remain valid.

Reference Point Approach

For reference point approach “with zero mark only”, you have to set machine data 8 to “2”.

Reference Point Proximity Switch

No reference point proximity switch is required.

NOTE

If the reference point approach is “with zero mark only”, the proximity switch is reproduced internally. In order that a specific reference point can be set, you need to change one setting in the MASTERDRIVES MC:

Parameter 178 = 0x308
Rough pulse parameterization. (proximity switch rough pulse is reproduced internally)

4.2.4 Reference Point Approach with Reverse Cam

NOTE

Reference point approach “with reverse cam” is only available in MASTERDRIVES MC from firmware version >= V1.4.

Function Description

The reference point approach with reverse cam option represents an addition to the reference point approach options “with proximity switch and zero mark” and “with proximity switch only”.

If the reverse cam is active (1 signal) when the reference point approach begins or if it is activated during the approach, then the traversing direction automatically switches if the reference point proximity switch is not tripped or does not trip during the approach. Then the axis moves in a negative direction until the reference point proximity switch trips.

If the proximity switch comes first after the start, the “reverse cam” input is not evaluated any more. Please refer to the chapters above for the sequence of the reference point approach from then on.

Reference Point Approach

The reference point approach with reverse cam can only be used in conjunction with the “with proximity switch and zero mark” or “with proximity switch only” reference point approach options. Machine data 8 must be set to “0” or “1”.

Reference Point Proximity Switch

Please refer to the above chapters on “with proximity switch and zero mark” or “with proximity switch only” reference point approach options.

Digital Inputs – Function 1

In this case, the value 87xxxx or x87xxx **must** be entered as machine data 45 in order to communicate the proximity switch connection point to the technology.

NOTE

The notes under above chapters on “with proximity switch and zero mark” or “with proximity switch only” are still valid.

Axis is Between Proximity Switch and Reverse Cam or on Reverse Cam MD5 = 1 and MD8 = 1

You have to activate the control signal “jog forwards” [J_FWD]. The axis moves in speed control mode at “Reference point – approach velocity" V_A [MD7]] in a positive direction. When the reverse cam is tripped, the axis is decelerated to a standstill, reversed and accelerated back up to the “Reference point – approach velocity" V_A [MD7].

When the proximity switch is tripped, the axis decelerated to a standstill, reversed and accelerated back up to the “Reference point – reducing velocity" V_R [MD6]. When the axis leaves the proximity switch, or more specifically with the falling edge of the proximity switch input, the measured value memory is read and the axis is brought to a standstill in speed control mode. The resulting deceleration path is cancelled by means of a position-controlled retraction movement (positioning) to the position of the transition of the proximity switch input (difference between measured value memory and rest position). The axis activates the checkback signals “axis referenced” [ARFD], "destination reached, axis stationary" [DRS] and "function terminated" [FUT].

4.2.5 Set Reference Point Mode

Function Description

In set reference point mode, the coordinate is set immediately by the user program immediately it is started up. This means that the reference point depends on the mechanical position of the axis on which it is when the reference is set.

Set Reference Point when MD 4 = 0

Activating the control signal “jog forwards” [J_FWD] or "jog backwards" [J_BWD] sets the position setpoint and the actual position value to the “reference point - coordinate" [MD3] and activates the checkback signals "Axis referenced" [ARFD] and "Function terminated" [FUT]. The checkback signal “Destination reached, axis stationary " [DRS] is not activated, as no positioning (axis movement) has taken place.

Set Reference Point when MD 4 >< 0

Activating the control signals “jog forwards” [J_FWD] or "jog backwards" [J_BWD] accelerates the axis in a positive or negative direction depending on the sign of the reference point shift to the “reference point – reducing velocity” V_R [MD6] in speed control mode and the reference point shift is implemented. Finally, the position setpoint and the actual position value is set to the “reference point - coordinate " [MD3] and the checkback signals “Axis referenced” [ARFD], "Destination reached, axis stationary" [DRS] and "Function terminated" [FUT] are activated.

If set reference point is used, input I6 can be used for other functions.

4.2.6 Special Cases

Simulation

If the technology is switched to simulation mode, digital input I6 (proximity switch) must be activated anyway in order to simulate the reference point approach. In simulation mode, the falling edge of the proximity switch is evaluated as the zero pulse.

Absolute Position Encoders

The “reference point approach” mode cannot be selected. Zero (no mode) is output as the mode checkback signal [MODE_OUT]. The checkback signal "Axis referenced " [ARFD] is always activated.

Line Fed

The “reference point approach” mode cannot be selected. Zero (no mode) is output as the mode checkback signal [MODE_OUT]. The checkback signal "Axis referenced " [ARFD] is never activated.

4.2.7 Handling by the User

Pulse Diagram

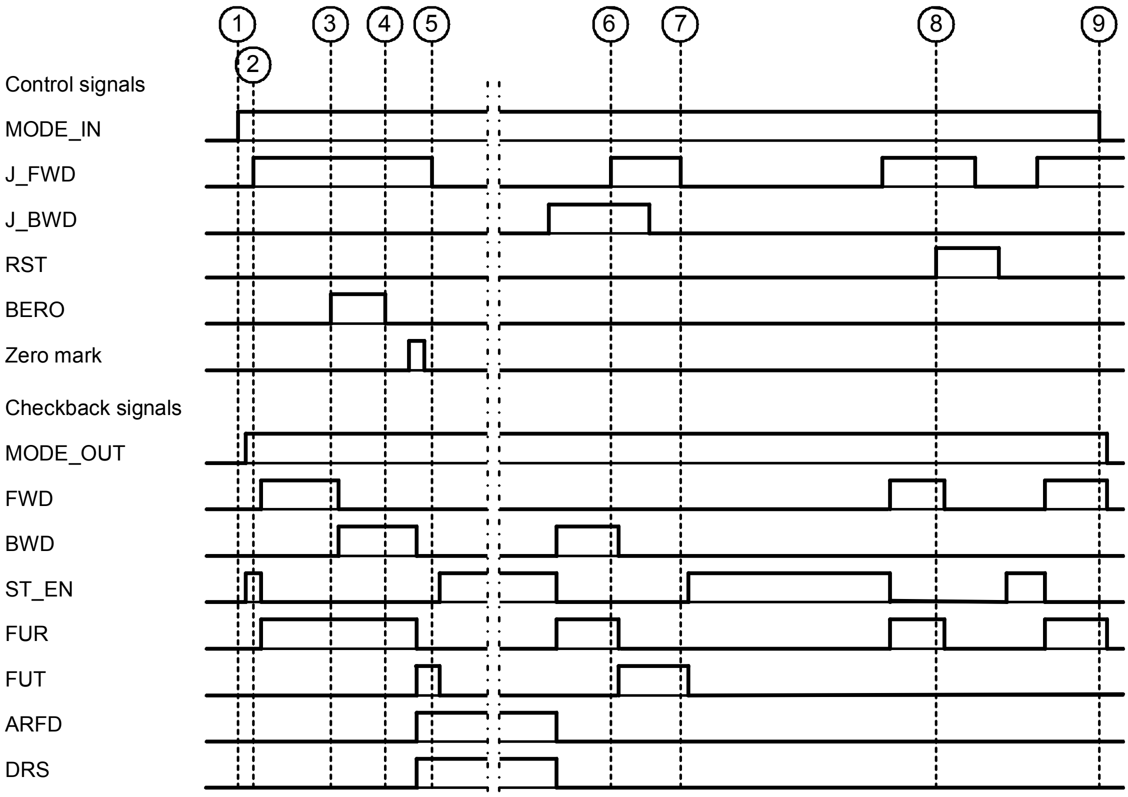


Fig. 4-1 Pulse Diagram for Reference Point Approach Mode

Pulse Diagram Sequence

① The user selects the mode ([MODE_IN] = 0010). The axis returns the mode ([MODE_OUT] = 0010) and the start enable ([ST_EN] = 1).

② If the start enable is active ([ST_EN] = 1), "jog forwards" is activated ([J_FWD] = 1). The axis accelerates to the "reference point – approach velocity", cancels the start enable ([ST_EN] = 0) and activates the checkback signals "function running" ([FUR] = 1) and "axis moves forwards" ([FWD] = 1).

③ When the proximity switch is reached, the axis stops, "axis moves forwards" is canceled ([FWD] = 0), "axis moves backwards" is activated ([BWD] = 1) and the axis accelerates to the "reference point – reducing velocity".

④ When the axis moves away from the proximity switch, the search begins for the next zero pulse of the encoder. The measuring system is synchronized when the zero pulse is detected. Checkback signals "axis moves backwards" and "function running" are canceled ([BWD] = 0 and [FUR] = 0). "Axis referenced", "destination reached, axis stationary" and "function terminated" are activated ([ARFD] =1; [DRS] = 1; [FUT] = 1).

⑤ When the axis has been referenced, "jog forwards" can be canceled ([J_FWD] = 0). The axis activates the "start enable" again ([ST_EN] = 1) and cancels "function running" ([FUR] = 0).

Pulse Diagram Sequence, Special Situations

⑥ While the axis is moving with jog backwards ([J_BWD] = 1), “jog forwards” is activated ([J_FWD] = 1). The axis is stopped because of the ambiguous direction command. "Axis moves backwards" and "function running" are canceled ([BWD] = 0 and [FUR] = 0). "Function terminated" is activated ([FUT] = 1).

⑦ The start enable is not activated ([ST_EN] = 1) until "jog forwards" and "jog backwards" have been canceled ([J_FWD] = 0 and [J_BWD] = 0). "Function terminated" is canceled ([FUT] = 0).

⑧ While the axis is moving, the reset axis signal is activated ([RST] = 1). The axis stops abruptly. "Axis moves forwards" and "function running" are canceled ([FWD] = 0 and [FUR] = 0). "Function terminated" is activated ([FUT] = 1). An existing "axis referenced" ([ARFD] = 0) is lost. The axis has to be referenced again.

⑨ The mode is changed during the traversing movement ([MODE_IN] <> 0010). The axis is stopped via the deceleration ramp. The mode checkback signal is switched ([MODE_OUT] <> 0010); "axis moves backwards" and "function running" are canceled ([BWD] = 0 and [FUR] = 0).

5 Manual Data Input Mode

Inhalt

In this chapter you will find all the information about manual data input mode.

Related Terms

Manual input, incremental traversing

Overview

MDI with axis type "axis with incremental or absolute measuring system":

- Individual MDI blocks consisting of a position and a velocity can be executed in MDI mode.
- A total of 10 MDI blocks per axis are available for loading and permanent storage on the technology.
- You can select an MDI block by specifying an MDI number between 1 and 10, and execute it by activating the start signal.
- The traversing velocity can also be controlled with the override.
- Software limit switches prevent the mechanical limits from being crossed. If a software limit switch is reached in MDI mode, the software limit switch can only then be left by moving in the opposite direction.
- If an MDI block with incremental dimensions is loaded, the programmed distance is traversed on every start (incremental traversing).

MDI with axis type "roll feed":

With the "roll feed" axis type, the positioning movement is not triggered by an active start signal until the read-in enable or an external read-in enable (programmable digital input) is activated. This achieves very high signal processing rates. The actual value is set to zero on each positioning movement. It is also possible to specify an MDI loop count, in order to restrict the number of positioning movements.

The roll feed can be operated with the MDI numbers 0 to 10. However, a floating MDI using MDI number 0 is not required.

MDI number 0 / MDI on-the-fly:

In order to further accelerate the transfer of MDI data, it is possible to place the MDI data in the optional control signal area → MDI number 0.

The Toggle-In [TGL_I] control signal can be used to store new MDI data on top of MDI positioning movements which have already been started. This process is performed on-the-fly → MDI on-the-fly.

5.1 Function Parameters

5.1.1 MDI Block

MDI Block

Up to 10 NC blocks can be initialized per axis for MDI mode. These can then be selected by specifying the MDI number.

The following input can be implemented in the MDI block:

Name	Input Limit		Unit	Pre-assign-ment
	Lower	Upper		
1. G function preparatory function (U550.01 to U559.01)	G90 absolute dimensions (not with roll feed) G91 incremental dimensions (not with MDI on-the-fly)		-	G90 G91 with RF
2. G function acceleration / deceleration response (U550.01 to U559.01)	G30 100 % acc./dec. G31 10 % acc./dec. : G39 90 % acc./dec.		-	G30
Position (U550.02 to U559.02)	-999 999.999	999 999.999	1000*LU	-
Velocity (U550.03 to 559.03	0.01	MD23	1000*LU/min	-

The first G function and the 2nd G function are initialized with the default settings in the table and only need to be entered if the values are to be changed. The position and velocity must be entered at least once. Programming G30 causes the axis to traverse at 100 % acceleration (MD18) and 100 % deceleration (MD19).

Input/output of MDI blocks is performed using the standard user interfaces available or in STEP 7 with the "input/output MDI block" task.

NOTE

The MDI number is stored in parameter P2540, Index 13. If nothing is traveling, the pre-selected MDI number is displayed. During traversing movement, the actual MDI number in use is displayed.

Further Information

The "input/output MDI block" task is described in the "Task Description" in the chapter entitled "MDI Data Tasks".

Optional Control Signals

Example: use for MDI data
The data for MDI number 0 (MDI and floating MDI) are pre-set in the optional range for the control signals.

	7 6 5 4 3 2 1 0		Axis_n.
DBBx+10	1st G function	DEC	IN_9_2
DBBx+11	2nd G function	DEC	IN_9_2
DBDx+12	Position in [LU]	DEC	IN_10
DBDx+16	Velocity in [10 * LU/min]	DEC	IN_11

Assignment of Optional Control Signals

Since the assignment of the optional control signal area is configurable, it is necessary to inform the technology which data are available on the optional control signals.

♦ Technology on the MASTERDRIVES MC (MCT):
The optional values are transferred to the MASTERDRIVES MC via the bus. PZD words 6 to 10 represent the optional values and must be wired to MDI block number 0.

The wiring is as follows for the above configuration:

Word 6 (1st and 2nd G functions) is wired to parameter U531
→ U531 = K3006

Words 7 and 8 (position) are wired to parameter U532
→ U532 = KK3037

Words 9 and 10 (velocity) are wired to parameter U533
→ U533 = KK3039

Further Information

For possible circuits, please refer to the MASTERDRIVES MC documentation entitled "Compendium".

♦ Technology on the SIMATIC Motion Control:
(under development)

5.1.2 Roll Feed Loop Count

MDI Loop Counter

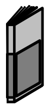
A loop counter can be initialized specially for the "roll feed" version. The counter is decremented by 1 on each positioning movement of the axis.

Name	**Input Limit**		**Unit**	**Pre-assignment**
	Lower	**Upper**		
MDI loop counter	0	2147483647	-	0

When the loop counter reaches 0, further positioning movements cannot be performed until the start control signal is activated again.

If no loop count or a loop count of 0 is entered, there is no limit to the number of positioning movements. The entered loop count can be changed at any time and is active immediately.

Input/output of the MDI loop count is performed using the standard user interfaces available or in STEP 7 with the "input/output roll feed loop count MDI" task.



Further Information

The "input/output roll feed loop count MDI" task is described in the "Task Description" in the chapter entitled "MDI Data Tasks".

5.1.3 Control and Checkback Signals

General Information

The control signals are used to operate the axis in manual data input mode. The checkback signals indicate the current status of the axis.

Control Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to manual data input mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx						LB=1			BIN	IN_1
DBBx+1	ACK_F				ENC	OFF3	OFF2	OFF1	BIN	IN_2
DBBx+2	MODE_IN = 0011				J_FWD		J_BWD		BIN	IN_3
DBBx+3	OVERRIDE								DEC	IN_4
DBBx+4	MDI_NO								DEC	IN_5
DBBx+5		RST	FUM		CRD	STA		TGL_I	BIN	IN_6
DBBx+6	R_VM	S_VM	EN_RF						BIN	IN_7
DBBx+7	ST_VM								BIN	IN_8

DBB address x is calculated as follows:
x = (axis number -1) * 80 + 26

Checkback Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to manual data input mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy					OTM	OTC	OLC	SMAX	BIN	OUT_1
DBBy+1		WARN	OFF3	OFF2	FAULT	IOP	RDY	RTS	BIN	OUT_2
DBBy+2	FAULT_NO								DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4
DBBy+4		ARFD	FUR_VM	OTR	FUT	BWD	FWD	DRS	BIN	OUT_5
DBBy+5									DEC	OUT_6
DBBy+6	MODE_OUT = 0011				FUR	ST_EN		TGL_O	BIN	OUT_7
DBBy+7									DEC	OUT_8

DBB address y is calculated as follows:
y = (axis number -1) * 80 + 54

5.2 Description of Function

General Information

The following control signals should always be activated:

- ♦ Controller enable [ENC] = 1
- ♦ OFF1 to OFF3 [OFF1..OFF3] = 1

Practical Tips

MDI mode is mainly used for individual positioning movements where a completely variable sequence, coordinated by the user program, is possible.

G91 (incremental dimensions) can be used for incremental traversing.

e.g.: G91 X0.01 [mm] F10000 is transferred once to the axis.
Each axis start initiates a movement of 10 µm

5.2.1 Handling by the User for Axis Type with Incremental or Absolute Encoder

Pulse Diagram

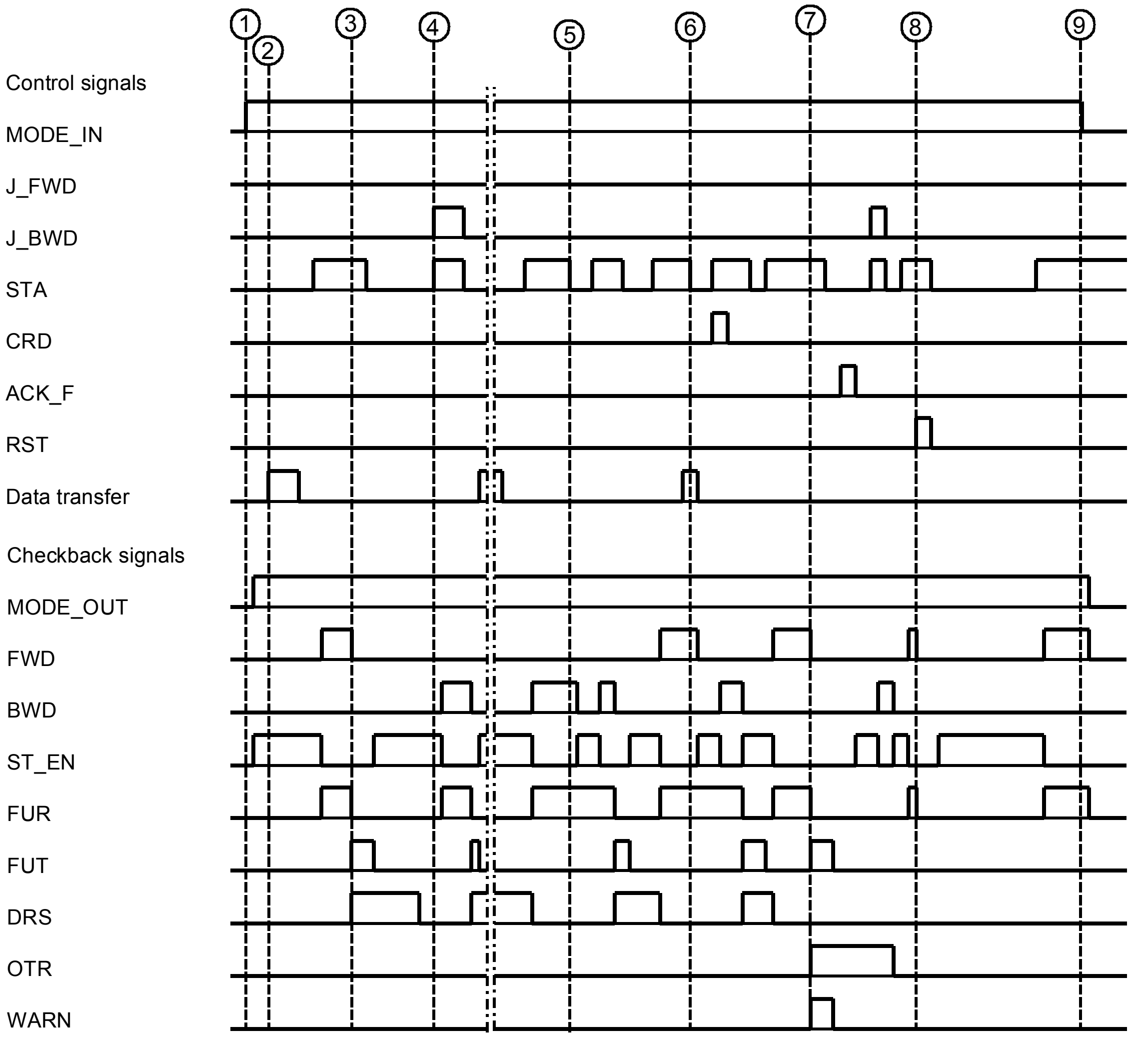


Fig. 5-1 Pulse Diagram for MDI Mode with Axis Type "Axis"

Pulse Diagram Sequence

1. The user selects MDI mode ([MODE_IN] = 0011). The axis returns the mode ([MODE_OUT] = 0011) and the start enable ([ST_EN] = 1).
2. The start signal can be activated ([STA] = 1) when the MDI block has been loaded. The axis cancels the start enable ([ST_EN] = 0) and activates the signals "axis moves forwards" ([FWD] = 1) or "axis moves backwards" ([BWD] = 1) and "function running" ([FUR] = 1).

3. When the specified position is reached, the axis activates "destination reached, axis stationary" ([DRS] = 1); checkback signals "axis moves forwards" and "function running" are reset ([FWD] = 0 and [FUR] = 0). "Function terminated" ([FUT] = 1) is activated. The user can now cancel the "start" control signal again ([STA] = 0). Canceling the "start" signal [STA] resets the "function terminated" signal ([FUT] = 0) and reactivates the "start enable" ([ST_EN] = 1).
4. The direction of rotation on rotary axes can be forced when programming with absolute dimensions. In this case, for example, [J_BWD] = 1 must be activated in addition to the "start" signal ([STA] = 1).

NOTE

Because 10 different MDI blocks are available, you must specify the MDI number when loading an MDI block. When the MDI block is started, the associated MDI number [MDI_NO] must also be specified in the control signals.

Pulse Diagram Sequence, Special Situations

5. If the "start" control signal is canceled during the positioning movement ([STA] = 0), the axis stops. "Axis moves backwards" is reset ([BWD] = 0), and the start enable is activated ([ST_EN] = 1). "Function running" ([FUR] = 1) remains active, and "destination reached, axis stationary" is not output ([DRS] = 0) because the positioning movement is not complete. When the "start" signal is activated again ([STA] = 1), the positioning movement is completed and "axis moves backwards" is activated again ([BWD] = 1). The start enable is reset ([ST_EN] = 0).
6. If the "start" signal is canceled during the positioning movement ([STA] = 0), the axis stops. "Axis moves forwards" is reset ([FWD] = 0), and the start enable is activated ([ST_EN] = 1). When a new MDI block has been loaded, "start" is activated again ([STA] = 1) and "cancel remaining distance" is selected ([CRD] = 1). The axis cancels the remaining distance of the old positioning movement and starts executing the new MDI block. "Axis moves backwards" is activated ([BWD] = 1) and the start enable is reset ([ST_EN] = 0). If a new MDI number is specified following an interruption in an MDI positioning operation, any remaining distance is always canceled. The remaining distance is also canceled by a mode change.
7. If the software limit switch is approached ([OTR] = 1) during a positioning movement, "axis moves forwards" and "function running" are canceled ([FWD] = 0 and [FUR] = 0). Checkback signals "function terminated" and "warning" are activated ([FUT] = 1 and [WARN] = 1). The warning number is entered in [WARN_NO]. Since "function terminated" is active ([FUT] = 1), the "start" signal can be canceled ([STA] = 0). The start enable is not output again ([ST_EN] = 1) until the fault has been acknowledged with "acknowledge fault" [ACK_F]. "Overtravel" remains active ([OTR] = 1) and the axis must move away from the software limit switch in the opposite direction before the "overtravel" signal is canceled again ([OTR] = 0).

8. If the "reset technology" signal is activated ([RST] = 1) during a positioning movement, the position controller is disabled abruptly. "Axis moves forwards" and "function running" are canceled ([FWD] = 0 and [FUR] = 0).
9. The mode is changed during the traversing movement ([MODE_IN] <> 0011). The axis is stopped via the deceleration ramp. The mode checkback signal is switched [MODE_OUT]. The "axis moves forwards" and "function running" signals are reset ([FWD] = 0 and [FUR] = 0).

MDI number 0:
MDI number 0 is transferred to the technology via the optional range for the control signals. The optional range must be hard wired to the MDI traversing set 0 on the MASTERDRIVES MC.

MDI on-the-fly:
With MDI on-the-fly, the MDI block is also supplied via MDI block 0. The only difference is that the toggle bit is used for activation. If the state of the toggle bit [TGL_I] is inverted when the "start" signal is activated ([STA] = 1), the MDI positioning operation is changed on-the-fly. In other words, if [TGL_I] is inverted during an MDI positioning operation, the new positioning movement is performed immediately using MDI block 0. Both the position and the velocity are accepted on-the-fly (i.e. without stopping if no direction reversal is necessary).
When [TGL_I] is inverted, MDI block 0 is adopted, irrespective of the selection in [MDI_NO].
A toggle bit checkback signal also exists [TGL_O]. This signal switches in the same way as [TGL_I] and confirms the data transfer.

Special situation: combination of MDI and MDI on-the-fly:
If a value between 1 and 10 is specified in [MDI_NO], the corresponding MDI block is started when [STA] = 1. If [TGL_I] is inverted during the positioning movement, the positioning operation is replaced immediately with the MDI block from the optional control signal area.

5.2.2 Handling by the User with Roll Feed Axis

Pulse Diagram

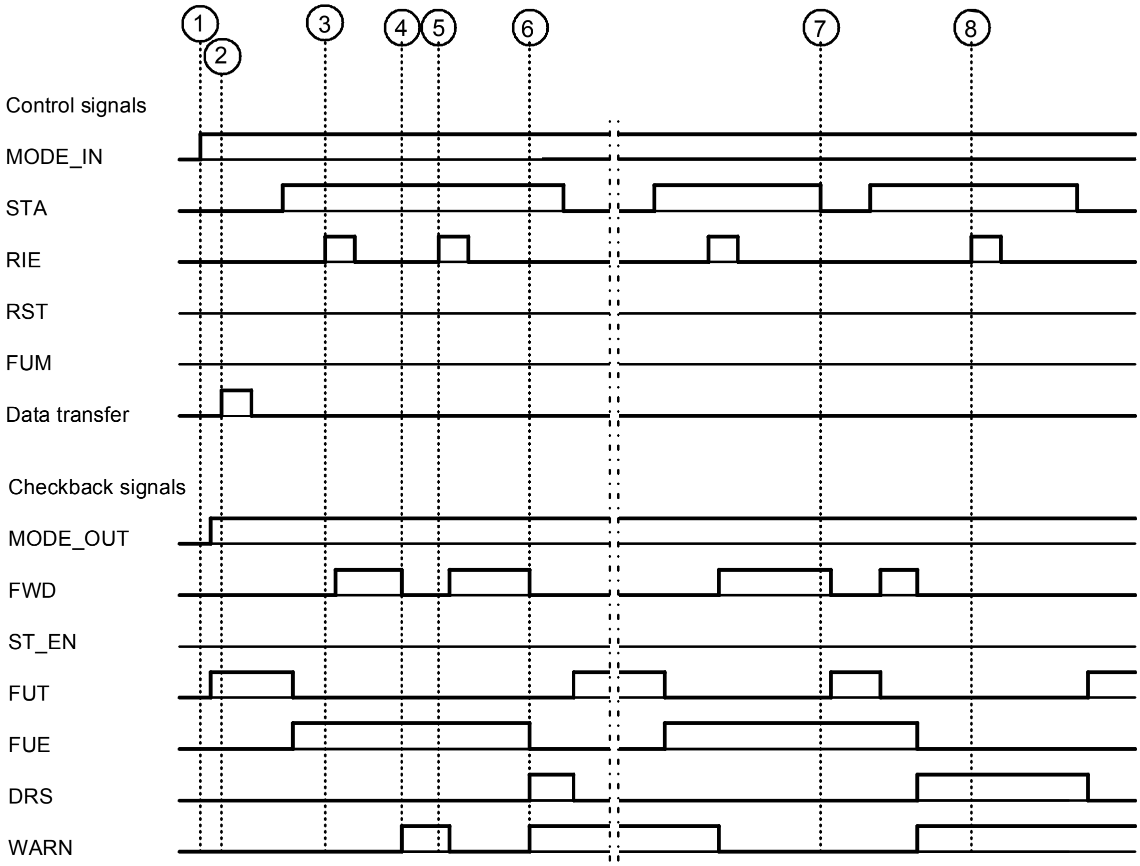


Fig. 5-2 Pulse Diagram for MDI Mode with Axis Type "Roll Feed"

Pulse Diagram Sequence

1. The user selects the mode ([MODE_IN] = 0011). The roll feed returns the mode ([MODE_OUT] = 0011) and the start enable ([ST_EN] = 1).
2. When the MDI block and the MDI loop count have been transferred, the "start" signal can be activated ([STA] = 1). The roll feed cancels the start enable ([ST_EN] = 0) and activates "function running" ([FUR] = 1). The positioning movement is prepared but is not triggered yet.
3. The user activates the read-in enable ([RIE] = 1). The rising edge of the read-in enable triggers the start of the positioning movement. The roll feed starts the positioning operation and activates the motion signal "axis moves forwards" ([FWD] = 1).
4. When the positioning movement has been successfully completed, "destination reached, axis stationary" is activated and "axis moves forwards" is reset ([DRS] = 1 and [FWD] = 0). "Function running" remains active ([FUR] = 1).

5. The read-in enable is activated again ([RIE] = 0->1). The roll feed is repositioned and "axis moves forwards" is activated ([FWD] = 1). "Destination reached, axis stationary" is reset ([DRS] = 0).
6. When the positioning movement has been successfully completed, "destination reached, axis stationary" is activated and "axis moves forwards" is reset ([DRS] = 1 and [FWD] = 0). If the MDI loop counter has reached 0, "function running" is canceled ([FUR] = 0) and "function terminated" is activated ([FUT] = 1). The "start" signal can now be canceled ([STA] = 0). The "function terminated" checkback signal is reset ([FUT] = 0) and the start enable is activated ([ST_EN] = 1).

Pulse Diagram Sequence, Special Situations

7. If the "start" signal is canceled ([STA] = 0) during the positioning movement, the roll feed is stopped. "Axis moves forwards" is reset ([FWD] = 0); the start enable is activated ([ST_EN] = 1). "Function running" remains active ([FUR] = 1) and "destination reached, axis stationary" is not output ([DRS] = 0) because the positioning movement is not complete. When the "start" signal is activated again ([STA] = 1), the positioning movement is continued and "axis moves forwards" is activated again ([FWD] = 1). The start enable is reset ([ST_EN] = 0).
8. A positioning movement is no longer triggered if the MDI loop counter has reached 0 and the read-in enable is activated again ([RIE] = 1).
9. If "reset technology" or "follow-up mode" is activated during the positioning movement ([RST] = 1 or [FUM] = 1), the position controller is disabled abruptly.
10. If the mode is changed ([MODE_IN] <> 0011) during the positioning movement, the roll feed is stopped via the deceleration ramp.

NOTE

The roll feed can be operated using MDI number 0 to 10. However, a floating MDI is not possible using MDI number 0. The toggle bit is not evaluated.

5.2.3 Set Floating Reference Point

NOTE

The "Set floating reference point" will only be available from MASTERDRIVES MC firmware version >=V1.4x.

General

Setting the floating reference point allows an axis to be referenced in any mode, including in MDI mode. The reference point is logged by means of a proximity switch (positive edge) without evaluating the sensor zero mark.

Setting the floating reference point has no influence on the course of the axis. It is the path conditions which are of relevance, as described in 5.2.1, the axis should be controlled by the user program.

The set position value and the actual position value are corrected on the basis of the discrepancy between the setpoint (reference point coordinate) and the measured value logged by the proximity switch. The function can also be influenced by an enable signal ([EN_RF]) and by the definition of an inner and an outer window.

G90: Absolute Programming

When programming with absolute data, the axis moves towards a defined target position. Setting the floating reference point therefore changes the remaining position path to be traversed. If the reference point coordinate is beyond the target position, the axis automatically changes direction.

Exception: if the direction of movement is set when positioning a rotary axis ([JVWD] or [JBWD] = 1), the axis does not change direction.

G91: Incremental Coordinates

When programming using incremental coordinates, a distance is traveled. Setting the floating reference point does not alter this distance. Therefore the target position reached depends on the adjustment to the measurement system caused by the floating reference system.

Parameters

The set floating reference point function is configured by means of the following parameters:

- Reference point – coordinate (MD3)
- Digital inputs – function 2 (MD46)
- Set floating reference point, inner window (FP2)
- Set floating reference point, outer window (FP3)
- Set correction value for floating reference point (FP4)

Further Information

A description of the parameters can be found in the function description in the chapter on “Machine data and technology parameters".

Pulse Diagram

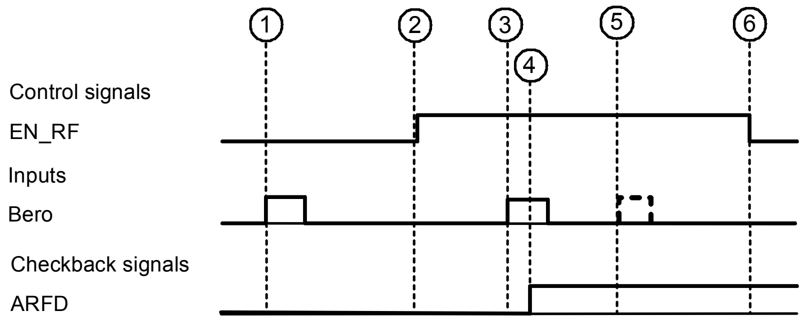


Fig. 5-3 Pulse diagram for setting floating reference point

Pulse Diagram Sequence

Conditions: The user has selected MDI mode, the axis is moving and the axis is not referenced.

1. The axis runs over the proximity switch and enable is not set. The axis is not referenced.
2. The user sets the enable ([EN_RF] = 1).
3. The next rising edge on the proximity switch triggers the evaluation. As the axis has not yet been referenced ([ARFD] = 0), the reference point is always set.
4. The checkback signal that the axis is referenced is set ([ARFD] = 1).
5. As long as the enable is active, the axis is referenced again with every rising edge of the proximity switch. The setting of the reference point depends on the evaluation of the discrepancy between the measured value and the reference point coordinate on the basis of the windows set. Repeated referencing has no influence on the ARFD signal.
6. The user resets the enable. The proximity switch is no longer evaluated.

6 Control Mode

Contents

In this chapter you will find all the information about control mode.

Related Terms

Test mode

Overview

In control mode, the axis is controlled by speed, not by position. You can switch between 2 speed levels (10 % n_{rated} and 100 % n_{rated}) which can be modified additionally by a speed override from 0 to 255 %. It is also possible to define the acceleration and deceleration ramps. While the axis is moving, the actual position value is tracked in order to retain the reference to the measuring system.

Control mode is suitable for commissioning the speed controller, for traversing to a fixed stop, or for applications using a motor which has to rotate at a specific speed.

WARNING

The software limit switches are not monitored in control mode, even if the measuring system has been synchronized.

6.1 Function Parameters

6.1.1 Acceleration Time / Deceleration Time

Acceleration Time / Deceleration Time

"Acceleration time, operating mode control" defines the acceleration ramp. "Deceleration time, operating mode control" defines the deceleration ramp.

- If a value of 0 is defined as the acceleration time, the speed controller accelerates abruptly to the speed setting.
- If a value greater than 0 is defined as the acceleration time, the specified time value defines the time taken to accelerate to the rated speed.
- If a value of 0 is defined as the deceleration time, the speed controller decelerates abruptly to setpoint 0.
- If a value greater than 0 is defined as the deceleration time, the specified time value defines the time taken to decelerate from the rated speed to a standstill.

MD No.	I	A	W	Name	Input Limit		Unit	Default
					Lower	Upper		
41				Acceleration time, operating mode "reference point approach / control" (U501.41)	0.001 0 = inactive	99.999	s	1.000
42				Deceleration time, operating mode "reference point approach / control" (U501.42)	0.001 0 = inactive	99.999	s	1.000

6.1.2 Control and Checkback Signals

General Information

The control signals are used to operate the axis in control mode. The checkback signals indicate the current status of the axis.

Control Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to control mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBx						LB=1			BIN	IN_1
DBBx+1	ACK_F				ENC	OFF3	OFF2	OFF1	BIN	IN_2
DBBx+2	MODE_IN = 0100				J_FWD	F_S	J_BWD		BIN	IN_3
DBBx+3	OVERRIDE								DEC	IN_4
DBBx+4									DEC	IN_5
DBBx+5		RST	FUM						BIN	IN_6
DBBx+6	R_VM	S_VM	EN_RF						BIN	IN_7
DBBx+7	ST_VM								BIN	IN_8

DBB address x is calculated as follows:
x = (axis number -1) * 80 + 26

Checkback Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to control mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBy					OTM	OTC	OLC	SMAX	BIN	OUT_1
DBBy+1		WARN	OFF3	OFF2	FAULT	IOP	RDY	RTS	BIN	OUT_2
DBBy+2	FAULT_NO								DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4
DBBy+4		ARFD	FUR_VM	OTR		BWD	FWD		BIN	OUT_5
DBBy+5									DEC	OUT_6
DBBy+6	MODE_OUT = 0100				FUR	ST_EN			BIN	OUT_7
DBBy+7									DEC	OUT_8

DBB address y is calculated as follows:
y = (axis number -1) * 80 + 54

6.1.3 Speed Value

The speed value is determined by the "fast / slow" [F_S] selection and the "override" [OVERIDE] as follows:

- [F_S] = 0: 10 % n_{rated} bei [OVERRIDE] = 100 %
- [F_S] = 1: 100 % n_{rated} bei [OVERRIDE] = 100 %

Examples:

[F_S] = 0, [OVERRIDE] = 25 % → n = 25/10 = 2,5 % n_{rated}

[F_S] = 0, [OVERRIDE] = 175 % → n = 175/10 = 17,5 % n_{rated}

[F_S] = 1, [OVERRIDE] = 25 % → n = 25/1 = 25 % n_{rated}

[F_S] = 0, [OVERRIDE] = 175 % → n = 175/1 = 100 % n_{rated} (value is limited to 100 %)

The output of the speed value to the speed controller takes place when the "jog forwards" [J_FWD] or "jog backwards" [J_BWD] control signal is activated.

NOTE

The position control system remains activated when control mode is selected. The switchover to speed control does not occur until "jog forwards" [J_FWD] or "jog backwards" [J_BWD] is activated. Similarly, the switchover to position control takes place when "jog forwards" [J_FWD] or "jog backwards" [J_BWD] is canceled. The switchover occurs either abruptly (if a value of 0 was entered for the deceleration time) or at the end of the deceleration ramp. In the event of an abrupt switchover from speed control to position control, the "following error monitoring, at standstill" position control monitoring circuit is active immediately. This circuit can trip in certain circumstances, because the motor cannot be reduced abruptly to zero speed.

The status of the "controller enable" [ENC] signal is not evaluated in control mode.

If one of the signals OFF1, OFF2 or OFF3 is set to "0" in control mode while the axis is moving, then the drive stops immediately. If the signals are set back to "1" afterwards and one of the control signals J_FWD or J_BWD is still set, then the drive starts moving again straightaway. This behavior depends on the MASTERDRIVES MC basic device.

6.2 Function Description

General Information

The following control signals should always be activated:

- OFF1 to OFF3 [OFF1..OFF3] = 1
- Controller enable [ENC] can be activated afterwards

6.2.1 Handling by the User

Pulse Diagram

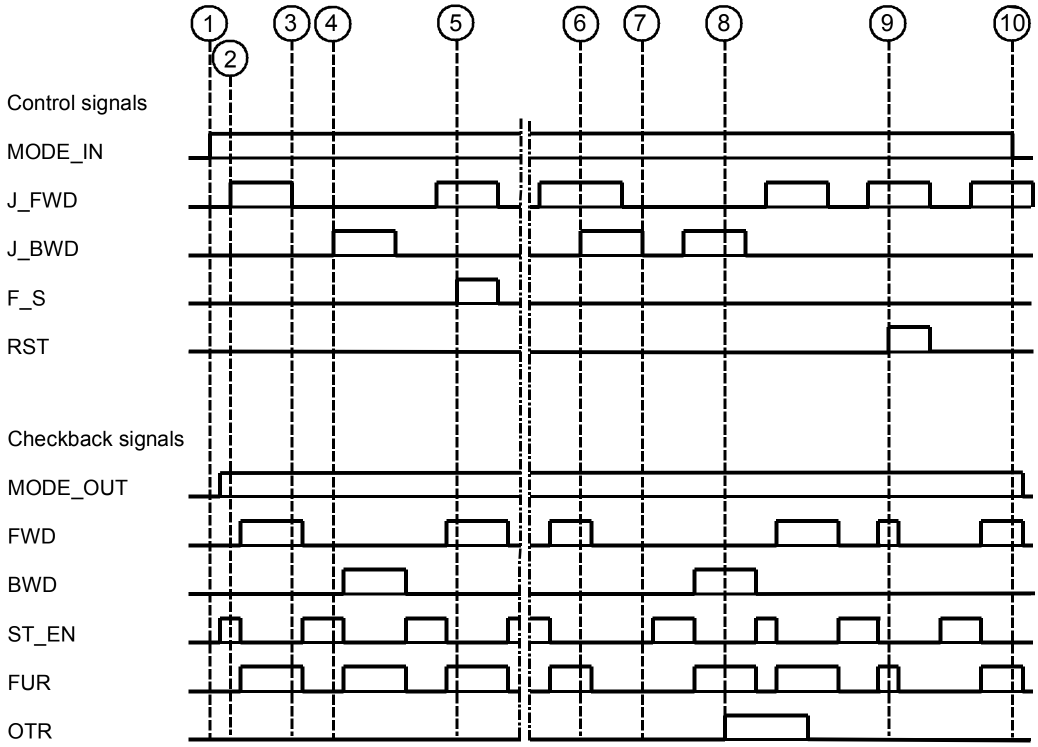


Fig. 6-1 Pulse Diagram for Control Mode

Pulse Diagram Sequence

(1) The user selects the mode ([MODE_IN] = 0100). The axis returns the mode ([MODE_OUT] = 0100) and the start enable [ST_EN] = 1).

(2) As soon as the start enable is available ([ST_EN] = 1), "jog forwards" can be activated ([J_FWD] = 1). The axis cancels the start enable ([ST_EN] = 0) and indicates "axis moves forwards" ([FWD] = 1) and "function running" ([FUR] = 1).

(3) When "jog forwards" is canceled ([J_FWD] = 0), "axis moves forwards" and "function running" are canceled ([FWD] = 0 and [FUR] = 0). The start enable reappears ([ST_EN] = 1).

(4) The "jog backwards" command ([J_BWD] = 1) cancels the start enable ([ST_EN] = 0). "Axis moves backwards" and "function running" are activated ([BWD] = 1 and [FUR] = 1).

(5) If the "fast/slow" [F_S] switchover or a change in override [OVERRIDE] is initiated during speed control mode, the new speed value is executed either abruptly or via the acceleration or deceleration ramp.

Pulse Diagram Sequence, Special Situations

(6) If "jog backwards" is selected ([J_BWD] = 1) while "jog forwards" is active ([J_FWD] = 1), the ambiguous direction command terminates speed control. "Axis moves forwards" and "function running" are canceled ([FWD] = 0 and [FUR] = 0) and position control is active again.

(7) The start enable is activated ([ST_EN] = 1) only when both jog directions have been canceled ([J_FWD] = 0 and [J_BWD] = 0).

(8) Since actual value detection is active in control mode, the crossing of the software limit switches is indicated by the "overtravel" checkback signal ([OTR] = 1). This has no effect on speed control, however.

(9) If "reset technology" is activated ([RST] = 1) during speed control, output of the speed value is terminated abruptly. "Axis moves forwards" and "function running" are canceled again ([FWD] = 0 and [FUR] = 0).

(10) If the mode is canceled ([MODE_IN] <> 0100) during speed control, the speed control is terminated. "Function running" and "axis moves forwards" are canceled ([FUR] = 0 and [FWD] = 0). The new mode is returned ([MODE_OUT] <> 0100).

6.2.2 Set Floating Reference Point

NOTE

The "Set floating reference point“ will only be available from MASTERDRIVES MC firmware version >=V1.4x .

General Information

Setting the floating reference point allows an axis to be referenced in any mode, including in control mode. The reference point is logged by means of a proximity switch (positive edge) without evaluating the sensor zero mark.

Setting the floating reference point has no influence on the course of the axis. As described in 6.2.1, the axis should be controlled by the user program.

The set position value and the actual position value are corrected on the basis of the discrepancy between the setpoint (reference point coordinate) and the measured value logged by the proximity switch. The function can also be influenced by an enable signal ([EN_RF]) and by the definition of an inner and an outer window.

Parameters

The set floating reference point function is configured by means of the following parameters:

- Reference point – coordinate (MD3)
- Digital inputs – function 2 (MD46)
- Set floating reference point, inner window (FP2)
- Set floating reference point, outer window (FP3)
- Set correction value for floating reference point (FP4)

Further Information

A description of the parameters can be found in the function description in the chapter on “Machine data and technology parameters".

Pulse Diagram

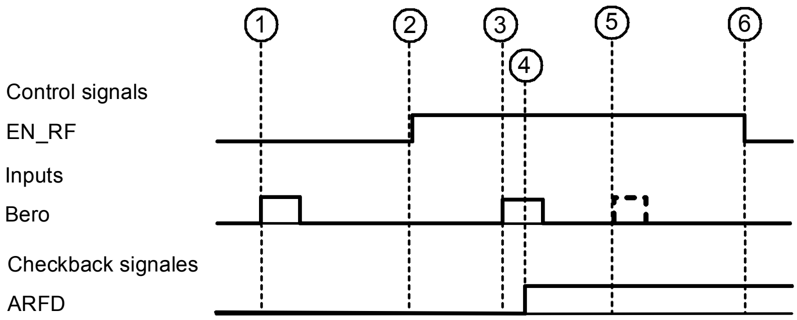


Fig. 6-2 Pulse diagram for setting floating reference point

Pulse Diagram Sequence

Conditions: The user has selected setup mode, the axis is moving and the axis is not referenced.

(1) The axis runs over the proximity switch and enable is not set. The axis is not referenced.

(2) The user sets the enable ([EN_RF] = 1).

(3) The next rising edge on the proximity switch triggers the evaluation. As the axis has not yet been referenced ([ARFD] = 0), the reference point is always set.

(4) The checkback signal that the axis is referenced is set ([ARFD] = 1).

(5) As long as the enable is active, the axis is referenced again with every rising edge of the proximity switch. The setting of the reference point depends on the evaluation of the discrepancy between the measured value and the reference point coordinate on the basis of the windows set. Repeated referencing has no influence on the ARFD signal.

(6) The user resets the enable. The proximity switch is no longer evaluated.

7 Automatic Mode

Contents In this chapter you will find all the information about automatic mode.

Related Terms Automatic cycle

Overview Complete NC programs are executed fully automatically in automatic mode. In order to start an NC program, the NC program data must first be transferred to the technology. Different NC programs are selected using different NC program numbers. It is also possible to execute an NC program step-by-step by activating the "single-step" control signal.

If the technology is installed on an M7-FM, the axes can be coordinated by one NC program. Alternatively, each axis can run its own NC program in parallel with other axes.

If the technology is installed on MASTERDRIVES MC, the NC programs refer exclusively to this axis.

If you need to use one NC program for multiple axes, the technology must be installed on an M7-FM (centralized technology).

With the roll feed axis type, each NC block is edge-triggered by an internal signal (control signal) and/or an external signal (digital input). This achieves very high signal processing rates. The actual position value is zeroed on each positioning movement. All movements are performed relatively (without an absolute reference to the mechanical system).

7.1 Function Parameters

7.1.1 NC Programs

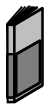
NC Program

The NC programs contain the information for the movements in a chained sequence. NC programs can respond to user input while they are running, and can output messages to the user.

Input/output of NC programs is performed using the standard user interfaces available or in STEP 7 with the "input/output NC program" task.

NOTE

If a movement program which is currently being executed is updated, the initial block programmed is always executed to completion before the newly programmed blocks are executed.
If one of the pre-coded blocks is also modified, the new version is only executed in the next cycle.



Further Information

The task is described in the "Task Description" in Chapter "NC Program Function Tasks" in Section "Input/Output NC Program".

The syntax of NC programs and their functional features are described in the "Programming Guide".

7.1.2 Control and Checkback Signals

General Information

The control signals are used to operate the axis in automatic mode. The checkback signals indicate the current status of the axis.

Control Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to automatic mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBx						LB=1			BIN	IN_1
DBBx+1	ACK_F				ENC	OFF3	OFF2	OFF1	BIN	IN_2
DBBx+2	MODE_IN = 0101							BLSK	BIN	IN_3
DBBx+3	OVERRIDE								DEC	IN_4
DBBx+4	PROG_NO								DEC	IN_5
DBBx+5	SIST	RST	FUM	ACK_M	CRD	STA	RIE		BIN	IN_6
DBBx+6	R_VM	S_VM	EN_RF						BIN	IN_7
DBBx+7	ST_VM								BIN	IN_8

DBB address x is calculated as follows:
x = (axis number -1) * 80 + 26

Checkback Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to automatic mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBy					OTM	OTC	OLC	S MAX	BIN	OUT_1
DBBy+1		WARN	OFF3	OFF2	FAULT	IOP	RDY	RTS	BIN	OUT_2
DBBy+2	FAULT_NO								DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4
DBBy+4	STR_M	ARFD	FUR_VM	OTR	FUT	BWD	FWD	DRS	BIN	OUT_5
DBBy+5	M_NO_1								DEC	OUT_6
DBBy+6	MODE_OUT = 0101				FUR	ST_EN	T_R		BIN	OUT_7
DBBy+7	M_NO_2								DEC	OUT_8

DBB address y is calculated as follows:
y = (axis number -1) * 80 + 54

7.2 Function Description

General Information The following control signals should always be activated:

- OFF1 to OFF3 [OFF1..OFF3] = 1
- Controller enable [ENC] = 1

7.2.1 Handling by the User

Pulse Diagram

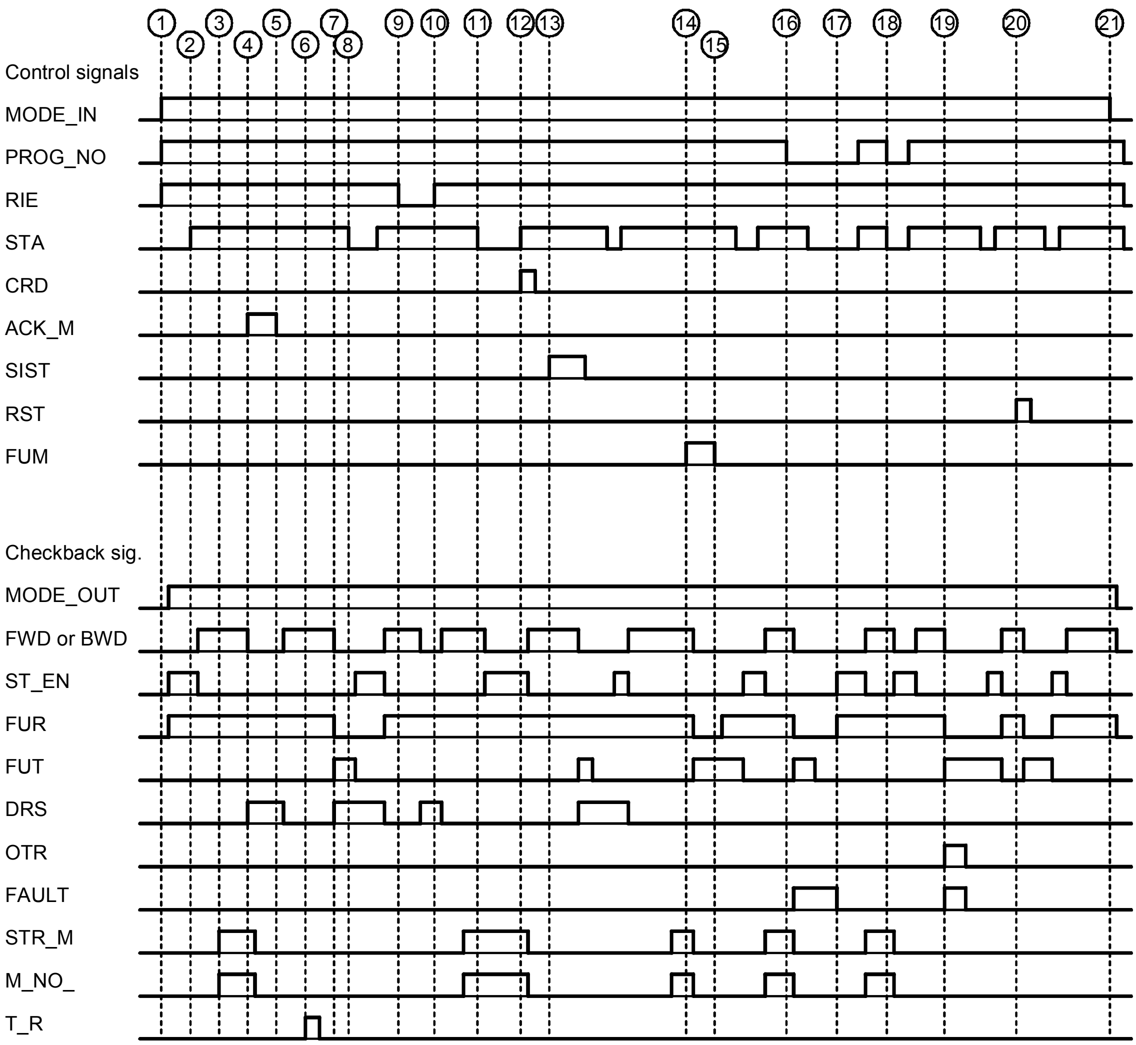


Fig. 7-1 Pulse Diagram for Automatic Mode

Pulse Diagram Sequence

1. The user selects the mode ([MODE_IN] = 0101), the NC program number [PROG_NO] and the read-in enable ([RIE] = 1). The axis returns the mode ([MODE_OUT] = 0101), the start enable ([ST_EN] = 1) and "function running" ([FUR] = 1), since the NC block decoder has already been started.
2. When the start enable is activated ([ST_EN] = 1) and the read-in enable is active ([RIE] = 1), NC program execution can be started with the "start" signal ([STA] = 1). Execution begins, for example with a positioning movement. "Axis moves forwards" ([FWD] = 1) or "axis moves backwards" ([BWD] = 1) is activated. The start enable is canceled ([ST_EN] = 0).
3. If M function output is acknowledge-driven, the user program can continue executing M function number 1/2 [M_NO_1 and M_NO_2] when the "strobe signal for M functions" is activated ([STR_M] = 1).
4. The M function is acknowledged by the "acknowledge M function" signal ([ACK_M] = 1), upon which the "strobe signal for M functions" and "M function number 1/2" are canceled ([STR_M] = 0 and [M_NO 1 to 2] = 0).
5. The "acknowledge M function" signal can now be canceled again ([ACK_M] = 0). Only at this point is the M function output complete. The NC program continues. "Axis moves forwards" or "axis moves backwards" ([FWD] = 1 or [BWD] = 1) is activated and, if the "destination reached, axis stationary" signal is active, it is now canceled ([DRS] = 0).
6. During execution of an NC block with dwell, the "dwell time running" checkback signal is output for the duration of the dwell ([T_R] = 1).
7. The end of the NC program is indicated by activating the "destination reached, axis stationary" signal ([DRS] = 1) and resetting "function running" ([FUR] = 0). The "function terminated" checkback signal is now output ([FUT] = 1), indicating to the user that the "start" signal [STA] is no longer required.
8. The user can then cancel the "start" signal ([STA] = 0). The axis cancels "function terminated" ([FUT] = 0) and activates the start enable ([ST_EN] = 1).

Pulse Diagram Sequence, Special Situations

9. If the read-in enable is canceled ([RIE] = 0) while the NC program is running, the NC block is finished and program execution is subsequently halted. "Axis moves forwards" or "axis moves backwards" is canceled ([FWD] = 0 or [BWD] = 0). "Destination reached, axis stationary" is activated ([DRS] = 1).
10. When the read-in enable is activated ([RIE] = 1), the program continues; "axis moves forwards" or "axis moves backwards" is activated ([FWD] = 1 or [BWD] = 1); and "destination reached, axis stationary" is reset ([DRS] = 0).
11. Canceling the "start" signal ([STA] = 0) interrupts the program. The "axis moves forwards" or "axis moves backwards" checkback signal is canceled ([FWD] = 0 or [BWD] = 0). "Function running" remains active ([FUR] = 1) and the start enable is activated ([ST_EN] = 1). There is no impact on M function output.
12. If "cancel remaining distance" is selected ([CRD] = 1) with "start" ([STA] = 1), the NC block which was previously interrupted by canceling the "start" signal [STA] is not finished. Instead, the program starts processing the next NC block. If an M function is active, it is also canceled.
13. Activating the "single-step" signal ([SIST] = 1) switches the automatic sequence to single-step mode, i.e. NC program execution stops at the end of the current NC block. The axis activates "function terminated" ([FUT] = 1). The user can then cancel the "start" signal ([STA] = 0), and the start enable reappears ([ST_EN] = 1).
14. "Follow-up mode" is activated ([FUM] = 1) while the axis is moving. The axes are stopped abruptly or by the deceleration ramp during error (MD43). NC program execution is interrupted. "Function running" is canceled ([FUR] = 0) and "function terminated" is activated ([FUT] = 1). If M functions are active, these are also canceled.
15. When "follow-up mode" is canceled ([FUM] = 0), the position controller is activated again. Since the read-in enable is active ([RIE] = 1), the NC block decoder is started again and "function running" is activated ([FUR] = 1).
16. The NC program number is changed while the NC program is running. The axes are brought to a standstill via the deceleration ramp and NC program execution is interrupted. "Function running" is canceled ([FUR] = 0), and "function terminated" is activated ([FUT] = 1). If M functions are active, these are also canceled. A warning [WARN] and [WARN_NO] are generated.
17. The start enable does not reappear ([ST_EN] = 1) until the warning has been cleared. The NC block decoder is started again.

18. If NC program execution is interrupted by canceling the "start" signal ([STA] = 0), and the NC program number is changed and "start" is activated again ([STA] = 1), the interrupted NC program is aborted and processing resumes with the new NC program. If an M function is active, it is canceled by changing the NC program number.
19. If the software limit switch is approached while the NC program is running, the axis is stopped via the deceleration ramp, the "overtravel" checkback signal is activated ([OTR] = 1) and a warning [WARN] and [WAR_NO] are generated. The axis cannot move away from the software limit switch in automatic mode.
20. If "reset technology" is activated ([RST] = 1) while the NC program is running, the position controller is disabled abruptly. NC program execution is interrupted, and "function running" is reset ([FUR] = 0). With incremental encoders, the measuring system must be synchronized again after "reset technology" [RST]. The NC block decoder is restarted when "reset technology" is canceled ([RST] = 0).
21. If automatic mode is canceled while the NC program is running ([MODE_IN] <> 0101), the axes are stopped via the deceleration ramp. NC program execution is interrupted, "function running" is canceled ([FUR] = 0) and the new mode is returned ([MODE_OUT] <> 0101).

7.2.2 Set Floating Reference Point

NOTE

The “set floating reference point” function will only be available from MASTERDRIVES MC firmware version >=V1.4x.

General Information

Setting the floating reference point allows an axis to be referenced in any mode, including in automatic mode. The reference point is logged by means of a proximity switch (positive edge) without evaluating the sensor zero mark.

As described in 7.2.1, the axis should be controlled by the user program.

The set position value and the actual position value are corrected on the basis of the discrepancy between the setpoint (reference point coordinate) and the measured value logged by the proximity switch. The function can also be influenced by an enable signal ([EN_RF]) and by the definition of an inner and an outer window.

Setting the floating reference point influences how the axis continues to move. This influence depends on the programming. There are essentially two different cases.

Depending on the time and the magnitude of the discrepancy, the correction affects:

- the block currently being executed
- more than one block

Correction of the Block Being Executed

If the setting of the floating reference point does not mean that the block limit is exceeded, the set position value and the actual position value are corrected. Irrespective of the programming of the block (absolute / incremental), the axis moves towards the target position which has been either programmed or calculated, i.e. the correction influences the path to be traversed in the block.

Correction of More Than One Block

If the setting of the floating reference point means that the block limit is exceeded, the system automatically switches to the block to be processed. To do this, it can jump one or more blocks. The "search" is subject to the following framework conditions:

- The traverse program is searched in ascending block number order, starting with the current block. During this process, control instructions, such as jump to start of program (M18), programmed loops, sub-programs and skippable blocks, are obeyed.
- If the end of the program is reached before the search is complete, the axis is stopped and the program ended.
- The search is automatically interrupted if a programmed direction change or a block without a target position is detected. Execution begins again with the block which contains the direction change or no target position.
- The block to which the program jumps is executed from the start of the block, i.e. with incremental programming, the programmed incremental coordinates are executed (in the same way as for block search).

NOTE

Depending on the size of the traverse program, the movement of the axis can be interrupted briefly. The interruption time depends on the number of blocks which need to be pre-decoded because of the "block search". There is no parameterization for continuation of movement during this time and the axis is stopped using the parameterized acceleration. As soon as the decoder has decoded the block to be executed, movement can continue.

Parameters

The set floating reference point function is configured by means of the following parameters:

- Reference point – coordinate (MD3)
- Digital inputs – function 2 (MD46)
- Set floating reference point, inner window (FP2)
- Set floating reference point, outer window (FP3)
- Set correction value for floating reference point (FP4)

Further Information

A description of the parameters can be found in the function description in the chapter on "Machine data and technology parameters".

Pulse Diagram

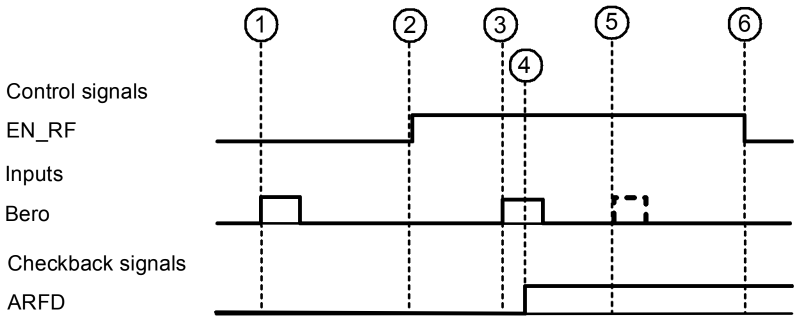


Fig. 7-2 Pulse diagram for setting floating reference point

Pulse Diagram Sequence

Conditions: The user has selected automatic mode, the axis is moving and the axis is not referenced.

1. The axis runs over the proximity switch and enable is not set. The axis is not referenced.
2. The user sets the enable ([EN_RF] = 1).
3. The next rising edge on the proximity switch triggers the evaluation. As the axis has not yet been referenced ([ARFD] = 0), the reference point is always set.
4. The checkback signal that the axis is referenced is set ([ARFD] = 1).
5. As long as the enable is active, the axis is referenced again with every rising edge of the proximity switch. The setting of the reference point depends on the evaluation of the discrepancy between the measured value and the reference point coordinate on the basis of the windows set. Repeated referencing has no influence on the ARFD signal.
6. The user resets the enable. The proximity switch is no longer evaluated.

8 Automatic Single-Block Mode

Contents

In this chapter you will find all the information about automatic single-block mode.

Related Terms

Single-block

Overview

One NC block is executed at a time in automatic single-block mode. The NC block must be transferred to the technology before each start. The NC block data are identical to the data of an NC block in automatic mode. Only one NC block can be stored on the technology for automatic mode. If the start is initiated without transferring a new NC block, the old NC block is executed again.

With the roll feed axis type, each NC block is edge-triggered by an internal signal (control signal) and/or an external signal (digital input). This achieves very high signal processing rates. The actual position value is zeroed on each positioning movement. All movements are performed relatively (without an absolute reference to the mechanical system).

8.1 Function Parameters

8.1.1 NC Block

The NC block contains the information for the sequence of movements.

If the technology is installed on the MASTERDRIVES MC, the NC block with block number 1 must be entered as NC block number 21.

If the technology is used on a SIMATIC Motion Control system, a traversing program number will be available for every SIMATIC Motion Control axis for the automatic single block mode. SIMATIC Motion Control axis with local axis number 1-8/16/32 → traversing program number 201 to 204.

NOTE The local axis numbers are defined during commissioning in the axis descriptions in GMC_DB_ORG.

Input/output of an NC block is performed using the standard user interfaces available or in STEP 7 with the "input/output NC block" task.

Further Information

The task is described in the "Task Description" in Chapter "NC Program Function Tasks" in Section "Input/Output NC Block".

The syntax of NC blocks and their functional features are described in the "Programming Guide".

8.1.2 Control and Checkback Signals

General Information

The control signals are used to operate the axis in automatic single-block mode. The checkback signals indicate the current status of the axis.

Control Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to automatic single-block mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBx						LB=1			BIN	IN_1
DBBx+1	ACK_F				ENC	OFF3	OFF2	OFF1	BIN	IN_2
DBBx+2	MODE_IN = 0110								BIN	IN_3
DBBx+3	OVERRIDE								DEC	IN_4
DBBx+4									DEC	IN_5
DBBx+5		RST	FUM	ACK_M	CRD	STA			BIN	IN_6
DBBx+6	R_VM	S_VM							BIN	IN_7
DBBx+7	ST_VM								BIN	IN_8

DBB address x is calculated as follows:
x = (axis number -1) * 80 + 26

Checkback Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to automatic single-block mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBy					OTM	OTC	OLC	S MAX	BIN	OUT_1
DBBy+1		WARN	OFF3	OFF2	FAULT	IOP	RDY	RTS	BIN	OUT_2
DBBy+2	FAULT_NO								DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4
DBBy+4	STR_M	ARFD	FUR_VM	OTR	FUT	BWD	FWD	DRS	BIN	OUT_5
DBBy+5	M_NO_1								DEC	OUT_6
DBBy+6	MODE_OUT = 0110				FUR	ST_EN	T_R		BIN	OUT_7
DBBy+7	M_NO_2								DEC	OUT_8

DBB address y is calculated as follows:
y = (axis number -1) * 80 + 54

8.2 Function Description

General Information The following control signals should always be activated:

- OFF1 to OFF3 [OFF1..OFF3] = 1
- Controller enable [ENC] = 1

8.2.1 Handling by the User

Pulse Diagram

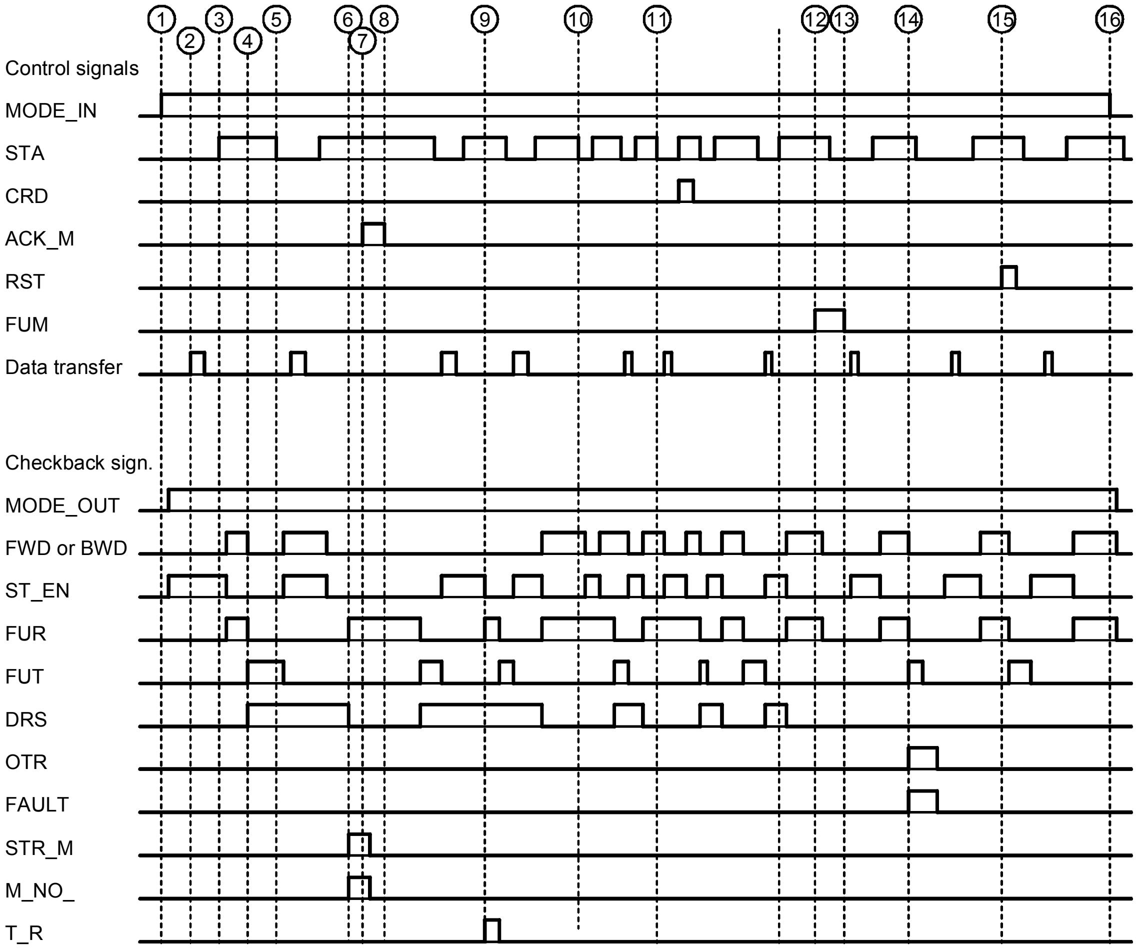


Fig. 8-1 Pulse Diagram for Automatic Single-Block Mode

Pulse Diagram Sequence

1. The user selects the mode ([MODE_IN] = 0110). The axis returns the mode ([MODE_OUT] = 0110) and the start enable ([ST_EN] = 1).
2. Once the automatic single-block mode has been selected, an NC block can be transferred.
3. When the transfer has been successfully completed and the start enable is activated ([ST_EN] = 1), the NC block can be started with "start" ([STA] = 1). If the NC block contains a positioning movement, the "axis moves forwards" or "axis moves backwards" and "function running" signals are activated ([FWD] = 1 or [BWD] = 1 and [FUR] = 1). The start enable is canceled ([ST_EN] = 0).
4. The end of the NC block is indicated by activating the "destination reached, axis stationary" signal ([DRS] = 1) (after a positioning movement) and resetting "function running" ([FUR] = 0). The "function terminated" checkback signal is now output ([FUT] = 1), indicating to the user that the "start" signal [STA] is no longer required.
5. The user can then cancel the "start" signal ([STA] = 0). The axis cancels "function terminated" ([FUT] = 0) and activates the start enable ([ST_EN] = 1).
6. If M function output is acknowledge-driven, the user program can continue executing M function number 1/2 [M_NO_1 and M_NO_2] when the "strobe signal for M functions" is activated ([STR_M] = 1).
7. The M function is acknowledged by the "acknowledge M function" signal ([ACK_M] = 1), upon which the "strobe signal for M functions" and "M function number 1/2" are canceled ([STR_M] = 0 and [M_NO 1 to 2] = 0).
8. The "acknowledge M function" signal can now be canceled again ([ACK_M] = 0). Only at this point is the M function output complete.
9. If a dwell is programmed in the NC block, the "dwell time running" checkback signal is output for the duration of the dwell ([T_R] = 1).

Pulse Diagram Sequence, Special Situations

10. If the "start" signal is canceled ([STA] = 0) while an NC block is running, NC block execution is interrupted. The "axis moves forwards" or "axis moves backwards" checkback signal is canceled ([FWD] = 0 or [BWD] = 0). "Function running" remains active ([FUR] = 1) and the start enable is activated ([ST_EN] = 1). Block execution continues on the next "start" signal ([STA] = 1).
11. If execution of an NC block is interrupted by canceling the "start" signal ([STA] = 0), two options are available after transferring a new NC block:
    - Only "start" is activated ([STA] = 1); the interrupted NC block is finished. The new NC block is not executed until the third "start" ([STA] = 1).
    - "Start" and "cancel remaining distance" are activated ([STA] = 1 and [CRD] = 1). The next NC block starts running immediately.
12. "Follow-up mode" is activated ([FUM] = 1) while the axis is moving. The axis or axes are stopped abruptly as a result of the position controller disable. NC program execution is interrupted. "Function running" is canceled ([FUR] = 0) and "function terminated" is activated ([FUT] = 1). If M functions or a dwell time are active, these are also canceled.
13. When "follow-up mode" is canceled ([FUM] = 0), the position controller is activated again.
14. The axis can only move away from the software limit switch in setup and MDI modes.
15. If "reset technology" is activated ([RST] = 1) while the NC block is running, the position controller is disabled abruptly. NC block execution is interrupted, and the block memory is cleared. "Function running" is reset ([FUR] = 0). With incremental encoders, the measuring system must be synchronized again after "reset technology" [RST].
16. If automatic single-block mode is canceled while the NC block is running ([MODE_IN] <> 0110), the axis or axes are stopped via the deceleration ramp. "Function running" is canceled ([FUR] = 0) and the new mode is returned ([MODE_OUT] <> 0110).

9 Slave Mode

Contents

In this chapter you will find all the information about slave mode.

Related Terms

Following axis

Overview

If the technology is installed on a SIMATIC Motion Control system, you can program NC blocks with multiple axes for automatic and single-block mode. One of the axes performs the leading function (master), and the remaining axes in the NC block/NC program are thus automatically slave axes.

The master axis is the axis in which automatic or single-block mode is selected. It processes the NC blocks and distributes the traversing tasks to the slave axes. The slave axes must be prepared to receive commands from a master axis. This condition is achieved by selecting slave mode in the slave axis or axes.

Only the axis-specific control signals (e.g. controller enable) have to be activated in the slave axes.

Only the axis-specific variables (e.g. destination reached, axis stationary) are output in the checkback signals.

If the technology is operated on a MASTERDRIVES MC, it is only ever possible to use one axis per unit. Slave axes are thus not supported in automatic and single-block modes. The slave mode is rejected by the technology.

The roll feed axis type is operated exclusively as a master axis. The slave mode is rejected by the technology.

NOTE

"Slave" mode is only available in conjunction with SIMATIC Motion Control.

9.1 Function Parameters

9.1.1 Control and Checkback Signals

General Information

The control signals are used to operate the axis in slave mode. The checkback signals indicate the current status of the axis.

Control Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to slave mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx						LB=1			BIN	IN_1
DBBx+1	ACK_F				ENC	OFF3	OFF2	OFF1	BIN	IN_2
DBBx+2	MODE_IN = 1010								BIN	IN_3
DBBx+3									DEC	IN_4
DBBx+4									DEC	IN_5
DBBx+5		RST	FUM						BIN	IN_6
DBBx+6	R_VM	S_VM							BIN	IN_7
DBBx+7	ST_VM								BIN	IN_8

DBB address x is calculated as follows:
x = (axis number -1) * 80 + 26

Checkback Signals in Data Block "GMC_DB_CMD"

In the bitmap below, the signals related to slave mode are highlighted in dark gray. The signals highlighted in light gray are always effective.

	7	6	5	4	3	2	1	0		Axis_n.
DBBy					OTM	OTC	OLC	SMAX	BIN	OUT_1
DBBy+1		WARN	OFF3	OFF2	FAULT	IOP	RDY	RTS	BIN	OUT_2
DBBy+2	FAULT_NO								DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4
DBBy+4		ARFD	FUR_VM	OTR		BWD	FWD	DRS	BIN	OUT_5
DBBy+5									DEC	OUT_6
DBBy+6	MODE_OUT = 1010								BIN	OUT_7
DBBy+7									DEC	OUT_8

DBB address y is calculated as follows:
y = (axis number -1) * 80 + 54

9.2 Function Description

General Information

The following control signals should always be activated:

- OFF1 to OFF3 [OFF1..OFF3] = 1
- Controller enable [ENC] = 1

9.2.1 Handling by the User

Since the function handling takes place exclusively in the master axis, the user only needs to ensure that the slave axis is switched to slave mode. The checkback signals in the slave axis respond in exactly the same way as in the master axis, according to the mode selected. Command errors are indicated by clear error messages.

10 Synchronization Mode

Contents

In this chapter you will find all the information about synchronization mode.

Related Terms

Synchronism , angle synchronization, rated velocity, rated master velocity, standardizing velocity, Vrated, maximum traversing velocity

Overview

In synchronization mode, the axis is controlled externally as a slave axis. The position setpoint is received from a master axis (the master value source). A large number of axes can be coupled in synchronization mode. They are either all controlled by the same master axis, or are organized into groups with different master axes.

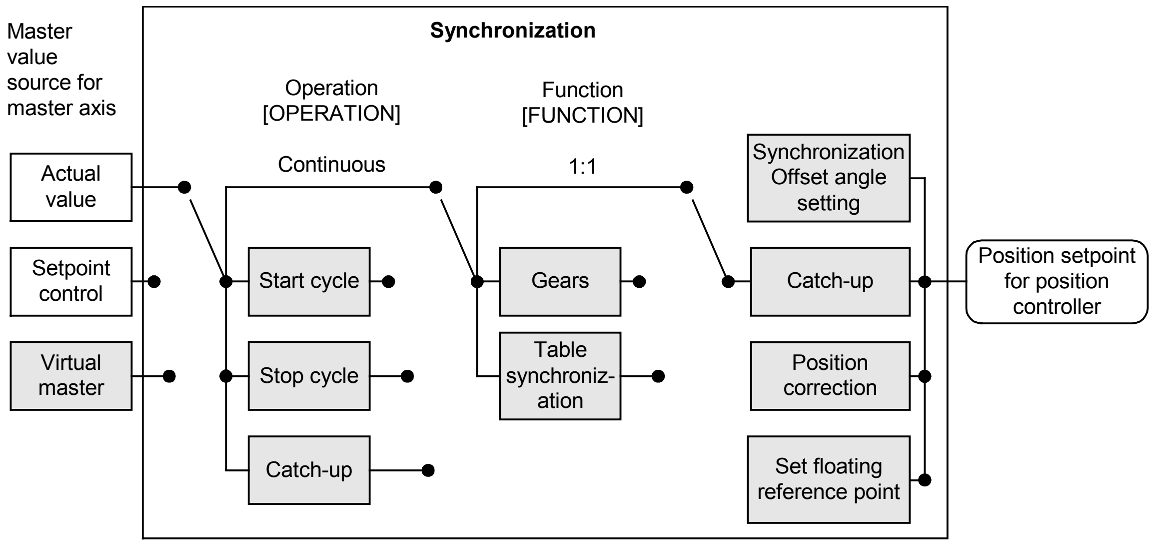


Fig. 10-1 Overall functionality of synchronization

The catch-up can either be connected upstream of the synchronization functions as an alternative mode or downstream as a mode-independent function.

This chapter describes the synchronization sub-functions. These include the following functions:

- Virtual master
- Master value selection
- Operating cycles
  - Continuous cycle
  - Start cycle
  - Stop cycle
  - Catch-up
- Synchronization functions
  - Synchronization 1:1
  - Gear synchronization
  - Table synchronization / cam disk
- Position correction
- Set floating reference point
- Master value synchronization / offset angle setting
- Catch-up

The MASTERDRIVES MC incorporate both the basic functions of the synchronization module and additional modules which can be connected to synchronization. These modules allow additional functions, and include:

- Real master
  The real master supports the processing of external sensor (master value sources). It allows smoothing and speed-dependent dead time compensation.
- Master value correction
  The master value correction module supports the processing of non-constant master value courses, such as those which result when using floating referencing for a master axis, and dynamic switching between two master values, or the superimposition of a master value with a parameterizable compensation movement.

Operating Principle

The synchronization routine acts as a setpoint generator for the axis. The master value applied to the synchronization input is converted by the synchronization functions into a setpoint for the synchronized axis. The axis is thus operated as a slave axis controlled by the setpoint of the master axis. The master value and the setpoint are positional values. The synchronization is also referred to as angular synchronization because of the positional coupling.

During internal processing, the change in the master value is converted into a change in the setpoint for the slave axis. The absolute position setpoint of the slave axis is calculated by integration at the output of the synchronization routine.

Additional Functions from MCT V1.4

Instead of setting the master value as a position setpoint, it can be set using the position setpoint and the speed set point. This procedure enables more stable axis movement because of the highly constant speed pilot control, which makes it particularly suited to processes where extreme stability and precision are priorities, such as printing presses.

General Features

This process is characterized by the following general features:

- ♦ The synchronization routine does not define an absolute position reference between the master axis and the slave axis. A fixed reference can be established, for example, by defining a starting position (default setting). The synchronization routine ensures that this initial reference is maintained.
- ♦ Axes can be coupled in any combination in synchronization mode. The master axis and slave axis can be defined independently of each other as linear or rotary axes. Rotary axes of different axis lengths can be coupled with each other.
- ♦ The position setpoint can be corrected while the synchronization mode is active without canceling the general synchronization reference. This can happen, for example, during position correction using print marks or during offset angle adjustment.
- ♦ Because the slave axis is algorithmically dependent on the master axis, the acceleration and velocity of the slave axis cannot be set by parameters, but are derived from the master value and the parameters of the synchronization functions.

CAUTION

Please ensure that your choice of master value and synchronization parameters does not result in impermissible acceleration rates.

Position setpoint jumps can occur both at the input and at the output of the synchronization routine as a result of manual control actions or parameter changes.

10.1 Synchronization Application

General

You can apply synchronization in two different modes, either as an operating mode or as a free block. The option of using synchronization as a free block is available only in MASTERDRIVES MC.

Operating Mode

If you are using synchronization as an operating mode, then all the general framework conditions for modes are applicable.

These include:

- ♦ Select the mode using [MODE_IN] and display the active mode using [MODE_OUT]
- ♦ Activate the mode using the [STA] signal in conjunction with the start enable [STA_EN]
- ♦ Display the current execution status using the signals [FUR], [FUT], [FWD] and [BWD]
- ♦ Monitoring the movement sequence and / or the axis position using the central functions of following error monitoring and software limit switch.
- ♦ Stop the axis on change of mode and or if an error occurs.

NOTE

In order to be able to select and start a mode, the converter must be set to "Operation" (Signal [IOP] = 1). If this condition is not fulfilled, a technology error is displayed.

Free block

If you are using synchronization as a free block, then the functions detailed in the operating mode section are not available.

In this case, coordination relies entirely on the "enable synchronization" signal.

If you link this signal to another signal, for example the status signal "enable position controller", then synchronization is automatically enabled if the converter is in "operation" mode and the position controller is activated.

10.1.1 Handling Synchronization as a Mode

The following monitoring systems are active in synchronization mode.

Following Error Monitoring

The following error monitoring function (MD15) is always active in synchronization mode, if you are using a centralized solution with an analog drive connection via DP/IM178, then dynamic following error monitoring (MD59) is active.

Software Limit Switch

The software limit switches are active providing the axis displays a referenced status ([ARFD] = 1). The status is displayed if the axis has an absolute value sensor, a reference search has been carried out or the axis is given a floating reference in synchronization point.

The software limit switches are parameterized using machine data MD12 and MD13. Addressing the software limit switch monitoring system stops the axis and gives an appropriate error message.

Mode Change and Error Behavior

If the mode is changed, the axis is stopped using the ramp parameterized in machine data MD42. If a technology error occurs, the axis is stopped using the ramp parameterized in machine data MD43.

Further Information

A definition of the machine data, its limits and further notes on the functions can be found in Chapter 1: "Machine data".

Notes on the technology errors (~warnings), their causes and effects and how to remedy them can be found in Appendix A: error messages.

Pulse Diagram

The pulse diagram below show the sequence and coordination for selecting and starting synchronization mode.

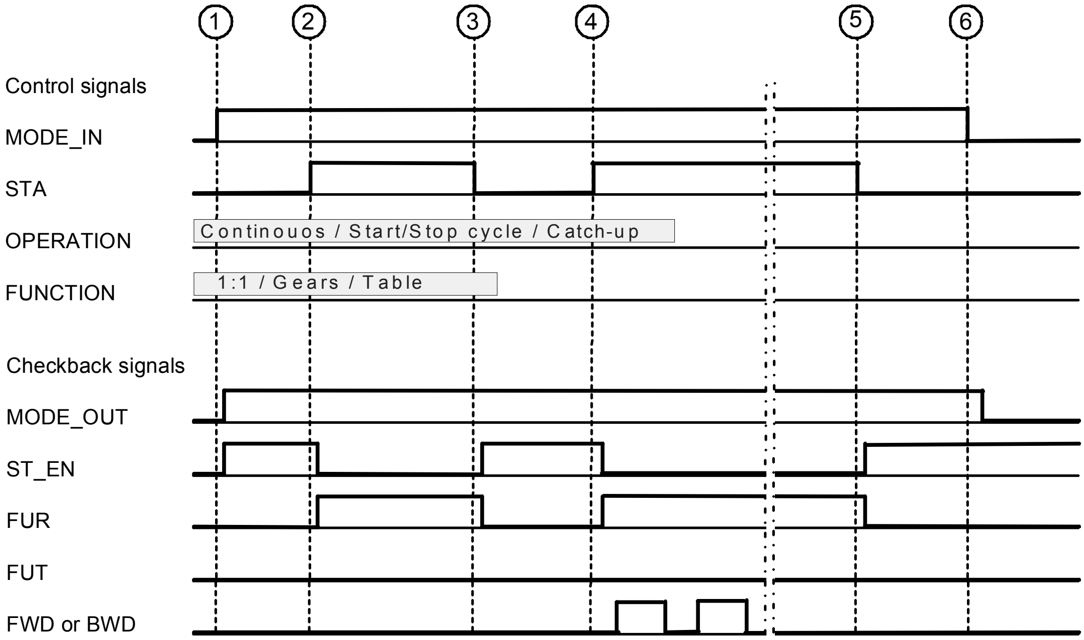
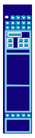


Fig. 10-2 Pulse diagram for synchronization mode

Pulse Diagram Sequence

1. The user sets the mode ([MODE_IN] = 1011). The axis reports the mode ([MODE_OUT] = 1011) and returns the "start enable". At this point, synchronization is not yet active. The axis is motionless in the position controller.
2. The start signal ([STA] = 1) activates synchronization and execution begins ([FUR] = 1). The start enable ([ST_EN] = 0) disappears. The slave axis remains still as it has not yet received a path pulse from a master axis.
3. If the start signal ([STA] = 0) is removed, synchronization is deactivated. Execution in progress ([FUR] = 0) disappears and the start enable ([ST_EN] = 1) appears.
4. Synchronization is active. As soon as the master axis begins to move, the slave axis moves as well. Depending on the direction of motion, the axis starts to move forwards ([FWD] = 1) or backwards ([BWD] = 1).
5. If the start signal is removed ([STA] = 0) while the axis is moving in synchronization mode, synchronization is ended and the axis brought to a halt by means of a controlled ramp. The same thing happens if an error occurs.
6. The mode checkback message [MODE_OUT] is removed at the same time as the synchronization mode [MODE_IN] is removed.

10.1.2 Handling Synchronization as a Free Block



Applying synchronization as a free block is only possible in MASTERDRIVES MC. Synchronization is enabled or disables using the control input **Enable Synchronization** .

If enable synchronization = 0, synchronization is not processed. The setpoint output is taken from the value closest to the "output set position setpoint value" input (which is generally the actual position). If enable synchronization = 1, synchronization is processed as an active function in accordance with its parameterization.

No other signals are available for the co-ordination and / or display of the operating status.

NOTICE

Coordination with the operating status of the converter must be ensured by switching the enable input accordingly.

It is sensible to connect the enable input with status signal "enable position control" (B220) and "output set position setpoint value" (U671) with the actual position value of the axis. This ensures that synchronization can only be activated if the converter is in operation mode and position control is activated.

Otherwise, switching on the converter can trigger an OFF1 signal, which can cause a sudden compensation movement from the slave axis if at the moment it is switched on there is a discrepancy between the position setpoint of the synchronization output and the actual position.

10.1.3 Control and Checkback Signals

General

In synchronization mode, the axes are operated by means of control signals. The checkback signals show the execution status of the axes.

If synchronization is to be applied as a free block, the mode selection and coordination signals are irrelevant.

Control Signals in Data Block "GMC_DB_CMD"

The following bit pattern diagram shows the signals which are relevant to synchronization mode, which are shaded dark gray. The signals shaded in light gray are always effective.

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx						LB=1			BIN	IN_1
DBBx+1	ACK_F				ENC	OFF3	OFF2	OFF1	BIN	IN_2
DBBx+2	MODE_IN = 1011								BIN	IN_3
DBBx+3									DEC	IN_4
DBBx+4									DEC	IN_5
DBBx+5		RST	FUM			STA			BIN	IN_6
DBBx+6	R_VM	S_VM	EN_RF	SSC	OPERATION		FUNCTION		BIN	IN_7
DBBx+7	ST_VM	TABLE_NO			SYN_T	SST	ST_S	SET_T	BIN	IN_8
DBBx+8	CU_DR	CU_EN	CU_SP	SYNC	DI_RN	DI_RP	DI_JN	DI_JP	BIN	IN_9_0
DBBx+9							S_DSP	CU_TR	BIN	IN_9_1

The DBB address x is calculated as follows:

x = (axis number-1) * 80 + 26

Checkback Signals in Data Block "GMC_DB_CMD"

In the following bit pattern diagram, the signals relevant to synchronization mode are shaded dark gray. The light gray signals are always effective.

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy					OTM	OTC	OLC	S MAX	BIN	OUT_1
DBBy+1		WA RN	OFF3	OFF2	FAU LT	IOP	RDY	RTS	BIN	OUT_2
DBBy+2	FAULT_NO								DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4
DBBy+4		ARFD	FUR_ VM	OTR		BWD	FWD		BIN	OUT_5
DBBy+5									DEC	OUT_6
DBBy+6	MODE_OUT = 1011				FUR	ST_E N			BIN	OUT_7
DBBy+7									DEC	OUT_8
DBBy+8	CU_ TE	CU_ VR	CU_ PR	SYNC	DI_A	POS_ A	CL_A	VM_ RA	BIN	OUT_9_0
DBBy+9	RESERVIERT									OUT_9_1

The DBB address y is calculated as follows:
y = (axis number-1) * 80 + 54

NOTE

If synchronization is deactivated, the checkback signals remain, which means they display the last valid status. When synchronization is activated, the checkback signals are deleted and all functions are set back to their initial status.

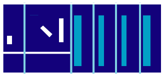
This is particularly valid for the discontinuous synchronization functions

- Start / stop cycle
  The start / stop cycle is not active.
- Cam disc
  The cam disc is started when x=0.
- Position correction
  Any remaining correction path is deleted.
- Offset angle adjustment
  The remaining path is deleted.
- Synchronization
  The function displays "not synchronized".

All other functions are in the status in accordance with their controls.

NOTE

The functions "set floating reference point", "master value synchronization", "catch-up" and "offset angle adjustment" are only available in MASTERDRIVES MC from firmware version ≥ V1.4x. This is also true for the corresponding checkback signals.



The functions "Master value synchronization", catch-up and offset angle settings are available with firmware version ≥ V1.00.48 of SIMATIC Motion Control.

10.2 Virtual Master

Overview

In applications where no real master axis is available, the master value is calculated as a reference variable and is thus made available as a "virtual master".

The virtual master is available as a separate function for each axis. A virtual master can act simultaneously as a master value generator for any number of axes.

The virtual master comprises a velocity ramp-function generator with definable velocity and variable acceleration/deceleration, and a down-circuit integrator for position setpoint generation.

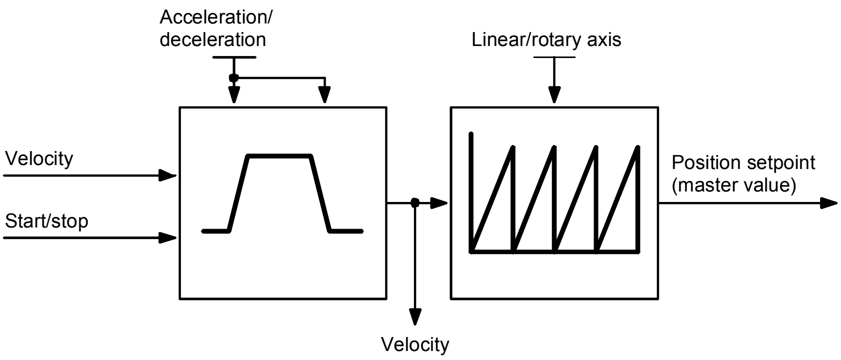


Fig. 10-3 Principle of the Virtual Master

10.2.1 Function Description

Enable Virtual Master

The virtual master function is always executed, irrespective of the current mode. You only need to enable the virtual master for the axis on which you plan to use it.

NOTE

The enable is used exclusively for optimization of the processing time and not for dynamic control of the setpoint generation. Activation of the enable can cause abrupt changes in outputs, depending on the last actuation (e.g. start active, velocity <> 0).

Start/Stop [ST_VM]

You control the virtual master using the start/stop command. It switches between the configured velocity setpoint and zero velocity, depending on the command. The velocity setpoint is defined using a parameter.

Reset [R_VM]

The reset control signal allows you to set the virtual master back to its initial status. As long as the reset virtual master control signal [R_VM] is active, the speed output and the position output are set to "0". If the input is deactivated, the velocity ramp function generator starts up again when the start command is active.

Set start position [S_VM]

The positive edge of the "set start value" sets the position output to a parameterizable start value.

CAUTION

You should only use the reset and set start position control signals if no axes are activated in synchronization mode. Otherwise, the rapid change in the virtual master's output leads to a sudden compensation movement from the slave axis.

Velocity Ramp-Function Generator

The velocity ramp-function generator matches the velocity at the output to the velocity setpoint with reference to the acceleration setting. The status is displayed via outputs.

The velocity output is normalized to 10*LU/min.

Position Setpoint Generation

Downstream of the velocity ramp function generator is an integrator which is used for position setpoint generation. The integrator integrates the current velocity and adjusts it to the length of the linear / rotary axis.

The path setpoint is normalized to LU.

Linear / Rotary Axis

In order to generate the position setpoint, the length of the virtual master axis can be set using the linear / rotary axis parameter.

NOTE

If the virtual master is connected to the synchronization block, it is necessary to ensure that the "virtual master linear/rotary axis" parameter matches the setting in the "master axis linear/rotary axis" parameter. In the event of a mismatch, the position setpoint generation on the virtual master axis causes setpoint jumps on the slave axis.

10.2.2 Differences Between MASTERDRIVES MC / SIMATIC Motion Control

Velocity Setpoints

In MASTERDRIVES MC you can specify the setpoint velocity either via a parameter connector as an absolute velocity, or via a connector. The velocity specified via the connector is a percentage of a settable rated velocity.

The default setting on Motion Control is parameter definition as an absolute velocity.

NOTE

If the default setting for the velocity setpoint definition is changed or the parameter connection for defining the absolute velocity is modified, the velocity setpoint can continue to be read and written via the task interface. The content of the parameter is not relevant for the function.

Velocity Setpoint

The current setpoint velocity is made available additionally as a percentage value referred to rated velocity.

Enable

In MASTERDRIVES MC, there is an "enable virtual master" input. If this input is not activated, the outputs are set to "0". No further execution takes place. Internal statuses are not changed. If the enable is activated, the virtual master carries on working, starting at its last internal status.

In the basic configuration of the standard application, the enable is set as static.

In the SIMATIC Motion Control solution, the enable is activated using the axis management task. Here, too, you should only enable the virtual master in the axis for which you need it.

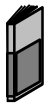
Linear/rotary axis

Since SIMATIC Motion Control enables each axis to access machine data MD11 and the "Linear/rotary axis" parameter of the virtual master of the other axis, the "Linear/rotary axis" parameter is acquired as follows for the master value source in synchronized operation:

Master value source	**"Linear / rotary axis" parameter ..**
Virtual master	.. from appropriate virtual master
Actual-value control	.. evaluation of MD11 of appropriate axis
Setpoint control	.. evaluation of MD11 of appropriate axis

10.2.3 Function Parameters

Various parameters must be set in order to define the virtual master as a master value source.



Further Information

The parameters are described in the "Task Description" in the chapter entitled "Synchronization Parameter Tasks".

Parameters

Name	Input Limit		Unit	Default
	Lower	Upper		
Linear/rotary axis (U687)	0 0: Linear axis >0: Rotary axis	2^{31}-1	LU	4096
Acceleration/deceleration (U685)	0 0: Step function >0: Defined acceleration	2^{31}	1000 LU/s^2	204
Set value virtual master axis (U693)	0	2^{31}-1	LU	0
Velocity setpoint (U679)	-2^{31}	2^{31}-1	10 LU/min	0
Mode for defining the velocity setpoint (MASTERDRIVES MC only) (U683)	0: Input in % of rated velocity 1: Definition in [10 LU/min]		-	0
Rated velocity (MASTERDRIVES MC only) (U682)	0	2^{31}-1	10 LU/min	0
Velocity setpoint (MASTERDRIVES MC only) (U681)	0	200	%	0

Actual Values

The table below describes the virtual master status bits.

Bit	Meaning
0	Virtual master stationary
1	Acceleration active
2	Velocity setpoint reached
3	Deceleration active
4	Virtual master moving
5 to 31	Reserved

10.2.4 Control Signals

Start Virtual Master [ST_VM] (U684.02)
The control signal start virtual master [ST_VM] starts or stops the velocity ramp function. The starting velocity is adjusted to the set velocity or to v = 0 at the acceleration set.

Reset Virtual Master [R_VM] (U684.01)
As long as the reset virtual master control signal [R_VM] is active, the velocity output and the position output are set to 0.

Set Virtual Master [S_VM] (U684.03)
The positive edge of the set virtual master control signal [S_VM] sets the position output to the set value. The velocity output is set to the velocity setpoint.

Enable Virtual Master (MASTER-DRIVES MC only) (U689)
If the enable is not set, the outputs are set to zero. The block is not calculated. The default entry for the circuit is "1".

10.2.5 Checkback Signals

Virtual Master Active [FUR_VM] (B815)
The FUR_VM signal indicates that the virtual master is active, i.e. the velocity output is not equal to 0.

Additional Checkback Signals
The following checkback signals are available internally for additional circuits. The checkback signals are grouped together in the actual value interface as virtual master status.

Acceleration Ramp Active (B816)
The virtual master accelerates to the velocity setpoint.

Velocity Setpoint Reached [VM_RA] (B817)
The acceleration or deceleration is complete, the velocity at the output matches the velocity setpoint.

Deceleration Ramp Active (B818)
The virtual master decelerates to the velocity setpoint.

Output = 0 (B819)
The velocity output is = 0, the virtual master is stationary.

10.2.6 Pulse Diagram

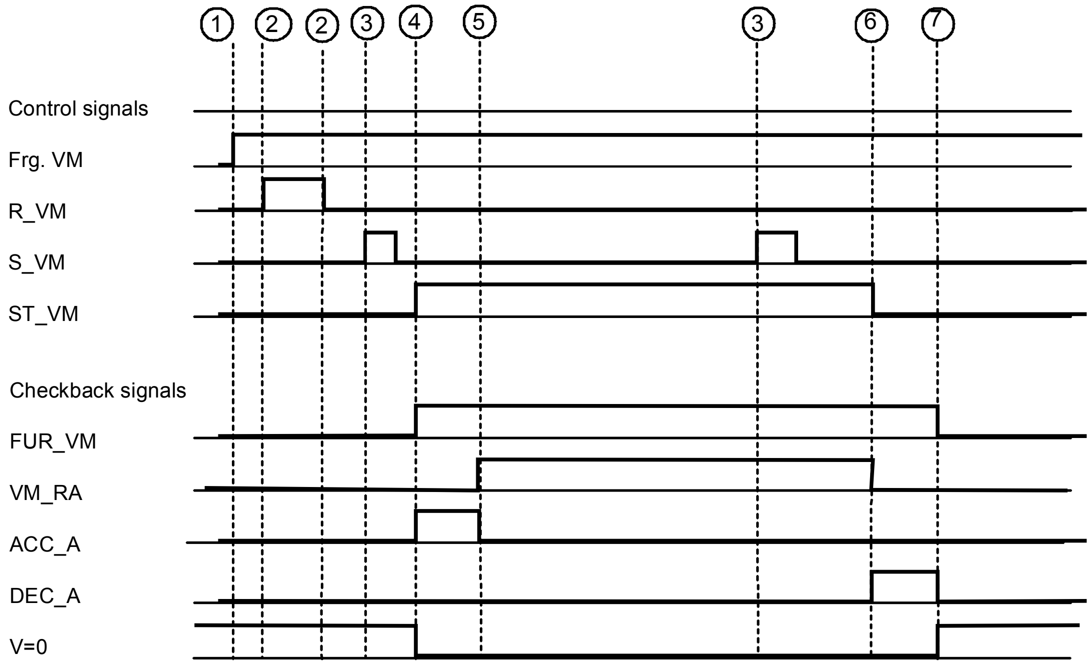


Fig. 10-4 Virtual master pulse diagram

Sequence

1. The virtual master is enabled. As long as the enable is inactive, the block is not calculated.
2. The outputs are reset using the reset control signal ([R_VM] = 1). The velocity output and the set position are set to "0" for as long as the signal remains.
3. The positive edge of the set virtual master control signal ([S_VM] = 1) sets the velocity output to the initial value and the position output to the set value.
4. Activating the start/stop control signal ([ST_VM] = 1) starts the virtual master. The velocity output is calculated in accordance with the velocity setpoint and acceleration entered, and the path setpoint is integrated on the basis of the actual speed. The virtual master active ([FUR_VM] = 1) and acceleration ramp active are set.
5. Once the velocity setpoint is reached, the output is activated ([VM_RA] = 1). The acceleration ramp active output is switched off.

6. The virtual master is switched off using the start/stop control signal ([ST_VM] = 0). The velocity output is reduced by the deceleration from the current setpoint to "0". The velocity setpoint reached output is deactivated ([VM-RA] = 0), and the deceleration ramp active output is activated ([DEC_A] = 1).
7. The deceleration process is complete. The velocity output becomes "0", and the position setpoint retains its actual value. The virtual master active ([FUR_VM] = 0) and deceleration ramp active ([DEC_A] = 0) outputs are deactivated and the virtual master stationary output is activated ([V=0] = 1).

10.3 Master Value Selection

Overview

You can choose between various master value sources for the coupled motion:

- Actual value control :
  If the master axis is derived from the actual position value of another axis, or, expressed in more simple terms, from an external position encoder, we refer to actual value control.
- Setpoint control :
  If the master axis is derived from the position setpoint of another axis, we refer to setpoint control.
- Virtual master :
  The virtual master calculates a position setpoint depending on a defined velocity.

These master value sources are preconnected as an operating mode in synchronized operation, but you can set 3 optional sources as the master setpoint via parameter U600 [834.1]. You can select one of these three sources with parameter U606. This can be:

- The output of the virtual master axis

  Output connector K817 [832] of the virtual master axis is connected to a SIMOLINK send word for the following drives. It is also essential that you make the connection to the synchronization block for the master drive via the "detour" through the receive buffer of SIMOLINK (e.g. KK7031 [150.7]) and not directly from the virtual master. In other words, do not use KK817. In this way you can guarantee that the master drive receives its position setpoint from the virtual master axis at the same time as all the slaves.
- The output of a position sensor as real master

  For synchronization with a real master, the measured actual value is connected to the input of the synchronization block. This value can be supplied by SIMOLINK or a position sensor.

  There are also options for adjusting the position setpoint externally (U460 and U461) for synchronization mode.

The mechanisms for selecting the master axis are different in the MASTERDRIVES Motion Control (Technology option F01) solution to the SIMATIC Motion Control solution.

10.3.1 Function Description

The master value selection is determined by the "master value source" parameter. The synchronization routine handles the different master value sources in exactly the same way. It is possible to switch between the master values during operation.

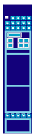
The "path setpoint" master value is reflected in the cycle length for the master axis. The path setpoint is differentiated. In turn, a velocity setpoint in per cent of rated velocity is generated as a precontrol value from the positional change per cycle. The quality of this signal depends on the resolution setting. As an alternative, you can also link a velocity setpoint master U600.04-06 in per cent of rated velocity directly to the synchronization block in parallel to the path setpoint. You should always use the velocity input for synchronization applications requiring higher velocities. Path setpoint and velocity setpoint must correspond to one another, the normalization velocity master (U607.2) must be correctly parameterized.

The path setpoint, the change in path per cycle and the velocity setpoint in percent are made available to the other synchronization functions.

NOTE

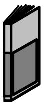
When switching between master values during operation, you should make sure that the master values are as similar as possible at the time of the switchover, in order to prevent a position setpoint jump on the slave axis.

10.3.2 Differences Between MASTERDRIVES MC / SIMATIC Motion Control



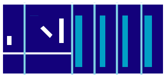
- Master axis: linear/rotary axis (U601)
  On the MASTERDRIVES MC, the cycle length of the master axis is set on the synchronization block using the "master axis: linear/rotary axis" parameter.
- The master axis is selected by setting the BICO parameters of the SIMOLINK interface.

With MASTERDRIVES MC, you have additional master value processing options in the form of the "real master" and "master value correction" blocks.



Further information

The functions of the blocks are described in detail in Chapters 10.10 "Real Master" and 10.11 "Master Value Correction.



Master axis: axis number
With SIMATIC Motion Control you can select the master axis in addition to the master value source. The axis selection transfers both the master value and the cycle length of the selected master axis.

10.3.3 Function Parameters

Various parameters must be set for the master value selection.

Further Information

The parameters are described in the "Task Description" in the chapter entitled "Synchronization Parameter Tasks".

Parameters

Name	Input Limit		Unit	Default
	Upper	Lower		
Master axis: linear/rotary axis (U601)	0 0: Linear axis >0: Rotary axis	2^{31}-1	LU	4096
Master axis: axis number selection (with SIMATIC Motion Control only)	1	8/16/32	-	0
Master axis: master value source (U606)	0: Actual value control 1: Setpoint control 2: Virtual master		-	0

10.4 Operating Cycles

Overview

The following different cycles are available in synchronization mode:

- Continuous cycle
  In the continuous cycle, the slave axis is always operational and follows every change in the master value. This is the standard situation for all permanently coupled axes where the synchronization acts as a substitute for mechanical gears or longitudinal shafts.
- Start cycle
  The start cycle allows an axis to be coupled dynamically and synchronized to a running master axis for a definable cycle. The initial state of the slave axis is stationary. The start cycle is typically used for axes which are mostly stationary and which only need to be activated in response to an explicit request, such as the initiation of a reject movement on detection of a defective part.
- Stop cycle
  The stop cycle allows dynamic decoupling of an axis and desynchronization from an in-progress movement for a definable cycle. The initial state of the slave axis is synchronized. The stop cycle is typically used for axes which are mostly operated in synchronism, and which only need to be stopped in response to an explicit request, e.g. in the event of missing products on the conveyor.
- Catch-up
  In addition to the start / stop cycle, the catch-up function provides another option for decoupling or recoupling a drive from or to an assembly of motors working in perfect synchronization (e.g. a shaftless printing press). The synchronization (closing up) does not require synchronization with the master axis. After the catch-up, there is no defined relationship between the master and the slave axis. There are independent functions which allow synchronization after catch-up (see 10.8 "Master Value Synchronization for Offset Angle Setting"). The catch-up is described in Chapter 10.9 "Catch-up".

10.4.1 Continuous Cycle

In the continuous cycle, the slave axis is always synchronized, irrespective of control commands or other signals.

Therefore, the best time to use this mode is when the synchronization relationship between the master axis and the slave axis is to be generated when they are stationary. If the master axis is then moved in any operating mode, the slave axis follows its movement directly.

Example

The master and slave axis are moved into their starting positions independently of one another (e.g. in MDI mode). This starting position also defines the angular relationship between the axes when they are in synchronization. If you now switch the slave axis to synchronization mode, the slave axis will follow all the movements of the master axis. The relationship between the two axes, as defined by their starting positions, is maintained.

NOTE

If you switch to synchronization mode from another mode, e.g. MDI, when the master axis is moving, the slave axis will accelerate rapidly to the velocity of the master axis. In this case, the positional relationship between the master axis and the slave axis is defined by their positions with respect to one another when you switch mode. A specific relationship can only be achieved by means of subsequent synchronization using print marks.

In order to change into synchronization mode when the master axis is moving, you should use the start or stop cycle or the catch-up.

10.4.2 Start Cycle

In the start cycle, the initial state of the slave axis is stationary. The slave axis can be moved to its initial position in advance using, for example, MDI mode. The start cycle is activated by a trigger signal (Start/Stop Trigger [SST] or Start/Stop continuous [SSC]). The start cycle begins when the master axis crosses the coupling position.

The start cycle allows the exact number of pulses of the master axis specified in the "on/off length" parameter to pass. Synchronization and desynchronization is path-dependent. The "path for ramp" parameter defines the synchronization or desynchronization distance.

With MASTERDRIVES version 1.6 and later, you can specify a decoupling position (special configuration U475=1) as an alternative to the ON/OFF length. The constant velocity phase of the start cycle ends at this position (see also function diagram 834a, column 1, annotation <3>).

During the coupling ramp and the decoupling ramp, the master axis traverses through twice the distance. The effect of this is as follows:

- ♦ During the complete coupling process, the master axis traverses through a distance equal to the on/off length + the distance for the ramp.
- ♦ The slave axis only traverses through a distance equal to the on/off length.

The sequence of movements of the master axis and slave axis is shown in the diagram below.

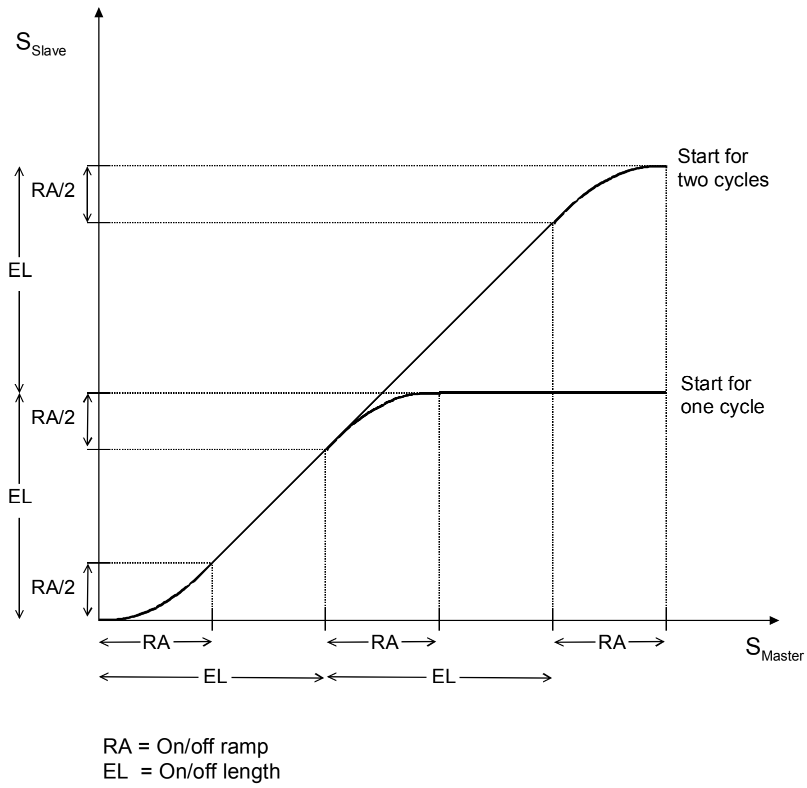


Fig. 10-5 Principle of the Start Cycle

Notes

- ♦ The coordinate origin represents the beginning of a start cycle.
  - The master axis is on the couple position
  - The slave axis is in its starting position
- ♦ Synchronization is reached when
  - Smaster = couple position + ramp
  - Sslave = starting position + ramp/2
- ♦ Synchronized motion is maintained over the distance: n * start cycle length – ramp, i.e. desynchronization begins when
  - Smaster = couple position + n * start cycle length
  - Sslave = starting position + n * start cycle length – ramp/2
- ♦ The start cycle finishes when
  - Smaster = couple position + n * start cycle length + ramp
  - Sslave = starting position + n * start cycle length

	Master axis distance	**Slave axis distance**
Synchronization	RA	RA/2
1^{st} cycle synchronization	EL – RA	EL – RA
2^{nd} – n^{th} cycle	(n-1) * EL	(n-1) * EL
Desynchronization	RA	RA/2
Total	n * EL + RA	n * EL

Where EL = start cycle length and RA = ramp

Table 10-1 Start cycle distances

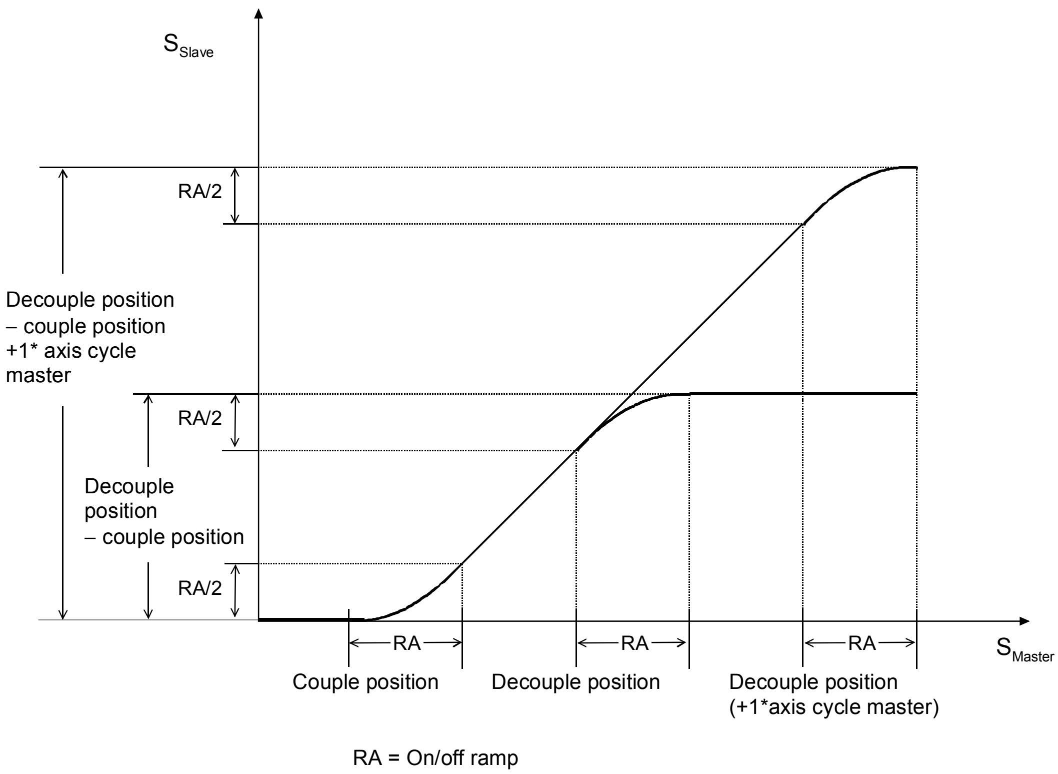


Fig. 10-6 Start cycle with special configuration U475=1

Explanation

- The slave axis is initially stationary at the starting position
- Synchronization is reached when
  - Smaster = couple position + ramp
  - S_{slave} = starting position + ramp/2
- Synchronized motion is maintained until the decouple position (+ n * master axis cycle)
  - S_{master} = decouple position + n * master axis cycle
  - S_{slave} = starting position + decouple position – couple position +n * master axis cycle – ramp/2
- The start cycle finishes when
  - S_{master} = decouple position + n * master axis cycle + ramp
  - S_{slave} = starting position + decouple position – couple position +n * master axis cycle

	Master axis position	**Slave axis position**
Beginning of start cycle	COUP	START
Beginning of synchroniz.	COUP + RA	START+RA/2
End of synchronization	DECOUP + n*MACl	START + DECOUP – COUP + n*MAZL - RA/2
End of start cycle	DECOUP + n*MACl+RA	START + DECOUP – COUP + n*MACl

Table 10-2 Start cycle distances

10.4.3 Stop Cycle

In the stop cycle, the initial state of the slave axis is moving synchronous to the master axis. The stop cycle is activated by a trigger signal (Start/Stop Trigger [SST] or Start/Stop Continuous [SSC]). The stop cycle begins when the master axis crosses the coupling position.

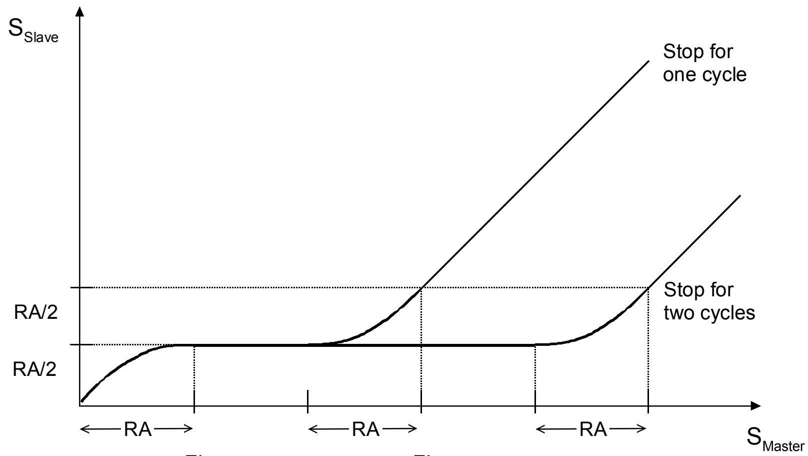
The stop cycle stops the slave axis for the exact number of pulses of the master axis specified in the "on/off length" parameter. Synchronization and desynchronization is path-dependent. The "path for ramp" parameter defines the synchronization or desynchronization distance.

With MASTERDRIVES version 1.6 and later, you can specify a decoupling position (special configuration U475=1) as an alternative to the ON/OFF length. The constant velocity phase of the start cycle ends at this position (see also function diagram 834a, column 1, annotation <3>).

During the coupling ramp and the decoupling ramp, the master axis traverses through twice the distance. The effect of this is as follows:

- ♦ During the complete coupling process, the master axis traverses through a distance equal to the on/off length → the distance for the ramp.
- ♦ The slave axis continues traversing through a distance equal to the on/off ramp (→ the slave axis stops for the on/off length).

The sequence of movements of the master axis and slave axis is shown in the diagram below.



RA = On/off ramp
EL = On/off length

Fig. 10-7 Principle of the Stop Cycle

Notes

- The coordinate origin represents the beginning of a stop cycle.
  - The master axis is on the couple position
  - The slave axis operating synchronously at the position defined by the synchronization reference (see Principle of Synchronization).
- The slave axis comes to a halt when
  - Smaster = couple position + ramp
  - Sslave = starting position + ramp/2
- The slave axis remains stationary until the master axis has covered the distance: n * start cycle length – ramp, i.e. synchronization begins at
  - Smaster = couple position + n * start cycle length
  - Sslave = starting position + ramp/2
- The stop cycle has finished when
  - Smaster = couple position + n * start cycle length + ramp
  - Sslave = starting position + ramp

	Master axis distance	**Slave axis distance**
Synchronization	RA	RA/2
1st cycle synchronization	EL – RA	0
2nd – nth cycle	(n-1) * EL	0
Desynchronization	RA	RA/2
Total	n * EL + RA	RA

Where EL = start cycle length and RA = ramp

Table 10-3 Stop cycle distances

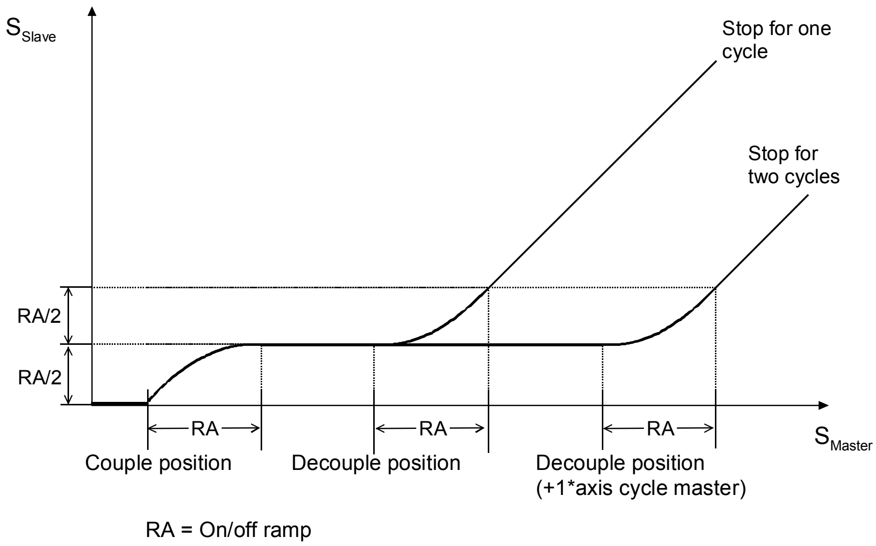


Fig. 10-8 Stop cycle with special configuration U475=1

Explanation

The slave axis is operating synchronously at the position defined by the synchronization reference (see Principle of Synchronization).

- ♦ The coordinate origin represents the beginning of a stop cycle.
  - The master axis is at the couple position
  - The slave axis operating synchronously at the position defined by the synchronization reference (see Principle of Synchronization).
- ♦ The slave axis comes to a halt when
  - S_{master} = couple position + ramp
  - S_{slave} = starting position + ramp/2
  - The slave axis remains stationary until the master axis has covered the distance: n * start cycle length – ramp, i.e. synchronization begins at
  - S_{master} = couple position + n * start cycle length
  - S_{slave} = starting position + ramp/2
- ♦ The stop cycle has finished when
  - S_{master} = couple position + n * start cycle length + ramp
  - S_{slave} = starting position + ramp

	Master axis distance	**Slave axis distance**
Synchronization	RA	RA/2
1st cycle synchronization	OL – RA	0
2nd – nth cycle	(n-1) * OL	0
Desynchronization	RA	RA/2
Total	n * OL + RA	RA

Table 10-4 Stop cycle distances

10.4.4 Retriggering

The start or stop cycle can be extended by one "on/off length" by further rising edges on the control signal [SST] if the following conditions are met:

- The second/further edges occur after the first start/stop cycle has started, i.e. the coupling position must have already been crossed by the master axis, and
- The return to the resting position (ramp at the end of the start/stop cycle) has not yet begun.

Trigger signals outside the permissible range are rejected.

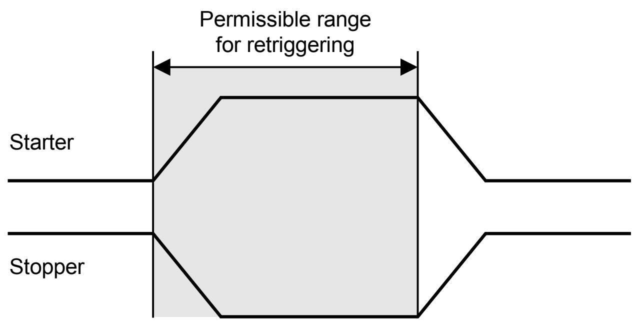


Fig. 10-9 Permissible Range for Retriggering

NOTE

In order to minimize the reaction time of the [SST] control signal, it can be connected directly to a digital input on the MASTERDRIVES MC.

With the SIMATIC Motion Control application, there is the option of using MD46 to parameterize a digital input of the MASTERDRIVES MC so that it can be used as a trigger signal for the start / stop cycle.

10.4.5 Continuous Start/Stop Cycle - Intermittent cycle

If the start/stop cycle is started with the SSC signal, there are two different types of operation:

- ♦ Intermittent cycle
- ♦ Continuous cycle

Intermittent Cycle

The intermittent cycle is activated automatically if the following conditions are fulfilled:

- ♦ The master axis is a rotary axis, i.e. it passes the coupling position on a cyclical basis
- ♦ The sum of the on/off length and the ramp is less than the length of the master axis, i.e. the slave axis is in its starting position before the coupling position is passed again
- ♦ The control signal [SSC] = 1 (as long as SSC = 1, the SST control signal has no effect)

The slave axis starts or stops each time the coupling position is crossed for as long as the SSC signal is active.

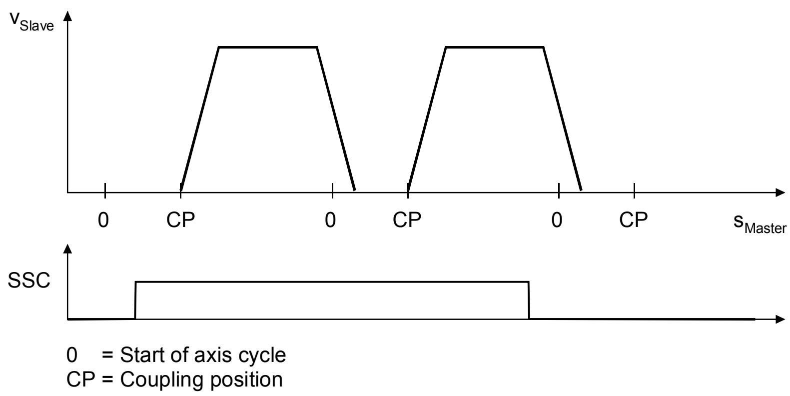


Fig. 10-10 Intermittent Start Cycle

Continuous Cycle

The continuous cycle is activated automatically if the conditions for intermittent operation are not fulfilled, i.e.:

- ♦ The master axis is a linear axis, or
- ♦ The master axis is a rotary axis and the sum of the on length and the ramp is greater or equal to the length of the master axis, i.e. the slave axis reaches its start position before the coupling position is passed again.

The slave axis starts or stops when the coupling position is crossed and the SSC signal is activated. The start/stop cycle is extended for as long as the SSC signal is active.

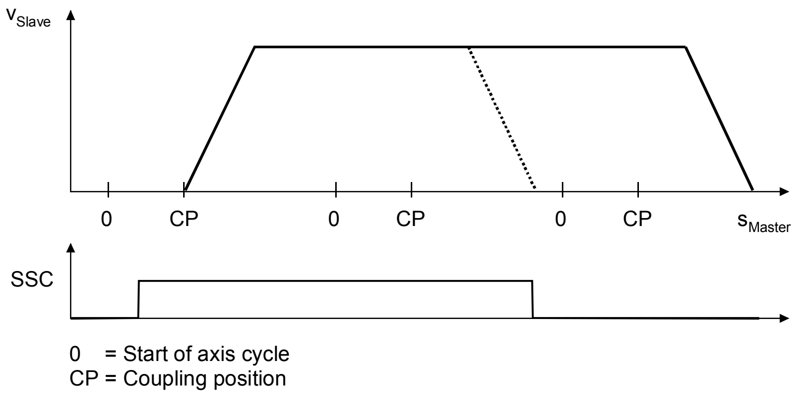


Fig. 10-11 Continuous Start Cycle

Conditions

The following conditions apply to the configuration of the start/stop cycle:

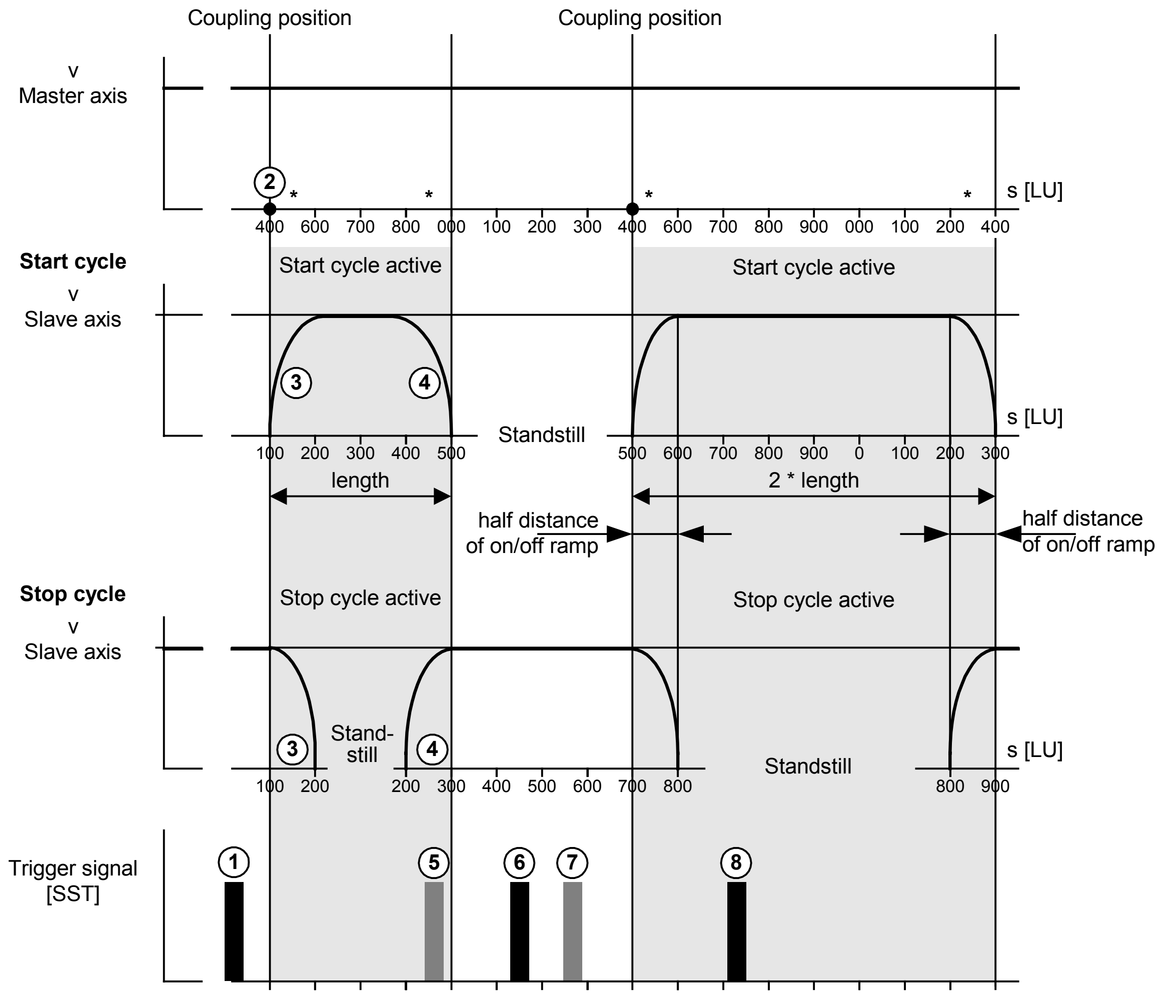
- ♦ 1 ≤ on/off ramp ≤ on/off length
- ♦ On/off ramp = 1 produces a very rapid coupling/decoupling process
- ♦ If on/off ramp = on/off length, there is no true angular synchronization, because there is no boundary between the coupling and decoupling process. The trapezoidal movement is transformed into a delta shape.

10.4.6 Examples

Example for Start/Stop Cycle

The following specifications are used for the example below:

- Synchronization function = 1:1
- v_{Master} = constant
- Master axis: linear/rotary axis = 1000 [LU]
- Slave axis: linear/rotary axis = 1000 [LU] (MD11)
- Coupling position CP = 400 [LU]
- Start/stop cycle: on/off length = 400 [LU]
- Start/stop cycle: on/off ramp = 200 [LU]
- The master and slave axes have the same path resolution [LU]



* **NOTE:** Scaling of the master axis is not linear in these areas

Fig. 10-12 Example Start/Stop Cycle

① The rising edge of the trigger signal prepares the start/stop cycle.

② The start or stop cycle is started when the coupling position is crossed.
The current data assumed for the slave axis at the beginning of the start cycle (standstill) are:
- Current position 100 [LU]
- The slave axis crosses position 100 [LU] at the beginning of the stop cycle (coupled-motion)

③ Start cycle: for acceleration, the slave axis needs "on/off ramp" / 2 and thus 100 [LU].
Stop cycle: for deceleration, the slave axis needs "on/off ramp" / 2 and thus 100 [LU].
The master axis traverses through twice the distance during synchronization and desynchronization.

④ The start cycle also needs 100 [LU] for deceleration and the stop cycle needs 100 [LU] for acceleration. The master axis traverses through twice the distance during synchronization and desynchronization.

⑤ The trigger signal has no effect, because the start cycle is already in the deceleration phase or the stop cycle is already in the acceleration phase.

⑥ The trigger signal prepares the next start/stop cycle.

⑦ The trigger signal has no effect, because the previous trigger signal has already activated the preparation but the coupling position has not yet been crossed by the master axis.

⑧ The start/stop cycle is active. Reactivation of the trigger signal extends the start/stop cycle by a further "length".

10.4.7 Example Calculations for Start/Stop Cycle

Example 1 - Start Cycle

Specifications:

- ♦ The master and slave are identical rotary axes (in each case one revolution equals 1000 [LU]).
- ♦ The slave axis traverses with synchronization 1:1 to the master axis and is to perform exactly one revolution after the start position "0" when the master axis crosses the coupling position 400 [LU].
- ♦ 10 % of the revolution of the slave axis is to be used for soft coupling/decoupling.

Parameters:

- ♦ Start/stop cycle:
  Length = 1000 [LU] = one revolution of the slave axis
- ♦ Start/stop cycle:
  Coupling position = 400 [LU] = start of slave axis movement
- ♦ Start/stop cycle:
  On/off ramp = 10 % of length = 100 [LU]

Result:

- ♦ The slave axis is stationary at position 0 [LU].
- ♦ When coupling position 400 [LU] is crossed in the master axis, the slave axis starts to accelerate.
- ♦ After 50 [LU], the slave axis has reached the velocity of the master axis. From this point on, the master and slave axes are in synchronism.
- ♦ When the slave axis reaches position 950 [LU], the synchronization is terminated and the deceleration process is initiated.
- ♦ After exactly one revolution, the slave axis comes to a standstill again at position 0.

Example 2 – Start Cycle with Down-Circuit Gearbox

Specifications:

- A gear ratio of 2:1 is used between the slave axis and master axis.
- Angular synchronization is required starting at position 500 [LU] in the master axis.
- During the start cycle, the slave axis is to traverse through a distance of 2000 [LU].
- 10 % of the distance is to be used for soft coupling/decoupling.

Parameters:

- Gear ratio denominator (slave axis) = 2 (see below)
- Gear ratio numerator (master axis) = 1 (see below)
- Start/stop cycle: length =
  slave axis distance / gear ratio = 2000/2/1 = 1000 [LU]
- Start/stop cycle: on/off ramp =
  10 % of length = 100 [LU]
- Start/stop cycle: Coupling position = 500-length = 400 [LU]

Result:

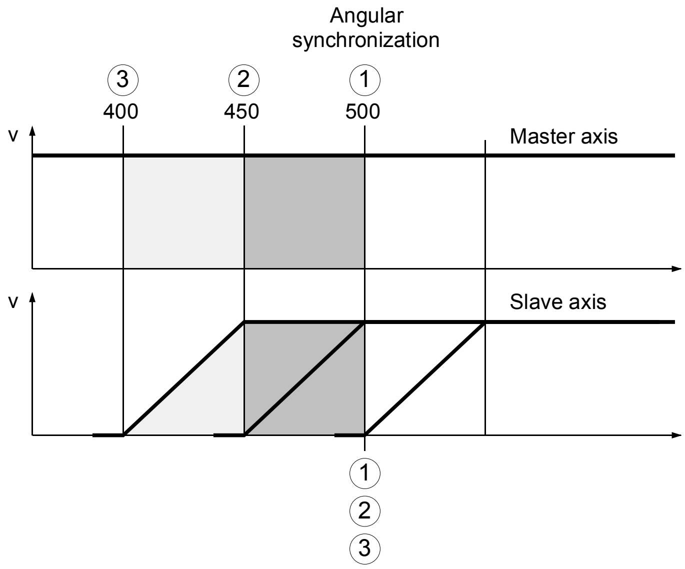


Fig. 10-13 Procedure for Determining the Coupling Position for Angular Synchronization

① If a coupling position of **500** is defined, synchronization only takes place after 500, since the slave axis first has to accelerate from standstill to the required velocity.

② If you now move the coupling position forward by a distance equal to the acceleration path (i.e. at **450** (500 – on/off ramp / 2), the velocity synchronism is achieved at position 500 of the master axis, however the slave trails behind by 50, because the master axis traversed through twice the distance of the slave axis (area = distance).

③ The coupling position is therefore moved forward again by 50 to **400**. In this case, the slave axis reaches angular synchronization at position 500 of the master axis.

Example 3 – Stop Cycle

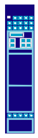
Specifications for µm normalization [LU = 1 µm]

- The master and slave are identical rotary axes (in each case one revolution equals 10000 [µm].
- No gear ratio is used between the slave and master axes (1:1).
- The slave axis is to remain stationary for exactly one revolution of the master axis.
- The slave axis standstill starts at position 250 [µm] of the master.
- The deceleration time of the slave axis is 100 [ms].
- The master axis traverses at a velocity of 600 [mm/min].

Parameters

- During the deceleration time, the slave axis decelerates from the velocity of the master axis to a standstill and traverses through the following distance:
  Distance for deceleration = ½ * 0.1 [s] * (600 [mm/min] / 60) = 500 [µm]
- The master axis traverses through twice the distance during this period, since it continues to move at a constant velocity.
  Distance of master axis = 0.1 [s] * (600 [mm/min] / 60) = **1000** [µm]
- Start/stop cycle:
  On/off ramp = distance of master axis = **1000** [µm]
- Start/stop cycle:
  Length = 10000 [µm] (length of standstill) + 1000 [µm] (on/off ramp) = **11000** [µm]
- Start/stop cycle:
  Coupling position = 250 [µm] (master position for standstill) – 1000 [µm] (on/off ramp) = -750 → transposed onto a full circle → **9250** [µm]]

10.4.8 Differences Between MASTERDRIVES MC / SIMATIC Motion Control

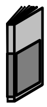


On the MASTERDRIVES MC, the default setting for the coupling position is taken from parameter U613/KK822. Parameter U609 is linked to connector KK822 for this purpose.

You can change the standard setting in order to route the offset to a different connector. In this case, parameter U613 is irrelevant. Controlling the parameter via the task interface has no effect on the function.

10.4.9 Function Parameters

Various parameters must be set for the start/stop cycle.



Further Information

The parameters are described in the "Task Description" in the chapter entitled "Synchronization Parameter Tasks".

Parameters

Name	Input Limit		Unit	Default
	Lower	Upper		
On/off length (U611)	0	2^{31}-1	LU	0
Length for on/off ramp (U610)	0	2^{31}-1	LU	0
On/off coupling position (U608)	2^{31}	2^{31}-1	LU	0
Offset for coupling position (U613/KK822/U609)	2^{31}	2^{31}-1	LU	0

Actual Values

Name	Meaning
Slave axis status	Bit 2=1: Start / stop cycle active Bit 3=1: Start cycle active Bit 4=1: Stop cycle active Bit 5=1: Start / stop cycle ramp inactive

10.4.10 Control Signals

Start/Stop Trigger [SST] (U612.02)

The control signal Start/Stop Trigger [SST] starts the start / stop cycle once. The cycle starts on the positive edge of the signal. The axis begins the start / stop cycle the next time it passes over the coupling position. During the ongoing start / stop cycle, the start / stop cycle can be retriggered by another positive edge, provided it has not yet started to return to the resting position (ramp at the end of the coupling process). Every retrigger extends the start / stop cycle by a start / stop cycle length.

Start/Stop Continuous [SSC] (U612.01)

The control signal Start/Stop Continuous [SSC] activates the start / stop cycle to run on a continuous (static) basis. The signal is evaluate before the axis passes over the coupling position and / or before the return to a resting position.

NOTE

The trigger signals [SST] and [SSC] must be active for at least one controller cycle to ensure that they are recognized.

10.4.11 Checkback Signals

Start/Stop Cycle Active [CL_A]

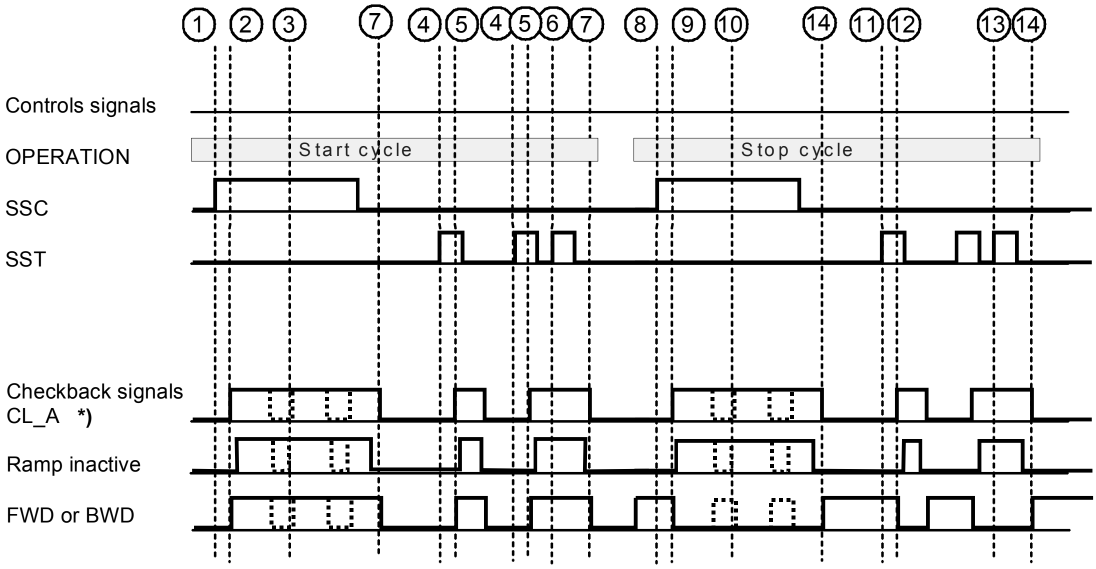
This checkback signal indicates that a start or stop cycle is active.

With MASTERDRIVES MC and SIMATIC Motion Control ≥ V1.00.48, you are given three additional binectors or signals which annunciate the current status in more detail (separately according to modes Start Cycle Active (B832) and Stop Cycle Active (B833)).

Ramp Inactive (constant traversal) (B831) indicates that the axis is either synchronized (start cycle) or has stopped (stop cycle).

These 3 binectors or signals are stored in the actual synchronization values in the slave axis status word.

10.4.12 Pulse Diagram



***)** In addition to the start/stop active signal [CL_A], the signals start cycle active (B832) and stop cycle active (B833) are also available in MASTERDRIVES MC.

Fig. 10-14 Start / stop cycle pulse diagram

Sequence

1. The start cycle is statically triggered by the signal [SSC]=1. The axis is currently stopped ([FWD], [BWD]=0).
2. The start cycle starts the next time the coupling position is passed. Depending on the direction in which the slave axis is moving, the checkback signal triggered is either axis moving forwards ([FWD] = 1) or axis moving backwards ([BWD] = 1). The start / stop cycle active checkback signal is also triggered ([CL_A] = 1). At the end of the synchronization ramp, the output ramp inactive = 1 is displayed until the desynchronization ramp starts.
3. Depending on how it is parameterized, the start cycle works in continuous or intermittent mode.
4. One start cycle is triggered by the signal [SST]=1.
5. The start cycle starts the next time the coupling position is passed. The start cycle is active for a single cycle. Depending on the direction in which the slave axis is moving, the checkback signal triggered is either axis moving forwards ([FWD] = 1) or axis moving backwards ([BWD] = 1). The start / stop cycle active checkback signal is also triggered ([CL_A] = 1). At the end of the synchronization ramp, the output ramp inactive = 1 is displayed until the desynchronization ramp starts.
6. The start cycle is retriggered by the signal [SST]=1.
7. The start cycle ends after the 2^{nd} (n^{th}) cycle.

8. The stop cycle is statically triggered by the signal [SSC]=1. The axis is currently moving ([FWD],[BWD]=1).
9. The stop cycle starts the next time the coupling position is passed. The checkback signal axis moving forwards ([FWD] = 0) or axis moving backwards ([BWD] = 0) is reset. The start / stop cycle active checkback signal is also triggered ([CL_A] = 1). At the end of the desynchronization ramp, the output ramp inactive = 1 is displayed until the synchronization ramp starts.
10. Depending on how it is parameterized, the stop cycle works in continuous or intermittent mode.
11. One stop cycle is triggered by the signal [SST]=1.
12. The stop cycle starts the next time the coupling position is passed. The start cycle is active for a stop cycle. The checkback signal axis moving forwards ([FWD] = 0) or axis moving backwards ([BWD] = 0) is reset. The start / stop cycle active checkback signal is also triggered ([CL_A] = 1). At the end of the desynchronization ramp, the output ramp inactive = 1 is displayed until the synchronization ramp starts.
13. The stop cycle is retriggered by the signal [SST]=1.
14. The stop cycle ends after the 2^{nd} (n^{th}) cycle.

10.5 Synchronization Functions

Overview

Synchronization mode supports various functions for determining the relationship between the master axis and the slave axis.

- Synchronization 1:1
  With 1:1 synchronization, the slave axis follows changes in the master value exactly.
- Gear synchronization
  In synchronization mode, changes in the master value are converted to setpoint changes of the slave axis via a configurable gearbox.
- Table synchronization / cam disk
  With table synchronization, the sequence of movements on the slave axis is defined by an NC table. The NC table contains configurable points between which a linear interpolation is performed. The position setpoint of the slave axis is calculated from the change in the master value with allowance for the table. Table synchronization can be used, for example, to reproduce a mechanical cam disk.

10.5.1 Synchronization 1:1

Function

The synchronization 1:1 function passes on changes in the master value without converting them into a setpoint change for the slave axis. This function is suitable for rigidly coupled axes with no gear transmission.

10.5.2 Gear Synchronization

Function

The master value is geared up or down flexibly using an electronic gearbox. The gear function behaves exactly like a mechanical gearwheel assembly. The ratio between master and slave can be modified with high resolution during operation and has long-term stability. The gear ratio is specified as a fraction (denominator, numerator) with two signed 16-bit numbers.

Transmission Ratio

The transmission ratio is defined as follows:

$$i = \frac{\text{Gear ratio numerator}}{\text{Gear ratio denominator}}$$

Example: i = 1 : 2 → Slave traverses at half the rate of the master

Example Applications

Several drives are to be operated in synchronism but with different speeds. This is necessary, for example, if a conveyor is to be driven across rollers of different diameters (D1, D2). Factor i for the gear ratio is calculated on condition that the peripheral speed of the rollers is identical. The speed of revolution (increments/time) is inversely proportional to the diameter of the roller. Assuming that the axes are mechanical:

$$i = \frac{\text{Gear ratio numerator}}{\text{Gear ratio denominator}} = \frac{\varnothing\ \text{Master roller}}{\varnothing\ \text{Slave roller}} = \frac{D1}{D2}$$

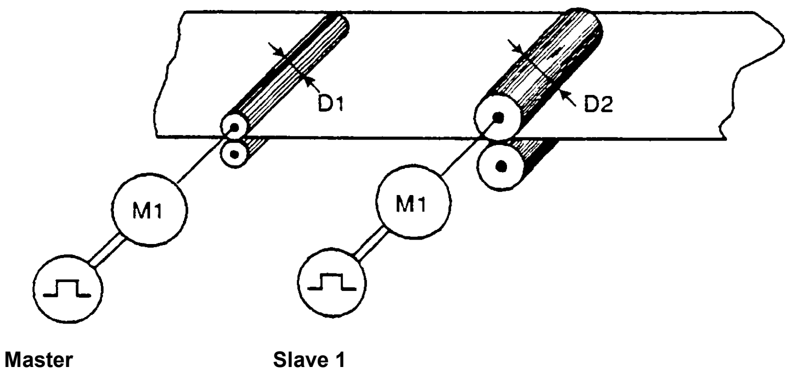


Fig. 10-15 Gears with different roller diameters

Limits

The following transmission ratio is permitted:

$$i = \frac{\text{Gear ratio numerator}}{\text{Gear ratio denominator}} = \frac{0 \ldots \pm 32767}{1 \ldots 32767} \le \frac{5}{1}$$

The limit of slave axis: master axis ≤ 5:1 (velocity of slave axis ≤ 5 x velocity of master axis) is imposed by the mechanical system, not the module. Higher transmission ratios cause slave axis "flutter" when the master axis is moving slowly.

Examples

i = 1 : 2 → The master axis traverses through twice as many increments as the slave axis

i = 2 : 1 → The slave axis traverses through twice as many increments as the master axis

i = 9998:9999 → Very fine gear adjustment (example application: roller wear)

Dynamic Gearbox

The transmission ratio i can also be modified or adjusted during the operating phase.

NOTE

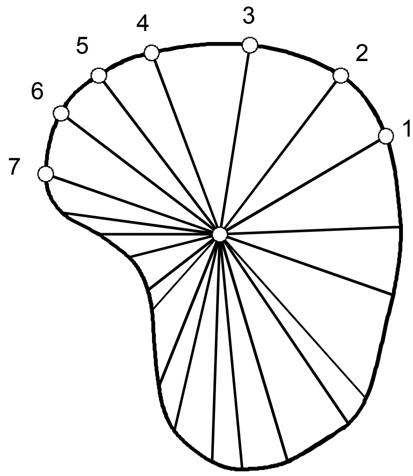
Any change has an abrupt impact on the slave axis.

10.5.3 Table Synchronization / Cam Disk

Description

An NC table contains interpolation points (pairs of positional values) which are used to define the assignment between a master axis position (plotted along the x axis) and a slave axis position (plotted along the y axis). Linear interpolation is used between two points in order to determine the value for constant velocity between two interpolation points for the slave axis. The interpolation points (pairs of positional values) define the movement profile of the slave axis, depending on the movement of the master axis.

A possible application of table synchronization is the electronic reproduction of a cam disk.



Position Master	Position Slave
0°	100 mm
30°	95 mm
60°	90 mm
90°	95 mm
:	:
:	:

Fig. 10-16 Cam disk and Associated Table

Corners occur at the transitions due to the linear interpolation. These can be minimized by using an appropriately large number of interpolation points.

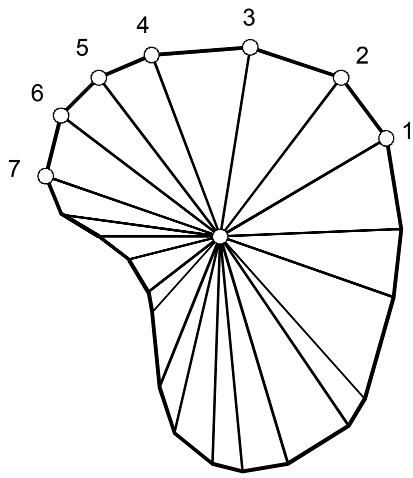
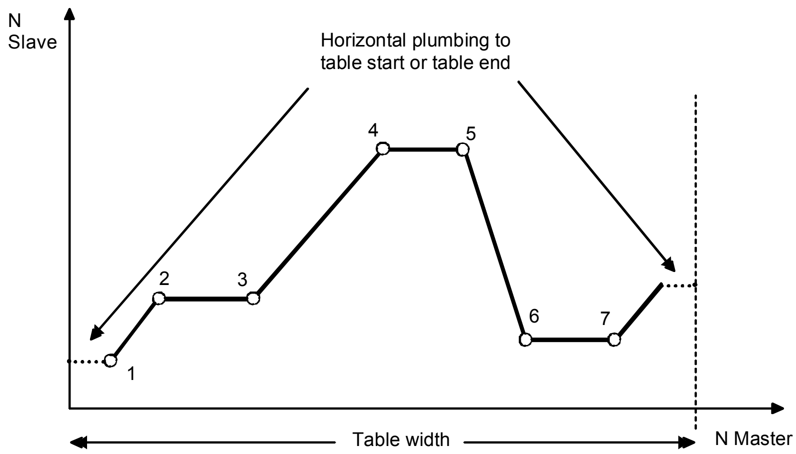


Fig. 10-17 Cam disk after Machining

Table Definition

Please note the following when using tables:

- ♦ The data sets must be sorted in ascending order of interpolation points (not interpolation values) when entered. Interpolation points correspond to the positions on the X axis (master), while interpolation values represent the positions on the Y axis (slave).
- ♦ Only data sets containing interpolation points within a range from 0 to the table width are accepted. If one interpolation point is outside the range, the table is rejected with an error.
- ♦ Horizontal plumbing takes place from the first data set (x^1, y^1) to the first (imaginary) interpolation point (0, y^o) if x^1 <> 0. The same applies at the end of the table. If necessary, the table is therefore continued horizontally (at the start or at the end).



NOTE

The width of the table and the length of the master axis do not necessarily have to agree. If there is a discrepancy, there is a different relationship to the slave axis in every master axis cycle.

For perfect synchronization, please refer to the notes in Chapter 10.8 "Master Value Synchronization for Offset Angle Setting".

Scaling

In order to program the machines as flexibly as possible, it is possible to scale the tables both in the X direction (master) and in the Y direction (slave). This means it is possible to machine a product in different sizes using the same table. Please note, however, that the permissible ranges for the position and velocity must not be exceeded.

Scaling in the X direction has the same effect as upstream gears. Any changes made to the X direction scaling factor are adopted immediately. This can lead to jerky movements on the slave axis in the event of improper use, since there is no ramp generation.

Scaling in the Y direction has an absolute effect. You can choose whether the scaling should become effective immediately or when the table next starts. Changing the scaling while the able is moving can mean that the slave axis has to make a sudden, jerky movement to compensate.

Continuous Output

During continuous output, there is an automatic jump back to the start of the table when the end of the table is crossed.

In the following example, the master axis moves at constant velocity. Continuous output is selected.

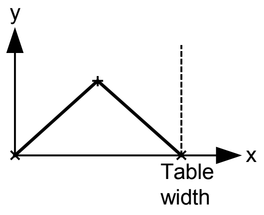


Fig. 10-18 Interpolation Points (Slave Position y, Master Axis Position x)

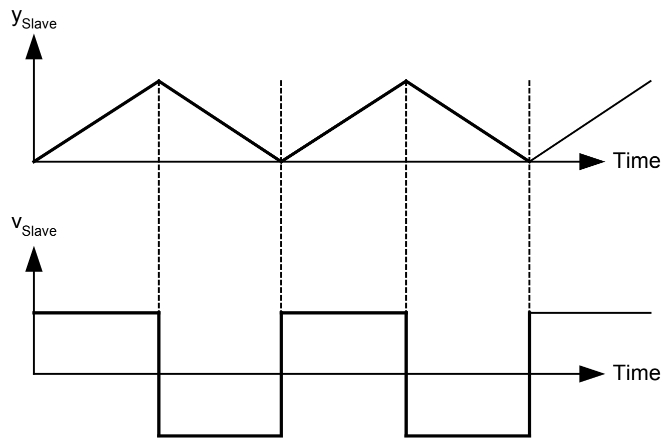


Fig. 10-19 Time Chart for Position y and Velocity v of Slave Axis

This setting is especially worthwhile if the slave axis is carrying out a cyclical execution which repeats automatically.

Stop at End of Table

With "stop at end of table", the table pauses at the last interpolation value when the end of the table is crossed. The return jump to the start of the table takes place after external synchronization by the **SYN_T** control signal "synchronize table".

In the following example, the master axis moves at constant velocity.

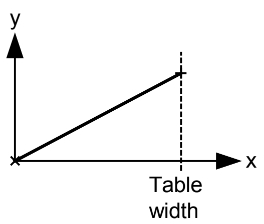


Fig. 10-20 Interpolation Points (Slave Position y, Master Axis Position x)

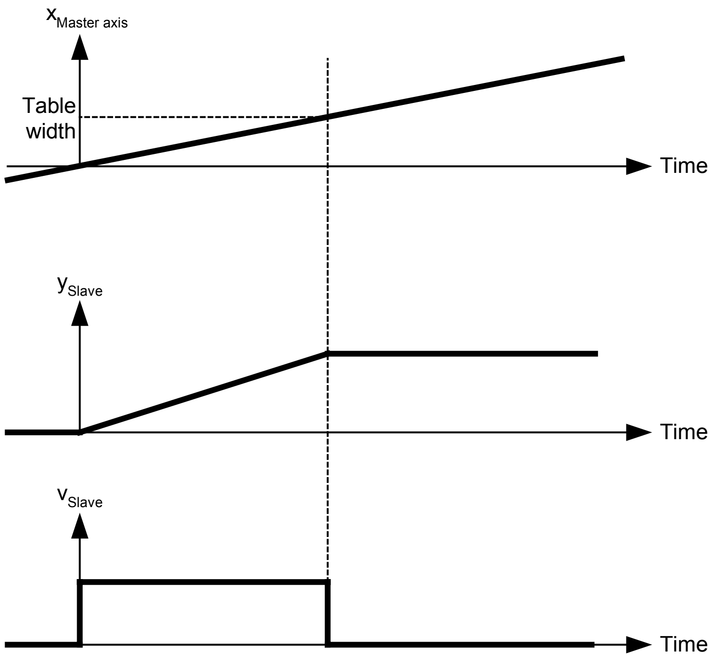


Fig. 10-21 Time Chart for: Position x of Master Axis; Position y and Velocity v of Slave Axis

When the master axis reaches table position x = table width, the slave axis stops. This setting is particularly suitable if slave axis dynamics are only required within a certain range (e.g. 0 to table width) (e.g. execution over the length of a product, with variable distances between products).

Window Table Actual-Value Control

If you are using the actual-value control option, parameter "Window Table – Actual-Value Control" is available with MASTERDRIVES MC with firmware version V1.51 and later and SIMATIC Motion Control ≥ V1.00.48.

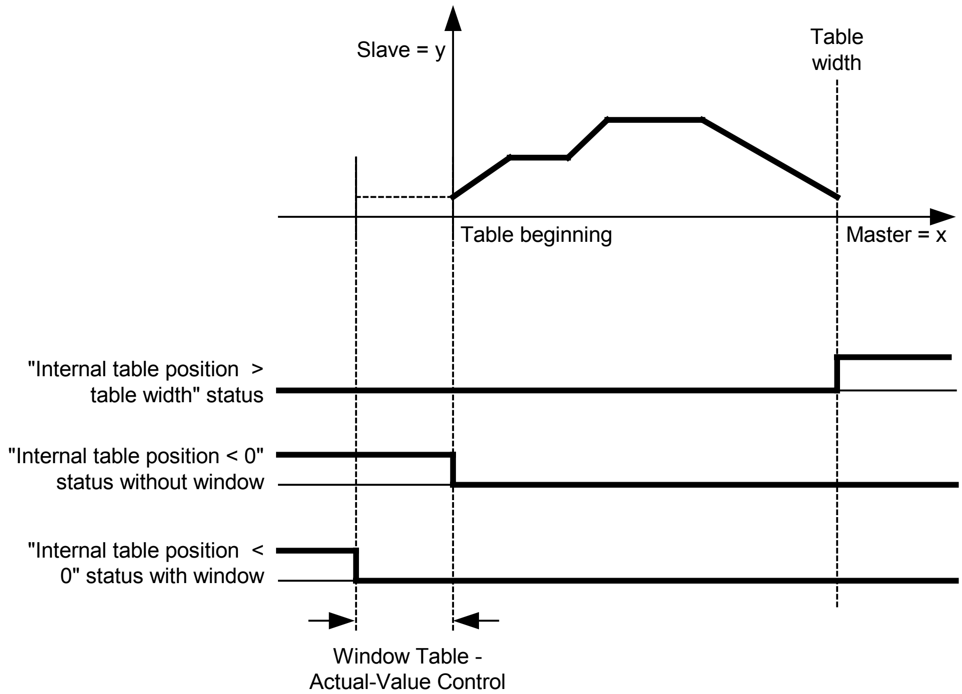
In table mode "Stop at table end" and actual-value control of the axis, the table can reach negative x coordinates at the instant of synchronization if the master axis is vibrating slightly. As a result, the NC table is deactivated. To avoid this problem, you can parameterize a suitable value (>0) for your installation in parameter "Window Table Actual-Value Control".

Traversal without Window

You are traversing without window if parameter "Window Table Actual-Value Control" is set to zero. In this case, the function is not activated.

If the table end is crossed in the positive direction with an NC table in "Stop at end of table" mode, status bit "Bit 7 = internal NC table position > NC table width" is set. The NC table is then deactivated. You can reactivate it by means of control signal "[SYN_T] Synchronize NC table" or by selecting another NC table.

If the table beginning is crossed in the negative direction (x = zero), status bit "Bit 6 = internal NC table position < zero" is set in the synchronization status word. The NC table is also deactivated in this case provided that synchronization is not in progress or you do not select another NC table.



Traversal with window

You are traversing with window if parameter "Window Table Actual-Value Control" is set to more than zero. In this case, the function is active.

The window specifies the area which can be traversed by the table in the negative direction (less than zero).

As long as the axis does not exit the window, the x position migrates into the negative range. The table output value is frozen at the y value associated with zero.

If the x position crosses the limit, status bit "Bit 6 = internal NC table position < zero" is output as before. The x position of the table is reset to zero at the same time.

Absolute/Relative Output Modes

- Relative output:
  On a return jump, the slave continues to traverse at a new velocity from its current position (infinitely rotating axis with movement profile).
- Absolute output:
  On a return jump, the slave jumps back to the absolute position. If there is a large difference between the last interpolation value and the new one, the slave axis traverses abruptly to the new position.

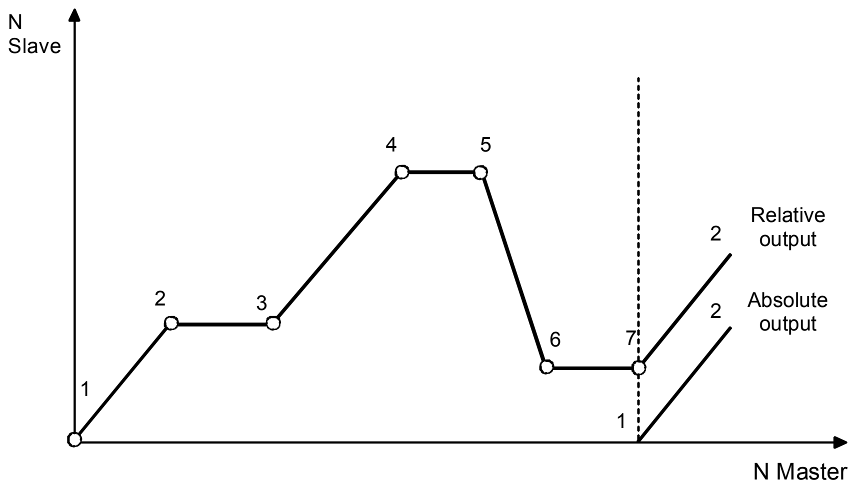


Fig. 10-22 Movement Profile of Slave: Absolute/Relative Output

Example: Relative Output

A moving sheet is to be cut using a rotating cutter.

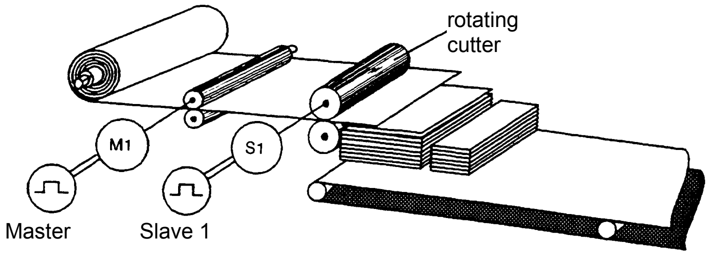


Fig. 10-23 Cutting a Moving Sheet with a Rotating Cutter

The axes operate as follows:

- Master axis
  Continuous axis (constant speed)
- Slave axis
  - NC table with 4 interpolation points
  - Relative output of slave table values
  - Continuous output of table values

The peripheral speed of the cutter must be equal to the velocity of the sheet during the cutting process, i.e. synchronization is required between the master axis and the slave axis.

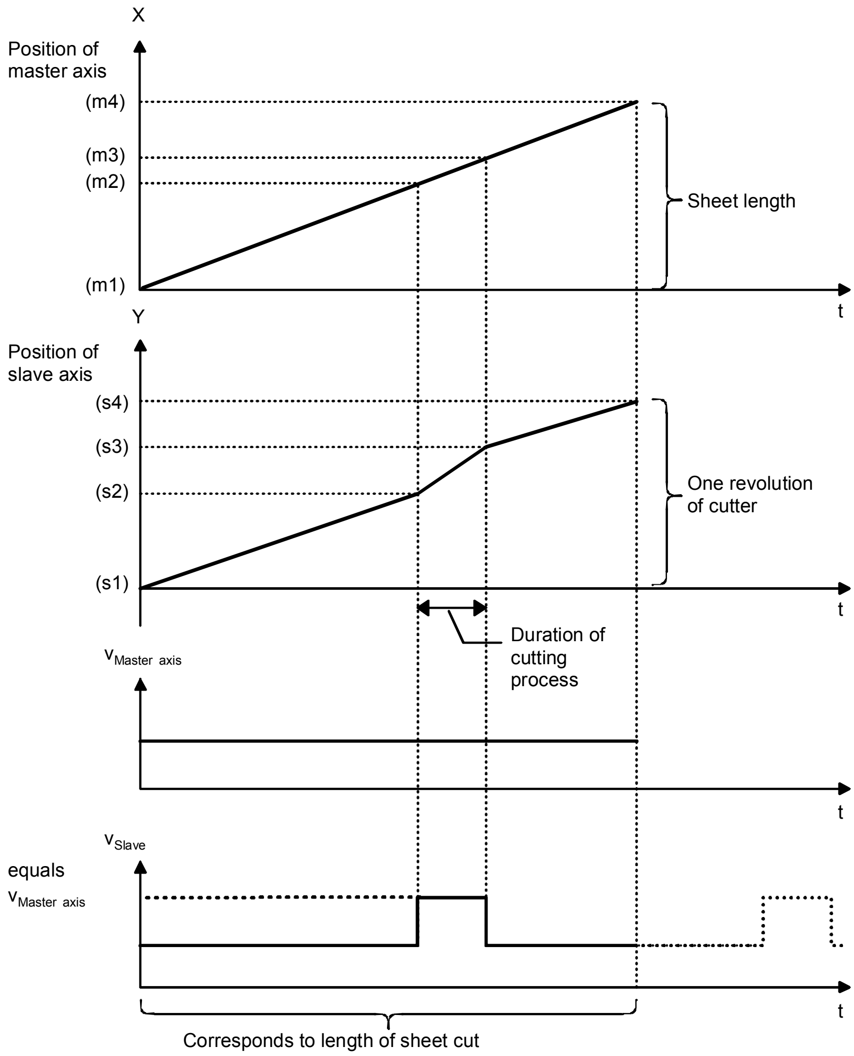


Fig. 10-24 Movement Charts: Relative Output Example

Slave axis positions m_1 to m_4 are assigned to master axis positions s_1 to s_4. This yields 4 pairs of values, $m_{1;}s_1$ to $m_{4;}s_4$, termed interpolation points.

Unlike relative output, the selection of absolute output can cause position jumps at the end of the table to be passed freely to the position controller without being suppressed.

The use of "absolute output" is only suitable in certain circumstances where the table data reveal that jumps cannot occur, e.g. where $y_{end} = y_{start}$.

Example: Absolute Output

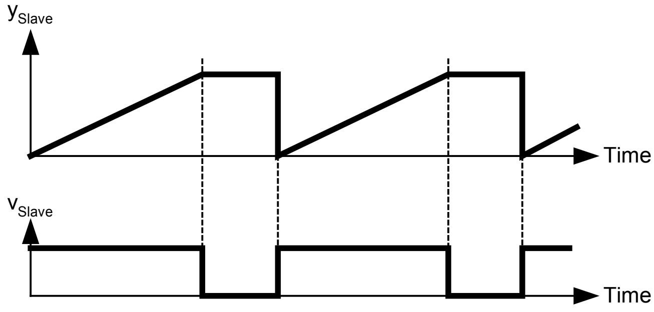


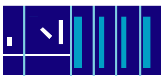
Fig. 10-25 Time Chart for Position y and Velocity v of Slave Axis

Table Change Absolute/Relative

Table change "absolute" means:
Selecting a new, active table causes a setpoint step change on the slave axis if the slave position corresponding to the current master position in the new table is not the same as the slave position in the previously active table.

Table change "relative" means:
Selecting a new, active table does not cause a setpoint step change on the slave axis. The difference between the old and new y coordinates is not included in the axis setpoint. The step change within the table is evident only from the display "Output table position slave axis" within task "Synchronization actual values and diagnostics output".

NOTE



This mode of table change is available on MASTERDRIVES MC with firmware version V1.5x and later and SIMATIC Motion Control ≥ V1.00.48.

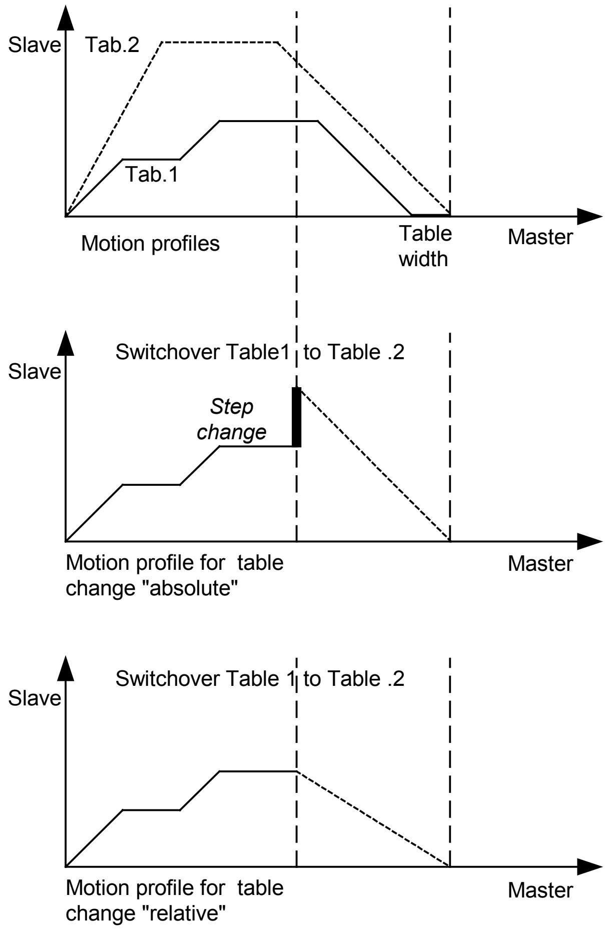


Fig. 10-26 Motion Profile of Slave with Absolute/Relative Table Change

10.5.4 Table Editor

Overview

The table editor supports the input and modification of NC tables via the task interface from an S7 user program or operator panel (OP) (e.g. using the GMC-OP-OAM standard software).

The table editor includes the following functions:

- Delete NC table and output using OP
- Delete NC table interpolation point and output using OP
- Insert NC table interpolation point and output using OP
- Output NC table to OP / S7
- Input NC table to OP / S7

Delete NC Table and Output to OP

The task deletes the x and y values of the NC table. The table width and the total number of interpolation points are not deleted.

The selected table area (starting at the interpolation point number) is then prepared for output.

Delete NC Table Interpolation Point and Output to OP

The task deletes the specified interpolation number. The interpolation points from this point to the end of the valid NC table (total number of interpolation points) are shifted forward one value pair, thereby overwriting the interpolation point which is to be deleted. The selected table area (starting at the interpolation point number) is then prepared for output.

Insert NC Table Interpolation Point and Output to OP

The task inserts a pair of values after the specified interpolation point. The value pair is initialized with 0. The interpolation points after the new point are shifted back one value pair.

The selected table area (starting at the interpolation point number) is then prepared for output.

Output NC Table to OP / S7

The task outputs the specified number of interpolation points from the specified table, starting at the specified interpolation point number. The NC table width and the total number of interpolation points are output as prefixes to the table.

You can also specify for the "NC table output OP" task whether the output is to be performed forwards (from interpolation point number to interpolation point number + total number–1) or backwards (from interpolation point number–total number+1 to interpolation point number). This feature supports scrolling functions for table output on an OP.

Input NC Table to OP/S7

The task accepts the specified number of interpolation points in the specified table starting at the specified interpolation point number. The NC table width is input as a prefix to the table.

General Functions of the Table Tasks

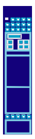
- ♦ Any table length (up to the maximum length of a table) can be handled by a table task. If the total length of data exceeds the maximum length of an individual communication task, the data are automatically blocked into subtasks. The NC table width and the total number of interpolation points is transferred with the first subtask only.
- ♦ In OP tasks, a table interpolation point always comprises the following triplet
  - Interpolation point
  - Position of master axis
  - Position of slave axis

The interpolation point is omitted in S7 tasks

10.5.5 Cam Disc Project Data

NOTE

The volume of project data for the available NC tables (cams) and interpolation points is not the same for the MASTERDRIVES Motion Control solution (technology option F01) and the SIMATIC Motion Control solution.



MASTERDRIVES MC up to Version 1.3

The MASTERDRIVES MC gives you 200 parameterizable table interpolation points. You can choose between the following configuration options using the table configuration parameter (U615):

- ♦ Table configuration = 1
  1 NC table with a maximum of 200 interpolation points
- ♦ Table configuration = 2
  2 NC tables with a maximum of 100 interpolation points each

For configuration 1, the table selection [TABLE_NO] is irrelevant. There is only one NC table, and this is always processed. For configuration 2, only bit 0, which differentiates between table 1 and table 2, of the table selection [TABLE_NO] is evaluated.

MASTERDRIVES MC from Version 1.4

The MASTERDRIVES MC gives you 400 parameterizable table interpolation points. You can choose between the following configuration options using the table configuration parameter (U615):

- Table configuration = 0
  1 NC table with a maximum of 400 interpolation points
- Table configuration = 1
  2 NC tables with a maximum of 200 interpolation points each
- Table configuration = 2
  4 NC tables with a maximum of 100 interpolation points each
- Table configuration = 3
  8 NC tables with a maximum of 50 interpolation points each
- Table configuration = 4
  Variable table configuration with a total of 8 NC tables and 400 interpolation points

The table selection [TABLE_NO] is evaluated for all the configurations. If a table is selected which is not available in the current configuration, the selection or the start is rejected, and a technology warning is displayed.

In configuration 4 (variable configuration), the distribution of the interpolation points across the maximum of 8 NC tables is determined on the basis of the "number of interpolation points" parameter. The number of interpolation points still available is displayed as an observation parameter. If the number of interpolation points in a NC table is changed, the remaining interpolation points in the other NC tables are shifted into memory.

NOTE

If you handle the NC tables using the parameter interface (parameters number, index, value), you can use the observation parameter "table info" (n639.1..16) to find out information about the start and end parameters of each individual NC table.

If you are using the S7 task interface, access is exclusively via the NC table number and the interpolation point number.

In the SIMATIC Motion Control solution, you can use 8 tables with 1022 interpolation points each for each axis.

The cam disc data is stores in data blocks. The "adopt NC tables" command is used to allocate the data blocks to the axes and NC table numbers. The default setting is as follows:

- Axis 1 Table 1..8 → DB 1011 ..1018
- Axis 2 Table 1..8 → DB 1021 ..1028
- ...
- Axis 16 Table 1..8 → DB 1161 ..1168

NOTE The allocations can be changed using "adopt NC tables" task. This means it is possible to use one NC table jointly for more than one axis.

10.5.6 Differences Between MASTERDRIVES MC / SIMATIC Motion Control

The operating principle of the functions implemented in MASTERDRIVES MC and SIMATIC Motion Control is identical. The differences relate mainly to the capacity of the functions and the option in MASTERDRIVES MC to redefine the sources for parameters by configuring them directly in the BICO system.

NOTE If you change the factory settings for the configuration of the parameters, the parameters affected can not be read or changed using the task interface.

10.5.7 Function Parameters

Various parameters must be set for the synchronization functions.

Further Information

The parameters are described in the "Task Description" in the chapter entitled "Synchronization Parameter Tasks".

Parameters

Name	Input Limit		Unit	Default
	Lower	Upper		
Gear ratio denominator (U604.01)	-32768	32767	-	1
Gear ratio numerator (U604.02)	-32768	32767 \{0}	-	1
NC table configuration (SIMATIC Motion Control only) (U615)	0	4	-	2
NC table mode (U614 and U616)	Bit 0=0: Slave axis interpolation points absolute Bit 0=1: Slave axis interpolation points relative Bit 4=0: Continuous output (automatic jump back to start of NC table) Bit 4=1: Stop at NC table end Bit 8=0: Master axis scaling off (X axis) Bit 8=1: Master axis scaling on (X axis) Bit 12=0: Master axis scaling off (Y axis) Bit 12=1: Master axis scaling on (Y axis) **As of MASTERDRIVES MC V1.4 (using task interface only)** Bit 16=0: Changes in Y scaling effective immediately Bit 16=1: Changes in Y scaling effective on next table start (x=0) Bit 24=0: Bit 16 not evaluated (setting through U614) Bit 24=1: Bit 16 is evaluated (U614 is irrelevant) **As of MASTERDRIVES MC V1.5 (using task interface only) and SIMATIC Motion Control ≥ V1.00.48** Bit 2=0: Table change absolute Bit 2=1: Table change relative Bit 25=0: Bit 2 is not evaluated Bit 25=1: Bit 2 is evaluated		-	0

Name	Input Limit		Unit	Default
	Lower	Upper		
NC table width (U620.01..08)	0	2^{31}-1	LU	4096
Number of interpolation points (U629.01..08)	1	50/100/200/400 (depends on U615)	-	0
Interpolation points X axis (U630.01-50/U631.01-50) (U640.01-50/U641.01-50) (U632.01-50/U633.01-50) (U642.01-50/U643.01-50)	0	2^{31}-1	-	0
Interpolation points Y axis (U635.01-50/U636.01-50) (U645.01-50/U646.01-50) (U637.01-50/U638.01-50) (U647.01-50/U648.01-50)	2^{31}	2^{31}-1	-	0
Set value NC table (U622)	0	2^{31}-1	LU	0
X axis scaling (denominator/numerator) (U623.01/U623.02)	-32768 / -32768	32767 / 32767 \{0}	-	1 / 1
Y axis scaling (denominator/numerator) (U651.01/U651.02)	-32768 / -32768	32767 / 32767 \{0}	-	1 / 1
Y axis scaling (U614)	0: Scaling effective directly 1: Scaling effective with SYN_T (if NC table mode Bit 24 = 1, U614 is irrelevant)		-	0
Window Table – Actual-Value Control (as of MASTERDRIVES MC V1.5 and SIMATIC Motion Control ≥ V1.00.48)	1 0: = switched off	32767	LU	0

Actual Values

Name	Value / Meaning	
Slave axis status	Bit 6=1: Internal table position < 0 Bit 7=1: Internal table position > NC table width	
Master axis internal table position	-2^{31} to 2^{31}-1	Current X position
Output slave axis table position	0 to 2^{31}-1	Current Y position
Number of active table	1 to 8	NC table active

10.5.8 Control Signals

Set Table [SET_T] (U619)

The positive edge of the SET_T control signal sets the X coordinate of the table to the value contained in the "set value table" parameter. The position setpoint at the output of the synchronization block is not changed.

Synchronize Table [SYN_T] (U621)

The positive edge of the SYN_T control signal sets the X coordinate of the table to the start of the table (X = 0). The SYN_T command is used to start a table in "stop at end of table" mode. In the standard circuit, the signal is received from the control interface. For time-critical responses, the signal can alternatively be connected to a direct input of the MASTERDRIVES MC.

In the SIMATIC Motion Control application, there is the option of using MD46 to parameterize a digital input of the MASTERDRIVES MC to make the signal available for time-critical reactions.

In "absolute" output mode, the resulting position setpoint change at the output of the synchronization block is passed on to the slave axis. In "relative" output mode, there is no setpoint jump at the output.

NOTE

Position setpoint jumps can occur on the slave axis as a result of "set table" or "synchronize table".

TABLE_NO (U650)

The TABLE_NO signal selects the current table for the machining operation.

10.5.9 Checkback Signals

The synchronization functions do not make any direct checkback signals available.

10.5.10 Pulse Diagram

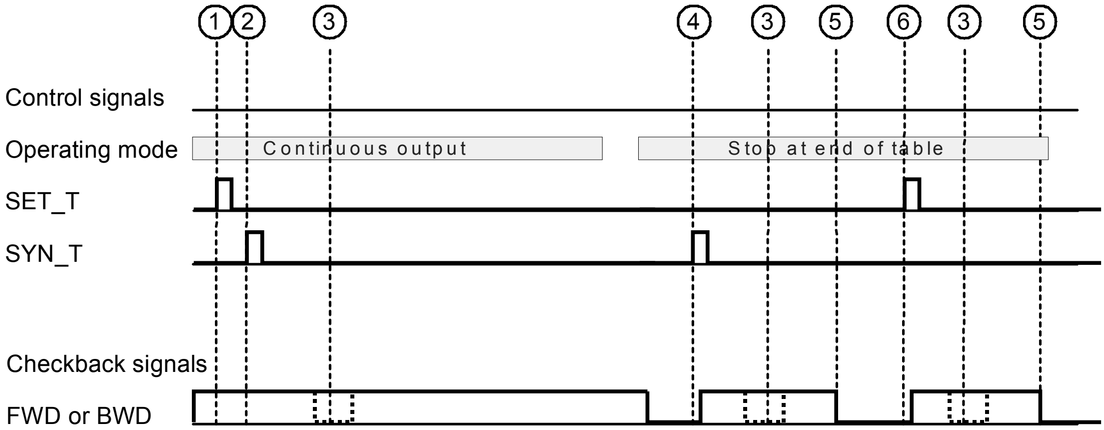


Fig. 10-27 Pulse diagram for cam disc

Sequence

The axis is operated in continuous cam disc operation mode. The master axis is moving. The slave axis is following its movements in accordance with the cam disc.

1. The cam disc is retriggered by the signal [SET_T]=1. The execution of the cam disc continues when x=table set value.
2. The cam disc is retriggered by the signal [SET_T]=1. The execution of the cam disc continues when x=0.
3. The slave axis follows the cam disc. Depending on the setpoint course, the checkback signal set or reset is either axis moving forwards ([FWD]) or axis moving backwards ([BWD]).
4. The slave axis stops. The cam disc is started by the signal [SYN_T]=1. The execution of the cam disc begins when x=0. Depending on the setpoint course, the checkback signal set or reset is either axis moving forwards ([FWD]) or axis moving backwards ([BWD]).
5. The cam disc has run through its course. The slave axis stays stopped. The checkback signal axis moving forwards ([FWD]) or axis moving backwards ([BWD]) is reset.
6. The slave axis is stationary. The cam disc is started by the signal [SET_T]=1. The execution of the cam disc begins when x=table set value. Depending on the setpoint course, the checkback signal set or reset is either axis moving forwards ([FWD]) or axis moving backwards ([BWD]).

10.6 Position Correction / Print Mark Synchronization

Overview

In synchronization applications, it is frequently the reference between a product and the master axis, and not the position between two axes, which is critical. For various reasons, shifts can occur between the product and the axis transporting the product.

Possible causes include:

- Slippage between axis and product
- Roller wear
- Material defects, Material tolerances
- Operator actions

The synchronization routine basically establishes a defined reference between the axes. The position correction function can be used to compensate for product shifts.

Operating Principle

The position correction function corrects the position of a slave axis in synchronization. The correction algorithm is designed such that a set working point can be maintained by means of corrective movements. The discrepancy between a prominent product position and a defined position setpoint is evaluated. The product position is detected by means of a suitable position sensor, such as a registration mark reader. The associated position setpoint is defined by parameter. Deviations between the position setpoint and the actual position are corrected by pulse injection. The pulse injection produces an offset between the master and slave axis.

Position correction is not designed to move to a working point in a short space of time. The application of position correction in conjunction with cam discs is only possible on the condition that the correction is effective on the position setpoint of the slave axis and not on the starting value (master value) of the cam disc.

Registration Mark Detection

The repeatability and accuracy of the registration mark detection determine the quality of position correction. On the MASTERDRIVES MC, registration mark detection is supported by the position measurement memory function. The measured value memory makes available a position value with reference to an interrupt-triggering digital input.

10.6.1 Function Description

Enable

The position correction function is enabled by the "activate position correction" parameter.

No position correction takes place while the function is not enabled. The measured value memory is not enabled and is not evaluated.

When the function is enabled, the measured value memory is enabled. A new measured value is detected on a positive edge at the "start position correction input".

Correction Value Calculation

The correction value is calculated as the difference between the parameterized position set point and the actual measured position.

Inner / Outer Window

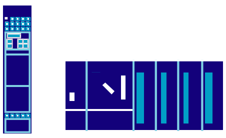
Position correction can be influenced by two adjustable tolerance windows, but only if the axis is homed. The windows relate to the set position and define the allowable discrepancy between the set position and the measured position. The windows allow position correction to be suppressed if the discrepancy is either too small or too large.

If the discrepancy is within the inner window (window1), the position is not corrected.

If the discrepancy is outside the outer window, the position is not corrected either. Binector output "Synchronization mark outside window 2" is activated at the same time. The output is displayed until the next time a synchronization mark is evaluated.

Either window can be switched off separately by parameterizing them with value = 0.

NOTE



The inner and outer windows are available on MASTERDRIVES MC with firmware version $\geq$ V1.4x and later and SIMATIC Motion Control $\geq$ V1.00.48.

Correction

The calculated correction value determines the required position correction. The correction is implemented in the form of a pulse injection at the position setpoint output. The pulse injection is performed with a configurable maximum correction velocity.

There are two types of position correction.

Correction on Slave Axis

With "correction on slave axis", the registration mark is transported by the master axis. The deviation occurs on the master axis. The slave axis has to be adjusted according to the change in movement of the material.

In this case, the correction causes a position shift on the slave axis relative to the master axis.

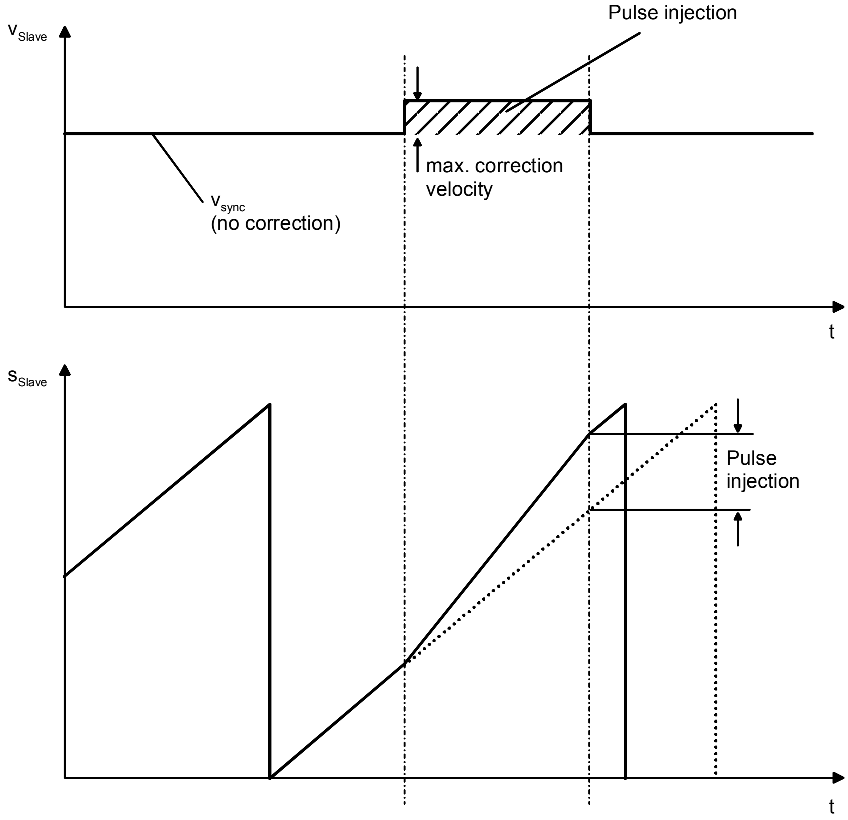


Fig. 10-28 Position Correction with Correction on Slave Axis

The diagram shows the effect of position correction with "correction on slave axis", if slip occurs in the positive direction, i.e. the set position of the print mark is less than the actual position of the print mark.

In this case, the slave axis is slowed down, and therefore follows the amended course of the product moved by the master axis.

Correction on Own Axis

With "correction on own axis", the deviation occurs on the slave axis. The slave axis has to be corrected such that the material is positioned with the right reference to the master axis after the correction.

In addition to the pulse injection associated with this correction cycle, the position setpoint and actual position value are corrected by the same amount with opposite signs.

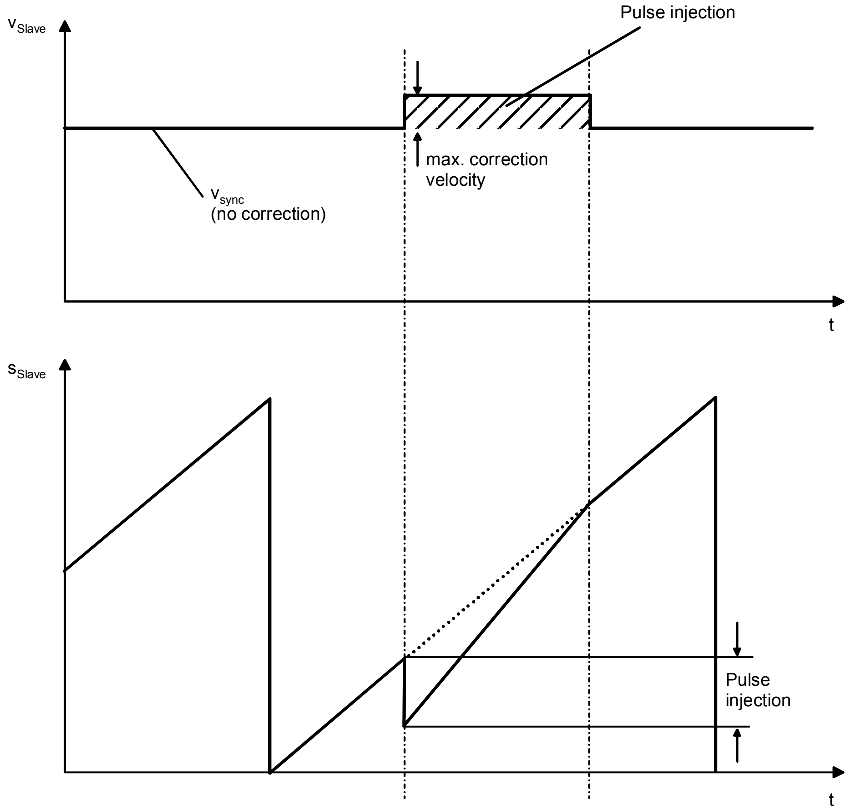


Fig. 10-29 Position Correction with Correction on Own Axis

The diagram shows the effect of position correction with "correction on own axis", if slip occurs in the positive direction, i.e. the set position of the print mark is less than the actual position of the print mark.

In this case, the slave axis is accelerated in order to bring the position of the product back into sync with the movement of the master axis. The simultaneous correction of the position setpoint and the actual position in opposite directions ensures that, once the correction is complete, the set position of the master axis and the set position of the slave axis have the same relationship to one another as they had before the correction.

NOTE

Correcting the set position and the actual position of the slave axis has the same effect as a floating actual value. Therefore, the relationship to a reference point is permanently changed.

Example

If slippage occurs, the registration mark signal arrives late (e.g. at 550), i.e. the slave axis is already further than the position setpoint.

With correction on the slave axis, this means that the slave axis must **decelerate**, so that the registration mark pulse occurs at 500 again on the next machine cycle.

With "correction on own axis", the consequence is that the slave axis has to **accelerate**, so that the registration mark pulse occurs at 500 again on the next machine cycle.

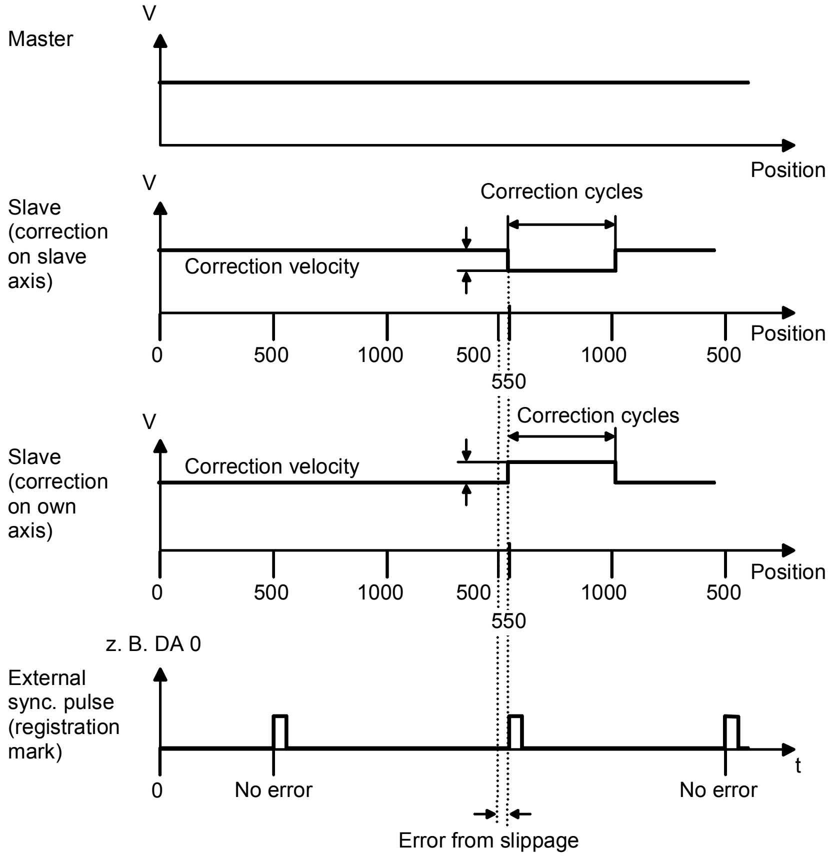


Fig. 10-30 Position Correction: Slave (Qualitative Representation)

10 increments are defined as the correction velocity.

Correction cycles (position control cycles): $Z = \frac{50\,\text{inc. deviation}}{10\,\text{inc. control cycle}}$

With "correction on slave axis", the slave velocity must be reduced for 5 cycles; with "correction on own axis" it has to be increased for 5 cycles.

Effects of Movement Direction

The effects of the position correction depend on the direction on which the actual axis is moving.

The relationships are summarized in the table below.

Correction	**v ≥ 0**		**v < 0**	
	Slippage DM too late actual > set	neg. slippage DM too early actual < set	slippage DM too late actual < set	neg. slippage DM too early actual > set
on own axis	→ Acceleration positive correction plus: negative position correction	→ Deceleration negative correction plus: positive position correction	→ Acceleration negative correction plus: positive position correction	→ Acceleration negative correction plus: positive position correction
on slave axis	→ Deceleration negative correction	→ Acceleration positive correction	→ Acceleration positive correction	→ Acceleration negative correction

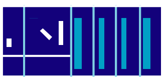
Table 10-5 Effects of position correction in relation to movement direction

10.6.2 Differences Between MASTERDRIVES MC / SIMATIC Motion Control

The operating principle of the functions implemented in MASTERDRIVES MC and SIMATIC Motion Control is identical.

A variable position setpoint is available as a connector (U663/KK) as an alternative to the fixed position setpoint. The switchover is performed using the "correction type" parameter.

In the default settings, the position correction enable is configured using parameter U660.1 and is displayed on binector B825. By reparameterizing parameter U675.1 (default: B825), the enable can be controlled dynamically.



The adjustable set position is not available.

10.6.3 Function Parameters

Various parameters must be set for the position correction.

Further Information

The parameters are described in the "Task Description" in the chapter entitled "Synchronization Parameter Tasks".

Parameters

Name	Input Limit		Unit	Default
	Lower	Upper		
Activate position correction (U660.1)	0: No position correction 1: Position correction with registration mark		-	0
Mode (U661)	0: Correction on slave axis (DM not transported on own axis) 1: Correction on slave axis (DM is transported by the axis itself)		-	0
Maximum correction velocity (not relevant if U467 > 0) (U667)	0	10 000	LU/ controller cycle	0
Maximum correction velocity (as of MASTERDRIVES MC V1.4 and SIMATIC Motion Control ≥ V1.00.48) (U467 or U467.1)	0	20 000 000.00	1000 LU/min	0
Acceleration (as of MASTERDRIVES MC V1.5 and SIMATIC Motion Control ≥ V1.00.48) (U467.2)	0	20 000 000.00	1000 LU/s²	0
Velocity adaptation in per cent for maximum correction velocity (as of MASTERDRIVES MC V1.5) (U468)	0	65535	-	1
Correction type (U664)	0: Correction to fixed position setpoint 1: Correction to variable position setpoint		-	0
Position setpoint, fixed (U662)	-2^{31}	2^{31}-1	LU	0
Inner window (as of MASTERDRIVES MC V1.3 and SIMATIC Motion Control ≥ V1.00.48) (U504.2)	-2^{31} 0= switched off	2^{31}-1	LU	0
Outer window (as of MASTERDRIVES MC V1.3 and SIMATIC Motion Control ≥ V1.00.48) (U504.3)	-2^{31} 0= switched off	2^{31}-1	LU	0

Actual Values

Name	Meaning	
Actual position last registration mark	-2^{31} to 2^{31}-1	Actual position
Slave axis status	Bit 1=1: position correction active **As of MASTERDRIVES MC V1.51 and SIMATIC Motion Control V1.00.48** Bit 0=1: Synchronization mark outside window 2	
Print mark discrepancy	-2^{31} to 2^{31}-1	Difference set to actual position
Adjustment velocity in per cent (KK0839)	-200% to 200%	Current velocity of compensation movement in per cent (as of MASTERDRIVES MC V1.5)

10.6.4 Control Signals

Enable Position Correction (U675.2)

Enable position correction.

Start Position Correction (measured value valid) (U666)

Trigger from measured value memory – measured value valid.

NOTE

The functions "position correction / print mark synchronization" and "set floating reference point" cannot be used at the same time, as they both evaluate functions of the measured value memory. If both functions are activated / enabled, the function "set floating reference point" is active.

10.6.5 Checkback Signals

[POS_A] (B800)

Position correction active. The checkback signal remains for as long as there is a correction output.

10.6.6 Pulse Diagram

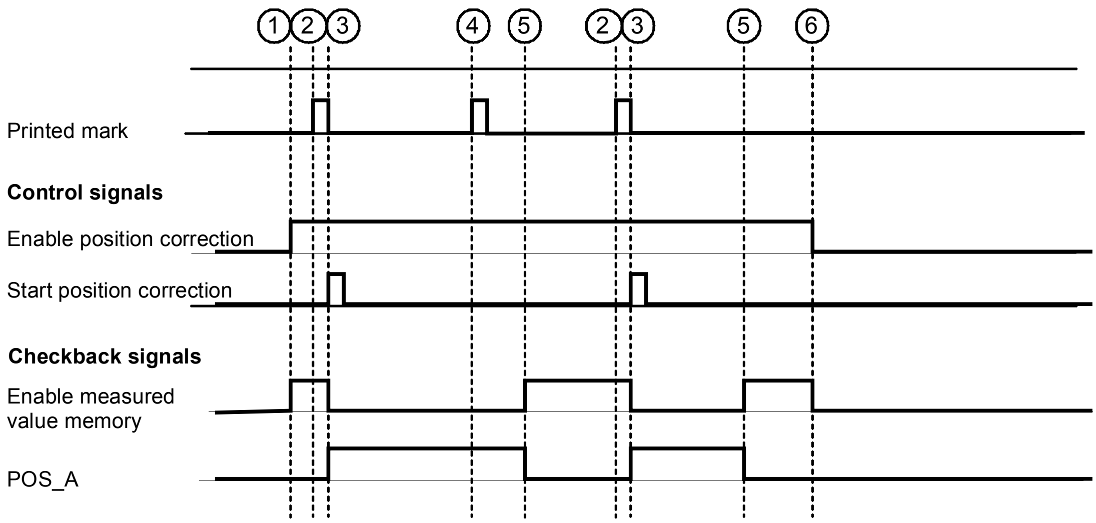


Fig. 10-31 Pulse diagram for position correction

Sequence

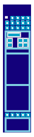
1. Position correction is activated by the enable position correction input.
2. A print mark is logged and triggers the measured value memory.
3. The measured value memory provides the measured value and starts evaluation using the start position correction signal (measured value valid). The discrepancy between the set position and the actual position results in a position correction. For the duration of the correction, the status position correction active ([POS_A] = 1) is output. The measured value memory is disabled for the duration of the correction.
4. Another print mark is logged. As the measured value memory is disabled, it is not evaluated.
5. Correction is complete. The position correction status output is switched off ([POS_A] = 0) and the measured value memory is enabled again.
6. Position correction is switched of, enable measured value memory is reset.

10.7 Set Floating Reference Point

Overview

For synchronization functions where it is not possible or not practical to run a reference search for individual axes, there is an option to set a ‘floating’ reference point for the axes in synchronization.

NOTE



The function "set floating reference point" is only available in MASTERDRIVES MC from firmware version ≥ V1.4x.

Operating Principle

In set floating reference point mode, the reference point is determined entirely using a synchronization mark (BERO). The sensor zero mark is not evaluated. As long as the function is enabled, a new reference is set with each positive edge of the synchronization mark. The synchronization mark is logged and evaluated by the measured value memory.

Setting a floating reference point has no influence on the movement of the axis. The position setpoint and the actual position of the axis are changed at the same time.

NOTE

Set floating reference point can be applied irrespective of the sensor type, and can also be used in conjunction with absolute value sensors. It effects a permanent change to the actual position.

10.7.1 Function Description

Enable [EN_RF]

The function is enabled by the control input [EN_RF]. The positive edge on the input resets the axis referenced checkback signal (B808/ARFD). The checkback signal is set again once the reference point has been set successfully. As long as the enable is active, a new reference is carried out for each synchronization mark logged. The status does not change again on the output.

Set Position

In the same ways as the position correction function, the reference point (reference point coordinate) is determined using the parameters type of correction, and fixed set position or variable reference position (see 10.6 "Position Correction / Print Mark Synchronization").

Inner / Outer Window (window1/2)

Set floating reference point can be influenced by two adjustable tolerance windows. The windows relate to the set position and define the allowable discrepancy between the set position and the measured position. The windows allow referencing to be suppressed if the discrepancy is either too small or too large.

NOTE

The windows are only evaluated if the axis is referenced ([ARFD] = 1). Therefore, a reference point will always be set the first time a synchronization mark is logged after start-up, irrespective of the window settings.

If the discrepancy is within the inner window (window1), the reference point is not set. If the discrepancy is outside the outer window, the reference point is not set either. At the same time, the output "error reference point BERO outside window 2" is activated. The output is displayed until the next time a synchronization mark is evaluated.

Either window can be switched off separately by parameterizing them with value = 0.

10.7.2 Function Parameters

Various parameters must be set for the floating referencing. The parameters are identical to the parameters for the position correction function (see 10.6.3 "Position Correction" / "Function Parameters").

Further Information

The parameters are described in the "Task Description" in the chapter entitled "Synchronization Parameter Tasks".

Parameters

Name	Input Limit		Unit	Default
	Lower	Upper		
Type of correction (U664)	0: Correction to fixed position setpoint 1: Correction to variable position setpoint		-	0
Position setpoint, fixed (U662)	-2^{31}	$2^{31}-1$	LU	0
Inner window (window 1) (U504.2)	-2^{31} 0: switched off	$2^{31}-1$	LU	0
Outer window (window 2) (U504.3)	-2^{31} 0: switched off	$2^{31}-1$	LU	0

Actual Values

Name	Meaning	
Actual position of last print mark	-2^{31} to $2^{31}-1$	Actual position
Print mark discrepancy	-2^{31} to $2^{31}-1$	Difference set to actual position

10.7.3 Control Signals

Enable [EN_RF] (U675.2)

Enable referencing

Start Referencing (measured value valid) (U666)

Trigger from measured value memory – measured value valid.

NOTE

The functions "position correction / print mark synchronization" and "set floating reference point" cannot be used at the same time, as they both evaluate functions of the measured value memory. If both functions are activated / enabled, the function "set floating reference point" is active.

10.7.4 Checkback Signals

Status Axis Referenced ARFD (B808,B809)

The output shows the status of the axis (1 = axis referenced). Binector 809 contains the inverse status.

The function also updates the [ARFD] output (B361).

10.7.5 Pulse Diagram

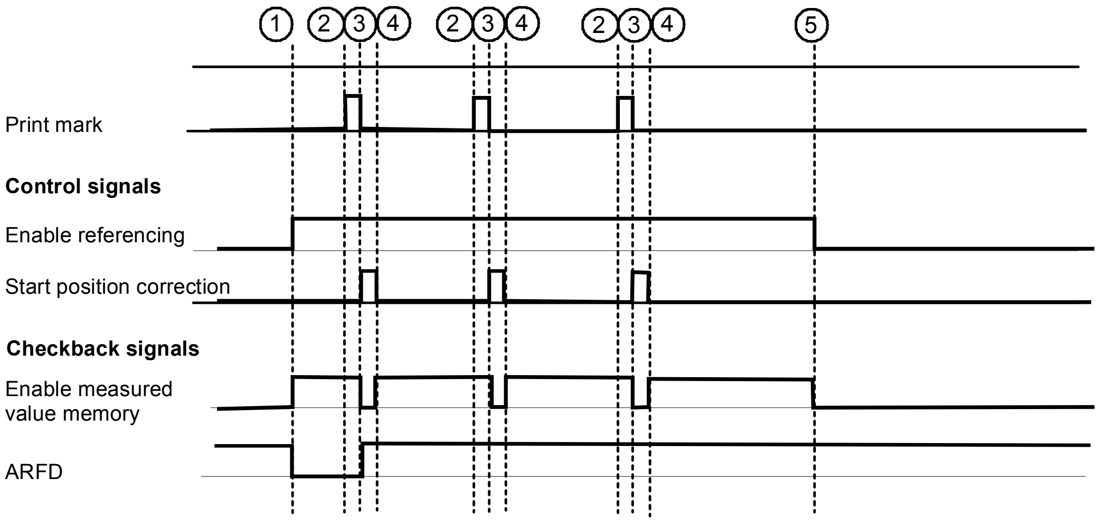


Fig. 10-32 Pulse diagram for set floating reference point

Sequence

1. The set floating reference point input is enabled. ([EN_RF] = 1). As long as referencing is enabled, position correction is not executed. The reference status output is reset. ([ARFD] = 0).
2. A print mark is logged and triggers the measured value memory.
3. The measured value memory provides the measured value and starts evaluation using the start position correction signal (measured value valid). The actual position is set as the position setpoint. The reference status output is set ([ARFD] = 1).
4. The measured value memory is enabled again immediately after the evaluation.
5. The enable referencing input is disabled ([EN_RF] = 0). Set floating reference point is switched off. The measured value memory is disabled.

10.8 Master Value Synchronization for Offset Angle Setting

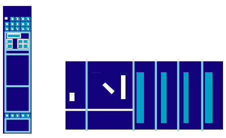
Overview

Synchronization mode does not generate an absolute relationship between the master value and the slave axis. Generally, the reference is created by running a reference search and positioning routines to move the axis to a specific stating position. Synchronization maintains the relationship set.

The master value synchronization function allows you to create the relationship between the master value and the slave axis in synchronization mode. Master value synchronization can be used anywhere where there is a defined relationship between the master value and the position setpoint of the slave axis, i.e. when, for example, the cycle length of the master axis and the slave axis are the same. For the purposes of the synchronization, an adjustable shift of the slave axis to the master value, the offset angle, is taken into consideration. Master value synchronization can be combined with all other synchronization functions (continuous start / stop cycle, 1:1, gears, cam disc).

The offset angle setting function allows you to set or change the offset angle while the system is running. Various interfaces are available which you can use to set the offset angle as an absolute value, or change it by a variable amount or change it in jog mode. The interfaces are designed so that they can be used both for manual adjustments and as an interface for external controllers.

NOTE



The functions "master value synchronization" and "offset angle adjustment" are available only in MASTERDRIVES MC from firmware version ≥ V1.3x and SIMATIC Motion Control ≥ V1.00.48.

Operating Principle

Master value synchronization takes place once after an external trigger. On the positive edge of the trigger signal, the difference between the current position setpoint of the axis and the position setpoint derived from the actual master value and the offset angle setpoint is calculated.

$$\Delta pos = \text{position setpoint}_{\text{master value}} + \text{offset} - \text{position setpoint}_{\text{actual}}$$

The difference is corrected by means of a parameterizable compensation movement from the axis. The compensation movement can be parameterized with respect to correction direction, acceleration and velocity.

Once the compensation movement is complete, the status of the synchronization is displayed by means of an output.

Status

The synchronization status is displayed by a binector. The status is reset in the following cases.

- Converter reset or switched on
- Mode changed
- Synchronization disabled
- Position setpoint output set
- Catch-up activated
- Synchronization activated
- Offset setpoint set

The binector indicates that synchronization has taken place. However, this display is only relevant if the framework conditions detailed in the chapter below are fulfilled. Otherwise, although the axis is synchronized once, the synchronization is not maintained because of the synchronization relationship set.

10.8.1 Function Description for Master Value Synchronization

Objective

The aim of master value synchronization is to create a specific relationship between the master axis and the slave axis. Synchronization can be carried out at any point. If synchronization is repeated while the system is running, there is no further compensation movement from the slave axis, as long as no settings have changed in synchronization mode which remove the synchronization status, such as changing the gear ratio or the cam disc (scaling, cam disc change) while the system is running.

Framework Conditions for Master Value Synchronization

Bearing in mind the above objective, clear master value synchronization is possible under the following framework conditions.

- no change in the gear ratio while the system is running
- no change in the cam disc while the system is running (cam disc selection, scaling on x or y axis)
- the coupling position is passed in a positive direction when the start / stop cycle is active
- the start / stop cycle is parameterized for intermittent operation

Generally, master value synchronization can take place whether the axis is stationary or in motion. For synchronization when the axis is moving, care should be taken to ensure that the compensation movement is added to the axis movement in synchronization. Under extreme circumstances, rapid changes in movement sequences can occur, especially in conjunction with cam discs. Therefore the following framework conditions should be taken into consideration:

- synchronization should only take place if the system is running at a low velocity or if the master axis is stationary.
- the parameters for the compensation payment should be selected such that the permissible axis velocities and accelerations are not exceeded, even if the master axis is in motion.

The following table gives an overview of the synchronization configurations in which master value synchronization is sensible, and which framework conditions should be fulfilled with respect to the parameter settings.

	Coupling master axis → slave axis (L= linear axis, R= rotary axis)			
Mode / function	**L → L**	**L → R**	**R → L**	**R → R**
Continuous mode 1:1	--	--	No clear image	AZL_S = AZL_M * 1/n
Continuous mode Gears	--	--	No clear image	AZL_S = AZL_M * 1/n * i
Continuous mode Table absolute	--	--	Tab width =AZL_M * 1/n	
Continuous mode Table relative	No clear image	AZL_S = ΔTab * 1/n	No clear image	Tab width = AZL_M * 1/n AZL_S = ΔTab * 1/n
Start / stop cycle 1:1	--	--	No clear image	AZL_S = EL * 1/n
Start / stop cycle Gears	--	--	No clear image	AZL_S = EL * 1/n * i
Start / stop cycle Table absolute	Tab width = EL * 1/n		AZL_M > EL + RA Tab width = EL * 1/n	
Start / stop cycle Table relative	Tab width = EL * 1/n	Tab width = EL * 1/n AZL_S = ΔTab * 1/n	No clear image	AZL_M > EL + RA Tab width = EL * 1/n AZL_S = ΔTab * 1/n

Table 10-6 Master value synchronization in the various synchronization configurations

Legend

--	No framework conditions
No clear image	The compensation movements could be differing depending on the time of synchronization or if synchronization is repeated.
AZL_M,AZL_S	Master axis cycle length, slave axis cycle length
Tab width	Table width
ΔTab	Overall table stroke ($y_{End} - y_{Start}$)
EL, RA	Start cycle length, start / stop cycle ramp
n	Natural numbers in {1;2;3....}
i	Gear ratio (numerator / denominator)

Master Value Synchronization Sequence

The diagram below shows the basic synchronization sequence for the following configuration

- ♦ Continuous mode
- ♦ 1:1 synchronization
- ♦ Master axis cycle length = slave axis cycle length
- ♦ Synchronization within one cycle

The compensation movement resulting from the synchronization can generally go beyond the cycle limits or take place over several cycles.

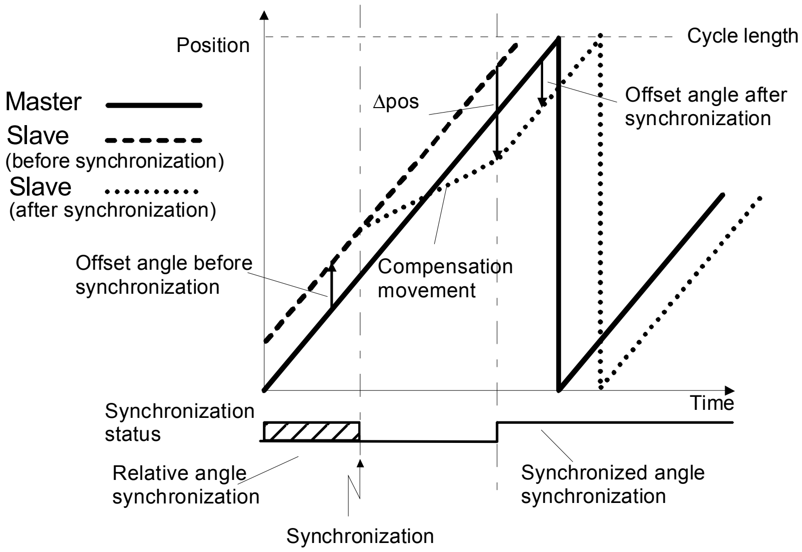


Fig. 10-33 Master value synchronization (qualitative representation)

Parameterization of Compensation Movement

A parameter can be used to define the direction of the compensation movement. The default setting is compensation by the shortest path, but you can set it so that the compensation can be only forwards or only backwards.

MASTERDRIVES MC V1.5x and SIMATIC Motion Control ≥ V1.00.48 feature a window function involving an inner and an outer window.

If the difference to be compensated is inside the outer window, the compensation movement is executed via the shortest route.

If the difference to be compensated is outside the outer window, the compensation movement is executed according to the direction of the selected window mode.

However, if the difference to be compensated is inside the inner window, no compensation movement is executed at all. The difference measured is included in the offset calculation.

Master Value Synchronization for Cam Discs

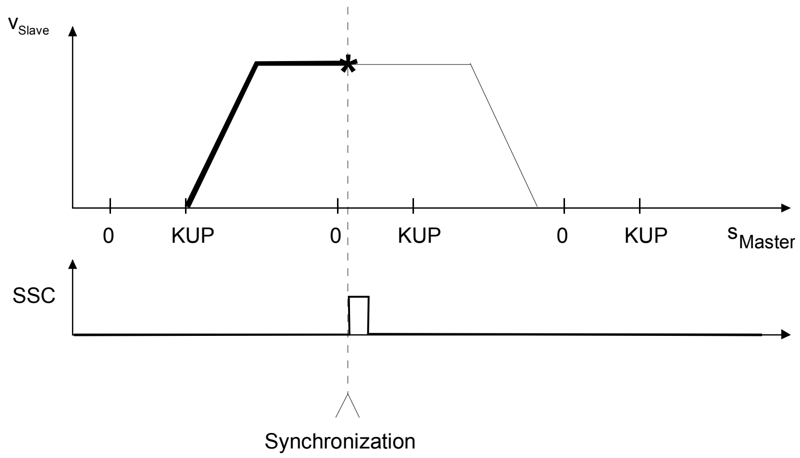
For cam discs, the position setpoint is calculated as a function of the master value, depending on the cam disc. If the master value is greater than the table width, the master value is calculated by means of modulo-division. The cam disc (x coordinate) is set to the master value calculated.

If the table is parameterized with "stop at end of table", the master value is not converted by means of the table width. In this case, the master value is adopted as the x coordinate and the last table value is used as the setpoint.

Master Value Synchronization in Start / Stop Cycle

In the start / stop cycle, the master value is calculated in the same way as for continuous mode, i.e. the start / stop cycle does not have its own transformation ratio or has the transformation ratio 1:1. The start / stop cycle is adjusted at the time of synchronization in either coupled or decoupled status depending on the [SSC] signal.

Example:
Synchronization takes place during the start cycle. At the point of synchronization, the signal [SSC] is equal to 1. After synchronization, the start cycles behaves as if it had been triggered and then the coupling position was passed over. The position setpoint is calculated in accordance with this configuration.



0 = beginning of axis cycle
KUP = coupling position

Fig. 10-34 Synchronization in start cycle

Master value synchronization with electronic gearing

In the case of gears, the position setpoint is multiplied at the input with the gear ratio numerator/denominator. The synchronization block output is synchronized with this multiplied value.

Master value synchronization with catch-up as operating mode

The synchronization output is synchronized with the "Setpoint for master value synchronization", i.e. with the setpoint after catch-up. When catch-up has ended as an operating mode, angular synchronism still exists between the "Setpoint for master value synchronization" and the synchronization output. Another "Synchronize master value" command does not therefore initiate a compensation movement.

Master value synchronization with catch-up at output

The synchronization output is synchronized with the "Setpoint for master value synchronization", i.e. with the setpoint before catch-up. When catch-up has ended, angular synchronism must be established explicitly, e.g. through connection of signal "Catch-up ended" to command "Synchronize master value" (U676= 820 [841.2]) which initiates the requisite compensation movement.

10.8.2 Function Description for Offset Angle Setting

Essentially, the offset setting consists of the offset memory functions, a positioning section for the execution of the compensation movement and various functions and / or interfaces to change the offset angle.

Offset Memory

Any changes to the offset angle are sent to the offset memory as input variables, which sets the changes in memory. The result offset angle is limited to parameterized length of the rotary axis. The actual offset angle is provided as an output variable.

In addition to the internal interfaces for adjusting the offset angle, there is also an interface which allows you to set the offset memory to a specific value. The memory is automatically set when the system starts up (power on). There is also an additional control input (set offset) which allows the memory to be set to the set value at any point.

Setting the offset memory does not result in a compensation movement of the slave axis.

Positioner

The positioner carries out all the compensation movements resulting from changes to the offset angle setpoint or the master value synchronization.

The acceleration and differential velocity for the compensation movement can be adjusted by means of parameters. The velocity can also be adjusted on a percentage basis (0-200%) using a factor. If the acceleration is equal to zero, the velocity output is in block form (≅ endless acceleration).

The positioner provides a status signal as an output variable, indicating that adjustment is active, and also outputs the remaining distance to be adjusted.

Setting the Offset Angle

There are three independent interfaces which you can use to set the offset angle. The interfaces can be used independently of one another (and even at the same time).

Absolute Offset

If the offset setpoint is changed, the offset memory is set to the new value. The change from the offset setpoint valid until this point to the new value is output as the offset change to the axis via the positioner.

The allowable value for the absolute offset setpoint is always within the rotary axis cycle. Values in excess of the rotary axis length or less than zero are corrected by means of modulo-division before execution to a value within the rotary axis cycle.

Therefore, with absolute offset, correction can only take place within a single axis cycle.

By default, the compensation movement is defined by specifying an amount and a direction for the change. This means that an adjustment from 10° to 270° leads to a positive adjustment of 260°. Alternatively, there is a parameter whereby you can select shortest path adjustment.

Relative Offset

A relative adjustment of the offset angle involves specifying the amount of the offset angle change and two control signals for the start of the change. With each positive edge of the inputs Start+ or Start-, the current offset angle changes by the appropriate amount. The changes are passed on to the positioner at the same time.

The relative offset angle can be greater than the parameterized rotary axis length.

All the changes to the offset angle are added up and output in the positioner. There is also a parameter which you can use to select the option of deleting the remaining traversing path of the positioner in advance when you are working with relative offset. The deletion of the remaining traversing path is also taken into consideration in the offset memory.

Jog Offset Angle

The offset angle can be changed continually in jog mode. The angle changes as long as you hold down either the Jog+ or the Jog- button.

The adjustment speed is parameterizable. It can also be adjusted on a percentage basis (0-200%) using a factor. There are also two parameters which can be used to set the acceleration or deceleration.

10.8.3 Function Parameters

Various parameters have to be set for the master value synchronization and offset angle adjustment.

Further Information

The parameters are documented in the task description in the chapter entitled "Synchronization parameter tasks".

Parameters

Name	Input Limit		Unit	Default
	Lower	Upper		
Synchronization mode (U699.1)	Correction direction selection 0: shortest path 1: positive direction 2: negative direction **As of MASTERDRIVES MC V1.5x and SIMATIC Motion Control ≥ V1.00.48** 3: Positive direction with window 4: Negative direction with window		-	0
Absolute offset angle correction (U699.2)	Correction direction selection 0: shortest path 1: as specified		-	0
Offset angle set value (SIMATIC Motion Control ≥ V1.00.48 only)	-2^{31}	$2^{31}-1$	LU	0
Absolute offset angle (U677.1)	-2^{31}	$2^{31}-1$	LU	0
Relative offset angle (U677.2)	-2^{31}	$2^{31}-1$	LU	0
Adjustment velocity jog offset angle (U695.1)	0	20 000 000.00	1000 LU/min	6 000 000
Percentage adjustment to adjustment speed in jog mode (SIMATIC Motion Control ≥ V1.00.48 only)	0	200	%	0
Offset angle acceleration in jog mode (U695.2)	0	20 000 000.00	1000 LU/s^2	0
Offset angle deceleration in jog mode (U695.3)	0	20 000 000.00	1000 LU/s^2	0
Adjustment velocity for compensation movement (U697.2)	0	20 000 000.00	1000 LU/min	6 000 000
Percentage adjustment of adjustment speed for compensation movement (SIMATIC Motion Control ≥ V1.00.48 only)	0	200	%	0

Name	Input Limit		Unit	Default
	Lower	Upper		
Compensation movement acceleration (U697.1)	0	20 000 000.00	1000 LU/s^2	0
Synchronization outer window (as of MASTERDRIVES MC V1.5x and SIMATIC Motion Control ≥ V1.00.48) (U462.1)	0	2^{31}-1		
Synchronization inner window (as of MASTERDRIVES MC V1.5x and SIMATIC Motion Control ≥ V1.00.48) (U462.2)	0	2^{31}-1		
Mode relative offset angle setting (as of MASTERDRIVES MC V1.5x and SIMATIC Motion Control ≥ V1.00.48) (U670)	0: Additive 1: Delete residual offset		-	0
Enable input for offset positioner (as of MASTERDRIVES MC V1.5x) (U464.1)	0: Offset positioner stopped, delete residual offset 1: Enable offset positioner		-	0
Include residual offset in 'current offset' if U464.1=0 (as of MASTERDRIVES MC V1.5x) (U464.2)	0: Do not include residual offset 1: Include residual offset in KK812		-	0

Actual Values

Name	Meaning	
Offset angle (KK812)	-2^{31} to 2^{31}-1	Current offset (setpoint)
Remaining offset path (KK827)	-2^{31} to 2^{31}-1	Distance still to be corrected (compensation movement)
Current velocity for offset (KK829 or n465)	-200 % to 200 %	The offset is applied at this current velocity in per cent related to U697.2
Slave axis status	Bit 8 = 1 Positioning active Bit 9 = 1 Synchronous to master value **As of MASTERDRIVES MC V1.5x and SIMATIC Motion Control V1.00.48** Bit 16 = 1 Synchronization in window 1 Bit 17 = 1 Synchronization in window 2	

10.8.4 Special features of MASTERDRIVES MC

In MASTERDRIVES MC, various parameters are available which allow free connection of the values. If these configuration options are used, the corresponding parameters in the task interface are irrelevant.

Free connection is possible for the following parameters.

- Absolute offset angle (U678.01)
- Relative offset angle (U678.3)

The following parameters are available exclusively as connection parameters in MASTERDRIVES MC.

- Offset set value (U678.02)
- Percentage adjustment of adjustment velocity in jog mode (U698.02)
- Percentage adjustment of adjustment velocity for compensation movement (U698.01)

10.8.5 Control Signals

Synchronize master value [SYNC] (U676)
Start synchronization. A positive edge on the control signal starts synchronization

Start + [DI_RP] (U694.1)
Start relative offset angle adjustment in positive direction

Start – [DI_RN] (U694.2)
Start relative offset angle adjustment in negative direction

Jog + [DI_JP] (U696.1)
Start jog mode relative offset angle adjustment in positive direction

Jog – [DI_JN] (U696.2)
Start jog mode relative offset angle adjustment in negative direction

Set offset [S_DSP] (U672)
Set offset memory

10.8.6 Checkback Signals

Axis Synchronized Status (B811)
The output shows the status of synchronization.

Adjustment active [DI_A] (B810)
The output shows that a compensation movement is being carried out by the positioner.

10.8.7 Pulse Diagram

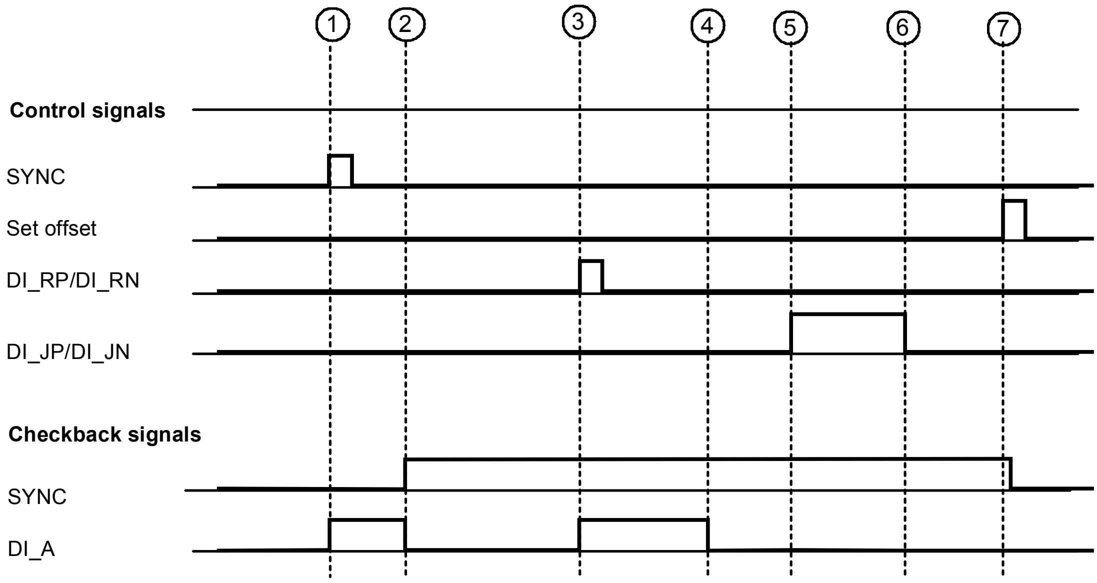


Fig. 10-35 Pulse diagram for master value synchronization / offset angle setting

Sequence

1. The synchronization input is activated. ([SYNC] = 1). The difference between the current position setpoint of the axis and the position setpoint based on the actual master value and the offset angle is calculated. The correction movement required from the slave axis is implemented. The adjustment active output is set. ([DI_A] = 1).
2. Once the compensation movement is complete, the axis status is set to synchronized. ([SYNC] = 1). The adjustment active output is reset. ([DI_A] = 0).
3. The actual offset angle is changed by a positive edge on either the relative offset angle + input ([DI_RP] = 1) or the relative offset – input ([DI_RN] = 1). The compensation movement to correct the position of the axis is implemented. The adjustment active output is set. ([DI_A] = 1).
4. The compensation movement is complete. The adjustment active output is reset. ([DI_A] = 0).
5. The actual offset angle is changed by activating either the jog offset angle + input ([DI_JP] = 1) or the jog offset angle – input ([DI_JN] = 1). The offset angle is changed on an ongoing basis for as long as the input is activated.
6. The jog offset angle + input ([DI_JP] = 0) or the jog offset angle – input ([DI_JN] = 0) is deactivated. The adjustment of the offset angle is complete.
7. The actual offset angle is set to the set value by activating the set offset input. The position of the axis is not corrected. The axis synchronized status is reset. ([SYNC] = 0).

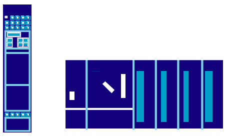
10.9 Catch-up

Overview

In addition to the start / stop cycle, the catch-up function provides another alternative for decoupling or recoupling a drive from or to an assembly of motors running in perfect synchronization (e.g. a shaftless printing press).

When it is stopped, the axis is decelerated to a specific velocity setpoint ("isolated setpoint"). It can also be stopped at a specific angular position. Starting from a stopped position, or from the current insular operation speed, the axis can catch-up to the running machine: once the catch-up command is given, the axis accelerates to the master velocity. Once the synchronization velocity is reached, the catch-up function is complete. Then, you can synchronize the axis using the "Master value synchronization" function.

NOTE



The "Catch-up" function is available on MASTERDRIVES MC with firmware version ≥ V1.31 and later and SIMATIC Motion Control ≥ V1.00.48.

Operating Principle

The catch-up behaves in the same way as an electronic coupling with an adjustable acceleration / deceleration ramp, roundings and a stoppage velocity. It also incorporates a positioner core to ensure the axis stops in precisely the right position.

The catch-up has three steady-state statuses:

- Catch-up inactive (closed up)
- Stopping
- Positioning

A control word is sued to select the statuses.

- In the inactive status (closed up), the catch-up is neutral, i.e. it has no influence on the movement of the axis.
- To stop it, the axis is braked to an isolated velocity. The isolated velocity can be selected to be either in the same or the opposite direction as the synchronization. Transitions are executed at a parameterizable deceleration, acceleration and rounding.

The axis can only move towards the target position in the specified direction of movement. Changes in direction are disabled. The insular velocity is important where the movement to the target position is concerned. The transition takes place at the set acceleration and deceleration. The rounding is not active for positioning.

Applications The catch-up function can be used for the following typical applications.

Application 1: Stop to v = 0

The axis is taken out of synchronization and braked to V = 0 or accelerated from a resting position to the synchronization velocity.

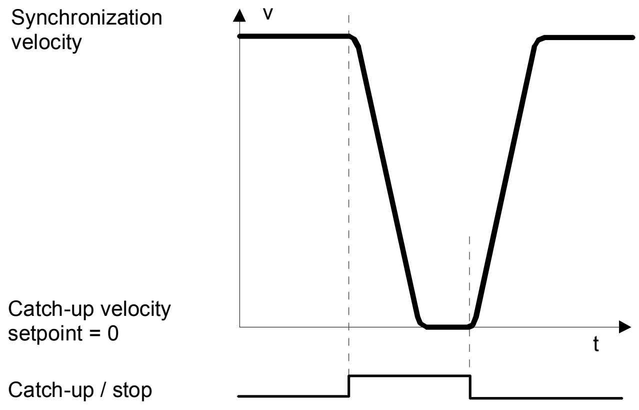


Fig. 10-36 Application 1: stop to v = 0

Application 2: Move at Isolated Velocity

The axis is taken out of synchronization and braked to the catch-up velocity setpoint (insular velocity) or accelerated from the insular velocity to the synchronization velocity.

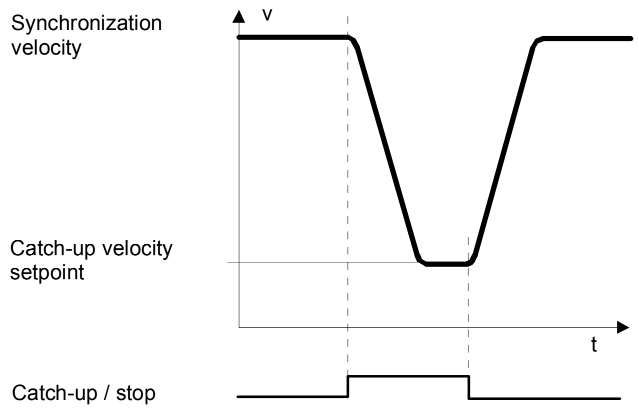


Fig. 10-37 Application 2: stop to v <> 0

Application 3: Stop to v=0 with Positioning

The axis is taken out of synchronization and braked to the catch-up velocity setpoint (insular velocity) and stopped at a specific target position or accelerated from a resting position to the synchronization velocity.

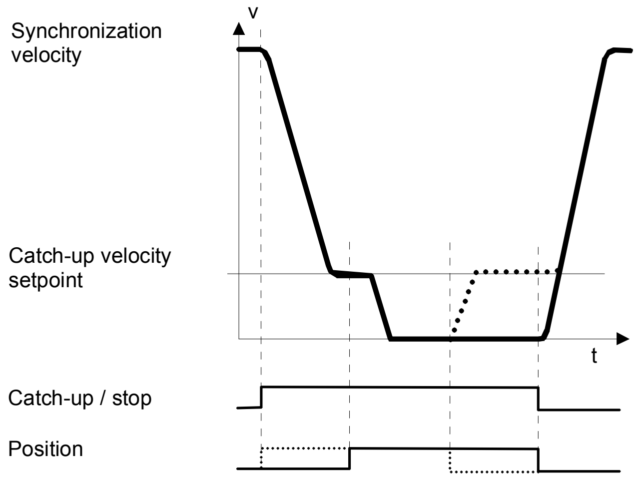


Fig. 10-38 Application 3: stop to v = 0 with positioning

Application 4: Stop to v=0 with Positioning in Opposite Direction

The axis is taken out of synchronization and braked to the catch-up velocity setpoint (insular velocity) and stopped at a specific target position or accelerated from a resting position to the synchronization velocity. The target position is approached in the opposite direction for the positioning.

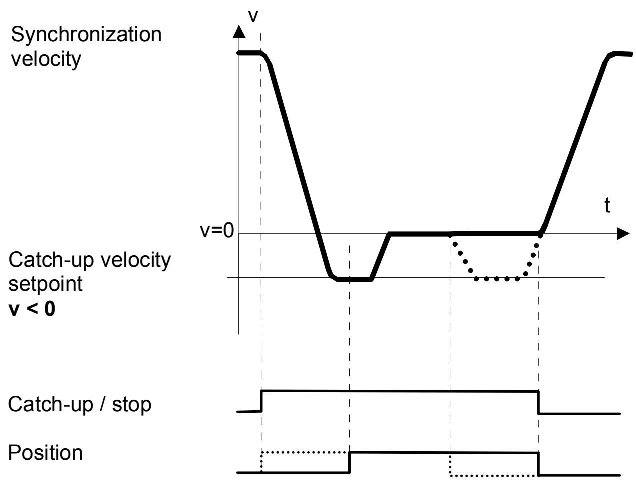


Fig. 10-39 Application 4: stop to v = 0 with positioning in opposite direction

10.9.1 Catch-up Function Description

Catch-up

To activate the catch-up you use the control command "catch-up / stop". A 1 ==> 0 signal from the control command couples a axis which has been stopped or is running at the "catch-up velocity setpoint" back into a synchronized assembly. The axis accelerates to the machine velocity specified by the master using an acceleration ramp. The movement of the axis is taken over by the master value without impact.

Once the axes are running at a synchronized speed, the output "Catch-up terminated" goes to "1". The output can be used, for example, in order to ensure perfect synchronization with the master by means of master value synchronization.

Stop

You can use a 0 ==> 1 signal from the "catch-up / stop" control command to decouple an axis from a synchronous group. The axis reduces its speed using the deceleration ramp and rounding to the "catch-up velocity setpoint". You can choose whether to set the velocity set point as an absolute value or a percentage value relative to the maximum velocity (MD23). The velocity setpoint can be = 0 (stop) or positive or negative (isolated velocity). It is possible to change direction while the axis is decelerating to the isolated velocity. The "Catch-up velocity setpoint reached" output displays that the "Catch-up velocity setpoint" has been reached.

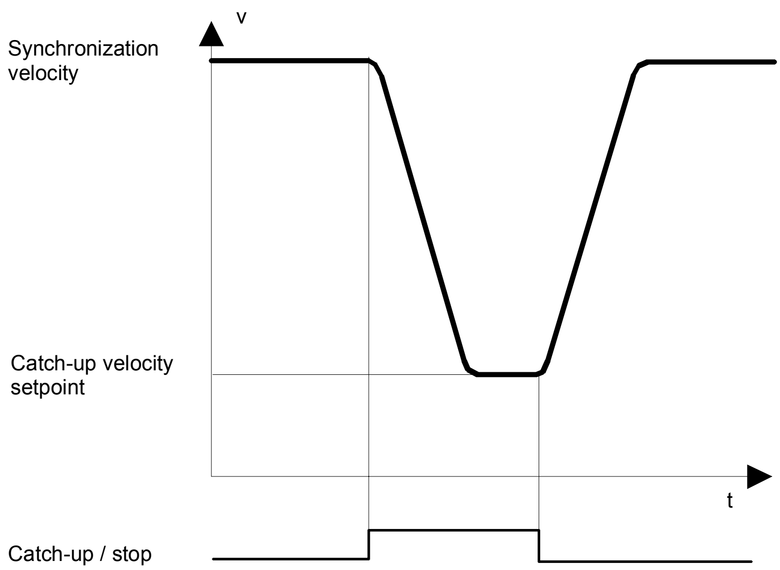


Fig. 10-40 Stop / catch-up sequence

Positioning

A 0 ==> 1 edge on the "enable positioning" control command enables you to stop the axis in its resting position. First, the axis continues to move at the "catch-up velocity setpoint" until the resting position can be reached using the ramp set without changing direction. The positioning enable should not be activated until the catch-up velocity setpoint has been reached.

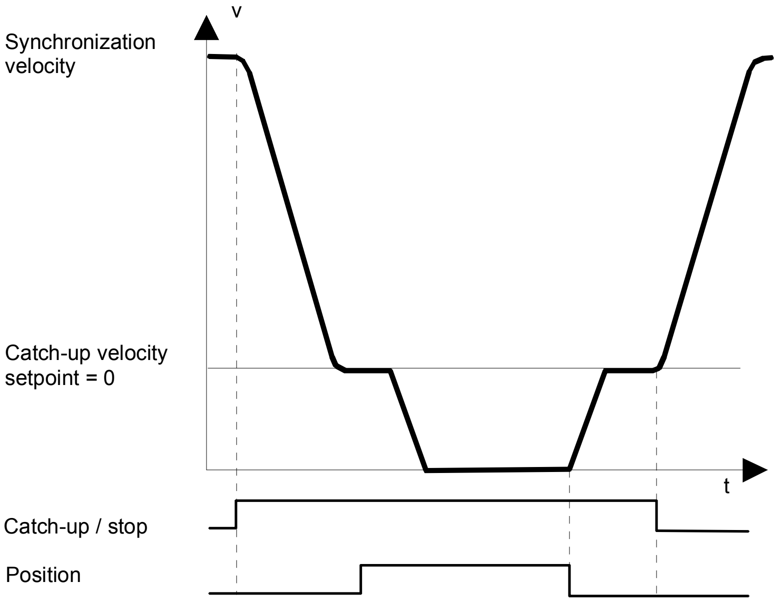


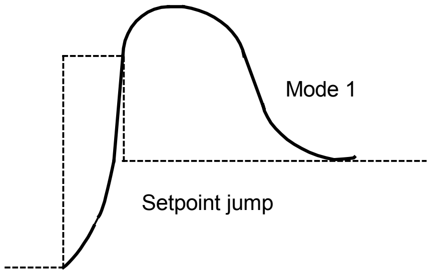
Fig. 10-41 Positioning sequence

Changing the Rest Position

If you change the rest position the axis is stationary, you can move the axis to the new position. To do this, you use the control signal [CU_TR]. The new rest position is adopted on a positive edge of this signal. The axis accelerates to the isolated velocity at maximum and moves to the new target position.

Velocity Transitions

- The transition from the synchronization velocity to the "catch-up velocity setpoint" is by means of the deceleration ramp and rounding set. You can use the rounding mode to determine whether the current acceleration for the synchronization speed should be adopted as the initial acceleration.



- If you change the catch-up velocity setpoint while the system is running, you can select whether the transition uses the acceleration / deceleration and rounding set, or whether it is sudden. You can make the selection using the "with / without acceleration / deceleration ramp" control bit.
- The transition from the "catch-up velocity setpoint" to the target position uses the deceleration ramp set. The rounding is not active.
- The acceleration from a resting position to the "catch-up velocity setpoint" uses the acceleration ramp set. The rounding is not active.
- The acceleration from the velocity setpoint to the synchronization velocity uses the acceleration ramp set and the rounding. The same is true if catch-up is immediately after the axis is in a resting position.

10.9.2 Function Parameters

The following parameters have to be set for the catch-up function.

Further Information

The parameters are documented in the order description in the Chapter on "Synchronization parameter tasks".

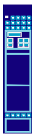
Parameters

Name	Input Limit		Unit	Default
	Lower	Upper		
Catch-up velocity setpoint (U688.02)	-2^{31}	$2^{31}-1$	10 LU/min	0
Resting position (U688.01)	0	$2^{31}-1$	LU	0
Return deceleration (U628.01)	0	20 000 000.00	1000 LU/sec^2	204.00
Rounding time constant for deceleration (U627.01)	0	60 000	msec	1000
Acceleration (U628.02)	0	20 000 000.00	1000 LU/sec^2	204.00
Rounding time constant for acceleration (U627.02)	0	60 000	msec	1000
Return deceleration for positioning (as of MASTERDRIVES MC V1.4) (U628.03)	0	20 000 000.00	1000 LU/sec^2	204.00
Acceleration for positioning (as of MASTERDRIVES MC V1.4) (U628.04)	0	20 000 000.00	1000 LU/sec^2	204.00
Rounding mode (U649)	0	1		0

Actual Values

Name	Meaning
Slave axis status	Bit 12 = 1 Catch-up ended Bit 13 = 1 Standstill position reached (catch-up axis is positioned) Bit 14 = 1 Setpoint velocity reached

10.9.3 Special Features of MASTERDRIVES MC



In MASTERDRIVES MC, various parameters are available which allow free connection of the values. If these configuration options are used, the corresponding parameters in the task interface are irrelevant.

Function	Parameter number
Catch-up velocity setpoint [10LU/min]	U626.01
Catch-up velocity setpoint [% of MD23]	U626.02
Rest position	U626.03

10.9.4 Control Signals

Catch-up / Stop [CU_SP] (U625.01)

You activate the catch-up using the [CU_SP] signal. If the signal = 0, the catch-up is inactive, i.e. after the acceleration ramp, the input signal is the same as the output signal, i.e. the axis is in synchronization. If the signal = 1, the axis is stopped.

Enable Positioning [CU_EN] (U625.02)

If the catch-up is activated ([CU_SP] = 1) and the stopping velocity has been reached ([CU_VR] = 1), you can hold the axis in the resting position by activating the control signal [CU_EN] =1.

Retrigger [CU_TR] (U625.04)

If you change the stopping position when the axis is stationary and you want to move the axis to the new stopping position, activate the control signal [CU_TR]. On the positive edge of the signal, the new position is adopted and the axis moves to it.

Enable / Disable Acceleration / Deceleration Ramp [CU_DR] (U625.03)

The with / without acceleration / deceleration ramp allows you to deactivate the ramps when the axis is moving at isolated velocity. This means you can follow a variable isolated velocity, i.e. that given by the comfort acceleration generator, without any delay. The ramps are always active in the transition phases (catch-up and positioning).

10.9.5 Checkback Signals

Catch-up Terminated [CU_TE] (B820)

The signal indicates that the catch-up routine has finished and that the synchronization speed has been reached. The axis is in synchronization.

Velocity Setpoint Reached [CU_VR] (B822)

The signal shows you that the stopping velocity has been reached.

Position Reached [CU_PR] (B821)

The signal indicates that the axis has been stopped in the stopping position.

10.9.6 Pulse Diagram

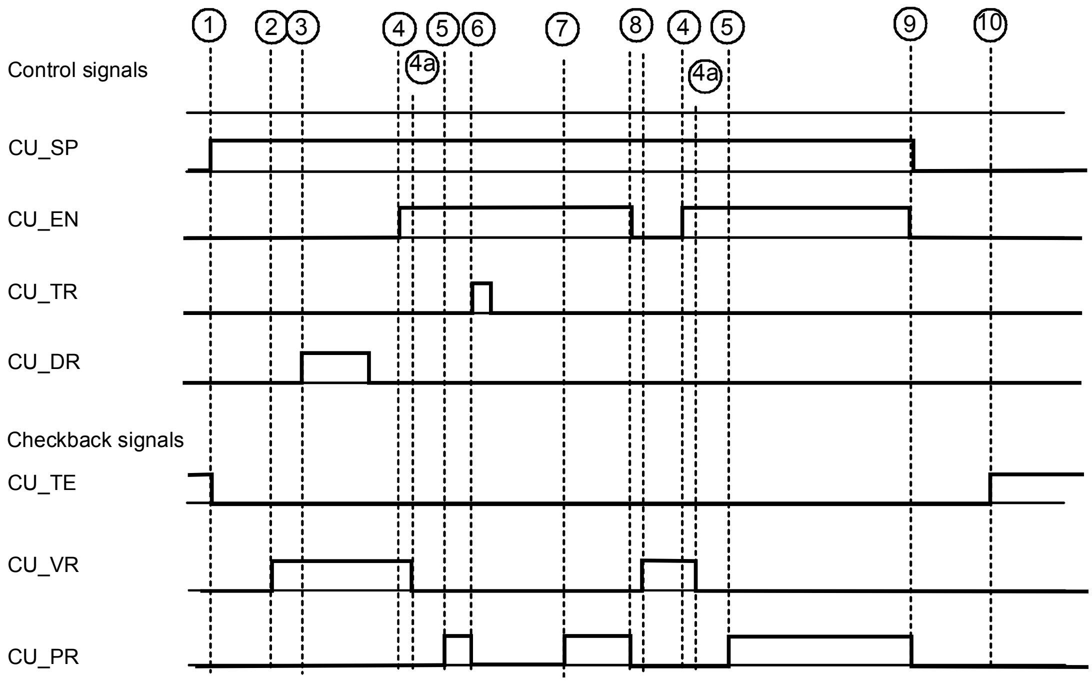


Fig. 10-42 Pulse diagram for catch-up

Sequence

Initial status:
The axis is operated in continuous mode with gears or 1:1. The master axis is moving. The movement of the slave axis is following the movement of the master axis. The catch-up is inactive, the signal "catch-up terminated" is active.

1. The catch-up is activated using the "catch-up / stop" control input. ([CU_SP] = 1). The "catch-up terminated" output is inactive. ([CU_TE] = 0). The axis is decelerated using the deceleration ramp set and the rounding time constant to the "catch-up velocity setpoint".
2. The "catch-up velocity setpoint" has been reached. The "velocity setpoint reached" is activated. ([CU_VR] = 1). If the velocity setpoint is changed during this phase, the "velocity setpoint reached" output is deactivated, [CU_VR] = 0, and the axis is accelerated / decelerated to the new velocity setpoint. Then the output is activated again. ([CU_VR] = 1).
3. In the "move at isolated velocity" status, the acceleration / deceleration ramp and the rounding are switched off. ([CU_DR] = 1). The axis follows any changes to the velocity setpoint without delays (the velocity setpoint can be generated by, for example, the comfort acceleration generator).

4. The "enable positioning" control signal is activated. ([CU_EN] = 1). The axis continues to move at the velocity setpoint until the parameterized target position can be reached using the deceleration set and without changing direction. Once the braking point has been passed, the axis brakes at the set deceleration rate (no rounding). The output "velocity setpoint reached" is deactivated. ([CU_VR] = 0).
5. The axis is held in the resting position. The "positioned" output is activated. ([CU_PR] = 1).
6. The parameterized target position is changed. A positive edge at signal [CU_TR] moves the axis to the new target position. The signal Is Positioned is deactivated ([CU_PR] = 0).
7. The axis is held in the resting position. The "positioned" output is activated. ([CU_PR] = 1).
8. The control signal "enable positioning" is deactivated, [CU_EN] = 0, the axis is accelerated at the set acceleration rate (no rounding) until the "catch-up velocity setpoint" is reached. When the velocity setpoint is reached, the "velocity setpoint reached" output is activated. ([CU_VR] = 1).
9. The control signals "enable positioning" ([CU_EN] = 0) and "catch-up / stop" ([CU_SP] = 0) are deactivated at the same time. The axis starts from a resting position and moves at the set acceleration rate, using the rounding, until the synchronization velocity is reached. The "positioned" output is deactivated. ([CU_PR] = 0).
10. The axis has reached the synchronization velocity. The catch-up becomes inactive. The axis is operated in angular synchronization. The "catch-up terminated" output is activated. ([CU_TE] = 1).

NOTE

After the catch-up routine, the axis is in angular synchronization without a definable angular relationship to the master axis. In this situation, synchronicity can be achieved using the "master value synchronization" function.

10.10 Real Master

Overview

The "real master" function supports you in adapting the master value to the synchronization master value input.

If the master value is read in via an external master value sensor which is built on to an upstream part of the machine, the block takes over the control of the position control block, e.g. to set the sensor to any rotary axis length. In addition, you can smooth the master value signal and compensate for the dead time caused by the smoothing.

Where the master value coupling, e.g. via SIMOLINK, is concerned, there is a time offset (dead time) between the master value source and the slave axis caused by the communication. This dead time results in a speed-dependent following error. The "real master" function allows you to compensate for the following error in the slave axis caused by the dead time.

NOTE

The "real master" function is only available in MASTERDRIVES MC from firmware version > V1.3x.
With firmware version V1.4 and later, the "real master" is also supported by the GMC task interface.

The block is described in the MASTERDRIVES MC compendium, in function plan 833.

Operating Principle

The initial position from the position sensor (or from SIMOLINK, PROFIBUS, etc.) is adapted to the adjustable axis cycle length. The necessary correction signals are made available for controlling the position logging.

In order to smooth out the input value, an adjustable smoothing section is available. Smoothing is especially worthwhile if the position value logged by the machine sensor position logging system incorporates errors which have a detrimental effect on the control of the slave axes.

The adjustable dead time compensation combines the master value with a velocity-dependent pilot control. The velocity signal can either be derived directly from the initial master value, or logged as an additional input signal. The input signal includes an adjustable smoothing and normalization system.

When logging an external sensor using the machine sensor position logging, it is generally preferable to log the velocity directly as the velocity signal is better in terms of quality than it would be if it were derived from the master value signal.

The output value of the real master can be set to any axis cycle length. The setting is independent of the input value. This means that it is also possible to use the real master to adapt rotary axes with varying axis cycle lengths. The output value can be connected directly to the master value input of the synchronization.

10.10.1 Function Description

Position Correction for Sensor Logging

An adjustable axis cycle is used to limit the position logging to the axis cycle for the external sensor logging system normally used. The position correction value and the +/- correction signals provide the axis cycle required for rotary axes.

The following configurations are possible:

- Position logging from the motor sensor:
  Input value provides KK120
  P174 = 614

  Correction signals to the motor sensor
  P175.02 = 581
  P175.01 = 582
- Position logging from the motor sensor
  Input value provides KK125
  P175.01 = 614

  Correction signals to the motor sensor
  P158.02 = 581
  P158.01 = 582

Smoothing the Actual Position Value

The fact that the actual position value signal is normally unstable means the signal has to be smoothed to a certain extent to suppress the jumps in the signal, which is normally constant. This is absolutely necessary if no velocity signal of the actual value is available.

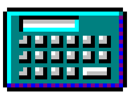
Smoothing the Actual Velocity

If the actual velocity is used, the internal velocity generation from the actual position value is skipped and the velocity signal is applied directly. This signal is generally more stable and only requires a limited degree of smoothing. This means that, to a large extent, actual position value smoothing is not required.

Dead Time Compensation

The actual velocity is multiplied by the dead time and compensates the path setpoint by adding an appropriate "path advance" to the master value. ($S^* = S + V_* t$)

Determining the dead time:
To do this, two different velocities are used. The change in their path is determined by measuring the zero impulses or using print marks.



Calculation of the dead time

$$v_{change}\,[mm/ms] = \frac{(v2\,[LU/min] - v1\,[LU/min])}{60\,000}$$

$$s_{change}\,[mm] = s2\,[LU] - s1\,[LU]$$

$$t_{dead\ time}\,[ms] = \frac{s_{change}\,[LU]}{v_{change}\,[LU/ms]}$$

Dead time compensation = $t_{dead\ time}$ = U424

Position Setpoint Generation

The compensated path is now limited by the adjustable axis cycle and output as the position setpoint. This integrator can also be set using a set value and a set command.

Set Start Position

The positive edge of the "set start position" command sets the position setpoint to the set value, and the setpoint is then recalculated on this basis.

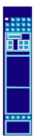
Linear / Rotary Axis

For position setpoint generation, the length of the master axis can be set using the linear / rotary axis parameter.

NOTE

When connecting the real master to the synchronization module, you must take care to ensure that the parameterizations for real master linear / rotary axis and master axis linear / rotary axis tally. Otherwise, position setpoint generation for the virtual master leads to jumps in the setpoint on the slave axis.

10.10.2 Differences Between MASTERDRIVES MC / SIMATIC Motion Control



The real master function is only available in MASTERDRIVES MC.

The function is not available in the SIMATIC Motion Control solution.

10.10.3 Function Parameters

Various parameters have to be set to enable the real master as the master value source.

Further Information

The parameters are documented in the task description in the chapter on "Synchronization parameter tasks".

Parameters

Name	Input Limit		Unit	Default
	Lower	Upper		
Actual position linear / rotary axis (U425.02)	0 0: Linear axis	2^{31}-1	LU	0
Position setpoint linear / rotary axis (U425.01)	0 0: Linear axis	2^{31}-1	LU	0
Input signal smoothing time (U423)	0	100	msec	0
Dead time compensation smoothing time (U427)	0	100	msec	0
Rated velocity (U428)	0	20 000 000.00	1000 LU/min	0
Dead time compensation (U424)	0	100.00	msec	0

Actual Values

Name	Meaning	
Position setpoint (KK608)	-2^{31} to 2^{31}-1	Dead time compensated position setpoint
Dead time compensation (KK609)	-2^{31} to 2^{31}-1	Difference position setpoint – actual position

10.11 Master Value Correction

Overview

The following chapter gives you an overview of the functions in the master value correction module. The master value correction function provides a range of functions for processing and / or influencing the master value in conjunction with the MASTERDRIVES MC synchronization functions.

It supports the following functions:

- ♦ Master value correction in conjunction with setting the floating reference point of the master axis
- ♦ Master value adjustment
- ♦ Master value switching
  No-impact switching between two master value sources.

The function is executed as a free module, which can be connected to the technology functions if required.

The master value correction and master value adjustment functions can be applied as different alternatives. Master value switching is available independently of this selection.

NOTE

The "master value correction" function is only available in MASTERDRIVES MC from firmware version ≥ V1.4.

With firmware version V1.4 and later, the "master value correction" is also supported by the GMC task interface.

The block is described in the MASTERDRIVES MC Compendium on function diagrams 845a, 845b and 845c.

Master Value Correction

This function allows a master axis to be homed while it is running and has slave axes connected in perfect synchronization.

It suppresses abrupt changes in the master value typical of those which occur when the master axis is homed on-the-fly in operation. Without this correction, the abrupt change in the master value would result in an equally abrupt compensation movement by the slave axis (axes).

As an option, the slave axes are made to follow the master axis by means of a parameterizable compensation movement when "Enable correction" U452.5 = 1. In this case, the master and slave are synchronized again on completion of the compensation movement.

Master Value Adjustment

The master value adjustment function allows you to correct the master value for a single slave axis or a group of slave axes. Unlike offset angle adjustment, the position of the slave axis is not corrected directly, rather an indirect correction is made, which also has an influence on the coupling position for the start / stop cycle or the orientation of a cam disc relative to the master axis.

Master Value Switching

The master value switching function enables you to switch dynamically between two different master values. The switchover is designed to be smooth.

Switchover involves two different modes:

Compatible mode up to version 1.4 (U458 = xx0x).
The velocity of the two master values must be identical.

Switchover is smooth in this mode, with the absolute difference between the master values corrected immediately by way of a parameterizable compensation movement.

New mode as of version 1.4 (U458 <> xx0x).
The velocities of the two master values can be different.

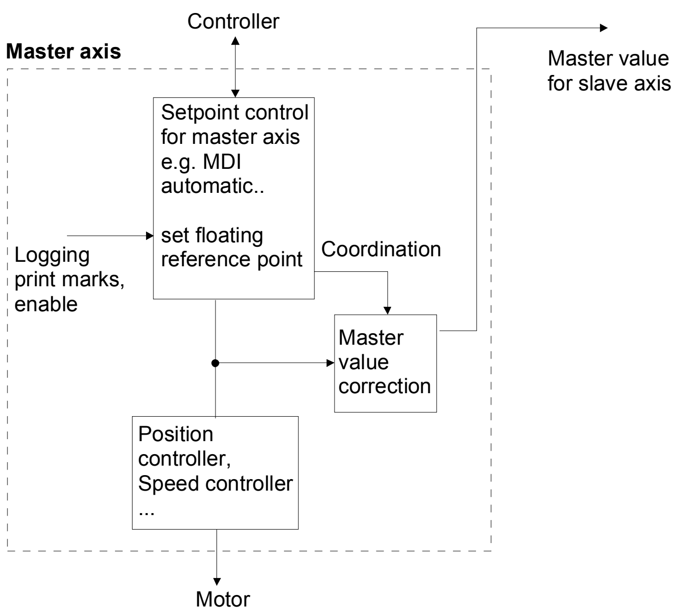
When you switch from one master value to the other, the master velocity (KK837) is immediately, but smoothly adjusted from the old to the new velocity value.

You then have the option of synchronizing the starting position with the position of the newly selected master value.
The master and slave(s) are synchronized according to the parameterized setting at exactly the moment the value at the Synchronize binector U452.6 = 1, and the velocity has been adjusted to the new value (value at KK866 = 0).

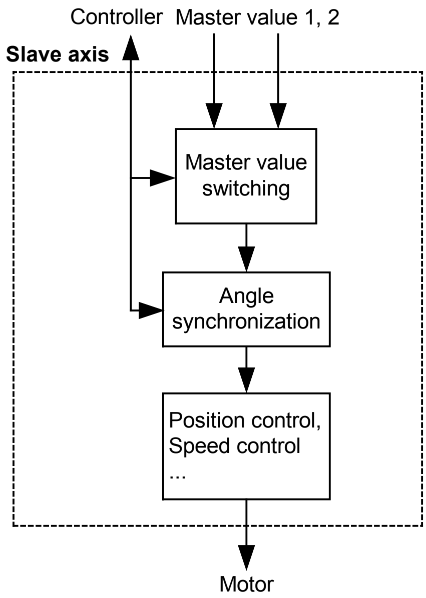
Application

The module can be connected to the master value branch from either the master side or the slave side.

When used on the master side, the module has a central effect on all the slaves depending on the master value. This mode is especially suitable for the application of master value correction in conjunction with setting the floating reference for the master axis.



When used on the slave side, the module can be connected directly before the synchronization module, in order to enable, for example, dynamic switching between two master values.



10.11.1 Function Description

Master Value Inputs / Outputs

This block provides two master value inputs (each with one position input and one velocity input) and one master value output (position and velocity output).

The velocity of master value 2 must be normalized such that it matches the normalization velocity of master value 1. The normalization velocity of master value 1 also acts as a normalization velocity at the output (for KK837). You can parameterize the inputs/outputs as linear axes (axis cycle length = 0) or as rotary axes (axis cycle length > 0) as you prefer.

Gear Adjustment of Master Value Input 2

If you wish, you can parameterize master value input 2 with a gear adjustment for adapting the resolution (increments per revolution) or the velocity.

Resolution adaptation:
Numerator and denominator are set such that the encoder resolution at the gear output matches the resolution of master value 1:

MW1 4096 Inc/rev
MW2 2048 Inc/rev
=> Gear ratio 2/1.

Velocity adaptation:
Numerator and denominator are set such that an existing encoder gear is taken into account (identical axis cycles).

Example:

Encoder gear 2:3
Numerator/denominator 3:2

Setting range
Gear ratio numerator - 32767 .. 32767
Gear ratio denominator - 32767 .. 32767

Master Value Output Set Value [U451.3]

You can set the master value output to any initial value regardless of the current input value(s). If the setting value is higher than the value parameterized for Axis Cycle Length Master Value Output, the setting value for Modulo Axis Cycle Length is corrected to within the 0 to axis cycle length range.

Function Selection (Synchronization Mode) [U458.THZE]

Using the last digit (units position) of parameter setting Function Selection, you can choose between the master value correction and the master value adjustment functions.
With version 1.5 and later, you can select the synchronizing mode for master value switchover by means of the penultimate digit (tens position).

Function selection: **E = "Units position"**	**Z = "Tens position"**
0: Master value correction 1: Master value adjustment	0: Without synchronization (compatible with V1.4) 1: Shortest path 2: Positive path 3: Negative path

Correction Offset / Trigger [U453 / U457.1]

These inputs acquire the correction variable for master value correction and adjustment, and initiate the correction. The applied correction value is evaluated as a signed position variable. The value is accepted and the correction function initiated when a positive edge is applied to the trigger input.

The technology functions provide you with two inputs for master value correction, i.e. for specifying the correction value and the trigger signal. You can activate the inputs for the master value correction function.

Selection Master Value 1/2 [U541.2]

This control input enables you to switch dynamically and smoothly between the two connected master value inputs. The difference in velocities is reduced first. Then, when synchronization commences, the positional difference between the two master values is compensated via parameterizable compensation movements.

Acceleration / Maximum Velocity [U455.01 / U455.02]

These parameters are used to set the compensation movements for all the functions of the module. The compensation movement is modulated on the current master value as a trapezoidal function.

Framework conditions:

- If acceleration = 0, endless acceleration is assumed. The velocity is switched up instantaneously.
- The parameterized velocity can be assessed using a connector input (velocity adjustment) with a percentage value.
- The correction velocity is limited dynamically so that the master value output does not reverse direction as a result of the correction. You can cancel this limitation via "Enable direction reversal" (U452.3) (see below).

Enable Direction Reversal [U452.03]

In order to prevent a correction causing the master value output to change direction, the output value is limited on a dynamic basis. This restricts the compensation movement output to the inverse of the master value, i.e. if the master value is positive, the correction is limited to a negative amount, and if the master value is negative, the correction is limited to a positive amount.

Example:
The actual master value velocity is v_{Master}. The parameterized correction velocity is double this.

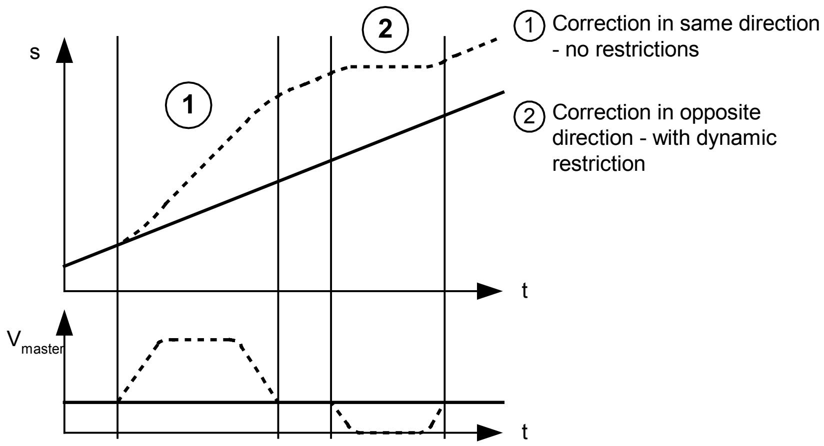


Fig. 10-43 Correction sequence with dynamic restriction

The restriction can be switched off using the "enable direction change" control input.

Example Application Master value correction

The following example shows the characteristic of the master value during on-the-fly homing of a master axis. This causes the master value input to change suddenly. The master value output is made to follow the change in accordance with the parameterized acceleration and velocity, which means that once the compensation movement is complete, the output corresponds to the input again and the slave axis is therefore synchronized with the master axis again.

The compensation movement can basically be executed over a number of axis cycles.

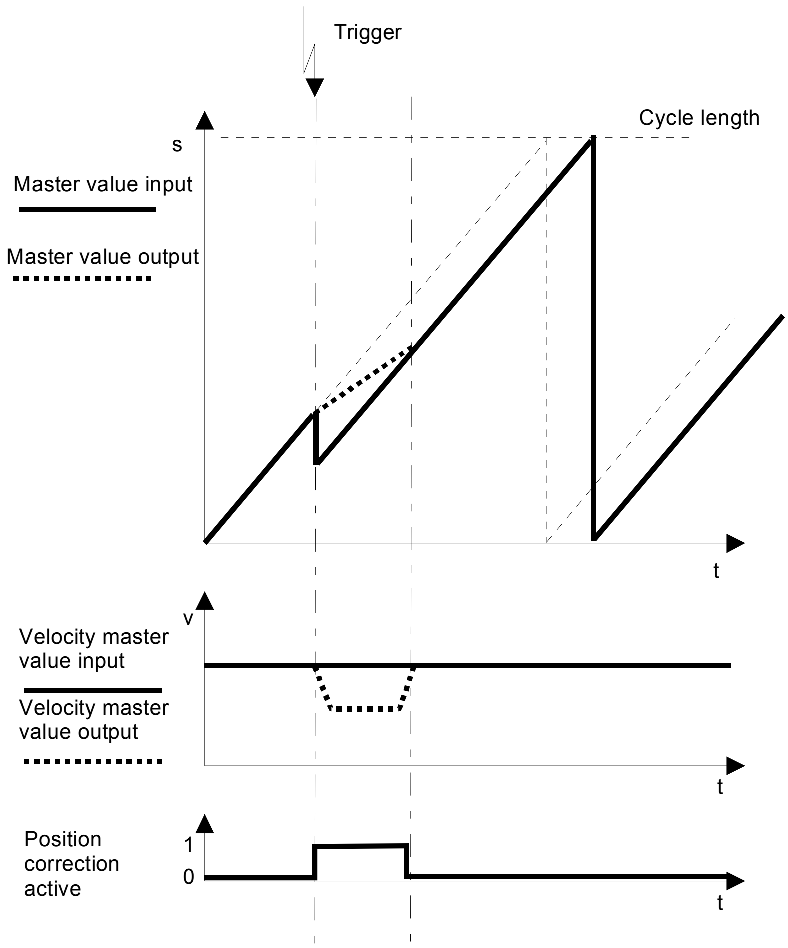


Fig. 10-44 Example of master value correction with setting of floating reference point

Example Application Master Value Adjustment

The following example shows the characteristic of the master value when master value adjustment is in progress. With a positive trigger edge (U452.1), the correction offset (U453) is applied according to the parameterized acceleration and velocity. On completion of the compensation motion, the output is offset in relation to the input by the correction value.

The compensation movement can basically be executed over a number of axis cycles.

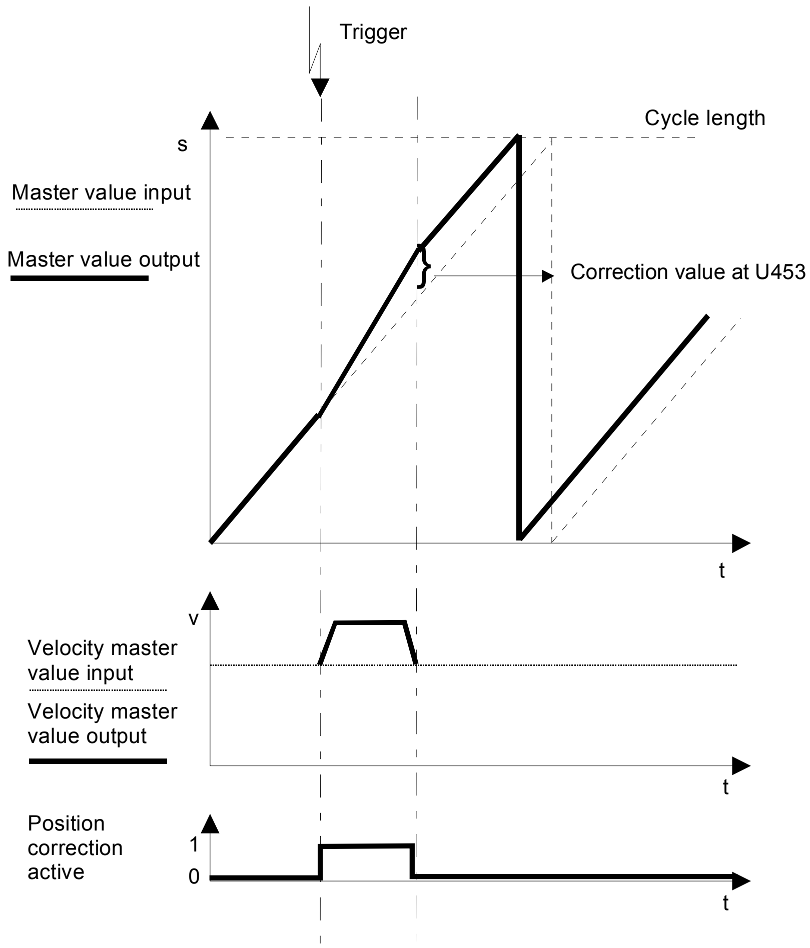


Fig. 10-45 Example of master value adjustment

Example Application Master Value Switchover with Synchronization

The example application shows switchover between two master values with different velocities. In phase 1 the velocity at the output of master value 1 is adjusted to match master value 2.
If synchronization is enabled, the output position is synchronized as a function of the synchronization mode with the input position of the newly selected master value in phase 2.

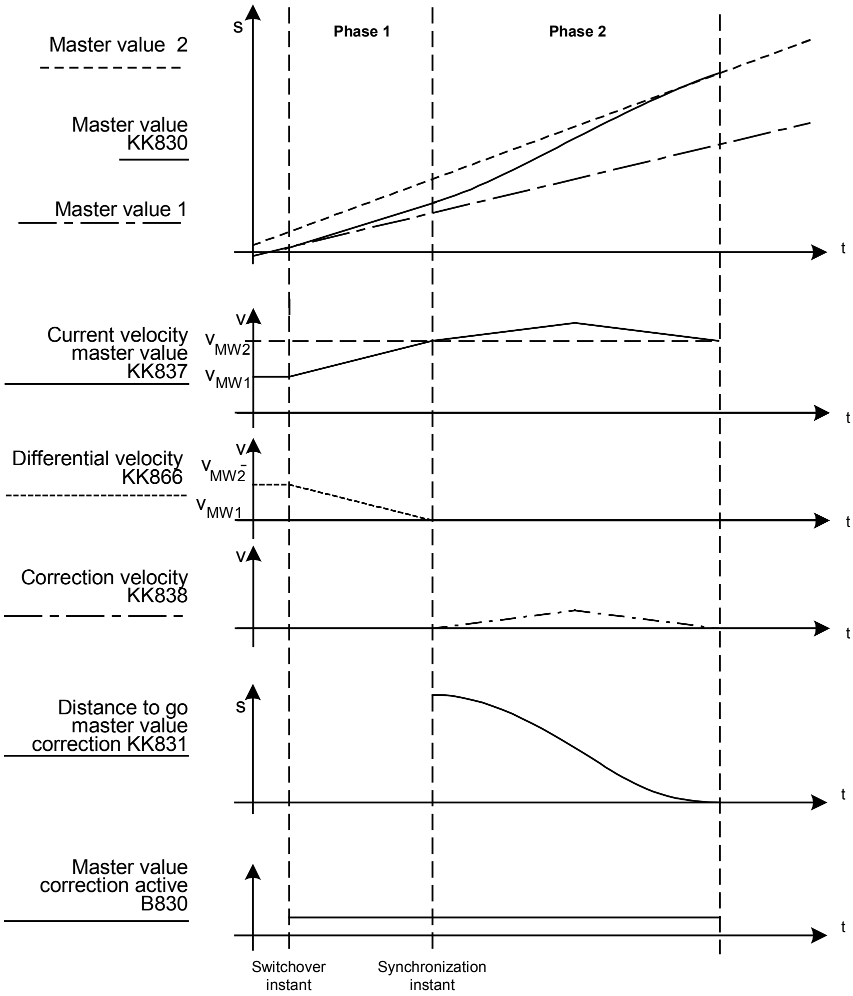


Fig. 10-46 Switchover with different velocities

Example Application Master Value Switchover (Compatible Mode)

The example application shows switchover between two master values with **identical** velocities (compatible mode).
With this method, a positional deviation is calculated from the absolute values of both masters at the instant of switchover and subsequently compensated by means of a compensation motion.
The differential values of velocity and distance can be evaluated via connectors to permit automatic switchover as a function of the "following error" between the two master axes.

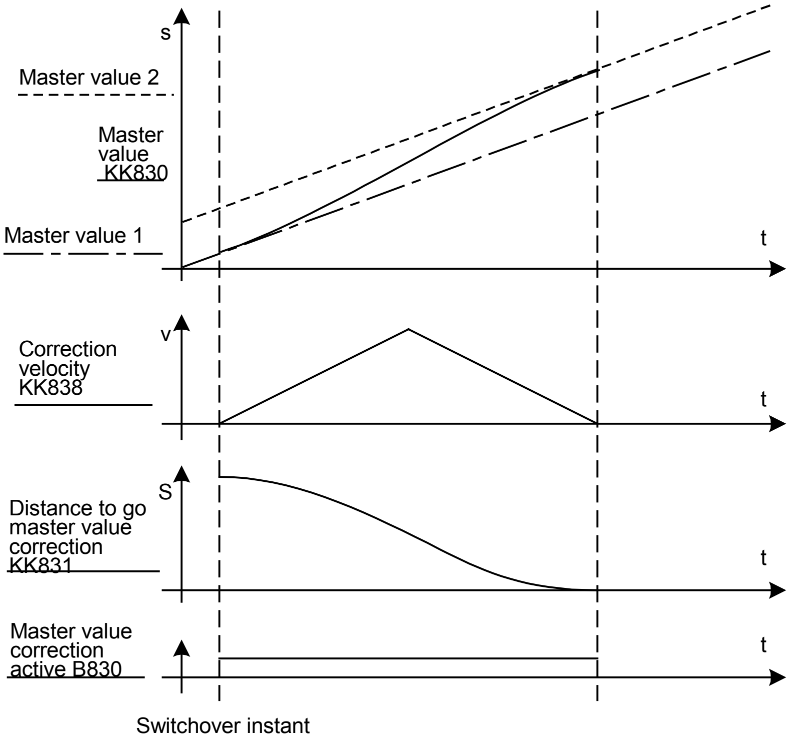


Fig. 10-47 Switchover with identical velocities

10.11.2 Behavior of Master Value Correction in Extended Mode (U458 <> xx00)

Behavior of master value correction in extended mode (U458 <> xx00), when the master value adjustment and the master value switchover functions are both used with synchronization.

Case 1

You have selected one of the two master values as a fixed setting, you do not switch over between master values. Every time a "correction offset" is accepted by trigger U452.1, the offset is applied. As a consequence, the position of the output is offset in relation to the position of the selected input by the sum of all previously accepted correction values. On synchronization at U452.6, the offset between the output and input is eliminated.

Case 2

A "correction offset" is accepted, the system switches from one master value to the other and the two axes are synchronized. The position of the output is subsequently synchronized with the position of the newly selected input, the "correction offset" is eliminated as in Case 1 above.

Case 3

The system first switches over from one master value to the other, the axes are synchronized and finally, a "correction offset" is accepted. As in Case 2 above, all correction values accepted prior to synchronization are deleted. When correction is initiated via trigger U452.1, only the "correction offset" currently applied at U453 is taken into account.

Case 4

The system switches simultaneously between both master values and a "correction offset" is accepted via trigger U452.1. In this instance, the "correction offset" is applied immediately, the axes are not synchronized until the differential velocity KK866 between the two master values has been reduced to zero [see function diagram 845c.2]. The effect of synchronization is that the output becomes positionally synchronized with the newly selected input, i.e. the "correction offset" is lost.

Case 5

The system first switches between both master values and a "correction offset" is accepted via trigger U452.1 provided that the differential velocity KK 866 between the two master velocities has not yet been reduced to zero. The function behaves in exactly the same way as in Case 4, i.e. the "correction offset" is lost.

Case 6

The system first switches between both master values and a "correction offset" is accepted via trigger U452.1 as soon as the differential velocity KK 866 between the two master velocities has been reduced to zero, but the "Distance to go master value correction" at KK831 has not been traversed to zero. In this case, the "correction offset" is added to the "Distance to go master value correction" during synchronization with the result that the output position is offset in relation to the position of the newly selected input by the new "correction offset" once the two axes are synchronized.

10.11.3 Parameters

Name	Input Limit		Unit	Default
	Lower	Upper		
Axis cycle length master value 1 (U456.1)	0	2^{31}-1	LU	4 096
Axis cycle length master value 2 (U456.2)	0	2^{31}-1	LU	4 096
Gear adjustment master value 2 Numerator output (U457.1)	-32768	32767	-	1
Gear adjustment master value 2 Denominator output (U457.2)	-32768	32767 {0}	-	1
Gear adjustment master value 2 Numerator input (U457.3)	-32768	32767	-	1
Gear adjustment master value 2 Denominator input (U457.4)	-32768	32767 \{0}	-	1
Axis cycle length Master value output (U456.3)	0	2^{31}-1	LU	4 096
Function selection (U458.0)	T | H | Z | E Function selection (units position E): 0 = Master value correction 1 = Master value adjustment Synchronizing mode (tens position Z): 0 = Without synchronization 1 = Shortest path 2 = Positive path 3 = Negative path		-	0000
Acceleration (U455.1)	0	20 000 000.00	1000 LU/sec^2	0
Correction velocity (U455.2)	0	20 000 000.00	1000 LU/min	60 000.00

Further Information

The parameters are documented in the task description in the chapter entitled "Synchronization parameter tasks".

Connection Parameters

Function	Parameter number
Q master value correction – master value 1	U451.1 and U451.4
Q master value correction – master value 2	U451.2 and U451.5
Setting value output	U451.3
Normalization velocity master value 1	U451.4
Normalization velocity master value 2	U451.5
Correction offset	U453
Velocity adjustment	U454

10.11.4 Control Signals

Select Master Value Source (U452.2]

You select the active master value source with control signal "Selection master value correction".
0 → Master value 1,
1 → Master value 2

Trigger (U452.1)

The correction offset (U453) is added to the internal correction value with the trigger signal.

The positioning technology supplies the "Trigger master value correction" B828 and the "Absolute value master value correction" KK308 for the homing function. If you wish to compensate the actual value step change using the master value correction function during homing, you can connect the "Trigger master value correction" B828 to "Trigger" U452.1, and the "Absolute value master value correction" KK308 to the "Correction offset" U453.

Enable Direction Change (U452.3)

This control input allows you to enable the master value output to change direction as a result of the correction movement. If the signal is 0, direction change is not enabled, i.e. the master value output can only be reduced to a velocity of 0 and it cannot change direction.

Set Output (U452.4)

A rising edge at the "Set output" control input transfers the setting value at interconnection parameter U451.3.

Enable Correction (U452.5)
You can activate and deactivate positioning via input "Enable correction". When positioning is deactivated, no compensation movement is executed and the "Distance to go master value correction" (KK831) is set to zero (prevents automatic starting of positioning after its activation). If you deactivate positioning while a compensation movement is being executed (B830=1), the velocity is first reduced to zero at the acceleration rate and the correction value then set to zero.

Synchronize (U452.6)
If this control signal is set, the position of the output is automatically synchronized with the position of the input after a switchover between master values depending on synchronizing mode U458. When the axes are already synchronized, you can only re-synchronize them with an edge at U452.6.

10.11.5 Checkback Signals

Master Value Correction Active (B830)
This output shows that a compensation movement is being carried out.

Master Value (KK830)
This output displays the result of function "position of master value correction".

Distance To Go Master Value Correction (KK831)
This output displays the distance still to go during the compensation movement.

V [%] (KK837)
This output displays the result of function "velocity of master value correction".

Delta Position (KK857)
This output displays the positional difference between master values 1 and 2.

Delta Velocity (KK858)
This output displays the differential velocity between master values 1 and 2.

Differential Velocity (KK866)
This output displays the differential velocity still to be eliminated between master values 1 and 2 in extended mode [U458 <> xx0x].

11 Simulation

Overview

Irrespective of the operating mode selected, and axis can be switched to simulation.

The command "Simulation Input" is used to select or deselect simulation. The selection is stored in EEPROM.

Switching simulation on

Once "Simulation on" has been set, the technology must be reset using the control signal RST (Reset) or by restarting the drive (network ON-OFF). Only then is simulation activated.

This process simulates the actual position of the position encoder, i.e. all the functions of the axis, including the set point output, can be tested without the position encoder and the drive. Even if a drive is connected, it is not driven (the drive does not move).

Switching simulation off

Once "Simulation off" has been set, the technology must be reset using the control signal RST (Reset) or by restarting the drive (network ON-OFF). Only then is simulation deactivated.

Simulation status

The simulation status can be determined using the "Simulation output" command, which gives the status as either "On" or "Off".

12 Advanced Functions with SIMATIC Motion Control

Contents

This chapter is a description of the additional functions incorporated into the SIMATIC Motion Control solution.

Overview

This chapter describes the advanced functions and the configuration options of the SIMATIC Motion Control solution.

The functions described herein are applicable exclusively to this solution. Similar or equivalent functions with MASTERDRIVES Motion Control (Technology option F01) are not affected.

The configuration chapter describes the standard configurations in which you can order the SIMATIC Motion Control solution, and the options for adapting the configuration to your specific requirements.

Further Information

The installation and configuration of the centralized solution is described in detail in Part 2 of the "Communication functions / Installation" manual.

The specific commands for the centralized solution can be found in the command description "Centralized solution orders" in Chapter 11.

12.1 Configuration

Standard Configuration

The SIMATIC Motion Control solution can be ordered in four standard configurations. These standard configurations can be used directly in your controller, without having to make adjustments on the M7 computer.

The following standard configurations are available

- SIMATIC Motion Control
  for 8 axes with analog drive interface (DP/IM178)
- SIMATIC Motion Control
  for 8 axes with digital drive interface (SIMOLINK)
  for MASTERDRIVES Motion Control
- SIMATIC Motion Control
  for 16 axes with analog drive interface (DP/IM178)
- SIMATIC Motion Control
  for 16 axes with digital drive interface (SIMOLINK)
  for MASTERDRIVES Motion Control

Installation / Configuration

The hardware configuration settings required are described in detail in Part 2 of the GMC basic / Installation manual.

The following sections of the installation manual are relevant to the different configurations.

- Section 3.2.2 Configuration of the M7-FM
- Section 3.2.3 Installation and configuration using SIMOLINK
- Section 3.2.4 Configuration using Profibus DP

12.1.1 Comparison of the Configurations

The diagram below gives an overview of the two configurations of the SIMATIC Motion Control solution.

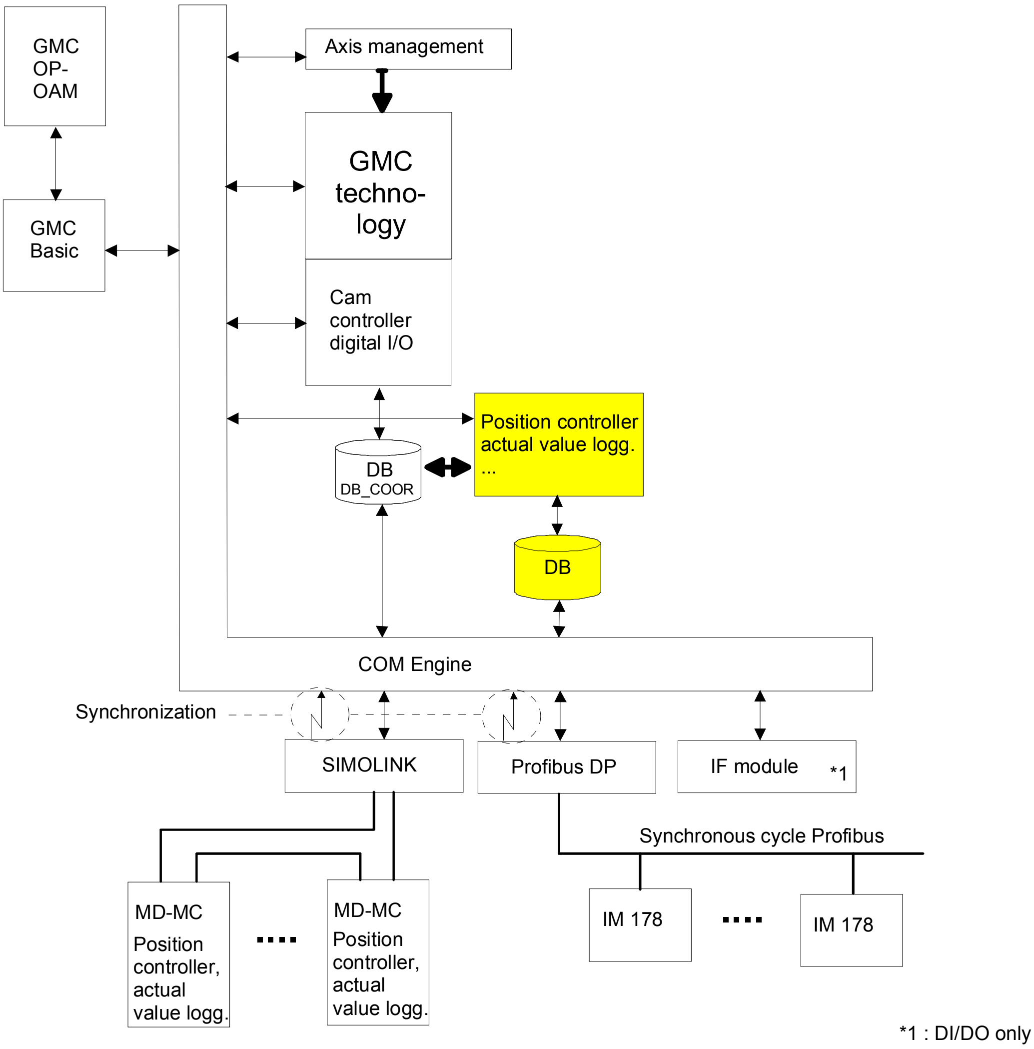


Fig. 12-1

General Features / Differences of the Configurations

The differences between the MASTERDRIVES Motion Control solution (Technology option F01) and the SIMATIC Motion Control solution are described in the introduction to the manual. The table below shows the differences between the different configurations of the SIMATIC Motion Control solution.

Function	**Configuration**	
	SIMOLINK	**DP/IM178**
Mixed configuration	With either SIMOLINK or DP/IM178	
Position logging / position control	in MASTERDRIVES MC (parameterization using basic device parameters)	in M7 (parameterization using command)
Acceleration pilot control (MD50)	Yes	No
Drum advance	Yes	No
Dynamic following error monitoring	No	Yes
Technology inputs / outputs (I1..I6,O1..O6) (without extension of inputs / outputs)	The inputs / outputs are available in the SIMOLINK interface and can be connected up in the basic device using BICO technology [1)]	Any two outputs can be connected up to the outputs Q0 and Q1.

1) This default setting is valid for inputs provided the inputs are not allocated elsewhere using the "Advanced digital inputs / outputs" function.

Table 12-1 Differences between the configurations

12.1.2 Analog Drive Interface Configuration with DP/IM178

The configuration with an analog drive interface is designed for analog converters with a ±10 Volt speed interface. Among the converters which can be connected are the SIMODRIVE 611A and 611U, and any other converters with an analog interface.

Either incremental detectors or absolute value detectors can be used for position logging. The solution consists of an M7 computer (FM456-2) with a Profibus DP master interface (IF964-DP) and one or more IM178 interface modules.

The IM178 interface module is operated as a slave on the Profibus DP. You can connect two axes to one IM178 station. The following interfaces are available for each axis:

- 1 detector input (you can choose between an incremental detector and an absolute value detector)
- 1 ± 10 Volt analog output
- 3 24 Volt digital inputs
- 3 24 Volt digital inputs

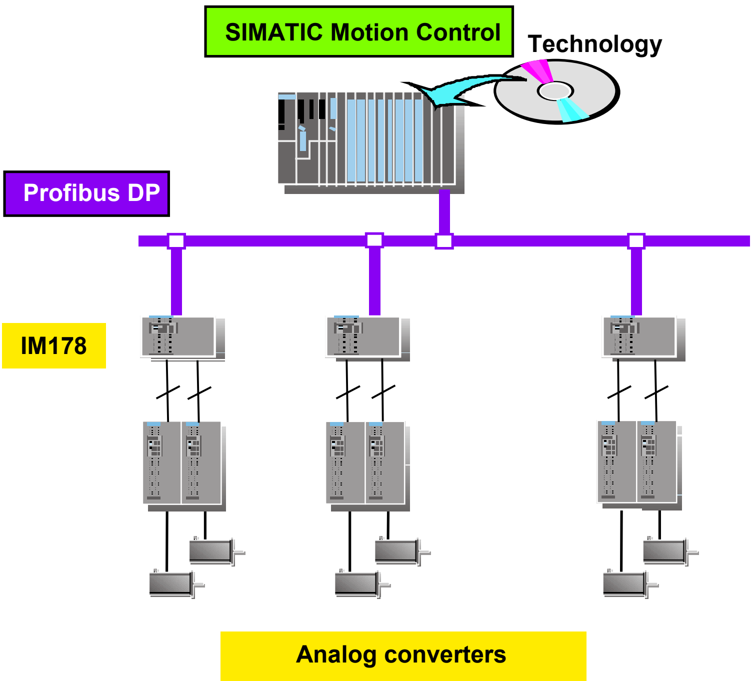


Fig. 12-2 Configuration with DP/IM178

Axis Configuration

The following configuration is the factory setting:

- ♦ 8 axes 4 IM178 (DP addresses 3 to 6)
- ♦ 16 axes 8 IM178 (DP addresses 3 to 10)

The axes are allocated on a sequential basis in ascending order.

	Address	Axes
1. IM178	3	1,2
2. IM178	4	3,4
...		
8. IM178	10	15,16

NOTE

The allocation of the axes to the modules is stored in the configuration data of the SIMATIC Motion Control application (see GMC-BASIC, Chapter 3.2.4). If necessary, you can adapt this configuration. To do this, you need a configuration tool, which is based on Excel. The tool is not supplied as standard, but can be made available on request.

IM178 Input / Output Allocation

The following table gives an overview of the allocation of the digital inputs / outputs of the IM 178.

Input	Function	Parameterization
I0	Pressure mark logging for floating measurements, floating reference points, pressure mark synchronization	fixed
I1	Rough impulse for reference point search	fixed
I2	Checkback converter is in service Checkback must be activated. Otherwise, no operating mode can be activated.	fixed

Output	Function	Parameterization
Q0	Rapid output (A1..A6) for direct control by the technology software	MD63
Q1	Rapid output (A1..A6) for direct control by the technology software	MD64
Q2	Controller enable The controller enable is set if • the axis is being processed (EN_AQ=1) and • the S7 controller enable is set (ENC=1) and • there are no technology errors	fixed

Please refer to the device manual for the front slot allocations of the IM178.

12.1.3 Digital Drive Interface Configuration with SIMOLINK

The configuration with the SIMOLINK digital drive interface is designed for MASTERDRIVES Motion Control converters.

SIMOLINK (Siemens Motion Link) is a digital, serial data transfer protocol using fiber-optic cables as the transfer medium. The SIMOLINK drive connection was developed to ensure extremely rapid, cyclical transfer of process data (control information, set points, actual values and status information) between the different MASTERDRIVES MC devices or between the MASTERDRIVES MC devices and a higher level control system at the same time as synchronizing all the connected stations to a joint system speed.

It is largely control and checkback signals for the controller and basis device and the data required for the control system which are transferred across the SIMOLINK interface.

The basic device can be parameterized either using SIMOVIS/DriveMonitor, which is then stored in the device, or there is also an option to use the Profibus interface of the SIMATIC S7.

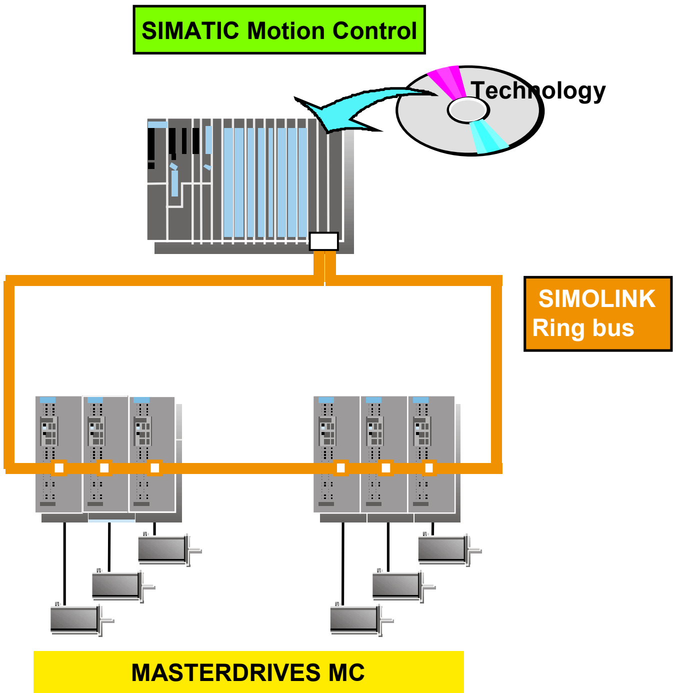


Fig. 12-3 Configuration with SIMOLINK

Axis Configuration

The following configuration is set on delivery:

- 8 axes transceivers 1 to 8
- 16 axes transceivers 1 to 16

The axes are allocated on a sequential basis in ascending order.

NOTE

The allocation of the axes to the modules (dispatchers) is stored in the configuration data of the SIMATIC Motion Control application. If necessary, you can adapt this configuration. To do this, you need a configuration tool, which is based on Excel. The tool is not supplied as standard, but can be made available on request.

12.1.4 Configuration with More Than 16 Axes

Additional axes cannot be added to the 8 axes version. The 16 axes version can be extended to up to 32 axes.

In order to do this, both the configuration data for the M7 application and the data interfaces (DBs) must be extended in order to store the command data in accordance with the desired number of axes.

The configuration tool required for this is not supplied as standard. It is available on request.

12.2 Axis Management

The SIMATIC Motion Control solution can be used to process a maximum of 32 axes on one M7 computer. The requirements of the individual axes can be different, especially when the systems are complex and have a large number of axes.

The axis management system enables the following settings / parameterization:

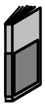
- number of axes to be processed
- processing sequence for axes
- reducing factor (scanning time)
- shift

In addition, the processing of a parameterized axis can be dynamically enabled or blocked using the control / checkback interface.

These access options enable optimum adaptation of the application to the requirements of the machine, and optimum usage of the resources of the M7 computer.

NOTE

The parameterization of the axis management system using the S7 is designed with one-off parameterization after system start-up in mind. At this point, no axes can be enabled for processing.



Further Information

A description of the command for parameterizing the axis management system using S7 can be found in the command description configuration introduction under "Commands for SIMATIC Motion Control".

The following is a detailed description of the individual parameterization settings.

Parameter File

Name		Input limits	
		Lower	Upper
Number of axis allocations calculated		1	to max. axes
1. axis	Local axis number on M7	1	to max. axes
	Axis enable	Bit 0 = virtual master Bit 1 = cam controller	
	Reducing factor	1	32
	Shift	0	31
Beginning of next definition			

Number of Axes

This parameter is used to set the number of axes which are actually processed in the concrete application.

Generally, you can only enable the axes which are available in the configuration, i.e. either a maximum of 8, 16 or up to 32 axes.

If the axis number is reduced, there is essentially the option of optimizing the scanning times.

Local Axis Number on the M7 (Processing Sequence)

This parameter sets the processing sequence for the axes. The default setting is for the axes to be processing in ascending axis number. However, for synchronization functions, it can be especially important for a master axis to be calculated before the slave axis in the same scanning level, in order to prevent unnecessary dead times in the conductivity connection. For individual axes in positioning mode, changing the processing sequence is generally irrelevant.

Reducing Factor

The reducing factor determines the scanning time for the axis. The scanning time can be set for each axis as an integral multiple of the basic scanning time. The reducing factor can be set between 1 and 32.

The basic scanning time is determined by synchronization with the bus systems or the converters.

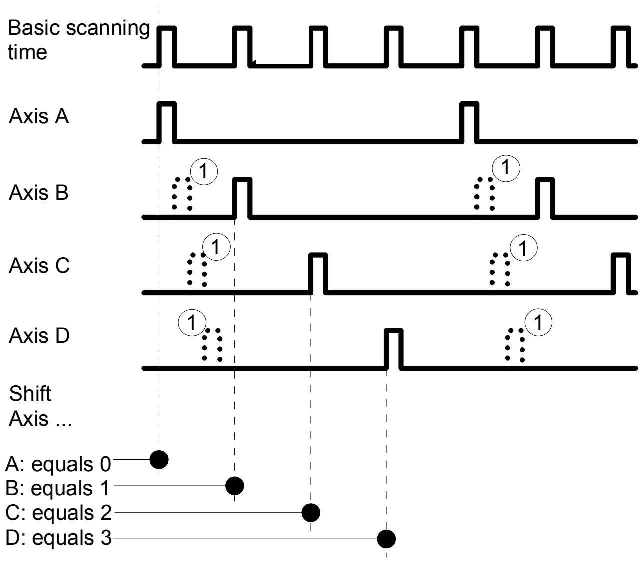
NOTE

If more than one axis is operated by an automatic program in automatic mode, all the axes processed by the program must have the same scanning time. If this is not the case, the selection or the start of the program will be refused with a warning message.

Shift

If several axes are processed in a scanning time which is longer than the basic scanning time, the point at which the axes are processed can be shifted relative to one another in order to prevent bunching of the processing at a certain time. Shift is parameterized as a whole number of basic scanning cycles for each axis.

Example with reducing factor = 4:



(1) Execution point of axes without shift

Fig. 12-4 Example of scanning shift

SIMOLINK

If there is a digital connection via SIMOLINK, the basic scanning time is set by parameterizing the IF module with the object manager (see Installation instructions). The basic scanning time must tally with one of the master levels of the MASTERDRIVES. If the basic frequency is 5KHz, scanning times of T3 = 1.6 ms or T4 = 3.2 ms are available. The scanning time entered is recognized automatically.

NOTE

The scanning time reduction between the actual scanning time for the axis and the position controller speed in the MASTERDRIVES MC must be set on the position set point interpolator of the MASTERDRIVES MC.

The default setting is 3.2 ms.

DP/IM178

If analog drives are connected via Profibus DP, the basic scanning time is set using the equidistance cycle of the Profibus master interface.

NOTICE

The equidistance cycle setting must be a multiple of 0.1 ms.

Default Setting

In the standard configuration, the axes are parameterized as follows:

Axis	Scanning time	Shift
1 ... 8	T0	0
9 ... 12	2 * T0	0
13 ... 16	2 * T0	1
17 ... 20	4* T0	0
21 ... 24	4* T0	1
25 ... 28	4* T0	2
29 ... 32	4* T0	3

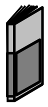
The parameterization can be read and altered through the command interface.

12.3 Control / Checkback Interface

The control / checkback interface of the SIMATIC Motion Control solution is largely in accordance with the allocation described in Chapter 2 "Control and checkback signals".

The differences relate to:

- the additional signals for axis management [EN_Ax, EN_AQ]
- the basic device interface for analog drive interfaces with DP/IM178 [ENC, IOP]
- the allocation of the "optional range"



Further Information

The allocations from DBBx+2 or DBBy+2 agree with the allocations detailed in the description in Chapter 2 "Control and checkback signals".

Dynamic Axis Enable

In addition to the axis processing determined by the parameterization, you can also enable the processing of each individual axis using a signal in the control interface [EN_Ax]. The status is displayed on the checkback interface, along with the signal [EN_AQ].

If an axis is not enabled for processing, it is not processed. The actual values and checkback signals are frozen as they were when it was last processed.

If you reset the dynamic enable when the axis is moving, the axis is only stopped once it has come to a standstill and the controller enable (Output Q2) is reset. Finally, the checkback message "Access being processed" [EN_AQ] is reset.

NOTE

If the processing of an axis is blocked, the cam controller and the virtual master in this axis are no longer processed either.

If the dynamic enable is reset while the virtual master is being processed, the axis is stopped, and the virtual master is not affected. As long as the virtual master is active, the checkback message "Axis in active processing" [EN_AQ = 1] remains.

12.3.1 SIMOLINK

The following table shows the control / checkback signals for the digital interface with SIMOLINK.

Control Signals

	7	6	5	4	3	2	1	0		Axis_n.
DBBx	RES	RES	RES	EN_AX	RES	LB	RES	RES	BIN	IN_1
DBBx+1	ACK_F	RES	RES	RES	ENC	OFF3	OFF2	OFF1	BIN	IN_2

EN_Ax Axis processing enable

Checkback Signals

	7	6	5	4	3	2	1	0		Axis_n.
DBBy	RES	RES	RES	EN_AQ	OTM	OTC	OLC	S MAX	BIN	OUT_1
DBBy+1	RES	WA RN	OFF3	OFF2	FAU LT	IOP	RDY	RTS	BIN	OUT_2

EN_AQ Checkback message - Axis being processed

Impulse Diagram

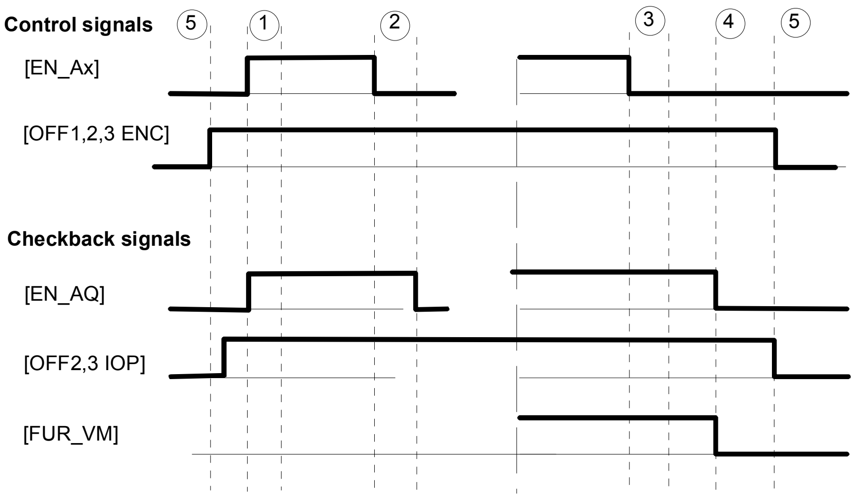


Fig. 12-5 Impulse diagram

The impulse diagram above shows the signal sequence for controlling an axis if there is a digital interface with SIMOLINK.

1. Processing of the axis is enabled [EN_Ax] = 1, the checkback signal "Axis is being processed" [EN_AQ] is set.
2. During the processing of an axis (BAx is active, the axis is moving), the axis enable [EN_Ax] is reset. The current processing of the axis is stopped, the axis is brought to a halt using a speed ramp. When the axis is at a standstill, the checkback message [EN_AQ] is reset.
3. The sequence is as described in point 2. In addition, the axis virtual master is in operation [FUR_VM] = 1. The axis is brought to a halt as described in point 2. Only the checkback message [EN_AQ] remains active for as long as the virtual master is processed.
4. As soon as the VM has stopped [FUR_VM] = 0, the checkback message [EN_AQ] is also reset.
5. The converter is controlled independently of the axis enable, using the signals [OFF1], [OFF2], [OFF3], [ENC], [ACK_F] and [LB]. The checkback signals in DBBy – DBBy+3 are updated accordingly.

12.3.2 DP/IM178

The following table shows the control / checkback signals for the analog interface with Profibus DP/IM178.

Control Signals

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBx	RES	RES	RES	EN_AX	RES	RES	RES	RES	BIN	IN_1
DBBx+1	ACK_F	RES	RES	RES	ENC	RES	RES	RES	BIN	IN_2

ENC Controller enable (output Q2)

EN_Ax Axis processing enable

Checkback Signals

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy	RES	RES	RES	EN_AQ	RES	RES	RES	RES	BIN	OUT_1
DBBy+1	RES	WARN	RES	RES	FAULT	IOP	RES	RES	BIN	OUT_2

EN_AQ Checkback signal - Axis being processed

IOP Controller ready for operation (checkback from converter via input I2)

Impulse Diagram

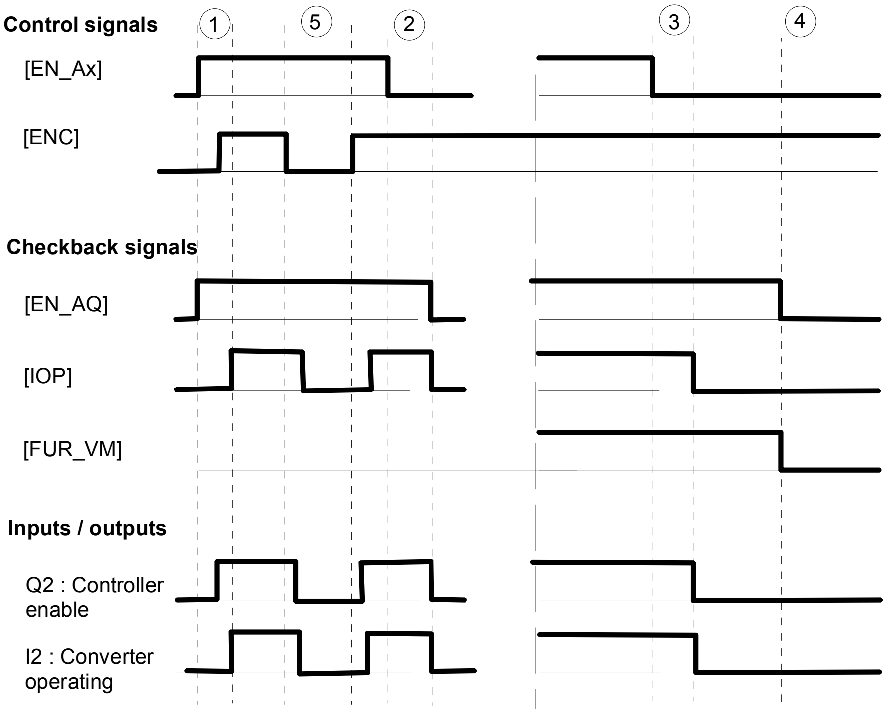


Fig. 12-6 Impulse diagram

The above impulse diagram shows the signal sequence for the control of an axis with an analog interface with IM178.

1. The processing of the axis is enabled [EN_Ax] = 1, the checkback message "Axis being processed" [EN_AQ] is set. When the controller is enabled [ENC] = 1, output Q2, "controller enable", is set. As soon as the converter reports back via input I2, "converter operating", the checkback message is activated.
2. During the processing of an axis (BAx is active, the axis is moving), the access enable [EN_Ax] is reset. The current processing of the axis is interrupted and the axis is brought to a halt using a speed ramp. When the axis is at a standstill, output Q2, controller enable, is reset. As soon as input I2 changes to 0, the checkback signals [IOP] and [EN_AQ] are reset.
3. The sequence is as described in point 2. In addition, the axis virtual master is in operation [FUR_VM] = 1. The axis is brought to a halt as described in point 2. Only the checkback message [EN_AQ] remains active for as long as the virtual master is processed.

4. As soon as the VM has stopped [FUR_VM] = 0, the checkback message [EN_AQ] is also reset.
5. The enable [ENC] is reset. Output Q2, "controller enable", becomes inactive. As soon as input I2 changes to 0, the checkback signal [IOP] is reset. When the enable [ENC] is set again, output Q2, "controller enable", is activated. As soon as the checkback message I2 changes to 1, the status [IOP]=1 is displayed.
   If the enable is reset during an active operating mode (moving axis), the mode is ended and an error message is given (130 operating condition not fulfilled), it must then be restarted once the error has been acknowledged.

12.3.3 Allocation of the Optional Range

The parameters OPTIONAL VALUE 1,2,3 INPUT in the control interface and OPTIONAL VALUE 1,2,3 OUTPUT in the checkback interface are defined by the application in the SIMATIC Motion Control solution.

In this version, the allocation is fixed, although flexible application is planned for a later upgrade version. The current fixed allocation is adopted as the default setting.

Control Interface

The MDI set 0 is available within the control interface.

	7 6 5 4 3 2 1 0		**Axis_n.**
DBBx+10	1. G function	DEC	IN_9_2
DBBx+11	2. G function	DEC	IN_9_2
DBDx+12	Position in [LU]	DEC	IN_10
DBDx+16	Speed in [10 * LU/min]	DEC	IN_11

Further Information

Instructions can be found in Chapter 5 of the description of function: Manual Data Input.

Checkback Interface

The following values are stored in the checkback interface.

DBWy+10	RES	OUT_9_2
DBDy+12	ACTUAL POSITION	OUT_10
DBDy+16	SPEED	OUT_11

The speed value is only updated on a cyclical basis for position-controlled operating modes.

12.4 Cam Controller

In each axis, there is a cam controller with four parameterizable position cams. The scanning time for the axis is used for processing the cam controller. If the axis is not enabled for processing (see axis processing), the cam controller cannot be processed either.

Further Information

A description of the command for parameterizing the cam controller using the S7 can be found in the description of commands in the configuration instructions under ‘Commands for SIMATIC Motion Control".

The cam controller can be parameterized as follows:

Enable

The command for parameterizing the axis management system incorporates not only the parameterization of the axis processing (scanning time, sequence, etc.), but also an enable for the cam controller. The cam controller is only processed if the enable is set.

Command: Axis management

	7	6	5	4	3	2	1	0	Axis_n.
DBBz+(n-1)*4+2	0	0	0	0	EN_CU	EN_DAS	**EN_CAM**	EN_VM	BIN

EN_CAM: Cam controller enable

Parameter File

Name	Input limits		Unit	Default
	Lower	Upper		
Local M7 axis number for input value	1	to max. axes	-	0
Input value options	0: Set position value 1: Actual position value 2: Virtual master 3: x coordinate of cam disc 4: y coordinate of cam disc		-	1
Hysteresis	-2^{31}	$-2^{31}-1$	LU	0
Start of cam 1 in [LU]	-2^{31}	$-2^{31}-1$	LU	0
End of cam 1 in [LU]	-2^{31}	$-2^{31}-1$	LU	0
Start of cam 2 in [LU]	-2^{31}	$-2^{31}-1$	LU	0
End of cam 2 in [LU]	-2^{31}	$-2^{31}-1$	LU	0
Start of cam 3 in [LU]	-2^{31}	$-2^{31}-1$	LU	0
End of cam 3 in [LU]	-2^{31}	$-2^{31}-1$	LU	0
Start of cam 4 in [LU]	-2^{31}	$-2^{31}-1$	LU	0
End of cam 4 in [LU]	-2^{31}	$-2^{31}-1$	LU	0
Reserved				
Reserved				
Reserved				
Reserved				
Reserved				
Reserved				
Reserved				
Reserved				

NOTE

In the case of rotary axes, the start and end values of the cams and hysteresis are limited to the axis cycle.

Local M7 Axis Number

The axis number sets the reference for the input parameters. By default (0), the input parameters of the current axis are processed. When an axis number is entered (1..8, 16, 32), the input parameters of the axis which is entered are processed. This gives you the option of parameterizing more than four cams for one axis. With regard to the reaction of the cam outputs, the scanning time of the axes and their processing sequence must be taken into consideration.

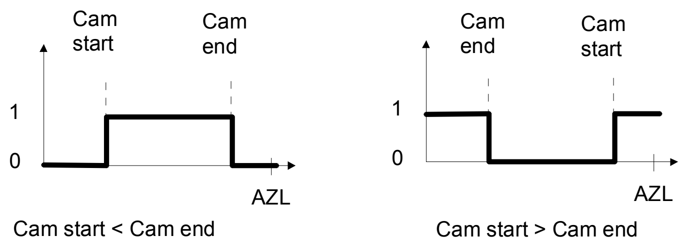
Input Parameters for Cam Controller

The following input parameters are available:

- Set position value
- Actual position value
- Set position of virtual master
- x coordinate of cam disc
- y coordinate of cam disc

Cam Start Value, Cam End Value

The cams are defined as two-way cams. If you swap the cam start and cam end values, the cam is inverted. You can also switch off each cam by defining the start and end value = 0.

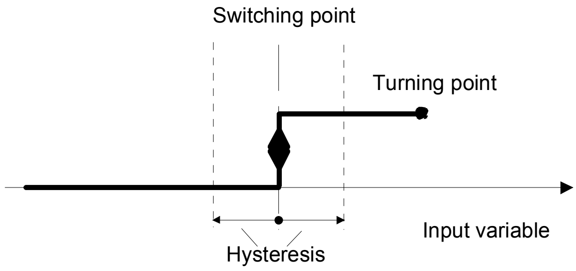


Hysteresis

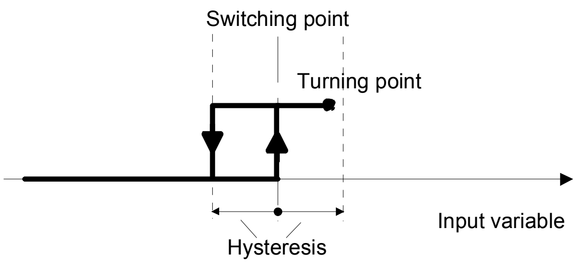
By entering a hysteresis, you can suppress the flickering of cam outputs if the cam comes to a standstill on a switching edge.

The hysteresis is defined as follows:

- ♦ If the direction changes outside the hysteresis range, the hysteresis has no influence on the switching position.



- ♦ If the direction changes within the hysteresis range, the defined switch point is adjusted by the magnitude of the hysteresis.



NOTE

If the hysteresis of one cam exceeds the hysteresis of a second cam, then no hysteresis is effective from the moment the axis crosses the second cam. Any active hysteresis will cease to be valid when parameters are reassigned.

Cam Outputs

The cam outputs (cam image) can be analyzed as follows:

Analysis in Programmable Controller

The cam outputs are made available in the P-Bus window to enable analysis in the programmable controller. You can read the cam images directly using peripheral commands (L PEB x).

The cam outputs are available at the following peripheral addresses:

PEB x+2 cam image for axis 1

PEB x+3 cam image for axis 2

...

PEB x+33 cam image for axis 32

Where x is the module starting address of the M7. You set this address in the HW Config under Properties FM456-2-Addresses (see Installation Description).

The bytes are allocated as follows:

	7	6	5	4	3	2	1	0	Axis_n.
PEB x+n	0	0	0	0	**CAM 3**	**CAM 2**	**CAM 1**	**CAM 0**	BIN

NOTE

The cams may fluctuate owing to the PLC reaction time. This time is determined by the analysis method used and the availability of CPU processing capacity.

Asynchronous analysis

With this method, the outputs are analyzed asynchronously to the processing of the cam controller on the M7, e.g. they are analyzed on a cyclic interrupt level or on the cyclic program execution level.

Synchronous analysis

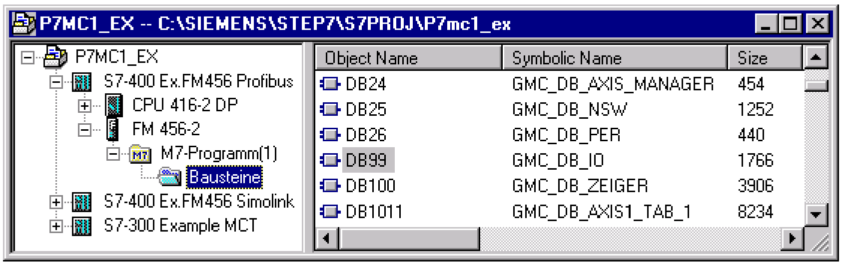
Synchronous analysis can be selected in cases where the M7 generates a process alarm (OB40) on the S7-CPU in every closed-loop control cycle.

However, to safeguard the performance of the whole system, the process alarm is not enabled on the M7 in the default setting.

If you need the cam outputs to be analyzed synchronously, you can set bit DBX37.1 in data block DB99, "GMC_DB_IO" on the M7 to enable alarm generation.

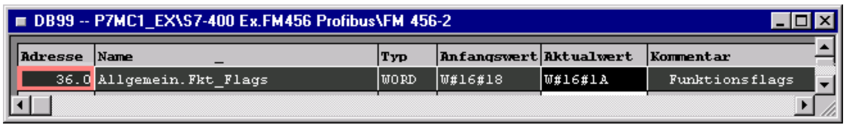
To do this, proceed as follows:
Illustrated using example program "S7-400 Ex.FM456 Profibus":

In the offline project, open DB99 in the block container of the M7 program.



Select "View – Data View" in the DB Editor. The avtual value W#16#18 is displayed for data word DBW36.

Now change the value in input field "Actual value" from W#16#18 to W#16#1A and save the modified block again. Then move the data block to the M7.



Once you have moved the block to the M7, restart the system.

CAUTION

An internal application error can occur if you change this data word as well.

You may only transfer the modified data block DB99, "GMC_DB_IO" to the M7 when the system is in the STOP state.

NOTE

Please also note that the CPU cycle time will increase as a result of this cyclic process alarm.

Direct Control of Peripheral Outputs using the M7

You can extend the M7 with additional digital inputs and outputs in the form of IF modules or distributed peripherals (see 12.5 "Advanced Digital Inputs / Outputs").

By parameterizing the extended peripherals, you can allocate individual cams to direct peripheral outputs and therefore allow direct control using cam outputs.

12.5 Advanced Digital Inputs / Outputs

To enable direct control of the technology functions and peripheral signals, there are 6 parameterizable inputs (I1 ..I6) and 6 parameterizable outputs (O1...O6) for each axis. The inputs / outputs are parameterized using technology machine data (MD45, MD46).

The outputs from the cam controller can also be used to control peripheral signals directly.

In order to use these functions, you can extend the M7 with digital I/O module. Digital I/Os are available in the form of IF modules which can be inserted into the M7-FM directly or in the form of add-on modules. You can also connect digital I/Os as distributed I/Os using Profibus DP.

Configuration

The following configuration steps need to be taken in order to extend the M7 configuration:

- ♦ Extend the hardware configuration using the STEP7 HW configuration utility
  (select module and configure address)
- ♦ Extend the M7 application configuration
  (process module input / output data)
- ♦ Parameterization of input / output allocation
  (allocate peripheries to axis inputs / outputs)

Hardware Configuration

You can use the following modules as additional digital I/Os:

- ♦ M7 – IF module (IF 961) in M7-FM or in an M7 add-on module (EXM 478)
- ♦ Distributed I/Os
  e.g. ET200M (IM153) with S7-300 peripheral (e.g. SM321, SM322, SM323), ET200 B,L etc.

NOTICE

When using distributed I/Os in conjunction with the configuration of analog axis connections with DP/IM178, the distributed I/O is operated on the same bus as the IM178 stations.

When selecting modules, the baud rate and equidistance cycle required on Profibus should be taken into consideration.

Extension of M7 Configuration

In the M7 application configuration, you set the general allocation of input / output data to an internal process image. The application makes available an image range for input and output data. The image ranges each represent 4 data words, i.e. you can configure a maximum of 64 digital inputs and 64 digital outputs.

The allocation of modules is on a word basis. For 8 bit modules, it is possible to allocate explicitly to the high or low byte of the word.

1. Word (Low)	7	6	5	4	3	2	1	0
1. Word (High)	15	14	13	12	11	10	9	8
2. Word (Low)	23	22	21	20	19	18	17	16
2. Word (High)	31	30	29	28	27	26	25	24
3. Word (Low)	39	38	37	36	35	34	33	32
3. Word (High)	47	46	45	44	43	42	41	40
4. Word (Low)	55	54	53	52	51	50	49	48
4. Word (High)	63	62	61	60	59	58	57	56

Table 12-2 Image range for input or output image

NOTE

In order to extend the configuration of the M7 application, you need a configuration tool, which is based on Excel. The tool is not supplied as standard but is available on request.

Parameterization of Input / Output Allocation

The input / output allocation is parameterized using a command. The allocation is for each individual channel. The reference for the allocation of the axis inputs / outputs is the input or output number in the image range in accordance with Table 12-2.

NOTE

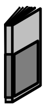
Please note that changing parameter settings when the system is running is equivalent to modifying the hardware configuration. Disconnected inputs and outputs can "freeze" at their last signal state. You can remedy this situation by restarting the system.

Parameter File

Name		Input limits Lower	Upper	Unit	Default
Allocation number for digital inputs / outputs		1	64	-	0
1^{St} allocation	Peripheral input / output no.	0	63	-	
	Axis number	1	8/16/(32) 1)		
	Axis input / output no.	1	6/10 2)		
$2^{N.D.}$ allocation	Peripheral input / output no.	0	63	-	
	Axis number	1	8/16/(32) 1)		
	Axis input / output no.	1	6/10 2)		
64^{Th} allocation	Peripheral input / output no.	0	63	-	
	Axis number	1	8/16/(32) 1)		
	Axis input / output no.	1	6/10 2)		
Reserved		-	-	-	0

1) depends on the configuration (upper limit = maximum number of axes)

2) Input no.: 1 .. 6 ↔ E1 .. E6
Output no. : 1 .. 6 ↔ A1 .. A6
7 .. 10 ↔ Cams 1 .. 4



Further Information

A description of the command for parameterizing the extended I/O peripheral using the S7 can be found in the description of commands in the configuration instructions under "Commands for SIMATIC Motion Control".

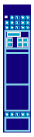
13 Extended Functions with MASTERDRIVES MC and F01

Contents

In this chapter you will find ...

13.1 Encoder Changeover

NOTE



	Function implemented as of:	**Function:**
MASTERDRIVES MC:		
	≥ V1.41	Encoder changeover, positioning with machine encoder
	≥ V1.51	Encoder changeover, positioning with motor encoder (U528), MD48: Variant 6, encoder monitoring signal

Overview

The encoder changeover function is implemented in the form of a BICO data set changeover. See also Chapter 13 "Configuring and Parameterizing the MASTERDRIVES MC (MCT) Converter".

You can use the encoder changeover function only in conjunction with roll feed axes, with the motor encoder connected to encoder evaluation "Motor encoder in Slot C" and the machine encoder as "External Encoder, Not Slot C".

If the motor encoder is switched over to the machine encoder (activate encoder changeover) from the user program, the machine encoder actual value is used for the following positioning operation(s). With MASTERDRIVES MC version V1.51 and later, the motor encoder actual value can continue to be used for positioning.

Encoders can be switched over only when the axis is at a standstill.

A comparison is made between the two actual values when the system is running. It is possible to read out the difference between them. The deviation between them is monitored for a maximum value. When the deviation exceeds the maximum value, the axis is either stopped and an error number output or a parameterizable output is set (via MD48) and positioning continues.

NOTE

The encoder changeover function requires an extended technology "connection" on the MASTERDRIVES MC. As a SIMOVIS/DriveMonitor scriptfile, this connection is included on CD SIMATIC Motion Control, GMC Basic Configuring Package.

13.1.1 Encoder Changeover, Positioning with Machine Encoder

13.1.1.1 Function Description

Function

To use the encoder changeover function, set machine data MD1 to "3" for roll feed.

To set up the axis, positioning initially commences with the motor encoder (in slot C) and the axis then switched over later to a machine encoder (external encoder).

Both measuring systems are then active in operation. To implement monitoring of the two actual-value encoders, parameterize "limit monitoring encoder changeover", function parameter FP6.
This monitoring function compares the difference between encoder 1 actual value minus encoder 2 actual value with function parameter FP6. If the deviation is higher than the value in function parameter FP6, the drive is stopped and alarm No. 139 "Difference between actual values of encoder 1 and encoder 2 too large" is output. You must perform a RESET to acknowledge this alarm.
Alternatively, a parameterizable output can be set (via MD48) and positioning continued.
This allows you to determine, for example, whether the parameterized acceleration is too high or whether slip is occurring on the machine encoder (odometer).

You can read out the encoder actual values using task "Actual value output", HA3.

NOTICE

Encoders can be switched over only when the axis is stationary and checkback signal [FUR] "Machining in Progress" = 0 signal.

13.1.2 Encoder Changeover, Positioning with Motor Encoder

13.1.2.1 Function Description

Function

To use the encoder changeover function, set machine data MD1 to "3" for roll feed.

However, this variant evaluates the actual value from the motor encoder while the machine encoder has a monitoring function only. Since axis positioning now continues via the motor encoder, you obtain a better dynamic response than with the machine encoder option.

To ensure that positioning with the motor encoder can continue after encoder changeover, binector U528 must be connected to value one in the MASTERDRIVES MC and you must make actual-value connection parameter P194.02 = KK120.

Both measuring systems are then active in operation. To implement monitoring of the two actual-value encoders, parameterize "Limit monitoring encoder changeover", function parameter FP6.
This monitoring function compares the difference between encoder 1 actual value minus encoder 2 actual value with function parameter FP6. If the deviation is higher than the value in function parameter FP6, the drive is stopped and alarm No. 139 "Difference between actual values of encoder 1 and encoder 2 too large" is output. You must perform a RESET to acknowledge this alarm.
Alternatively, a parameterizable output can be set (via MD48) and positioning continued.
This allows you to determine, for example, whether the parameterized acceleration is too high or whether slip is occurring on the machine encoder (odometer).

You can read out the encoder actual values using task "Actual value output", HA3.

NOTICE

Encoders can be switched over only when the axis is stationary and checkback signal [FUR] "Machining in Progress" = 0 signal.

13.1.3 Encoder Changeover - Tasks

Activate/deactivate encoder changeover

If the conditions for encoder changeover are fulfilled, you can initiate the changeover process with task "Activate/deactivate encoder changeover input".

Encoder changeover status

You can read out the encoder changeover status with task "Activate/deactivate encoder changeover output".

NOTE

Encoder changeover is always **inoperative** after power-up. In other words, the axis is positioned via the motor encoder.

13.1.4 Control Signals

No direct control signals are available for encoder changeover.

13.1.5 Checkback Signals

Machine encoder active [B359] (software limit switch reached OTR)

You can use the encoder changeover function only in conjunction with roll feed axes. Since a roll feed axis does not require a software limit switch, the checkback signal OTR "Software limit switch reached" [B359] has been used to display the encoder changeover status.

0 signal: Machine encoder deactivated

1 signal: Machine encoder active

NOTE

The connection for checkback signal "Machine encoder active" is taken into account in scriptfile "8_Scriptfile_MCT_sensor_changeover.ssc".

13.1.6 Function Parameters

Parameters

The encoder changeover function is configured via the following parameters:

- Machine data MD1, roll feed variant
- Machine data MD48, variant 6, encoder monitoring signal
- Function parameter FP6 (U504.6), limit monitoring encoder changeover
- Selection of position control via motor encoder or machine encoder:
  Binector U2528 = 0: Position control via machine encoder (compatible mode)
  Binector U2528 = 1: Position control via motor encoder
- Display difference between motor and machine encoders:
  KK308 = Encoder 1 actual value minus encoder 2 actual value

NOTE

If you require the current difference between the motor and machine encoders for further processing / display in your PLC, you can connect double connector KK308 to the PROFIBUS-DP.

Further Information

For a description of parameters, please refer to chapter "Technology Machine Data and Parameters" in the Description of Functions

13.1.7 Configuring and Parameterizing the MASTERDRIVES MC (MCT) Converter

Introduction

The GMC-BASIC standard software operates only in conjunction with technology option F01 on the MASTERDRIVES MC. You therefore need to activate technology option F01. You can check whether it has been activated by reading parameter n978 (parameter n978=1 : Is Activated). An extended technology connection must also be installed in the MASTERDRIVES MC for the encoder changeover function.

You will find the technology scriptfile for encoder changeover, 8_Scriptfile_MCT_sensor_changeover.ssc, on the SIMATIC Motion Control CD.

Hardware requirements

- ♦ You need a MASTERDRIVES MC with activated technology option F01
- ♦ An encoder evaluation circuit for an external encoder, not slot C.
- ♦ Connecting cable PG/PC – MASTERDRIVES MC (USS)

Software requirements

To load the extended technology connection, you must have installed the SIMOVIS/DriveMonitor tool on your PG/PC. SIMOVIS/DriveMonitor is not included in the scope of supply of GMC-BASIC.

- ♦ PG/PC with SIMOVIS/DriveMonitor ≥ V5.3
- ♦ SIMOVIS/DriveMonitor device files for firmware version ≥ V1.41

Encoder changeover scriptfile

Without an additional connection on the MASTERDRIVES MC, you will not be able to implement a motor encoder / machine encoder changeover.

For this reason, a preprogrammed SIMOVIS/DriveMonitor scriptfile is supplied for the encoder changeover function. It contains some parameters which you will need assign yourself and others which you must not change.

User parameters

The user parameters include configuration of the 2nd encoder, actual value weighting of the 2nd encoder, SBP configuration, reference speed for the machine encoder, actual-value connection and difference between encoder 1 and encoder 2 actual values.

NOTE

An encoder with 5V/RS422 and 5000 pulses per revolution is configured in Section "A3) SBP Configuration".

```
REM *****************************************************
REM **   A1) Configuration 2nd encoder [335,336]        **
REM *****************************************************
REM --- xxx1: Enable position sensing
REM --- xx0x: No reference point sensing
REM --- x0xx: Encoder CW rotation
REM --- 0xxx: Input of AVWF – integer/decimal positions
WRITE 166  1  0x1

REM *****************************************************
REM **   A2) Actual value weighting factor 2nd encoder
[335] **
REM
*****************************************************
REM --- AVWF integer position, example = 2
WRITE 152  1  2
REM --- AVWF decimal position, example = 44140625
WRITE 153  1  44140625
REM --- AVWF numerator, example = 10000um : 4096 pul/rev
REM WRITE 181 1 10000
REM --- AVWF denominator, example = 4096
REM WRITE 181 2 4096

REM *****************************************************
REM **   A3) SBP configuration [250]                    **
REM *****************************************************
REM ---Change to Drive Settings menu
WRITE 60  0  5
WAIT 60  0  5  5
REM  Input level: Example -> TTL
REM  Encoder voltage: Example -> 5V/RS422 [255]
WRITE 150  2  0x33
REM ---No. of encoder pulses: Example -> 5000
WRITE 151  2  5000

REM *****************************************************
REM **   A4) Machine reference speed [20,335]           **
REM *****************************************************
REM --- Reference speed for machine encoder
REM --- Example: 6000  rev/min
WRITE 355  0  6000
```

```
REM ****************************************************
REM **   A5) Select actual value for position controller**
REM ****************************************************
REM --- Actual position 1st encoder for position control
REM WRITE 2528 0 1
REM WRITE 194 2 120
REM --- Actual position 2nd encoder for position control
WRITE 2528 0 0
WRITE 194 2 125

REM ****************************************************
REM **   A6) Display difference (encoder 1 - encoder 2) **
REM ****************************************************
REM ---Connect difference to Profibus [817]
REM WRITE 734 7 308
REM WRITE 734 8 308
REM
REM  or
REM ---Connect difference to Profibus [817]
REM WRITE 734 9 308
REM WRITE 734 10 308
```

Technology connection

The technology connection parameters listed after the user parameters must be loaded unchanged to the drive with the user parameters.

13.2 Reversing Lockout

NOTE

The "Reversing Lockout" function is available only with MASTERDRIVES MC firmware version ≥ V1.5.

The block is described on function diagram 836 of the MASTERDRIVES MC Compendium.

The purpose of this function is to prevent the motor from running in the opposite direction to the enabled direction. If the motor runs in the reverse direction to the enabled direction when the lockout is active, the axis stops with 0 speed.

This means that master and slave become de-synchronized. Checkback signal "Axis is synchronized" [B811] is reset for this purpose (see also 10.8 "Offset Angle Setting / Synchronization".)

The loss of synchronization is displayed in the status memory for the reversing lockout function [B812= 1]. If the motor begins to rotate in the enabled direction again, the status memory for Reversing Lockout Active [B812= 1] remains set and is displayed by the status Reversing Lockout Not Currently Active [B813]. The status memory Reversing Lockout Active is not cleared again until command "Synchronize with master value", source U676 = 1, is issued.

Mode of operation

Enabled direction: V+ [U463.01] V- [U463.02]	Current direction [n653.02]	Effective direction [n653.01]	Reversing lockout active memory [B812]	Reversing lockout not curr. active memory [B813]
V+	V+	V+	0	1
V+	V-	0	1	0
V+	V+	V+	1	1
		Synchronize [U676] => 1	0	1
V-	V-	V-	0	1
V-	V+	0	1	0
V-	V-	V-	1	1
		Synchronize [U676] = 1	0	1

Table 13-1 Mode of operation and checkbacks of reversing lockout function

In the MASTERDRIVES MC factory setting, the enabling signals for reversing lockout [U463.01] and [U463.02] are preset to fixed binector 1, i.e. the reversing lockout is not active per default.

13.2.1 Control Signals

Synchronize with master value (U676)
This signal resets the reversing lockout memory again.

Enable ...
.. positive direction (U463.01)
.. negative direction (U463.02)
The Enable ... direction command defines the directions of rotation enabled for the drive.

13.2.2 Checkback Signals

The reversing lockout status is displayed in parameter "Slave axis status" of the actual synchronization values.

Reversing lockout active memory (B812)
Indicates that a correction with reversal of rotational direction has been detected.

Reversing lockout not currently active (B813)
Indicates that reversing lockout is not currently active.

13.3 Parallel Velocity Branch in Synchronized Operation

Function

Only one position setpoint is basically needed for synchronization to be fully functional. This is supplied by either a virtual or a real master. At its output, the synchronization function generates a velocity precontrol signal for closed-loop speed control in addition to the position setpoint.

You can use the parallel velocity branch additionally if you wish. In this case, the velocity setpoint is transferred to the synchronization as well as the position setpoint. In synchronized operation, all functions, such as gear functions, are then calculated in parallel for both position and velocity.

Advantages

The purpose of this parallel calculation is to achieve a high-quality velocity precontrol output for synchronization. When injected into the closed-loop control, this output can improve the accuracy of the closed-loop synchronization control.

We recommend use of the parallel velocity path in the following cases:

- If you require a very high synchronization quality.
  The high quality of the precontrol signal helps to reduce errors in the speed control loop.
- Axes are operating in synchronism with low encoder resolution.
  A low resolution in the actual position and setpoint channels impairs the quality of the velocity precontrol signal.

Application

It is very easy to apply the parallel velocity branch. When you connect the external velocity setpoint input, the internal calculation is adjusted automatically. The connection must be made at connector input U600.04 to 06.

Index

A

B

C

I

J

K

L

M

N

O

SIEMENS

System Solutions

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Programming Guide Edition 11.2002

Writing NC Programs

SIEMENS

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Part 1:Technology Functions

Programming Guide–
Writing NC Programs

Edition 11.2002

Documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the "Remarks" column.

Status code in the "Remarks" column:

A .... New documentation.
B .... Unrevised reprint with new Order No.
C .... Revised edition with new status.

Edition	Order No.	Remarks
04.98	6AT1880-0AA00-1BA0	**A**
10.99	6AT1880-0AA00-1BC0	**C**
11.2002	6AT1880-0AA00-1BE0	**C**

SIMATIC and SINUMERIK are registered trademarks of Siemens AG. Other names in this publication might be trademarks whose use by a third party for his own purposes may violate the rights of the registered holder.

We have checked that the contents of this document correspond to the hardware and software described. Nonetheless, differences might exist and therefore we cannot guarantee that they are completely identical. The information contained in this document is, however, reviewed regularly and any necessary changes will be included in the next edition. We welcome suggestions for improvement.

Subject to change without prior notice

Order No.: 6AT1880-0AA00-1BE0
Printed in the Federal Republic of Germany

Siemens-Aktiengesellschaft

Contents

1 Fundamentals

Contents

In this chapter you will find all the fundamental information about programming.

1.1 NC Program

General Information

The program design is based on DIN 66025. An NC program comprises a sequence of up to 200 NC blocks and is assigned to a program number.

Program Number / Subprogram Number

The numbers 1 to 200 (M7-FM) and 1 to 20 (MASTERDRIVES MC) are permitted for use as program numbers/subprogram numbers.

The numbers 201, 202, 203 and 204 (local axis 1 to 4 for M7-FM), and 21 (MASTERDRIVES MC) are reserved for the single block operating mode.

Program Structure

An NC program comprises a number of NC blocks which are executed in ascending order according to the block number. The start of program and end of program do not have to be labeled explicitly. The first NC block is automatically the start of program and the last NC block is the end of program. The special M functions which generate an end of program (M02, M30) or end of subprogram (M17, M29) in accordance with DIN 66025 can be used to enhance the legibility of the program.

There is no intrinsic difference between subprograms and main programs. A program only becomes a subprogram when invoked from an NC program. That means you can run a subprogram as a main program.

1.2 NC Block

General Information

An NC block contains all the data required for the execution of a machining step, e.g. positioning, dwell or output of switching functions.

Block Structure

An NC block comprises:

- ♦ A character for the start of block
- ♦ A block number
- ♦ A number of words containing the data for the execution of a machining step

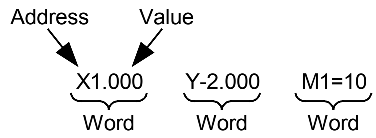
Programming:
N10 G.. X.. Y.. Z.. F.. M.. D..
N20 L.. P..
/ N30 ..

/	Skip identifier for skip block
N	Block number
G	G functions
X, Y, Z	Positional data of axes (preferred usage for linear axes)
A, B, C	Positional data of axes (preferred usage for rotary axes)
X	Parameter for dwell
F	Path velocity for interpolation
FX, FY, ...	Axis velocity for chaining
M	M function
D	Tool offset number
L	Subprogram number
P	Loop count

Word

A word is an element of an NC block and consists of an address character and a sequence of digits. The address character is an alphanumeric character string. The digit sequence can be signed and can include a decimal point. The sign appears between the address letter and the digit sequence; a positive sign can be omitted. There must be no space between the address and the value.

Example:



Block Number

> **Programming:**
> **N10** G90 G43 X100.000 F4000.00 M1=10 M2=11 M3=12 D1

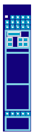
You can assign block numbers in any combination from 1 to 200, however you should remember that the program is executed in ascending order of block numbers. It is practical to define the block numbers in increments of five (5, 10, 15....) so that blocks can be inserted at a later stage.

Axis Assignment

> **Programming:**
> N10 G90 G01 **X**100.000 **Y**100.000 F10000.00
> N20 G77 A100.000 FA2000.00

Machine data 2 can be used to assign a logical name to an axis (X, Y, Z for linear axes and A, B, C for rotary axes). When you write an NC program, the axis parameters (positional data) refer to these logical axes.

MD No.	I	A	W	Name	Input Limit Lower	Input Limit Upper	Unit	Default
2				Axis assignment (U501.02)	1: X axis	7: XA axis	-	1,2,3,4
					2: Y axis	8: YA axis		
					3: Z axis	9: ZA axis		
					4: A axis	10: AA axis		
					5: B axis	11: BA axis		
					6: C axis	12: CA axis		
					13: XB axis			
					14: YB axis			
					etc. up to 32 axes max.			



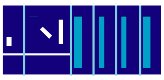
Because only one axis ever exists on a MASTERDRIVES MC, the NC program can only access this axis name defined in machine data 2. Functions such as interpolation are therefore not supported. On a MASTERDRIVES MC, only axis assignments 1 to 6 are supported.

NOTE

Note for the following travel program examples:

- Should the positions and speeds of several axes be shown in the NC block, these must be reduced to one axis.
- Should differing axes be shown within one travel program, these must be reduced to one axis.

This functional description continues to be valid. The representation, however, must also be reduced to one axis.



Different axis names must be assigned to the axes on an M7-FM.

G Functions

> **Programming:**
> N10 **G90** **G43** X100.000 F4000.00 M1=10 M2=11 M3=13 D1

The G functions are used, on the one hand, to specify the type of movement and, on the other, to call up offsets, shifts and special functions. The G functions are subdivided into G groups according to their meaning. Each G group has an initial state or default setting, i.e. this G function is active before selection.

Example:
G00 - Interpolation with rapid traverse for the 1st G group.

The scope of G functions is "modal" or "non-modal" (local), depending on whether they remain active after the end of block.

Several different G functions can be specified in an NC block. Each NC program is a self-contained unit separate from other NC programs. This means that all G groups are in the default setting at the start of an NC program and this setting must be switched if necessary.

G Functions - Modal

Modal G functions remain active until they are replaced by another G function of the same group.

Example:
G77 - Chaining with axis velocity cancels G76 - Chaining with rapid traverse.

G Functions - Local

Non-modal (local) G functions are only active in the block in which they are programmed.

G Functions - List

G functions which are active in the default setting are printed in bold; modal functions are identified by m; non-modal (local) functions are identified by l:

Group	G Function		Active	Roll feed
G group for preparatory function				
1	**G00**	Interpolation with rapid traverse or roll feed with rapid traverse	m	yes
	G01	Interpolation with path velocity or roll feed with axis velocity	m	yes
	G76	Chaining with rapid traverse	m	no
	G77	Chaining with axis velocity	m	no
G group for tool length compensation				
5	**G43**	Tool length compensation +	m	no
	G44	Tool length compensation -	m	no
G group for zero offset				
6	**G53**	Deselect zero offset	m	no
	G54	Zero offset 1	m	no
	G55	Zero offset 2	m	no
	G56	Zero offset 3	m	no
	G57	Zero offset 4	m	no
	G58	Zero offset 5	m	no
	G59	Zero offset 6	m	no
G group				
7	G04	Dwell	s	no
G group for block monitoring behavior				
8	G60	Exact stop	m	no
	G63	Position-dependent chaining	m	no
	G64	Maximum corner rounding	m	no
	G66	Corner rounding window 1	m	no
	G67	Corner rounding window 2	m	no
G group				
9	G68	Shortest path with rotary axis	s	no
G group for dimensions				
11	**G90**	Absolute dimensions	m	no
	G91	Incremental dimensions	m	yes

Group	G Function		Active	Roll feed
G group for special functions				
13	G50	External block change with absolute dimensions	s	no
	G51	External block change with incremental dimensions	s	yes
	G87	Set actual value on-the-fly - Deselection	s	no
	G88	Set actual value on-the-fly / Turning endlessly -	s	no
	G89	Set actual value on-the-fly / Turning endlessly +	s	no
	G99	Read-in enable, externally programmable	s	no
G group for collision detection				
14	G96	Select collision monitoring	m	no
	G97	Deselect collision monitoring	m	no
G group for acceleration/delay				
16	**G30**	100 % override acceleration/deceleration	m	yes
	G31	10 % override acceleration/deceleration	m	yes
	G32	20 % override acceleration/deceleration	m	yes
	G33	30 % override acceleration/deceleration	m	yes
	G34	40 % override acceleration/deceleration	m	yes
	G35	50 % override acceleration/deceleration	m	yes
	G36	60 % override acceleration/deceleration	m	yes
	G37	70 % override acceleration/deceleration	m	yes
	G38	80 % override acceleration/deceleration	m	yes
	G39	90 % override acceleration/deceleration	m	yes

All G functions can be programmed, even using the technology, on the MCT . G functions which are only activated when used together with several axes, are ignored during machining.

Positional Data

Programming:
N10 G90 G43 G01 **Z100.000** F4000.00 M1=10 M2=11 M3=13 D1

The positional data comprise an address letter identifying the axis (X, Y, Z, ...) and the value of the position. Positional data can be entered with a positive or negative sign. You can omit the sign when entering a positive value.

Name	Input	Unit
Positional data A, B, C, X, Y, Z	from -999 999,999 to +999 999,999	mm

Velocity

Programming:
N10 G90 G43 G01 X100.000 **F4000.00** M1=10 M2=11 M3=13 D1
N20 S1000.00 M1=3

We distinguish between a path velocity and an axis velocity.

- For interpolation with path velocity (G01), the velocity is specified together with the address letter F. The velocity value F that you enter refers to the programmed path. The individual axis velocities are calculated from the distances to be traversed. The resulting path is a straight line.
- For chaining with axis velocity (G77), the velocity is specified together with the address letter FX, FY, etc. The velocities that you enter refer to axes X and Y. Each axis travels at the programmed axis velocity. The resulting path depends on the distances to be traversed and the axis velocities.

During both interpolation and chaining, the control monitors that none of the axes traverses at a rate in excess of its own traversing velocity - a maximum of (machine data 23). All velocity values are modal.

Name	Input	Unit
Velocity F, FX, FY, ...	from 0,01 to 1000 000,00	mm/min

M Functions

Programming:
N10 G90 G43 G01 X100.000 F4000.00 **M1=10 M2=11 M3=13** D1

You can use the M functions to define the activation of machine functions at the time you write the NC programs.

The M functions comprise the address code M1=, M2=, M3= and a number between 0 and 255. You can program up to three M functions in each NC block. Any user M functions can be programmed. The M functions with special functions must be programmed in the M1 group (except for M97 and M98).

Assignment of M Functions

M1	M2	M3	Special Functions
0			Stop at end of block
1	1	1	User function
2			End of program
3 to 16	3 to 16	3 to 16	User functions
17			End of subprogram
18			Infinite loop
19 to 28	19 to 28	19 to 28	User functions
29			End of subprogram
30			End of program
31 to 96	31 to 96	31 to 96	User functions
97 to 98	97 to 98	97 to 98	Output programmable
99	99	99	User functions
100 to 255	100 to 255	100 to 255	User functions

Table 1-1 Meaning of the Individual M Functions

Tool Offset Number

Programming:
N10 G90 G43 G01 X100.000 F4000.00 M1=10 M2=11 M3=13 **D1**

The tool offset allows you to introduce different offset values without changing the NC program.

The tool length offset consists of an address letter D and a number between 0 and 20.

You select the appropriate offset by assigning D1 to D20. The direction of the tool offset is derived from G43 (tool length compensation positive) and G44 (tool length compensation negative).

Assigning D0 deselects the tool length offset.

Dwell

Programming:
N10 **G04 X1.000** M1=10 M2=11 M3=13

The dwell consists of the address letter X and the value of the dwell time.

An NC block with dwell may only contain M functions in addition to G function 04 and the time parameter.

Name	Input	Unit
Dwell	0.004 to 99.999	s

NOTE

If a block with dwell is defined as the last block in the NC program, an automatic jump back to the start of program is performed after the dwell time expires (infinite loop).

2 NC Program / Subprogram Execution

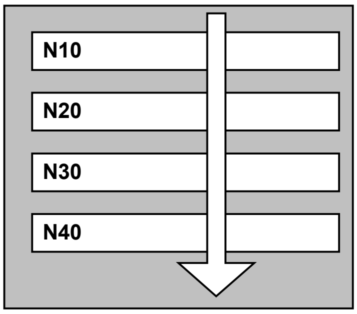
Contents

In this chapter you will find all the information about the execution of NC programs and subprograms.

2.1 NC Program Execution

Programming:

```
N10  G90  G01  X100.000  Y200.000  F4000.00  M1=10  M2=11  M3=13
N20  G90  X300.000  F3000.00
N30  G90  Y400.000  F2000.00
N40  M1=30
```

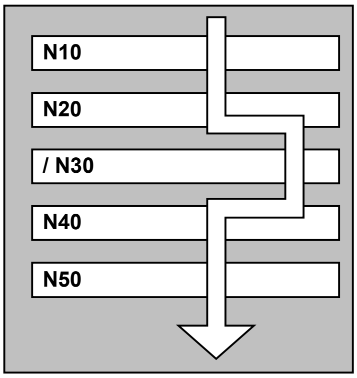


An NC program comprises a number of NC blocks, the direction of movement being determined by the ascending NC block numbers. The NC block decoder prepares the NC blocks by performing any necessary calculations or checks before executing the NC block. This achieves short block changes and execution times.

Before you can start an NC program which has already been stored, some basic conditions must be fulfilled. This includes specifying the NC block number, the read-in enable [RIE] and the start signal [STA]. While the read-in enable activates the NC block decoder, the start signal enables the actual execution of the NC blocks.

2.2 Skippable NC Blocks

Programming:
N20 G90 X1.000 F4000.00 M1=10 M2=11 M3=13
/ N30 G90 X2.000 F4000.00
N40 G90 X3.000 F4000.00
N50 M1=30



NC blocks which are not to be executed on every NC program pass can be identified as skippable NC blocks by means of the character "/". The "block skip" control signal [BLSK] determines whether or not these blocks are skipped.

NOTE You may not skip the last NC block of an NC program or subprogram.

2.3 Subprograms

Self-contained machining sequences which you need to call up or run several times can be implemented using subprograms.

Calling

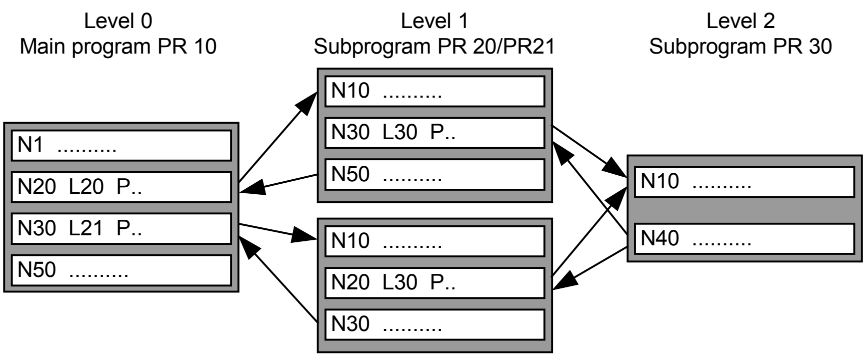
> **Programming:**
> N10 **L24 P23**

The subprogram call is identified by the address letter L, followed by a number from 1 to 200. The loop count is specified with the address letter P and a value for the loop count from 0 to 65535.

An NC block with a subprogram call must not contain any other words in addition to the loop count. If 0 is specified as the loop count, the subprogram is not called.

Execution

Two nested levels are supported, the number of subprogram calls being limited only by the memory capacity.



NOTE

If a subprogram is terminated with M18, the M18 is only output as an M function → there is no return jump function.

There is no difference between a main program and a subprogram. Like a main program, a subprogram can be started directly or as a result of a subprogram call. When a subprogram is called, it is executed block by block, starting with the first NC block number, until it jumps back to the level immediately above. The return jump is invoked by an NC block containing an end of subprogram command (M17 or M29) or when the last NC block in the subprogram has been executed. The loop count determines the number of repeats.

Modal G functions take effect in the block in which they appear. They remain active following a jump to a subprogram or a return jump to the main program.

Information for Programming

Please note the following points when using subprograms:

- You should not attempt to program more than two nested levels, otherwise the program is aborted and an error message is output.
- Calling the higher-level NC program (main program or subprogram) in the subprogram causes the maximum nesting depth to be exceeded and is therefore not allowed. Similarly, a program cannot call itself.
- An NC block change on-the-fly is not possible when changing the NC program level. An exact stop of the axes takes place before the NC program level is changed.
- If you enter a subprogram call in a skippable block, the subprogram is not executed whenever the "block skip" signal [BLSK] is enabled.
- All definitions in the main program (G functions, tool offset, zero offsets, etc.) are carried over into the subprograms.

2.4 Block Search

The "block search" function can be used to start execution of the NC program at any position. This may be necessary, for example, if the NC program was aborted and you don't want to restart it from the beginning.

An NC program abort occurs as a result of the following events:

- A specific error message
- A mode change

NOTE

An interruption in program execution caused by the cancellation of the start signal [STA] does not constitute a program abort. In this case, the NC program continues running when the start signal is enabled.

There are two types of block search: "manual block search" and "automatic block search".

Conditions

The block search must be activated in the master axis and can only be performed under the following conditions:

- Automatic mode is selected
- The NC program number of the main program (level 0) is selected
- Program execution is inactive, stopped or aborted
- The correct block search data were defined

Any operating error generates an appropriate warning. After successful activation, you can check the current block search data before the NC program starts running. After that, the block search output data are "0" (the data are cleared when the program starts running).

Program Start after Activation

Decoding of the NC program commences at the start of the program. Axis movements and the output of M functions are performed with effect from the specified entry point. Tool offsets, zero offsets, modal G functions and velocities are tracked up to the entry point.

NOTE

- If you jump to an NC block with incremental dimensions (G91), the entire traversing path is crossed even if part of the distance was already covered before the program was aborted.
- M functions in the NC block to which you have jumped are output even if they were already output before the program abort.
- If program execution is interrupted and M functions are still queued, these are cleared by the block search.

Automatic Block Search

The data of the interruption point are saved each time the program is aborted. When the automatic block search is activated, program execution resumes at the last point of interruption, that is at the NC block at which the NC program was interrupted.

In order to perform an automatic block search, a program interruption must already have occurred.

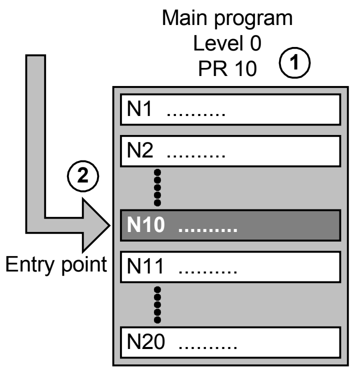
NOTE

An automatic block search cannot be performed after a RESET initiated by the "reset technology" control signal ([RST = 1)].

Manual Block Search

For a manual block search, you must provide the exact data identifying the entry point in the NC program.

Manual Block Search - Entry Point in Main Program

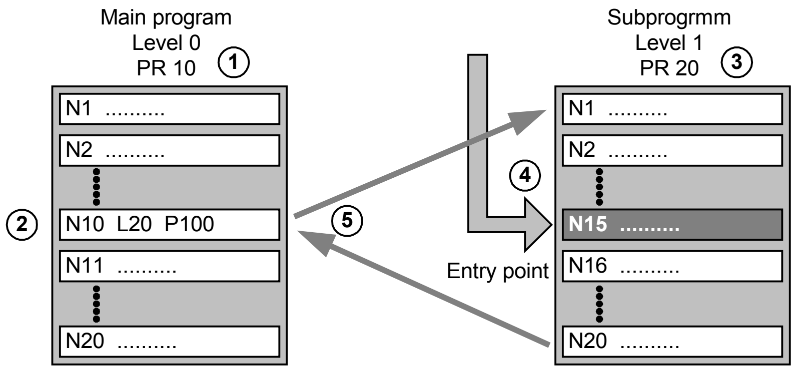


You must provide the following data:

- ① NC program number level 0 → 10
- ② NC block number level 0 at which program execution is to be resumed → 10

The remainder of the data must be initialized with "0".

Manual Block Search - Entry Point in Subprogram Level 1

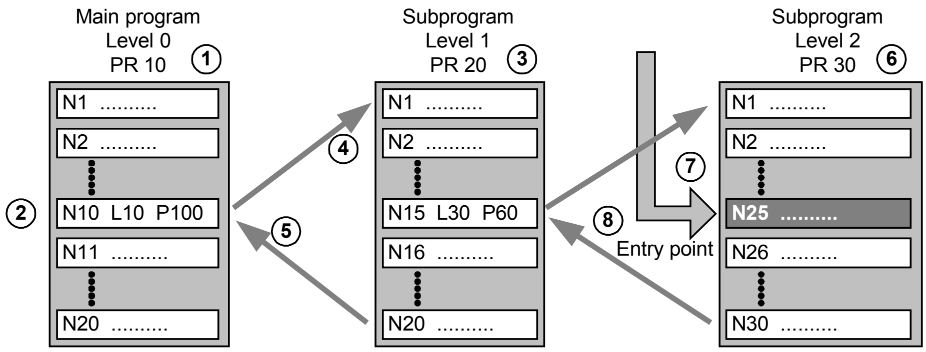


You must provide the following data:

- ① NC program number level 0 → 10
- ② NC block number level 0 at which the subprogram is called → 10
- ③ NC program number level 1 → 20
- ④ NC block number level 1 at which program execution is to be resumed → 15
- ⑤ Remaining loop count of level 1 → e.g. 20

The remainder of the data must be initialized with "0".

Manual Block Search - Entry Point in Subprogram Level 2



You must provide the following data:

- ① NC program number level 0 → 10
- ② NC block number level 0 at which subprogram level 1 is called → 10
- ③ NC program number level 1 → 20
- ④ NC block number level 1 at which subprogram level 2 is called → 15
- ⑤ Remaining loop count of level 1 → z. B. 20
- ⑥ NC program number level 2 → 30
- ⑦ NC block number level 2 at which program execution is to be resumed → 25
- ⑧ Remaining loop count of level 2 → e.g. 5

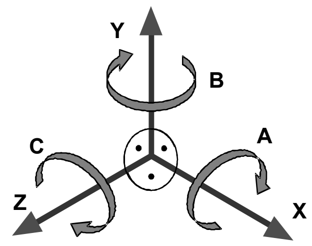
3 Programming NC Blocks

Contents

In this chapter you will find all the information about programming NC blocks.

3.1 Coordinate System and Reference Points

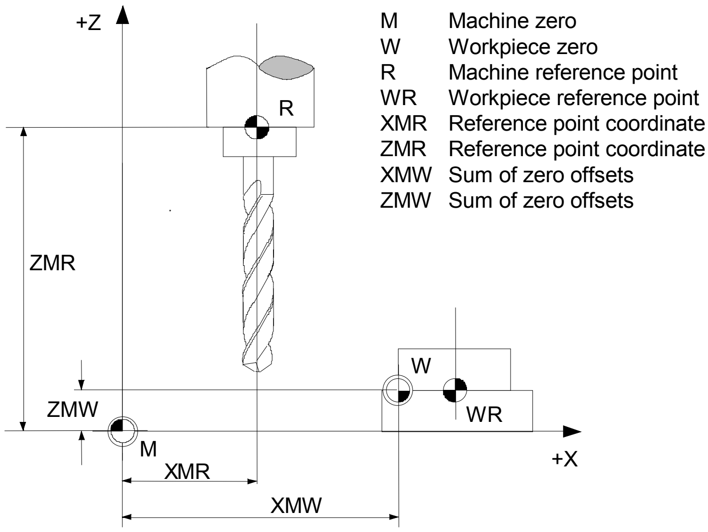
Coordinate System



The directions of movement of a machine tool can be mapped onto a coordinate system whose axes represent the axes of movement on the machine. We use a right-handed, rectangular coordinate system with the axes X, Y and Z. The coordinate system is oriented to the main axes of the machine.

Reference Points

A series of machine data is used to match the machine to the coordinate system. The data must be entered at the time of installation.



Adaptation

The specifications of DIN standard 66025 apply as a general rule. If the axis is moved in the direction of the workpiece, the actual value must become smaller (negative traversing direction). When the axis moves away from the workpiece, the actual value increases and the axis executes a positive traversing movement.

The definition of reference points varies from plant to plant, however the reference point is usually the workpiece zero. The programming instructions in the NC blocks also refer to this point. The same point is also referenced by the coordinate-specific machine data. These are:

- Reference point– coordinate (machine data 3)
- Reference point – approach direction (machine data 5)
- Software limit switches (machine data 12 and 13)

3.2 Dimensions G90, G91

The traversing movement at a specific point can be described by:

- Absolute dimensions G90 or
- Incremental dimensions G91

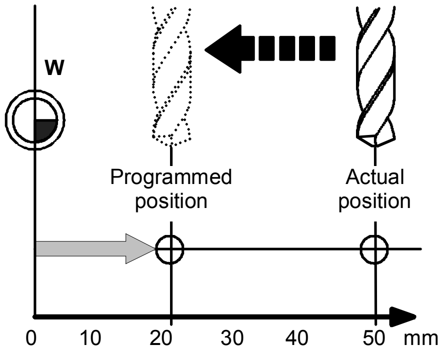
You can switch between absolute dimensions and incremental dimensions at any time. The initial setting is absolute dimension programming G90. G90 and G91 are modal. This means that you only need to program a G90 or G91 command if you want to switch the dimensioning system.

NOTE The roll feed axis type only supports G91 (incremental dimensions). The initial setting for this axis type is therefore G91.

3.2.1 Linear Axis

Absolute Dimensioning G90

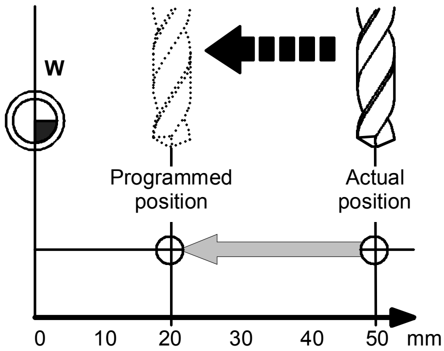
Absolute dimensions generally refer to the workpiece zero W or to the machine zero M.



Programming:
N10 G90 X20.000 FX100.00

Incremental Dimensioning G91

Incremental dimensions refer to the last actual position.



> **Programming:**
> N10 G91 X-30.000 FX100.00

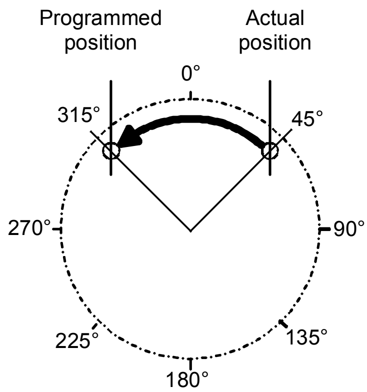
3.2.2 Rotary Axis, Shortest Path G68

If the axis is operated as a rotary axis, the measuring system should be adapted such that the subdivision of units relates to a full circle (e.g. 1°/1000). The positions which can be reached are then in the range between 0° and 360°. Any other adaptation is also possible, however.

Absolute Dimensioning G90

With a circle of 360°, absolute dimensioning (G90) has the special feature that there are always 2 ways to reach the setpoint.

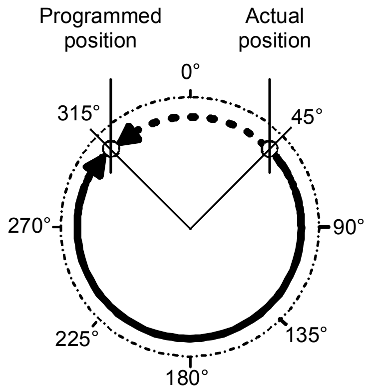
G68 - shortest path



When you program G68, the axis traverses across the shortest path to the programmed position, ignoring the sign of the position value. G68 is non-modal.

> **Programming:**
> N10 G90 G68 C315.000 FC1000.00

Programming the direction of rotation



The axis traverses to the programmed position; the sign of the position value determines the direction of movement:

- + : Movement towards increasing degree values (actual values)
- – : Movement towards decreasing degree values (actual values)

Programming:
N10 G90 C315.000 FC1000.00
(oder N10 G90 C-315.000 FC1000.00)

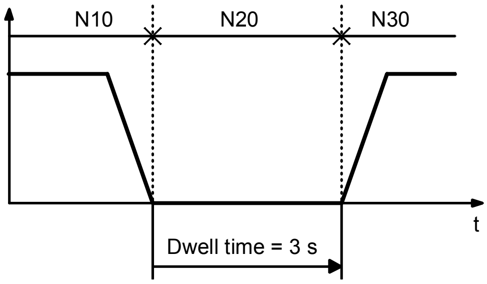
Incremental Dimensioning G91

With incremental dimension programming G91, the direction of rotation of the rotary axis is derived from the sign of the position setpoint. You can program several revolutions by specifying the position setpoint as a multiple of 360°. If G68 is programmed in combination with G91, a warning is output.

3.3 Dwell G04

Dwell times are required in order to stop the machine control for a specified time.

Programming:
```
N10 G90 G01 X50.000 F4000.00
N20 G04 X3.000
N30 G91 X2.000
```



Dwell times are only active non-modally and must be selected again if required more than once.

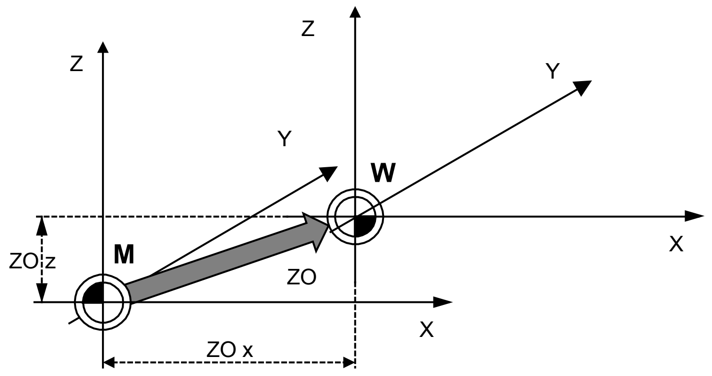
NOTE A dwell in the last NC block causes a jump back to the start of the NC program (infinite loop).

Programming:
```
N10 G90 G01 X10.000 F400.00
N20 X-2.000
N30 G04 X0.100
```

In this case, the end of the NC program is indicated by G04 and a time value. When the dwell time of 100 ms expires, there is an automatic jump back to the start of the NC program and the process is repeated.

3.4 Zero Offset G53, G54 to G59

The zero offset is the distance from the workpiece zero W to the machine zero M.



6 zero offsets are available for each axis; these are activated by G functions G54 to G59. The zero offsets in all axes can be deselected with G53.

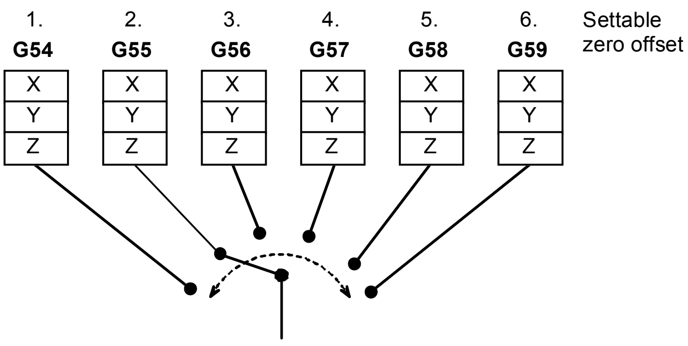
The following input limit applies to all zero offset values:

Name	Input	Unit
Zero offset	from -999 999,999 to +999 999,999	mm

NOTE

All values of the zero offsets used in an NC program must be transferred before the program is started. The absolute positions of the software limit switches are retained when the zero offset is used, because these positions refer to the machine zero.

Value Memory



Each zero offset is associated with a value memory for the axes. The zero offsets act alternately. Selecting another zero offset deactivates the previous zero offsets.

Programming

The zero offset is modal, that is it remains active until it is deselected. You can deselect a zero offset by:

- G53
- Selection of another zero offset
- End of NC program (M02, M30)
- Mode change
- NC program change
- "Reset technology" control signal [RST]

At the start of the NC program no zero offset is active. If a zero offset is needed, it must therefore be selected explicitly. Subprograms are an exception to this rule. A zero offset which is selected in the main program remains active in the subprogram. Similarly, a zero offset which is selected in the subprogram is also active in the main program.

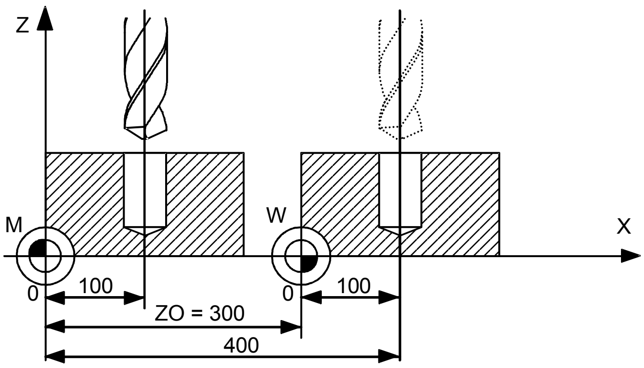
Traversing Distance

The traversing distance is calculated after allowing for all the shifts and offsets.

Absolute dimension programming G90:
ZO selection: Distance = new SP (setpoint) - old SP + ZO
ZO switchover: Distance = new SP - old SP + new ZO - old ZO
ZO deselection: Distance = new SP - old SP - ZO

Incremental dimension programming G91:
Distance = Incremental dimension

Example




```
Programming:
N10  G90  X100.000  FX5000.00
N20  G90  G54  X100.000
N30  G53
```

N10: Axis X traverses to 100.000.

N20: The G54 function is used to activate the zero offset. The zero point is shifted in the positive direction by 300 mm, and the axis traverses to position 100.000. All subsequent position values now refer to the new zero point.

N30: The zero offset is deactivated.

3.5 Chaining with Rapid Traverse G76

The path programmed with G76 "chaining with rapid traverse" is traversed at the maximum traversing velocity (machine data 23) in each axis, the resulting path being dependent on the distances to be traversed and the axis velocities.

The rapid traverse movements are programmed by entering the preparatory function G76 and specifying the target position. The target position can be specified in absolute dimensions G90 or incremental dimensions G91. G76 is modal. You can traverse one or more axes simultaneously using "chaining with rapid traverse".

NOTE

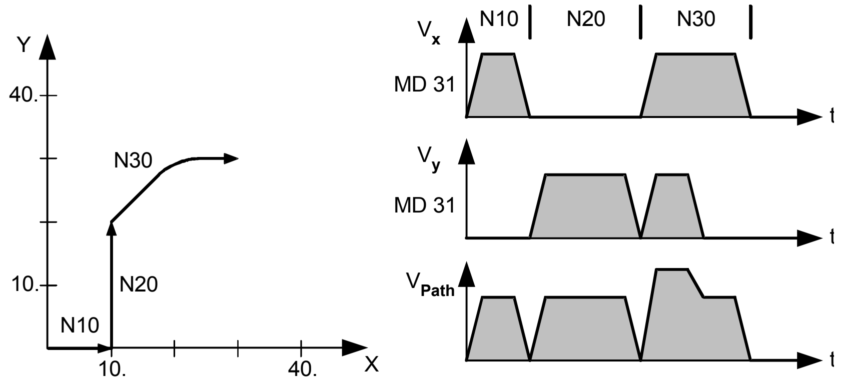
Since the override is active for "chaining with rapid traverse", the maximum traversing velocity is only attained with 100 % override.

Example

Programming:

```
N10 G76 X10.000
N20 Y20.000
N30 X30.000 Y30.000
```

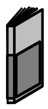
Movement in the coordinate system: v-t charts:



N10: Axis X moves with rapid traverse (MD 23 = e.g. 3000 mm/min) to position X10.000.

N20: Axis Y moves with rapid traverse (MD 23 = e.g. 3000 mm/min) to position Y20.000.

N30: Axes X and Y move with rapid traverse to position X30.000/Y30.000. Each axis traverses with its own maximum traversing velocity (MD 23). Since the distance in axis Y is shorter than the distance in axis X, axis Y reaches its destination sooner.



Further Information

In this example, the NC block transition response is assumed to be exact stop G60. The various options available for NC block transitions are described in detail in section 3.10, “NC Block Transitions”.

3.6 Chaining with Axis Velocity G77

The path programmed with G77 "chaining with axis velocity" is traversed in each axis at the programmed axis velocity, the resulting path being dependent on the distances to be traversed and the axis velocities.

"Chaining with axis velocity" is programmed by entering preparatory function G77 and specifying a target position and the axis velocities. The target position can be specified in absolute dimensions G90 or incremental dimensions G91. G77 is modal. You can traverse one or more axes simultaneously using "chaining with axis velocity".

NOTE

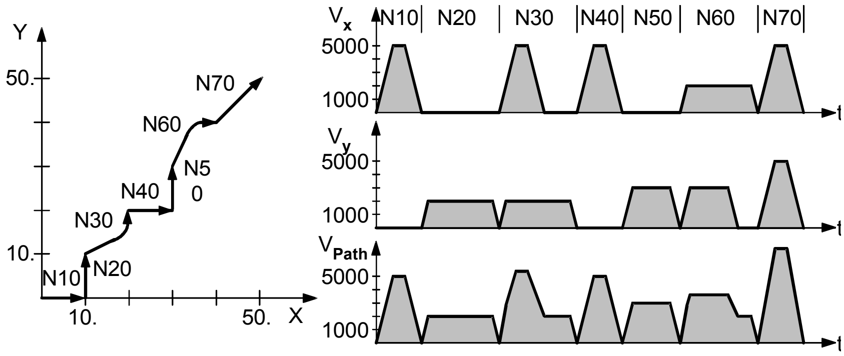
The programmed axis velocities refer to a 100 % override. Smaller or larger override values are included in the axis velocities.

Example

Programming:

```
N10  G90  G77  X10.000  FX5000.00
N20  Y10.000  FY2000.00
N30  X20.000  Y20.000
N40  G91  X10.000  FY3000.00
N50  Y10.000  FX2000.00
N60  G90  X40.000  Y40.000
N70  X50.000  Y50.000  FX5000.00  FY5000.00
```

Movement in the coordinate system: **v-t charts:**



N10: Axis X traverses in absolute dimensions G90 at its axis velocity FX5000.00 to position X10.000.

N20: Axis Y traverses in absolute dimensions G90 at its axis velocity FY2000.00 to position Y10.000.

N30: Axes X and Y traverse together in absolute dimensions to position X20.000/Y20.000, each axis moving at its own axis velocity FX5000.00/FY2000.00. Since axis X travels at a greater axis velocity across an identical distance, this axis reaches its destination first.

N40: Axis X traverses in incremental dimensions G91 by X10.000 at its axis velocity FX5000.00. The new velocity value FY3000.00 is also assigned to axis Y.

N50: Axis Y traverses in incremental dimensions G91 by Y10.000 at the new axis velocity FY3000.00 assigned in N40. The new velocity value FX2000.00 is also assigned to axis X.

N60: Axes X and Y traverse together in absolute dimensions G90 to position X40.000/Y40.000 at the axis velocities FX2000.00/FY3000.00. Since axis Y travels at a greater axis velocity across an identical distance, this axis reaches its destination first.

N70: Axes X and Y traverse together in absolute dimensions G90 at their new axis velocities FX5000.00/FY5000.00 to position X50.000/Y50.000. Since both axes move across identical distances at identical velocities, the resulting path is a straight line.

Further Information

In this example, the NC block transition response is assumed to be exact stop G60. The various options available for NC block transitions are described in detail in section 3.10, “NC Block Transitions”.

3.7 Interpolation with Rapid Traverse G00

The path programmed with G00 "interpolation with rapid traverse" is traversed at the greatest possible velocity across a straight line. The rapid traverse movements are programmed by entering preparatory function G00 and specifying the target position. The target position can be specified in absolute dimensions G90 or incremental dimensions G91.

If the rapid traverse movement is performed simultaneously in several axes, each axis is monitored to prevent any of the axes from traversing at a rate in excess of its own maximum traversing velocity (machine data 23).

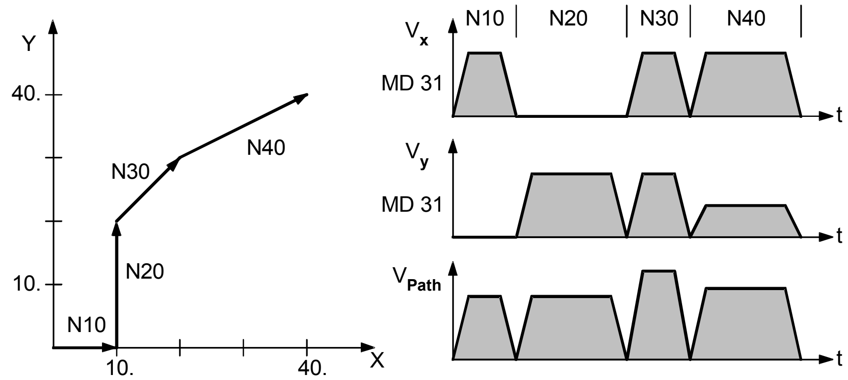
The initial state (default setting) is G00. G00 is modal. One or two axes can be traversed across a straight line with rapid traverse using "interpolation with rapid traverse".

NOTE Since the override is active for "interpolation with rapid traverse", the maximum traversing velocity is only attained with 100 % override.

Example

Programming:
N10 G90 (G00) X10.000
N20 Y20.000
N30 G91 X10.000 Y10.000
N40 G90 X40.000 Y40.000

Movement in the coordinate system: **v-t charts:**



N10: Axis X moves in absolute dimensions G90 with rapid traverse to position X10.000. G00 does not have to be specified, since it is active in the default setting.

N20: Axis Y moves in absolute dimensions G90 with rapid traverse to position Y20.000.

N30: Axes X and Y move in incremental dimensions G91 with rapid traverse across the distance X10.000/Y10.000. Since both axes have to cross the same distances, both axes can be traversed at their maximum traversing velocities.

N40: Axes X and Y move in absolute dimensions G90 with rapid traverse to position X40.000/Y40.000. Axis X has a longer distance to cross and thus moves at the maximum traversing velocity. Axis Y has a shorter distance to cross and must therefore reduce its velocity such that it reaches its destination at the same time as axis X.

Further Information

In this example, the NC block transition response is assumed to be exact stop G60. The various options available for NC block transitions are described in detail in section 3.10, "NC Block Transitions".

3.8 Interpolation with Path Velocity G01

The path programmed with G01 "interpolation with path velocity" is traversed across a straight line at the programmed path velocity.

The "interpolation with path velocity" is programmed by entering preparatory function G01 and specifying a target position and the path velocity. The target position can be specified in absolute dimensions G90 or incremental dimensions G91.

G01 is modal. One or two axes can be traversed across a straight line using "interpolation with path velocity".

NOTE

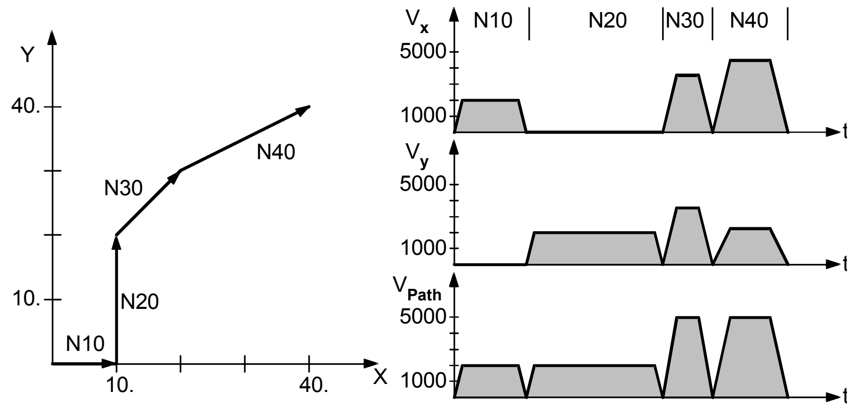
The programmed path velocities refer to a 100 % override. Smaller or larger override values are included in the axis velocities.

Example

Programming:

```
N10  G90  G01  X10.000  F2000.00
N20  Y20.000
N30  G91  X10.000  Y10.000  F5000.00
N40  G90  X40.000  Y40.000
```

Movement in the coordinate system: **v-t charts:**



N10: Axis X traverses in absolute dimensions G90 at the path velocity F2000.00 to position X10.000.

N20: Axis Y traverses in absolute dimensions G90 at the path velocity F2000.00 to position Y20.000.

N30: Axes X and Y traverse in incremental dimensions G91 across the distance X10.000/Y10.000. Since both axes cross the same distances, the axis velocities are the same.

N40: Axes X and Y traverse in absolute dimensions G90 to position X40.000/Y40.000. Since axis X has a longer distance to cross, it traverses at a greater axis velocity than axis Y.

Further Information

In this example, the NC block transition response is assumed to be exact stop G60. The various options available for NC block transitions are described in detail in section 3.10, “NC Block Transitions”.

3.9 Collision Monitoring G96, G97

The collision monitoring function can be used to stop the axes quickly in response to the actuation of a digital input.

- **G96: Select collision monitoring**
  Collision monitoring is activated when an NC block with G96 is read in
  → Scanning of the digital input is enabled
- **G97: Deselect collision monitoring**
  Collision monitoring is deactivated when an NC block with G97 is read in.
  → The digital input is no longer scanned

G97 is active in the default setting. G96 and G97 are modal.

Definition of Deceleration Value

The deceleration for a collision can be set separately for each axis in machine data 20.

MD No.	I	A	W	Description	Input Limit		Unit	Default
					Lower	Upper		
20				Deceleration for collision	1	99 999	1000*LU/s²	1000

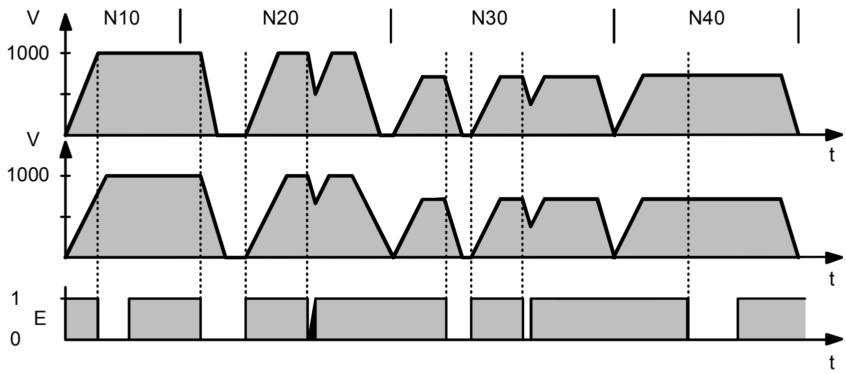
The digital input can be defined with machine data 45.

MD No.	I	A	W	Description	Input Limit		Unit	Default
					Lower	Upper		
45				Digital inputs – Function 1 (U501.45)	0: No function 1: Start ORed 2: Start ANDed 3: Set actual value on-the-fly 4: External NC block change 5: Inprocess measurement 6: Collision 7: Proximity switch for reference point approach 8: Reverse cam for reference point approach 9: External read-in enable, programmable		6-digit input	0

Example

Programming:
N10 G77 X100.000 Y100.000 FX1000.00 FY1000.00
N20 G96 X200.000 Y200.000
N30 G01 X300.000 Y300.000 F1000.00
N40 G97 X500.000 Y500.000

v-t charts:



N10: Since collision monitoring is not yet active in N10, disabling the high-speed input does not stop the axes.

N20: Collision monitoring is activated by G96 in N20. Since chaining is active, the cancellation of the signal causes the axes to decelerate to a standstill at the axis-specific rates defined in "deceleration for collision". When the signal is activated, the axes start up with the acceleration values. If the signal is canceled for a short period only, the axes decelerate at the rate defined in "deceleration for collision". They are subsequently started "on-the-fly" with the acceleration values.

N30: The interpolation is activated in N30. When the signal is canceled, the axes are stopped without violating the interpolation conditions (they must not leave the contour). In this case, the "deceleration for collision" of the "weaker" axis (deceleration value is smaller) is taken as the maximum permissible deceleration (axis Y in this example). When the signal is activated, the axes start up with the acceleration values. If the signal is canceled for a short period only, the axes decelerate at the "deceleration for collision" rate defined for the "weaker" axis without violating the interpolation conditions. They are subsequently started "on-the-fly" with the acceleration values.

N40: Since the collision monitoring is deactivated by G97 in N40, cancellation of the signal does not stop the axes.

3.10 NC Block Transitions

NC programs are executed in the programmed sequence of the NC blocks. Various NC block transitions are produced as a result of the content of the NC blocks.

3.10.1 Maximum Corner Rounding G64

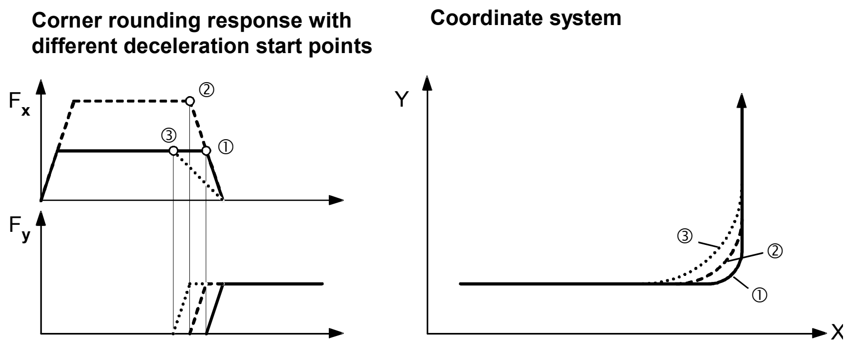
G64 is active in the default setting, and is modal. Maximum corner rounding means that consecutive NC blocks are executed in the shortest possible time with the lowest possible reduction in velocity. This makes it possible to achieve an extremely rapid sequence of movements.

The "deceleration start point" plays an important role during the block transition, because it determines the earliest possible time at which the block change can take place. The deceleration start point is the time at which the axis would have to start decelerating in order to stop at the programmed destination or in order to reach the new velocity. The deceleration start point thus depends on the selected deceleration (deceleration ramp) and the current velocity.

Corner Rounding Response with G64

Programming:

```
N10  X100.000
N20  Y100.000
```



① Deceleration start point at low velocity → low corner rounding
② Deceleration start point at high velocity → higher corner rounding
③ Deceleration start point for flat deceleration ramp → high corner rounding

NOTE

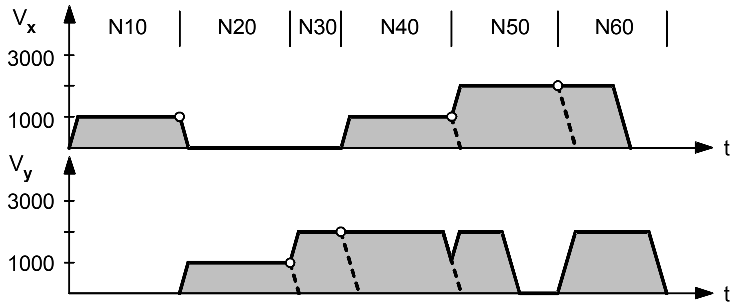
Maximum corner rounding (NC block change at deceleration start point) is only possible in association with chaining (G76, G77).

Chaining (G76, G77) with Maximum Corner Rounding (G64)

Programming:

```
N10  G90  G77  X100.000  FX1000.00
N20  Y100.000  FY1000.00
N30  Y200.000  FY2000.00
N40  X200.000  Y385.000
N50  X400.000  Y485.000  FX2000.00
N60  X500.000  Y650.000
```

v-t charts:



N10: Axis X traverses at velocity FX1000.00 to position X100.000.

N20: Axis Y is started at the deceleration start point of axis X. It traverses at velocity FY1000.00 towards position Y100.000.

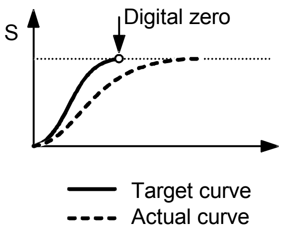
N30: At the deceleration point, axis Y accelerates to its new velocity FY2000.00.

N40: While axis Y is approaching the new position Y385.000 with no reduction in velocity, axis X is started. Since this axis needs more time to reach its destination, axis Y starts to decelerate.

N50: Axes X and Y accelerate to their respective velocities. Since axis Y reaches its destination sooner, it remains at this position and waits until axis X has reached the deceleration start point.

N60: Axis Y starts to accelerate again at the deceleration start point of axis X. Both axes traverse to their destination and stop.

Interpolation (G00, G01) with Maximum Corner Rounding (G64)

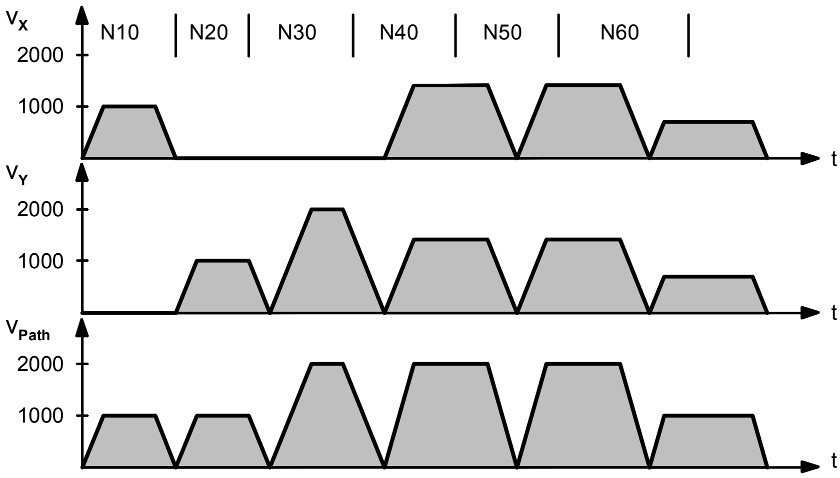


Corner rounding at the deceleration start point is not possible with interpolation, because the axis is not allowed to leave the programmed contour. With interpolation, G64 is always traversed to "digital zero". Traversed to "digital zero" means that the interpolator is required to have reached the programmed position.

Programming:

```
N10  G90  G01  X100.000  F1000.00
N20  Y100.000
N30  Y300.000  F2000.00
N40  X300.000  Y500.000
N50  X500.000  Y700.000
N60  X600.000  Y800.000  F1000.00
```

v-t charts:



N10: Axis X traverses at velocity F1000.00 to position X100.000.

N20: Axis Y is started at "digital zero" in axis X. Axis Y traverses at the velocity F1000.00 to position Y100.000.

N30: At "digital zero", axis Y accelerates to the new velocity F2000.00.

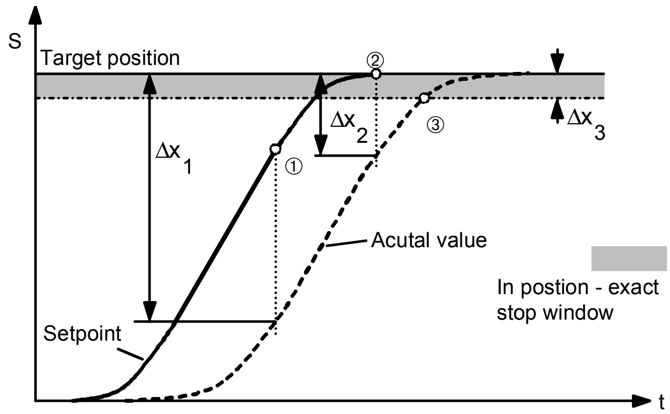
N40: At "digital zero" in axis Y, axes X and Y are accelerated in interpolating mode to the path velocity F2000.00.

N50: The NC block change also occurs at "digital zero".

N60: At "digital zero", axes X and Y are traversed at the new path velocity F1000.00 to positions X600.000 Y800.000.

3.10.2 Exact Stop G60

Exact stop G60 must be programmed whenever programmed positions are to be approached exactly. G60 is modal. When you program a G60 command, program execution is suspended until all axes involved in the movement have reached their respective positions and stop window (machine data 17).



① Maximum corner rounding occurs on a block change at the deceleration start point.
② Low corner rounding is achieved with traversing to "digital zero".
③ With exact stop G60, the axis approaches the exact position, because execution of the NC program is suspended until the actual value has reached the exact stop window.

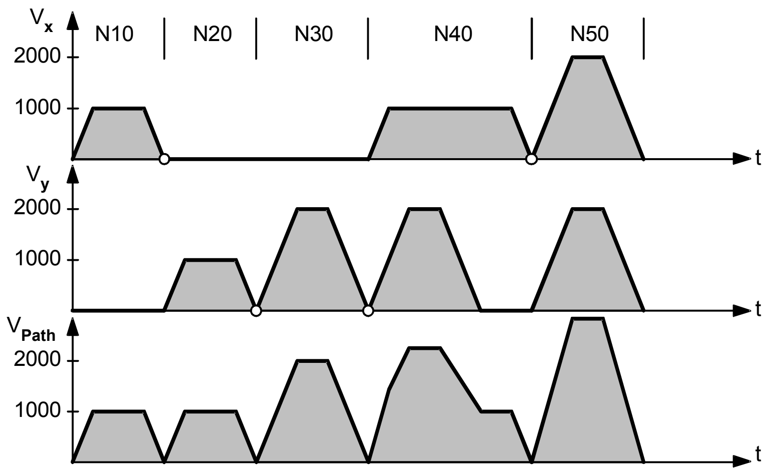
NOTE

If the position is reached and the stop window is greater than the following error at the time of the "digital zero" (②), the NC block change takes place at "digital zero".

Chaining (G76, G77) with Exact Stop (G60)

Programming:
N10 G90 G77 G60 X100.000 FX1000.00
N20 Y100.000 FY1000.00
N30 Y300.000 FY2000.00
N40 X300.000 Y500.000
N50 X500.000 Y700.000 FX2000.00

v-t charts:



N10: Axis X traverses at velocity FX1000.00 to position X100.000.

N20: When axis X has reached its position and exact stop window, axis Y traverses to position Y100.000.

N30: When axis Y has reached its position and exact stop window, it continues to position Y300.000.

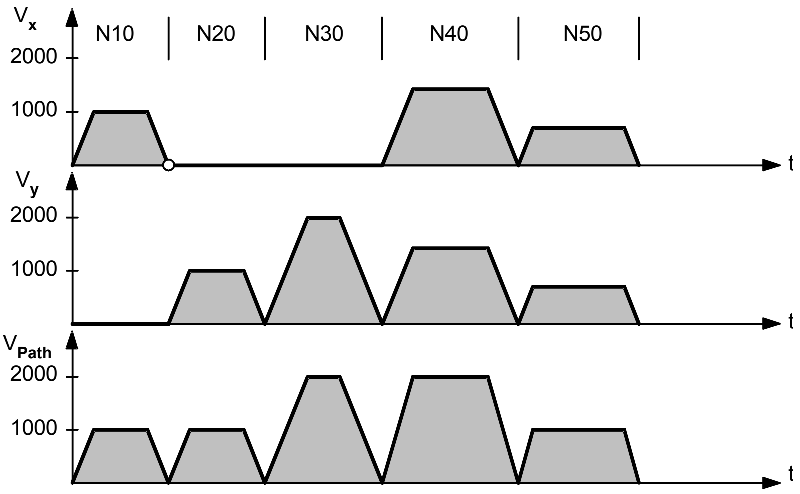
N40: When axis Y has reached its position and exact stop window, axes X and Y traverse to positions X300.000 and Y500.000. Since axis Y reaches its destination first, it stops and waits for axis X.

N50: When both axes have reached their respective positions and exact stop window, they traverse to their new positions X500.000 and Y700.000.

Interpolation (G00, G01) with Exact Stop (G60)

Programming:
N10 G90 G01 G60 X100.000 F1000.00
N20 Y100.000
N30 Y300.000 F2000.00
N40 X300.000 Y500.000
N50 X400.000 Y600.000 F1000.00

v-t charts:



N10: Axis X traverses at velocity F1000.00 to position X100.000.

N20: When axis X has reached its position and exact stop window, axis Y traverses to position Y100.000.

N30: When axis Y has reached its position and exact stop window, it continues to position Y300.000.

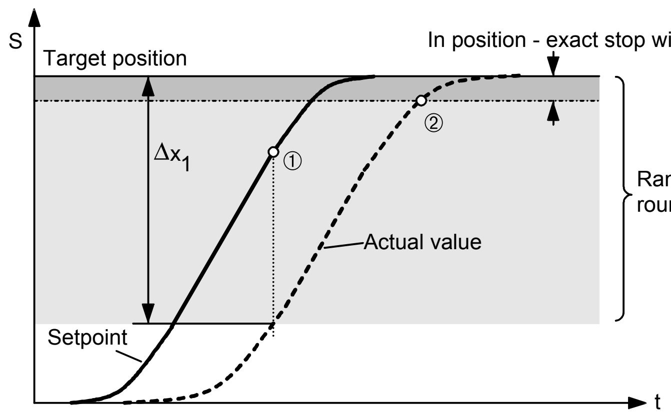
N40: When axis Y has reached its position and exact stop window, axes X and Y traverse in interpolating mode to positions X300.000 and Y500.000.

N50: When axes X and Y have reached their respective positions and exact stop window, they continue to positions X400.000 and Y600.000.

3.10.3 Corner Rounding Window 1 or 2 (G66, G67)

The NC block change at deceleration start point G64 does produce rapid sequences of movement, however the high corner rounding factor results in a large deviation from the programmed path. The NC block change at exact stop G60 allows the exact positions to be approached, however this takes place at the expense of an increase in the positioning time, because the movement takes the axes into the exact stop window and stops them.

To allow greater flexibility in programming, two corner rounding windows are available. Two dimensions between the deceleration start point ① and exact stop ② can be defined for corner rounding. G66 activates corner rounding window 1, G67 activates corner rounding window 2. G66 and G67 are modal.



NOTE

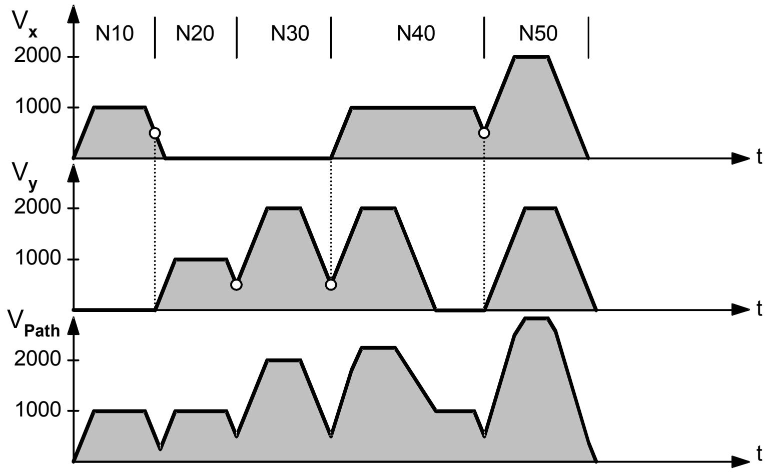
If the corner rounding window is greater than the actual value at the deceleration start point (①), the block change takes place at the deceleration start point. If the corner rounding window is less than the exact stop window (②), the block change takes place, as in the case of exact stop G60, when the exact stop window is reached.

Chaining (G76, G77) with Corner Rounding (G66, G67)

Programming:

```
N10  G90  G77  G66  X100.000  FX1000.00
N20  Y100.000  FY1000.00
N30  Y300.000  FY2000.00
N40  X300.000  Y500.000
N50  X500.000  Y700.000  FX2000.00
```

v-t charts:



N10: Axis X traverses at the velocity FX1000.00 to position X100.000.

N20: Since no axis change takes place on the transition from N20 to N30, that is the corner rounding refers to the same axis, the rounding is performed with no reduction in velocity (as with G64).

N30: Once the velocity of axis Y has been reduced until the actual value has reached corner rounding window 1, axis Y accelerates again.

N40: Axis X is also started in corner rounding window 1 of axis Y. Axes X and Y traverse together to positions X300.000 and Y300.000. Since axis X takes longer to cover its path, axis Y remains at a standstill.

N50: Axis Y is also started in corner rounding window 1 of axis X.

NOTE Interpolation (G00, G01) with Corner Rounding (G66, G67)

Corner rounding in corner rounding window 1 / 2 is only possible with interpolation if corner rounding window 1 / 2 is smaller than the rounding value resulting at "digital zero". This means it is not possible to increase the corner rounding response in excess of "digital zero" by increasing the size of the corner rounding window.

3.10.4 Path-Dependent Chaining (G63)

In certain applications you need to start an axis depending on the current position of another axis without stopping this axis. You can only achieve this with path-dependent chaining G63. G63 is modal.

Path-dependent chaining is not possible in the following circumstances:

- With interpolation (G00, G01)
- On a change between interpolation and chaining
- When using the special function "set actual value on-the-fly" and "external NC block change"
- If the next NC block only contains other axes
- If all programmed axes are in motion at the time of the NC block change (deceleration start point)

The following table shows the many possible combinations of NC block transitions and their effect in association with G63:

From ↓ To →	X	Y	Z	XY	XZ	YZ	XYZ
X	a	a	a	b	b	A	b
Y	a	a	a	b	a	B	b
Z	a	a	a	a	b	B	b
XY	a	a	a	c	b	B	b (Z) / c
XZ	a	a	a	b	c	B	b (Y) / c
YZ	a	a	a	b	b	C	b (X) / c
XYZ	a	a	a	c	c	C	c

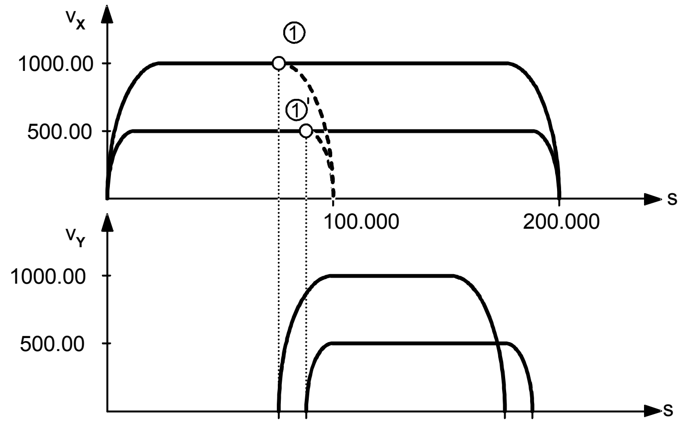
a G63 has the same effect as G64

b G63 causes path-dependent chaining

c G63 causes path-dependent chaining for the axis (axes) which has (have) come to a standstill at the deceleration start point of the axis (axes) which continue to move. G63 has the same effect as G64 in all other respects.

Programming with G64 (Maximum Corner Rounding)

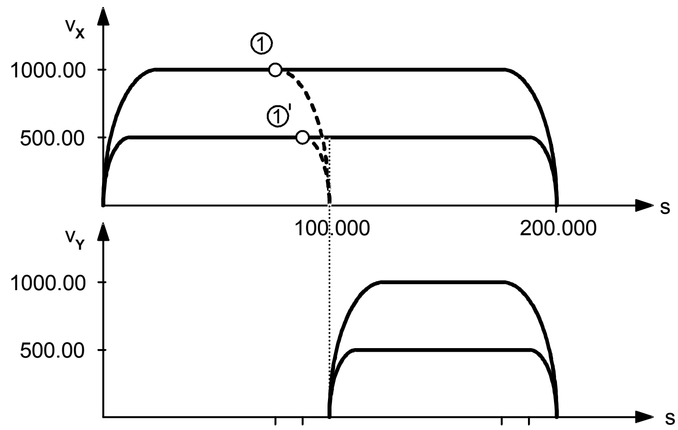
Programming:
N10 G77 G64 X100.000 FX1000.00 (FX500.00)
N20 X200.000 Y100.000 FY1000.00 (FY500.00)



With G64, the Y axis is started at the deceleration start point ① of the X axis. Since the deceleration start point depends on the magnitudes of the velocity, deceleration and/or time override. the coupling with the Y axis takes place at different positions in the X axis.

Programming with G63 (Position-Dependent Chaining)

Programming:
N10 G77 G63 X100.000 FX1000.00 (FX500.00)
N20 X200.000 Y100.000 FY1000.00 (FY500.00)



With G63, coupling of the Y axis is delayed until the actual value of the X axis has reached the programmed position value (100.000 in the example). The magnitudes of velocity, deceleration and time override therefore have no influence on the coupling. The coupling can only be delayed by the cycle time of the interpolator.

NOTE

Function G63 is derived from G64. This means that if path-dependent chaining is not possible, G63 has the same effect as G64.

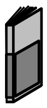
3.10.5 NC Block Transitions with Different Types of Coupling

The NC block transitions between different types of coupling always take place at "digital 0". That means the axes decelerate until the setpoint interpolator has reached its target.

Programming:
N10 G01 X100.000 Y100.000 F1000.00 → Interpolation
N20 G77 X200.000 Y200.000 FX1000.00 FY1000.00 → Chaining

3.10.6 M Functions

The M functions are optional components of an NC block. Different NC block transitions result from the time of the M function output.



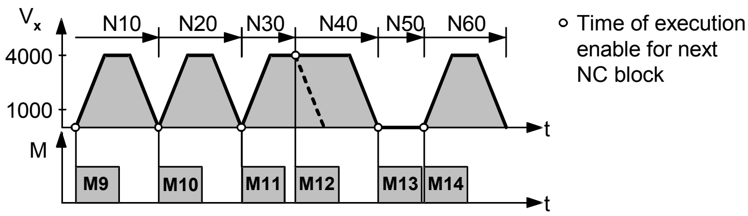
Further Information

The output of M functions is described in the Function Description in chapter 2.4.14 "M Function Number".

M Function Output During Positioning

The M function output and the traversing movement take place when the NC block is enabled for execution.

Programming:
N10 G01 X100.000 F4000.00 M1=9
N20 X200.000 M1=10
N30 G77 X300.000 FX4000.00 M1=11
N40 X400.000 M1=12
N50 M1=13
N60 X500.000 M1=14



The use of G60 (exact stop) or G66/G67 (corner rounding window 1/2) affects the execution enable of the NC block and thus the M function output.

N10: M function M9 is output when the axis is started.

N20: Since interpolation G01 is active, the NC block change from N10 → N20 takes place at "digital 0". M10 is thus output at "digital 0".

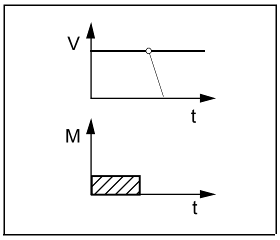
N30: A transition of coupling type from interpolation G01 to chaining G77 takes place. As a result, the output of M11 also takes place at "digital 0".

N40: The NC block transition from N30 → N40 is on-the-fly. The NC block change and hence the output of M12 therefore take place at the theoretical deceleration start point.

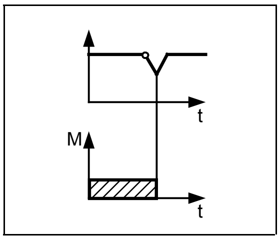
N50: Since only one M function is programmed in NC block N50, the output of M13 takes place at "digital 0" of NC block N40.

N60: N60 is read in and M14 is output simultaneously at the end of the M13 M function output.

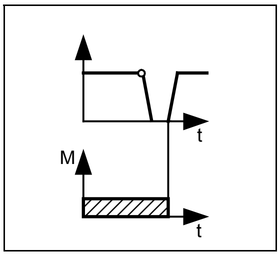
M Function Output During Positioning - Response with Block Change On-the-Fly



The M function output is complete before the deceleration start point for the block position has been reached. It is thus possible to perform an NC block change without reducing the velocity.



The M function output has not yet been completed at the deceleration start point. The axis starts to decelerate. If the M function output is completed during the deceleration process, the NC block change is performed and the axis starts to accelerate again.

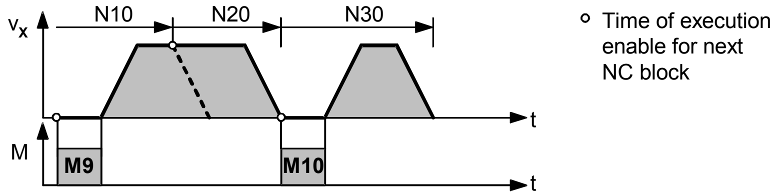


The axis comes to a complete standstill. The next NC block is not executed until the M function output has been completed.

M Function Output Before Positioning

The M function output takes place when the NC block is enabled for execution. The traversing movement takes place after the M function output.

Programming:
N10 G76 X100.000 M1=9
N20 X200.000
N30 X300.000 M1=10



The use of G60 (exact stop) or G66/G67 (corner rounding window 1/2) affects the execution enable of the NC block and thus the M function output.

N10: M9 is output when the axis is started, the traversing movement takes place after this.

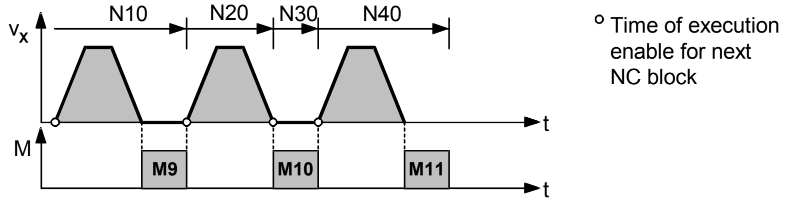
N20: The NC block transition from N10 → N20 is on-the-fly.

N30: Since "before positioning" is defined as the output type, M10 is output first, followed by the traversing movement of N30.

M Function Output After Positioning

The traversing movement takes place when the NC block is enabled for execution. The traversing movement takes place after the M function output.

Programming:
N10 G76 X100.000 M1=9
N20 X200.000
N30 M1=10
N40 X300.000 M1=11



If several axes are programmed in the NC block, the M function output takes place when all axes have reached "digital 0" or the exact stop window (with G60) or the corner rounding window (with G66/G67).

N10: The traversing movement takes place on "start". M9 is output when "digital 0" is reached. When G60 (exact stop) is programmed, M9 is not output until the exact stop window is reached. When G66/G67 (corner rounding window 1/2) is programmed, the output takes place in corner rounding window 1/2 if "digital 0" has already been reached.

N20: Traversing movement only.

N30: Since only one M function was programmed in NC block N30, the NC block change from N20 → N30 only takes place at "digital 0" for N20.

N40: M11 is output after the traversing movement.

M function output depending on actual value

In programming mode “M function output depending on actual value“ the output of the M function is delayed until the actual position (corresponding to the set position from the previous NC set) has been passed.

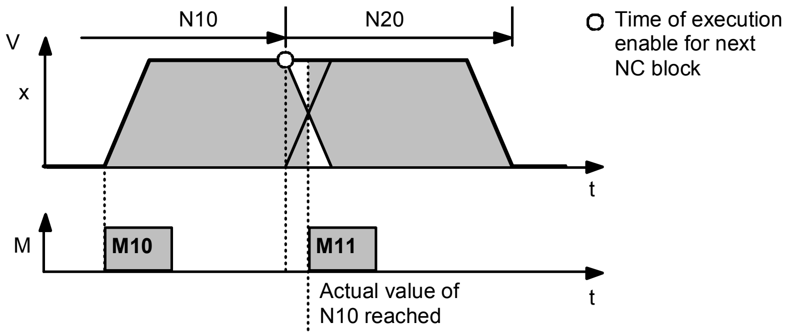
NOTE

"M function output depending on actual value" is available from firmware version V1.32 onwards.

"M function output depending on actual value“ is only effective during:

- NC block change on-the-fly with G76 and G77
- corner rounding functions G64, G66 and G67

Programming:
N10 G01 G77 X100.000 M1=10
N20 X200.000 M1=M11



The use of G64, G66 or G67 (corner rounding) influences the enabling of the NC block execution and therefore the M function output.

N10: Start initiates the output of M10 and the NC movement.

N20: The NC block transition N10 → N20 occurs on-the-fly. Output of M11 occurs when the actual position (corresponding to the set position from the previous NC set) is passed.

3.10.7 External NC Block Change (G50, G51)

The "external NC block change" function is used to perform a block change on-the-fly as a result of the actuation of a digital input. It is also possible to read out the actual value (inprocess measurement).

G50: Absolute dimensions with external NC block change

G51: Incremental dimensions with external NC block change

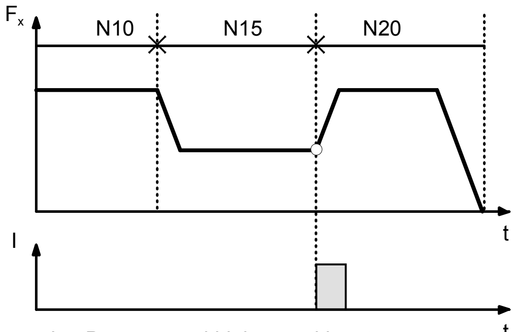
G functions G50 and G51 are non-modal and must be selected again if necessary.

NOTE

The digital input used with the "external NC block change" function must be defined in machine data 45 of the axis programmed in the NC block..

Example

Programming:
N10 G90 G77 X100.000 FX4000.00
N15 G50(G51) X200.000 FX2000.00
N20 G90 X300.000 FX4000.00



N15: The transition from N10 to N15 takes place on-the-fly.

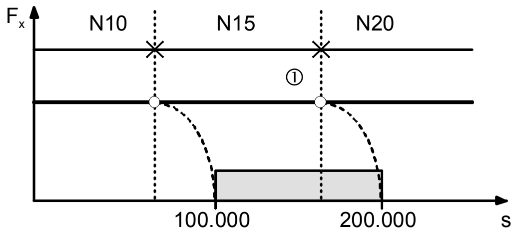
N20: The axis only traverses until a signal from 0 to 1 (24 V) takes place on the digital input. This triggers two reactions:

- NC block change on-the-fly and thus immediate execution of NC block N20.
- Saving of the actual value of the position encoder. The task "GMC status data output – identifier 8" can be used to read out the encoder actual value.

Response Depending on MD44

Machine data 44 (external NC block change – settings) can be used to determine the response of the axis when the digital input is not actuated.

Machine data 44 = 1: Warning at end of NC block (default)



The axis approaches the appropriate target position (position X200.000 for G50 or position X300.000 for G51). Within the area shaded in gray, a positive edge at the "digital input" triggers an NC block change. If the "digital input" has not been actuated by the time the deceleration start point (1) is reached, the axis starts to decelerate and waits for the signal change until the programmed position is reached. When the target position is reached, a warning is output and execution of the NC program is aborted. Queued M functions are deleted. In order to continue the NC program with the next NC block, you must initiate a block search after the warning has been cleared, otherwise the NC program resumes execution at the beginning of the program following the "start" command.

Machine data 44 = 2: No warning at end of NC block

If the "digital input" has not been actuated by the time the deceleration start point is reached, an NC block change takes place on-the-fly, and the NC program continues running without interruption. In this situation, G50 has the same effect as G90, and G51 has the same effect as G91.

NOTE Example, Special Situations

The special situations below refer to the example above.

- ♦ The NC block position in N20 is less than the actual position value at the time the "digital input" is activated:
  The axis must stop via the deceleration ramp in order to approach the position in the opposite direction.
- ♦ G91 is programmed instead of G90 in NC block N20:
  The position programmed in N20 refers to the actual position value at the time the "digital input" is activated (inprocess measurement).
- ♦ NC block N20 contains only M functions or a dwell:
  The axis is stopped via the deceleration ramp.

- NC block N20 is the end of the NC program:
  The axis is stopped via the deceleration ramp.
- NC block N15 contains an M function which has not yet been acknowledged:
  The traversing movement is stopped via the deceleration ramp, but the NC block change does not take place until the M function has been acknowledged.

Restrictions

- Only one axis can be programmed in an NC block with G50 or G51.
- Only one axis (the same axis) can be programmed in the subsequent NC block.
- In order to achieve a NC block change on-the-fly (without stopping the axis) between a block with G50/G51 and the following block, G76 (chaining with rapid traverse) or G77 (chaining with axis velocity) and G64 (maximum corner rounding) must be active.

3.10.8 Set Actual Value On-the-Fly (G87, G88, G89)

The function "set actual value on-the-fly" is triggered by a "digital input", the NC block change being performed on-the-fly and the actual position value being set simultaneously to a new dimension. It is also possible to read out the actual position value which was saved at the time the "digital input" was activated.

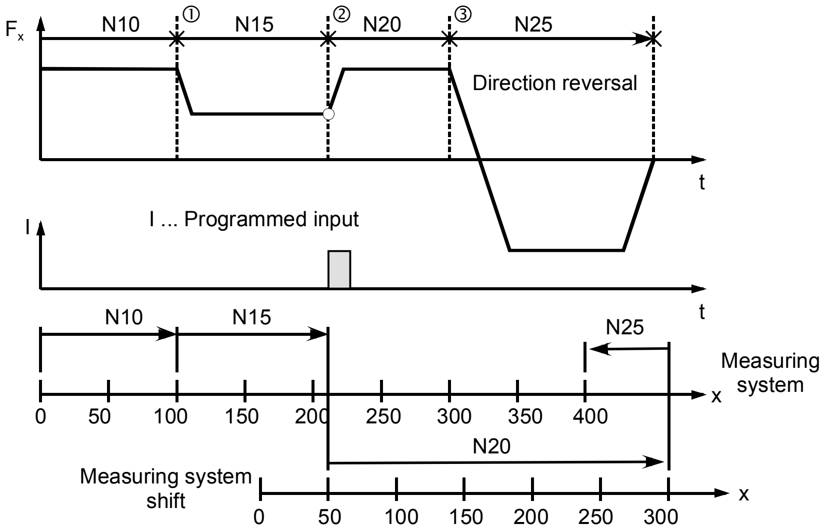
NOTE

The digital input used with the "set actual value on-the-fly" function must be defined in machine data 45 of the axis programmed in the NC block.

G functions G87, G88 and G89 are non-modal and must be selected again if necessary.

Example

Programming:
N10 G90 G77 X100.000 FX4000.00
N15 G89(G88) X50.000 FX2000.00
N20 G90 X300.000 FX4000.00
N25 G87 X400.000 FX4000.00



N15: NC block change on-the-fly from N10 to N15, with G89 causing a movement in the positive direction and G88 causing a movement in the negative direction with the programmed velocity of N15.

N20: The axis now traverses in the specified direction until a positive edge change occurs at the "digital input". This triggers the following reactions:

- ♦ Block change on-the-fly and immediate execution of NC block N20
- ♦ Set actual value on-the-fly to the block position in N15 and a consequent shift in the measuring system
- ♦ Saving of the actual position value. The task "GMC status data output – identifier 8" can be used to read out the actual value saved from the position encoder.
- ♦ The programmed position in NC block N20 refers to the shifted measuring system.

N25: NC block change on-the-fly from N20 to N25. G87 cancels the measuring system shift, and absolute dimensions are used for the NC block position in N25, since G90 is still active.

A measuring system shift initiated by G88/G89 remains active until it is deselected by G87 or a mode change.

It is possible to use the existing measuring system shift in different NC programs.

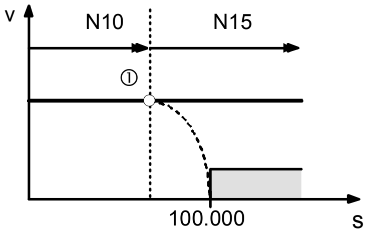
The measuring system can be shifted again without deselecting an existing measuring system shift. If the measuring system is shifted several times without deselecting the existing shift, the values refer to the original measuring system state once the measuring system shift is deselected.

NOTE
Example, Special Situations

The special situations refer to the example above.

- ♦ NC block N20 contains only M functions or a dwell:
  The axis is stopped via the deceleration ramp.
- ♦ NC block N20 is the end of the NC program:
  The axis is stopped via the deceleration ramp.

Measuring Range of High-Speed Input



Within the area shaded in gray, a positive edge at the "digital input" triggers a "set actual value on-the-fly" operation. The NC block change from N10 to N15 takes place at the time ① (deceleration start point). Enabling of the "digital input" detection is delayed until position 100.000 is crossed.

WARNING

If the "digital input" is not actuated, the axis continues traversing until it reaches the software limit switches.

Restrictions

- ♦ Only one axis can be programmed in an NC block with G87, G88 or G89.
- ♦ Only one axis (the same axis) can be programmed in the subsequent NC block.
- ♦ In order to achieve a block change on-the-fly (without stopping the axis) between a block with G88/G89 and the following block, G76 (chaining with rapid traverse) or G77 (chaining with axis velocity) and G64 (maximum corner rounding) must be active.
- ♦ It is not possible to use set actual on-the-fly with the roll feed axis type.

3.10.9 Read-In Enable, Externally Programmable (G99)

You can use the function "read-in enable, externally programmable" to stop or continue running the program with a "digital input".

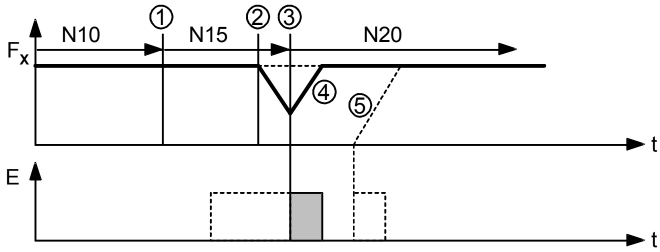
NOTE

The digital input used with the "read-in enable, externally programmable" function must be defined in machine data 45 of the master axis (in which the automatic mode is defined).

Example

Programming:

```
N10  G90  G77  X100.000  FX4000.00
N15  G99  X200.000
N20  X300.000
```



N10: NC block change on-the-fly from N10 to N15.

N15: The axis begins to decelerate at the deceleration start point of N15; G99 scans the "digital input", which has not yet been actuated at this time. The "digital input" is then actuated. The NC program changes to NC block N20 and the axis accelerates again. If the "digital input" has already been actuated at the deceleration start point of N15, a block change on-the-fly takes place (the axis continues traversing). If the "digital input" is not actuated, the axis decelerates to a standstill, remains at position X200.000, and signals "destination reached, axis stationary" [DRS]. Execution of the NC program continues when the "digital input" is actuated.

3.10.10 Acceleration Override (G30 to G39)

The acceleration and deceleration response of traversing movements can be controlled using the acceleration override. The acceleration override is practical for moving transport equipment alternately with and without load. A reduced acceleration/deceleration value is used for traversing under load. Both values can be reduced by a percentage factor using functions G30 to G39. Functions G30 to G39 are modal; the initial setting is G30.

Selection and effect of the acceleration override:

G30 → 100 % of acceleration (MD18) / deceleration (MD19)

G31 → 10 % of acceleration (MD18) / deceleration (MD19)

....

G39 → 90 % of acceleration (MD18) / deceleration (MD19)

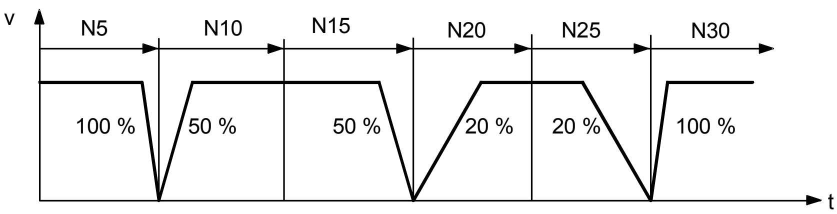
The acceleration override is deselected as a result of:

- A mode change
- Activation of the "reset technology [RST]" control signal

Example

Programming:

```
N5  G90  G77  X10.000  FX4000.00
N10  G35  X20.000
N15  X30.000
N20  G32  X40.000
N25  G32  X50.000
N30  G30  X60.000
```



N10: A block change on-the-fly does not take place between N5 and N10, because the acceleration and deceleration ramps change. The axis decelerates until the exact stop window is reached, and subsequently starts accelerating again at the new rate.

N15: An NC block change on-the-fly takes place from N10 to N15, because there is no change in the acceleration override.

N20: The change in the acceleration override inhibits the block change on-the-fly.

N25: An NC block change on-the-fly takes place, because there is no change in the acceleration override.

N30: The change in the acceleration override inhibits the block change on-the-fly.

4 M Functions with Special Functions

Stop at End of Block M00

If you program M number 0 as the first M function in an NC block, the program stops at the end of the NC block after the block has been executed. The NC program does not continue until the next time you actuate the "start" control signal [STA] (0 → 1).

Infinite Loop M18

If you program M function M18 as the first M function in the last NC block of an NC program, there is an automatic jump back to the start of the program (continuous execution of an NC program).

A distinction is made between the following two situations:

♦ M function M18 is output in the same way as any other M function. The return jump to the start of the program takes place only after complete execution of NC block N50 (including M18).

Programming:

```
N5  G91  X10.000  F100.00
..................
N50  G91  X50.000  F100.00  M1=18
```

♦ If only the M function M18 is programmed in the last NC block of an NC program, there is no M function output. Instead, the axis performs an immediate jump back to the start of the NC program.

Programming:

```
N5  G91  X10.000  F100.00
..................
N50  G91  X50.000  F100.00
N55  M1=18
```

In this example, an NC block change on-the-fly is executed at the start of the NC program, with no reduction in velocity in the X axis.

NOTE

An infinite loop can also be implemented by programming a dwell in the last NC block of an NC program.

End of Subprogram M17/M29

If M function M17 or M29 is programmed as the first M function in a subprogram, program execution jumps up one NC program level.

You only need to program M17 or M29 in order to terminate the NC program prematurely.

End of NC Program M02/M30

M functions M02 and M30 are used to terminate an NC program.
M functions M02 and M30 must be entered on their own (without any further NC block information). All G functions are restored to their default settings at the end of the NC program.

You only need to program M02 or M30 in order to terminate the NC program prematurely. Otherwise, the NC block with the highest block number is automatically the end of the program.

5 Tool Offsets

Contents

In this chapter you will find all the information about tool offsets.

5.1 Tool Offset G43, G44

General Information

With the tool offset, an existing NC program can continue to be used even after the tool dimensions have changed. Normally, the programming commands refer to the tool zero. The position of the tool tip can alter as a result of tool wear or a tool change. The tool offset allows you to compensate for this change without having to modify the NC program.

In general terms, a tool offset can be used to shift the programmed position of an NC block.

NOTE

Allowance is only made for the tool offset when absolute dimensions are programmed (G90).

Example

Programming:

```
N5  G90  (G43)  (G44)  G01  X50.000  F4000.00  D5      → Selection
N10  X80.000
N15  X150.000  D6                                      → Reselection
N20  X200.000
N25  X300.000  D0                                      → Deselection
```

N5: The tool offset is selected with D number D5. The optional parameters G43 and G44 can be used to define the direction in which the tool offset takes effect. Since G43 is active in the default setting, you only need to program G43 in order to deselect an active G44. The axis is positioned to X50.000 with allowance for tool offset memory D5.

N10: Tool offset D5 continues to remain active.

N15: The switchover to D6 deselects D5. The axis is positioned to X150.000 with allowance for tool offset memory D6.

N20: Tool offset D6 continues to remain active.

N25: Deselecting the tool offset with D0 deactivates D6. The axis is positioned to X300.000 without a tool offset.

Deselecting the Tool Offset

You can deselect an active tool offset by:

- Switching to another D number
- Deselecting with D0
- End of NC program or M02 or M30
- Mode change
- NC program abort as a result of a warning
- Block search, because the NC program is decoded again
- Actuating the "reset technology" [RST] control signal
- Activating follow-up mode [FUM]

A selected tool offset remains active after:

- Jumps to subprograms or return jumps to the main program
- Infinite NC program loops caused by M18 or G04 (dwell in the last NC block)

5.2 Tool Offset Memory

General Information

A total of 20 tool offset memories are available per axis. These are called up with D1 to D20. The tool offset memories are permanently assigned to the axes. You allocate axis names (X, Y, Z, A, B, C etc.) by defining the axis assignment. In the NC program, the respective tool offset memories are addressed using the assigned axis names.

NOTE

If you do not want a tool offset to be applied in all axes programmed in an NC block (including following block(s)), you must enter a zero (initial setting) in the appropriate tool offset memories.

Recommendation:
Reselect the tool offset to be applied in each block.

NOTICE

A tool offset is selected (D1 to D20) and deselected (D0) for a specific axis. For axes programmed in following blocks (slave axes), the D number selected for the master axis applies, i.e. the axis-specific tool offset number is computed.

You must not program the selection of a tool offset on its own in a block.

Example

Local M7 axis number	Axis assignment (MD2)	Tool offset D1	Tool offset D2
1	X	Value	Value
2	Y	Value	Value
3	Z	Value	Value

Programming:

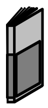
```
N5  G77  X100.000  FX2000.00  D1           ①
N10  Y100.000  FY2000.00                   ②
N15  Z100.000  FZ2000.00  D2               ③
N20  X200.000  Y200.000                    ④
N25  X300.000  Y300.000  Z300.000  D0      ⑤
```

N10: No tool offset is selected for the Y axis, i.e. "D0 is active", the Y axis is positioned at 100.000 ② .

N15: The appropriate D2 number ③ is applied to positioning of the Z axis.

N20: D1 is active for the X axis, no tool offset is selected for the Y axis. The appropriate D1 number is applied to positioning of the X axis, the Y axis is positioned at 200.000 ④.

N25: The tool offsets are deselected with D0 in the programmed axes X, Y and Z ⑤.Axes X, Y and Z are positioned at 300.000.

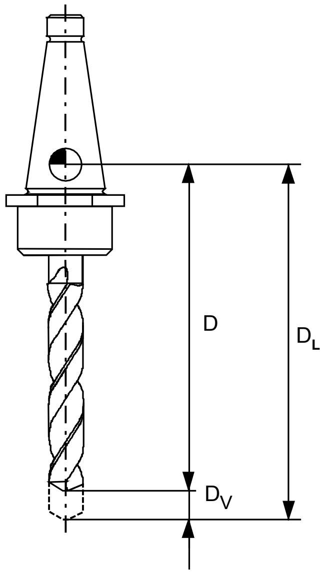


Further Information

In addition to using the standard user interfaces, you can also use the "input/output tool offset" task to input or output the tool offsets. The tasks are documented in the Task Description.

5.3 Tool Offset Variants

General Information

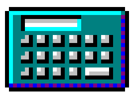


The tool offset is composed of the tool length offset and the tool wear.

D Nominal tool length

DL Effective tool length (tool length offset)

DV Tool wear (tool length wear)



EQUATION

D = DL + DV

Example

The tool used is 100 mm long (DL = 100 mm), for example. There is tool wear of 1 mm (DV = –1 mm).

The active tool offset is calculated as follows:
D = DL + DV = (100) + (–1) = **99 mm.**

Tool Length Offset

The tool length offset is defined as the actual tool length from the tool zero to the tool tip.

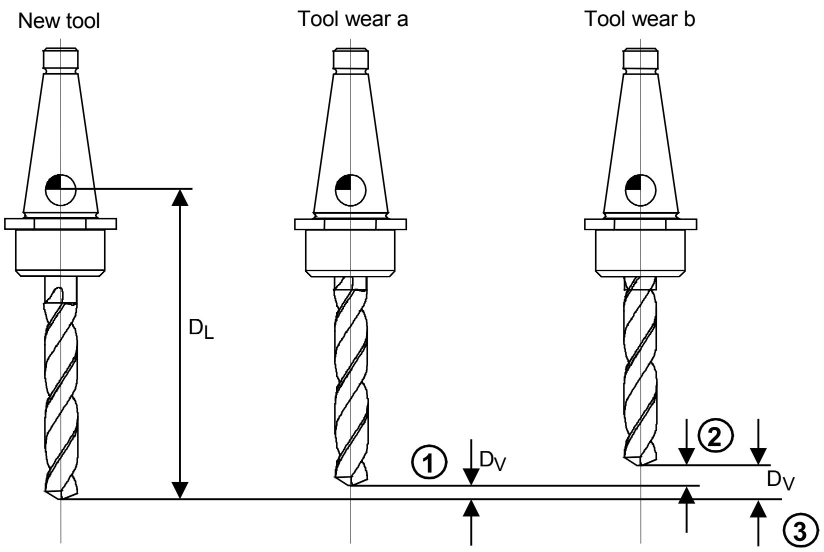
Name	**Input**	**Unit**
Tool length offset 1	from -9 999,999 to +9 999,999	mm

Tool Wear

The tool wear can be used to compensate for the change in tool length as a result of wear in two ways.

Absolute: Definition of a permanent tool wear value

Additive: Adding an offset value to the actual value of the tool wear.



① A tool wear D_V occurs during machining. This wear is calculated as an absolute reference to the new tool dimensions.

② If more wear occurs, this can be allowed for as an additive factor.

③ The tool wear can continue to be calculated in absolute terms.

Name	Input	Unit
Tool wear	from -9,999 to +9,999	mm

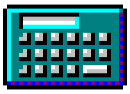
5.4 Direction of the Tool Offset

NOTE

In order to program a tool offset in the NC block, you need to enter at least the tool length offset. If no offset is to be allowed for even though the tool offset is selected, you must define a tool length offset and tool wear of 0. You can delete a tool wear by an absolute input of 0.

Negative Tool Offset G44

The tool generally points to the workpiece in the negative direction. The position value reduces with the infeed. The offset for the position value is calculated as follows when using G44.



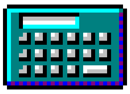
EQUATION

$X = X_{setpoint} + (D)$

X	Actual position approached
$X_{setpoint}$	Programmed position
D	Tool offset

Positive Tool Offset G43

It is also possible for the tool to point to the workpiece in the positive direction, however. The position value increases with the infeed. The offset for the position value is calculated as follows when using G43.



EQUATION

$X = X_{setpoint} - (D)$

X	Actual position approached
$X_{setpoint}$	Programmed position
D	Tool offset

6 Representation of NC Blocks in S7 Data Block Format

General Information

You only need to consult this chapter if you want to store NC programs directly in data blocks, or if you use the standard OP27/37 user interface.

All of the programming examples shown in the previous chapters are listed in this chapter in the S7 data block format.
The standard user interfaces convert the DIN representation to the S7 data block representation.

Guidelines

You must follow the guidelines below in order to convert NC programs from the DIN format to the S7 data block representation:

- If all the block information of a DIN NC block cannot be accommodated in an NC block in S7 data block format, you must subdivide the NC block into a main block and subsequent blocks.
- An NC block in S7 data block format can contain one main block and 19 subsequent blocks.
- M functions, D functions and subprogram calls must always be entered in the main block.

NC Block in S7 Data Block Format

DBWz	NC program number								DEC
DBWz+2	NC block number								DEC
DBWz+4	Continuation block identifier								DEC
DBWz+6	Reserved								
DBBz+8	**A**	**L**	0	**P**	**G4**	**G3**	**G2**	**G1**	BIN
DBBz+9	0	0	**FN**	**D**	**M3**	**M2**	**M1**	**F**	BIN
DBBz+10	Reserved								BIN
DBBz+11	Axis assignment								DEC
DBBz+12	G1 function								DEC
DBBz+13	G2 function								DEC
DBBz+14	G3 function								DEC
DBBz+15	G4 function								DEC
DBDz+16	Position in [AVWF] or dwell in [ms] or subprogram number								DEC
DBDz+20	Velocity in [10 AVWF/min] or subprogram loop count								DEC
DBBz+24	M1 function								DEC
DBBz+25	M2 function								DEC
DBBz+26	M3 function								DEC
DBBz+27	D function								DEC

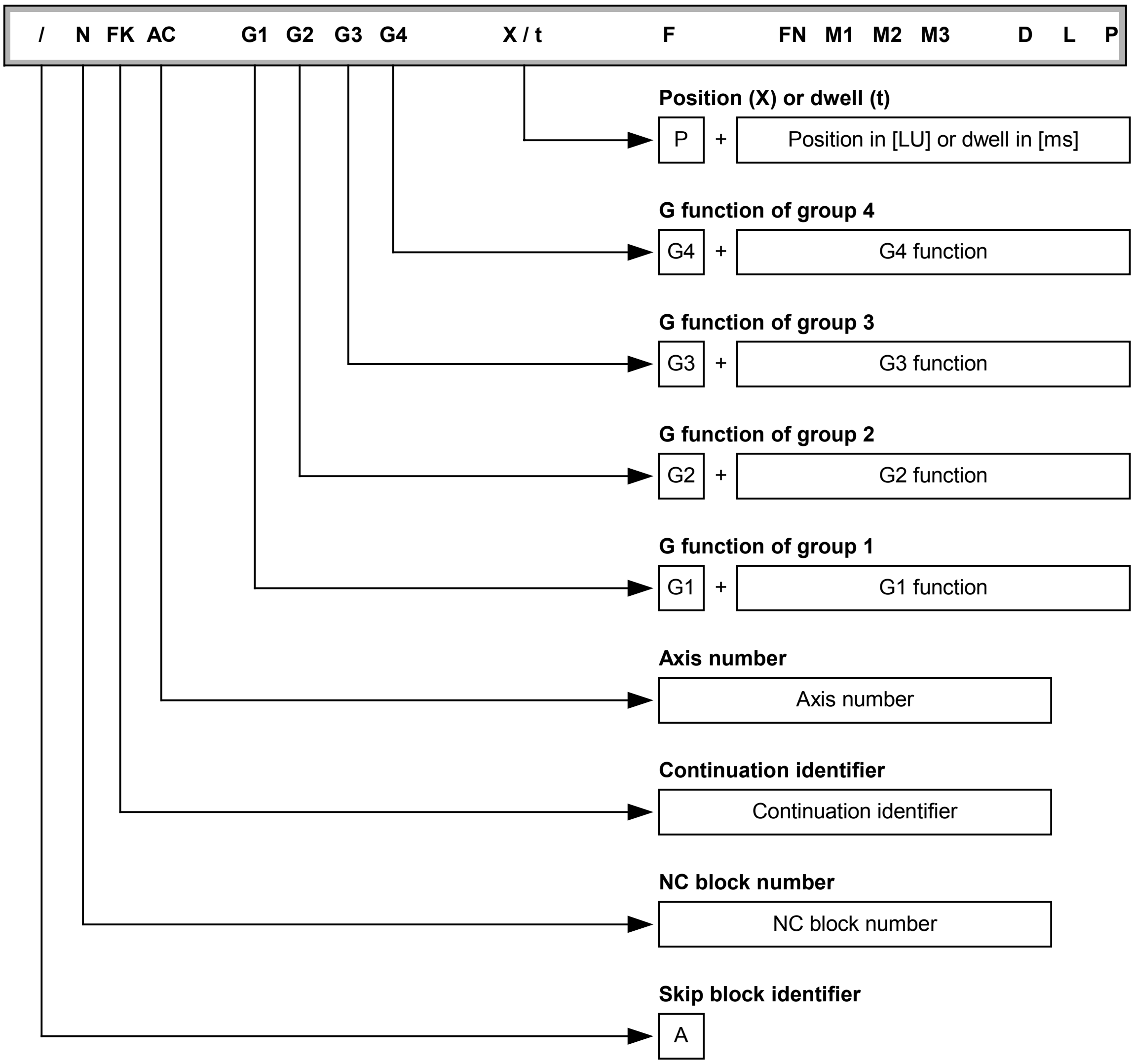
Representation of NC Blocks

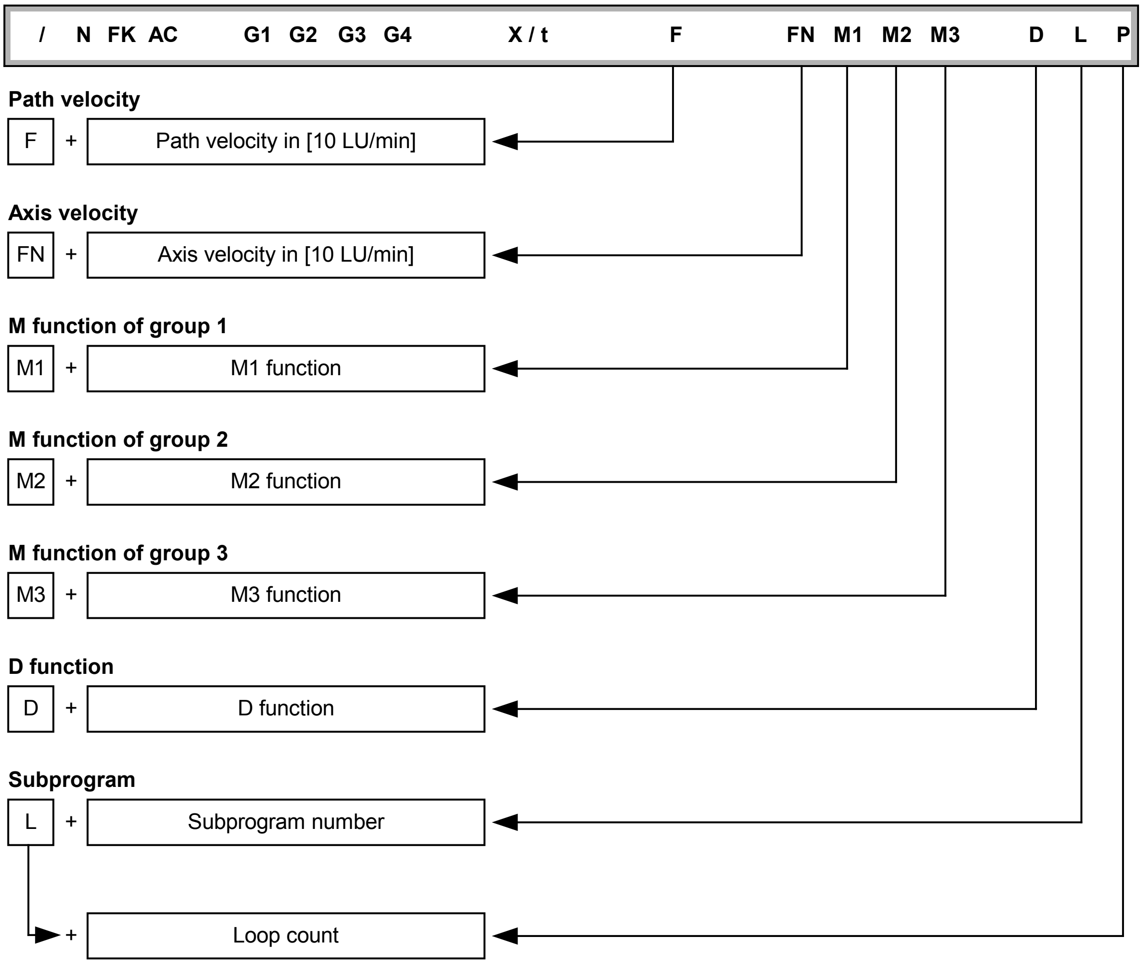
In order to simplify the descriptions below, the NC blocks are illustrated in the form of a table in which the columns show the NC block information and the lines represent the NC blocks.

NOTE

The NC program number is not shown in the table, since it always represents the same numeric value within an NC program.
In the data block, the NC program number in each NC block must be specified.
Input using the standard OP27/37 user interface is also shown in the table.

Data Representation In order to simplify the description, the data are represented in a table in which the columns show the NC block information and the lines represent the NC blocks.




/ N FK AC G1 G2 G3 G4 X / t F FN M1 M2 M3 D L P
Path velocity
F + Path velocity in [10 LU/min]
Axis velocity
FN + Axis velocity in [10 LU/min]
M function of group 1
M1 + M1 function
M function of group 2
M2 + M2 function
M function of group 3
M3 + M3 function
D function
D + D function
Subprogram
L + Subprogram number
+ Loop count

Index	Name	G Function		Active	Roll feed
G1	1st G function	G04	Dwell	s	no
		G50	External block change with absolute dimensions	s	no
		G51	External block change with incremental dimensions	s	yes
		G87	Set actual value on-the-fly - Deselection	s	no
		G88	Set actual value on-the-fly / Turning endlessly -	s	no
		G89	Set actual value on-the-fly / Turning endlessly +	s	no
		G90	Absolute dimensions	m	no
		G91	Incremental dimensions	m	yes
G2	2nd G function	**G30**	100 % override acceleration/deceleration	m	yes
		G31	10 % override acceleration/deceleration	m	yes
		G32	20 % override acceleration/deceleration	m	yes
		G33	30 % override acceleration/deceleration	m	yes
		G34	40 % override acceleration/deceleration	m	yes
		G35	50 % override acceleration/deceleration	m	yes
		G36	60% override acceleration/deceleration	m	yes
		G37	70 % override acceleration/deceleration	m	yes
		G38	80 % override acceleration/deceleration	m	yes
		G39	90 % override acceleration/deceleration	m	yes
G3	3rd G function	**G43**	Tool length compensation +	m	no
		G44	Tool length compensation -	m	no
		G53	Deselect zero offset	m	no
		G54	Zero offset 1	m	no
		G55	Zero offset 2	m	no
		G56	Zero offset 3	m	no
		G57	Zero offset 4	m	no
		G58	Zero offset 5	m	no
		G59	Zero offset 6	m	no
G4	4th G function	**G00**	Interpolation with rapid traverse or roll feed with rapid traverse	m	yes
		G01	Interpolation with path velocity or roll feed with axis velocity	m	yes
		G60	Exact stop	m	no
		G63	Position-dependent chaining	m	no
		G64	Maximum corner rounding	m	no
		G66	Corner rounding window 1	m	no
		G67	Corner rounding window 2	m	no
		G68	Shortest path with rotary axis	s	no
		G76	Chaining with rapid traverse	m	no
		G77	Chaining with axis velocity	m	no
		G96	Select collision monitoring	m	no
		G97	Deselect collision monitoring	m	no
		G99	Read-in enable, externally programmable	s	no

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	1	X	90			01	100 000	4000 00		10	11	13			
	10	0	Y					200 000								
	20	0	X	90				300 000	3000 00							
	30	0	Y	90				400 000	2000 00							
	40	0									30					

Fig. 6-1 Programming: NC Block Execution (Section 2.1)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	20	0	X	90				1 000	4000 00		10	11	13			
/	30	0	X	90				2 000	4000 00							
	40	0	X	90				3 000	4000 00							
	50	0									30					

Fig. 6-2 Programming: Skippable NC Blocks (Section 2.2)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0													24	23

Fig. 6-3 Programming: Subprograms (Section 2.3)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90				20 000	100 00							
	10	0	X	91				-30 000	100 00							

Fig. 6-4 Programming: Linear Axis (Section 3.2.1)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	C	90			68	315 000	1000 00							
	10	0	C	90				315 000	1000 00							
oder	10	0	C	90				-315 000	1000 00							

Fig. 6-5 Programming: Rotary Axis, Shortest Path (Section 3.2.2)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90			01	50 000	4000 00							
	20	0	X	04				3 000								
	30	0	X	91				2 000								
	10	0	X	90			01	10 000	400 00							
	20	0	X					-2 000								
	30	0	X	04				100								

Fig. 6-6 Programming: Dwell (Section 3.3)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90				100 000		5000 00						
	20	0	X	90		54		100 000								
	30	0				53										

Fig. 6-7 Programming: Zero Offset (Section 3.4)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X				76	10 000								
	20	0	Y					20 000								
	30	1	X					30 000								
	30	0	Y					30 000								

Fig. 6-8 Programming: Chaining with Rapid Traverse (Section 3.5)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90			77	10 000		5000 00						
	20	0	Y					10 000		2000 00						
	30	1	X					20 000								
	30	0	Y					20 000								
	40	1	X	91				10 000								
	40	0	Y							3000 00						
	50	1	Y					10 000								
	50	0	X							2000 00						
	60	1	X	90				40 000								
	60	0	Y					40 000								
	70	1	X					50 000		5000 00						
	70	0	Y					50 000		5000 00						

Fig. 6-9 Programming: Chaining with Axis Velocity (Section 3.6)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90			(00)	10 000								
	20	0	Y					20 000								
	30	1	X	91				10 000								
	30	0	Y					10 000								
	40	1	X	90				40 000								
	40	0	Y					40 000								

Fig. 6-10 Programming: Interpolation with Rapid Traverse (Section 3.7)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90			01	10 000	2000 00							
	20	0	Y					20 000								
	30	1	X	91				10 000	5000 00							
	30	0	Y					10 000								
	40	1	X	90				40 000								
	40	0	Y					40 000								

Fig. 6-11 Programming: Interpolation with Path Velocity (Section 3.8)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	1	X				77	100 000		1000 00						
	10	0	Y					100 000		1000 00						
	20	1	X				96	200 000								
	20	0	Y					200 000								
	30	1	X				01	300 000	1000 00							
	30	0	Y					300 000								
	40	1	X				97	500 000								
	40	0	Y					500 000								

Fig. 6-12 Programming: Collision Monitoring (Section 3.9)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X					100 000								
	20	0	Y					100 000								

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90			77	100 000		1000 00						
	20	0	Y					100 000		1000 00						
	30	0	Y					200 000		2000 00						
	40	1	X					200 000								
	40	0	Y					385 000								
	50	1	X					400 000		2000 00						
	50	0	Y					485 000								
	60	1	X					500 000								
	60	0	Y					650 000								

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90			01	100 000	1000 00							
	20	0	Y					100 000								
	30	0	Y					300 000	2000 00							
	40	1	X					300 000								
	40	0	Y					500 000								
	50	1	X					500 000								
	50	0	Y					700 000								
	60	1	X					600 000	1000 00							

Fig. 6-13 Programming: Maximum Corner Rounding (Section 3.10.1)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	1	X	90			77	100 000		1000 00						
	10	0					60									
	20	0	Y					100 000		1000 00						
	30	0	Y					300 000		2000 00						
	40	1	X					300 000								
	40	0	Y					500 000								
	50	1	X					500 000		2000 00						
	50	0	Y					700 000								

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	1	X	90			01	100 000	1000 00							
	10	0					60									
	20	0	Y					100 000								
	30	0	Y					300 000	2000 00							
	40	1	X					300 000								
	40	0	Y					500 000								
	50	1	X					400 000	1000 00							
	50	0	Y					600 000								

Fig. 6-14 Programming: Exact Stop (Section 3.10.2)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	1	X	90			77	100 000		1000 00						
	10	0					66									
	20	0	Y					100 000		1000 00						
	30	0	Y					300 000		2000 00						
	40	1	X					300 000								
	40	0	Y					500 000								
	50	1	X					500 000		2000 00						
	50	0	Y					700 000								

Fig. 6-15 Programming: Corner Rounding Window 1 or 2 (Section 3.10.3)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	1	X				77	100 000		1000 00						
	10	0					64									
	20	1	X					200 000								
	20	0	Y					200 000		1000 00						

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	1	X				77	100 000		1000 00						
	10	0					63									
	20	1	X					200 000								
	20	0	Y					200 000		1000 00						

Fig. 6-16 Programming: Path-Dependent Chaining (Section 3.10.4)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	1	X				01	100 000	1000 00							
	10	0	Y					100 000								
	20	1	X				77	200 000		1000 00						
	20	0	Y					200 000		1000 00						

Fig. 6-17 Programming: NC Block Transitions with Different Types of Coupling (Section 3.10.5)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X				01	100 000	4000 00		9					
	20	0	X					200 000			10					
	30	0	X				77	300 000		4000 00	11					
	40	0	X					400 000			12					
	50	0									13					
	60	0	X					500 000			14					
	10	0	X				76	100 000			9					
	20	0	X					200 000								
	30	0	X					300 000			10					
	10	0	X				76	100 000			9					
	20	0	X					200 000								
	30	0									10					
	40	0	X					300 000			11					
	10	1	X				01	100 000			10					
	10	0					77									
	20	0	X					200 000			11					

Fig. 6-18 Programming: M Functions (Section 3.10.6)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90			77	100 000		4000 00						
	15	0	X	50				200 000		2000 00						
	20	0	X	90				300 000		4000 00						

Fig. 6-19 Programming: External NC Block Change (Section 3.10.7)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90			77	100 000		4000 00						
	15	0	X	89				50 000		2000 00						
	20	0	X	90				300 000		4000 00						
	25	0	X	87				400 000		4000 00						

Fig. 6-20 Programming: Set Actual Value On-the-Fly (Section 3.10.8)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	10	0	X	90			77	100 000		4000 00						
	15	0	X				99	200 000								
	20	0	X					300 000								

Fig. 6-21 Programming: Read-In Enable, Externally Programmable (Section 3.10.9)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	5	0	X	90			77	10 000		4000 00						
	10	0	X		35			20 000								
	15	0	X					30 000								
	20	0	X		32			40 000								
	25	0	X		32			50 000								
	30	0	X		30			60 000								

Fig. 6-22 Programming: Acceleration Override (Section 3.10.10)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	5	0	X	91				10 000	100 00							
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
	50	0	X	91				50 000	100 00		18					
	5	0	X	91				10 000	100 00							
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
	50	0	X	91				50 000	100 00							
	55	0									18					

Fig. 6-23 Programming: M Functions with Special Functions (Section 4)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	5	0	X	90		(43)	01	50 000	4000 00					5		
	10	0	X					80 000								
	15	0	X					150 000						6		
	20	0	X					200 000								
	25	0	X					300 000						0		

Fig. 6-24 Programming: Tool Offset (Section 5.1)

/	N	FK	AC	G1	G2	G3	G4	X / t	F	FN	M1	M2	M3	D	L	P
	5	0	X				77	100 000		2000 00				1		
	10	0	Y					100 000		2000 00						
	15	0	Z					100 000		2000 00				2		
	20	1	X					200 000								
	20	0	Y					200 000								
	25	1	X					300 000						0		
	25	2	Y					300 000								
	25	0	Z					300 000								

Fig. 6-25 Programming: Tool Offset Memory (Section 5.2)

Index

L

M

N

P

R

S

T

V

W

Z

SIEMENS

System Solutions

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Part 2: SIMATIC S7 Communication Functions Edition 11.2002

GMC-BASIC Standard Software

SIEMENS

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Part 2: SIMATIC S7 Communication Functions

GMC-BASIC Standard Software

Edition 11.2002

Documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the "Remarks" column.

Status code in the "Remarks" column:

A .... New documentation.
B .... Unrevised reprint with new Order No.
C .... Revised edition with new status.

Edition	Order No.	Remarks
04.98	6AT1880-0AA00-1BA0	**A**
10.99	6AT1880-0AA00-1BC0	**C**
11.2002	6AT1880-0AA00-1BE0	**C**

SIMATIC and SINUMERIK are registered trademarks of Siemens AG. Other names in this publication might be trademarks whose use by a third party for his own purposes may violate the rights of the registered holder.

We have checked that the contents of this document correspond to the hardware and software described. Nonetheless, differences might exist and therefore we cannot guarantee that they are completely identical. The information contained in this document is, however, reviewed regularly and any necessary changes will be included in the next edition. We welcome suggestions for improvement.

Subject to change without prior notice

Order No.: 6AT1880-0AA00-1BE0
Printed in the Federal Republic of Germany

Siemens-Aktiengesellschaft

Contents

1 Introduction

Contents of this Manual

This documentation contains:

- ♦ A description of all procedures necessary for installation and configuration
- ♦ Configuration and operation of the "GMC-BASIC" standard software (user/application interface)

Structure of this Manual

The documentation is organized into the following sections:

- ♦ Overview: An overview of General Motion Control
- ♦ Installation/Configuration: Installation of the software; configuration of PROFIBUS and SIMOLINK
- ♦ A description of the GMC-BASIC standard software:
  - Data Structure: overview of the application interface
  - Parameter and Application Interface – Data Blocks: configuration of the system capacity (number of axes) and target systems (MASTERDRIVES MC, M7-FM). Description of the transfer of data tasks from/to the technology
  - Function Blocks: Calling and Parameter Assignment
  - Program and Configuration Examples: Examples of finished programs to help you get started.
  - Expert Knowledge/Tips: An in-depth examination of the system; changing the standard configuration, extended system diagnostics, technical specifications and much more.

2 Overview

Introduction

GMC (General Motion Control) describes the positioning and synchronization system solution for the application of the technology in SIMATIC Motion Control (M7-FM target systems) and in MASTERDRIVES Motion Control, Technology Option F01 (MASTERDRIVES MC target system).

SIMATIC Motion Control

In the solution with SIMATIC Motion Control, the technology is installed on one or more M7 computers. The M7-FM(s) is (are) connected to the individual MASTERDRIVES MC drive units across a high-speed fiber-optic link. The MASTERDRIVES MC units contain only the basic functionality with position control (MCB = MC with basic functionality)..

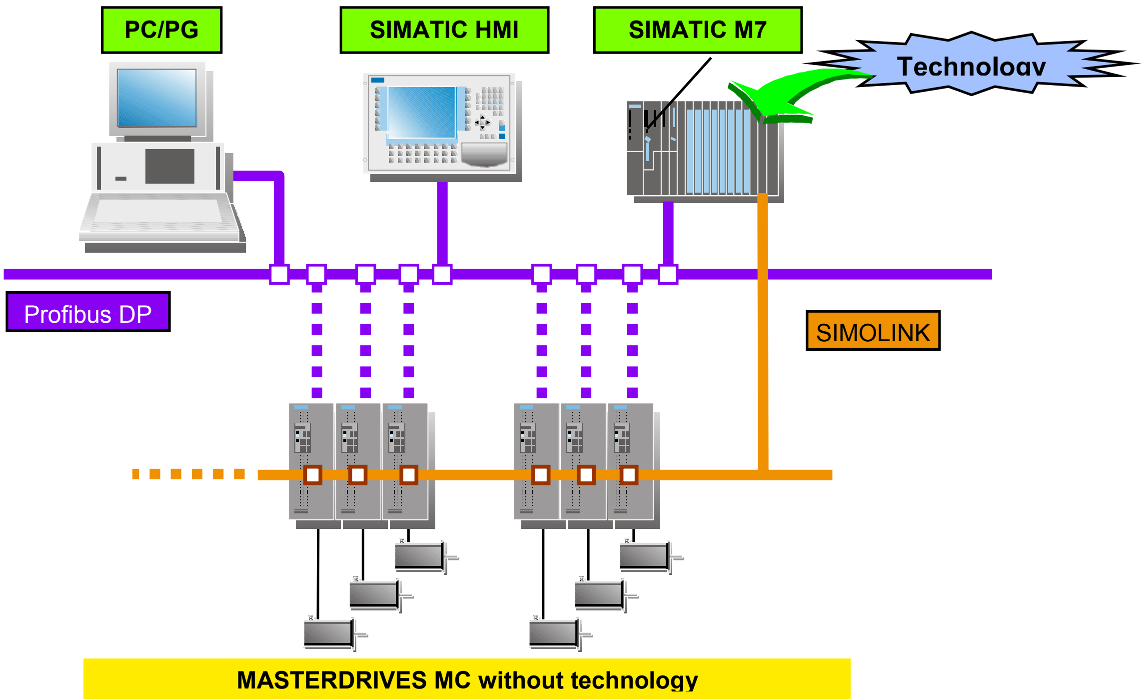


Fig. 2-1 Motion Control on M7-FM Target System with MASTERDRIVES MC

SIMATIC Motion Control with IM178

In the solution with SIMATIC Motion Control, the technology is installed on one or more M7 computers. M7-FMs are connected with the individual drive devices of the IM178 interface module via a synchronous Profibus module which only incorporate the basic functionality and the set speed output via an analog output.

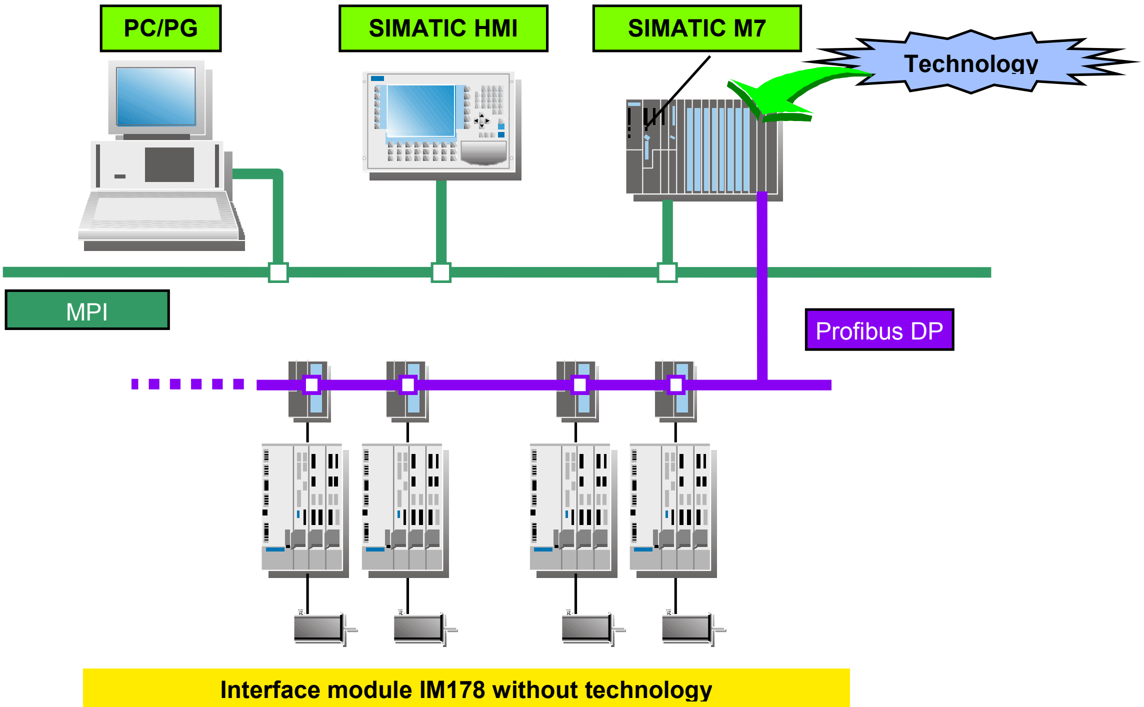


Fig. 2-2 Motion Control on M7-FM Target System with IM178

MASTERDRIVES Motion Control (Technology Option F01)

In the solution with MASTERDRIVES Motion Control (Technology Option F01), the technology is installed directly on the MASTERDRIVES MC (MCT = MC with technology). The S7-CPU is connected to the individual MASTERDRIVES MC drive units via PROFIBUS-DP. For the high-speed data communication required by the synchronization technology, the individual MASTERDRIVES MC units must be connected via the SIMOLINK fiber-optic link.

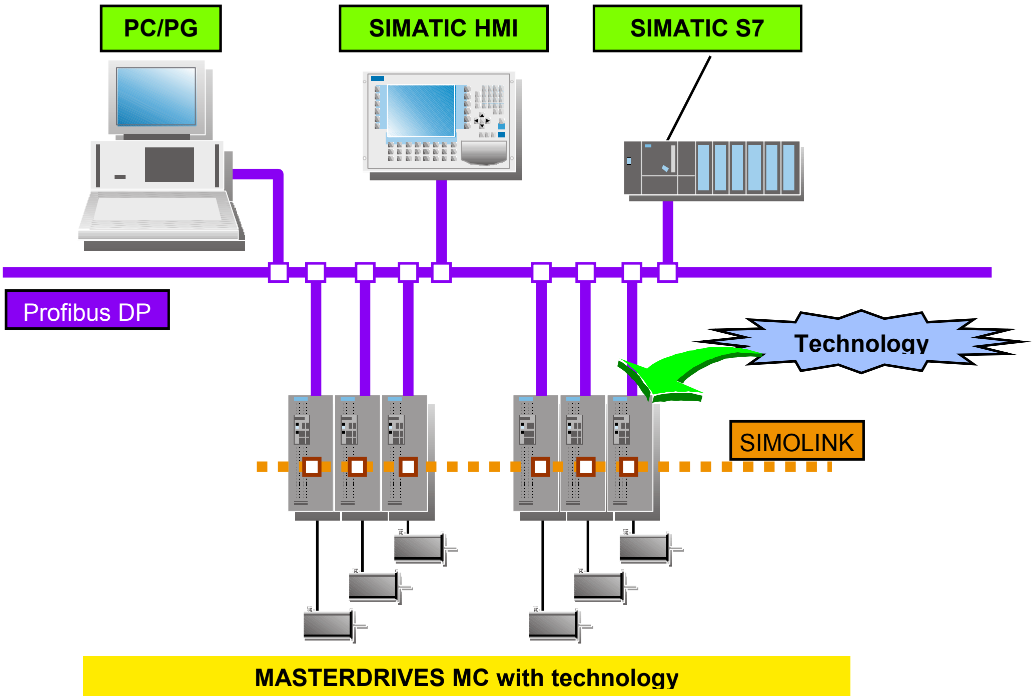


Fig. 2-3 Motion Control on MASTERDRIVES MC Target System

Extensions

MASTERDRIVES Motion Control and SIMATIC Motion Control can be used together. It is also possible for SIMATIC Motion Control (M7-FM) to communicate with the basic functionality of MASTERDRIVES MC via PROFIBUS-DP.

Whatever type of solution you use, the application interface is always centralized in the S7-CPU. The "GMC-BASIC" standard software, which allows the user to communicate with the application, is documented in this description.

The "GMC-BASIC" standard software provides all the drivers required for communication with the M7-FM and MASTERDRIVES MC target systems. The application interface (data blocks) is standardized and is independent of the target system. "GMC-BASIC" is used as a platform for both the application and optional standard user interfaces such as "GMC-OP-OAM" (e.g. user interface with COROS OP 27).

Hardware for SIMATIC Motion Control

You need the following hardware in order to operate the technology on an M7 computer:

- SIMATIC S7-400
- FM456 (M7-FM)
- SIMOLINK for the connection between M7 and MASTERDRIVES MC or Profibus for the connection between M7 and IM178 (IF modules)
- MASTERDRIVES MC with basic functionality (MCB) or interface module IM178

Hardware for MASTERDRIVES Motion Control (Technology Option F01)

You need the following hardware in order to operate the technology on a MASTERDRIVES MCT:

- SIMATIC S7-300/400 CPU with integrated DP interface
- MASTERDRIVES MC with enabled technology software (MCT)
- PROFIBUS for the connection between SIMATIC S7 and MASTERDRIVES MC

Software for SIMATIC Motion Control

You need the following software in order to operate the technology on an M7 computer:

- STEP 7 V5.0 or higher with the appropriate authorization
- Object manager for MASTERDRIVES and SIMOLINK
- M7-SYS, V5.0
- Technology software for M7-FM
- GMC-BASIC standard software for SIMATIC S7

Software for MASTERDRIVES Motion Control (Technology Option F01)

You need the following software in order to operate the technology on a MASTERDRIVES MCT:

- STEP 7 V3.x or higher with the appropriate authorization
- Object manager for MASTERDRIVES
- GMC-BASIC standard software for SIMATIC S7

Software Options

The following software products are available for operator control and parameter assignment:

- GMC-OP-OAM standard software user interfaces for different OPs
- SIMOVIS/DriveMonitor PC software
- PC software for parameter assignment

3 Installation / Configuration

Contents

In this chapter you will find the information required for the installation and configuration of the "GMC-BASIC" software.

Software Requirements

The installation diskettes for the GMC software packages can be read on a programming device/PC running the MS-Windows 95/98 and Windows NT operating system.

The software packages cannot be installed until STEP 7 has been installed completely. If STEP 7 is to be deinstalled, the GMC software packages must be deinstalled first.

NOTE

The technology (positioning, synchronization) can be operated centrally on an M7-FM and/or decentrally on a MASTERDRIVES MC. If the technology is operated on a MASTERDRIVES MC, the term **MCT** (Motion Control with Technology) is used in this manual. If the technology is operated on an M7-FM and the MASTERDIVES MC is used without technology as a drive controller, the term **MCB** (Motion Control with Basic Functionality) is used. In the text, the interface module IM178 is denoted by IM178 on its own.

Hardware Requirements MCT

- PLC with an approved CPU
- MASTERDRIVES MC with technology
- Approx. 4.0 MB free capacity for the MASTERDRIVES MC Object Manager and approx. 16 MB free capacity for the GMC-Basic Standard Software on your hard disk
- Programming device/PC with STEP 7, V3.1 or higher
- PROFIBUS connecting cable

Hardware Requirements MCB

- SIMATIC S7 PLC with an approved CPU
- M7-FM
- SIMOLINK IF module
- MASTERDRIVES MC with SIMOLINK-Board
- Optical fiber to set up the ring bus (connection of SIMOLINK modules)
- "SIMOLINK" object manager, which enables the integration of the SIMOLINK module after installation in the S7 hardware configuration
- Approx. 2 MB free capacity for the "SIMOLINK" Object Manager and approx. 16 MB free capacity for the GMC-Basic Standard Software on your hard disk
- Programming device/PC with STEP 7, V5.0 or higher

Hardware Requirements IM178

- SIMATIC S7 PLC with an approved CPU
- M7-FM
- Profibus-DP IF module
- Interface module IM178
- PROFIBUS connecting cable
- Approx. 16 MB free capacity for the GMC-Basic Standard Software on your hard disk
- Programming device/PC with STEP 7, V5.0 or higher, including service pack 2

Introduction

The SIMATIC S7 system is used as centralized application interface to the technology. The necessary S7 library and the example S7 projects are generated automatically during installation of the "GMC-BASIC" software.

If you use the technology centrally on one or more M7-FMs, you will receive a ready-to-operate M7-FM module with the "technology software" already installed.

If MASTERDRIVES MC units are used to control the drive, you will also need to configure the SIMOLINK connection between the M7-FM system and MASTERDRIVES MC (MCB).

If the interface module IM178 is to be used, you also need to configure the Profibus connection between the M7-FM and the IM178 and parameterize the IM178.

If you operate the technology decentrally on the MASTERDRIVES MC (MCT), you must configure the PROFIBUS connection between the SIMATIC S7 system and the MASTERDRIVES MC. If the synchronization function is used, you will also need to configure the SIMOLINK connections between the individual MASTERDRIVES MC units.

The following hybrid configurations are also possible:

- Technology on M7-FM and MASTERDIVES MC (hybrid centralized and distributed configuration)
- Technology on M7-FM (communication primarily with M7-FM) but also communication with the basic functionality of MASTERDRIVES MC via PROFIBUS-DP.

3.1 Installing the "GMC-BASIC" Standard Software

Preparations

Before you start the setup program, please close all applications (e.g. STEP 7, MS-Word etc.), because you will need to reboot your programming device/PC after installing the example program and library, so that all the system variables are entered completely.

Installation

Insert the CD, and start the setup program in the "GMC-BASIC" directory. Important operating instructions are provided by the setup program during the installation routine. The drive on which you install the software is determined automatically from the installed version of STEP 7. A library named "P7MC1LIB" is installed in directory STEP7\S7LIBS and a project named "P7MC1_EX" is installed in directory STEP7\EXAMPLES. The library contains all the blocks of the "GMC-BASIC" standard software. The project contains the example programs.

Further Information

The operation of the "GMC-BASIC" standard software is described in the following chapters. The chapter entitled "Program/Configuration Examples" puts you on the fast track to writing a functional program.

CAUTION

You should not use MS-Windows tools, such as the Explorer, to move STEP 7 files and folders or to modify the STEP 7 data in the MS-Windows registry. Such modifications may cause a program malfunction.

3.2 SIMATIC Motion Control with M7-FM

3.2.1 Technology Software for M7-FM

Defining the Properties

You can order the technology software only in conjunction with an M7-FM. You then receive a flash card containing the technology software, an FM456-2 and the desired interface module (SIMOLINK / PROFIBUS-DP).

The runtime version installed supports 8 or 16 axes and their associated onboard peripherals as standard. All you need to do is load your hardware configuration into the M7-FM.

Further Information

Further properties can be found in Chapter 3.2.2 "Configuring the M7-FM".

3.2.2 Configuring the M7-FM

Configuration of the GMC application

The configuration of the GMC application is within the following limits:

- Either an IF964-DP module (Profibus-DP) or an IF-SIMOLINK module (fiber-optic-cable ring bus)
- A max. of 10 further IF modules in the form of digital and analog peripheral equipment
- A max. of 32 DP slave nodes

NOTE

It is not permissible to use the connecting devices, Profibus-DP and SIMOLINK, together.

If necessary, you can adapt the standard configuration. To do this, a configuration tool based on Excel is necessary. The tool is not included in the scope of supply. It is provided on request.

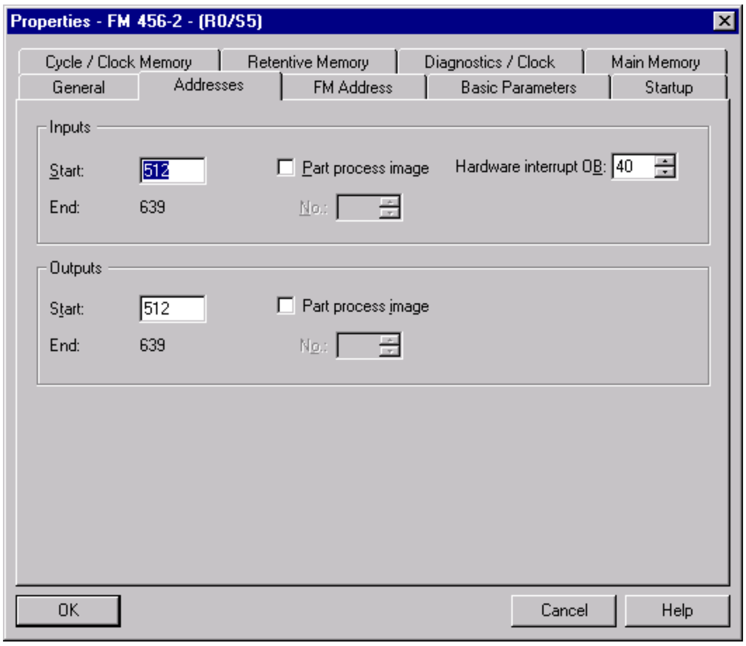
Defining the Properties

For the purposes of the following description, we are working on the assumption that you have started a project and set up a hardware configuration, using the template below, for example.

To configure the M7-FM, simply double-click on the installation position of the FM456-2 or select the installation position, right-click and select object properties.

Properties: Addresses

In the properties dialog, select "Addresses". If it is not already suggested as a default, enter the starting address for the inputs and outputs from the S7-CPU as 512 for the first M7-FM. The OB40 should be parameterized as the S7-CPU process alarm OB.



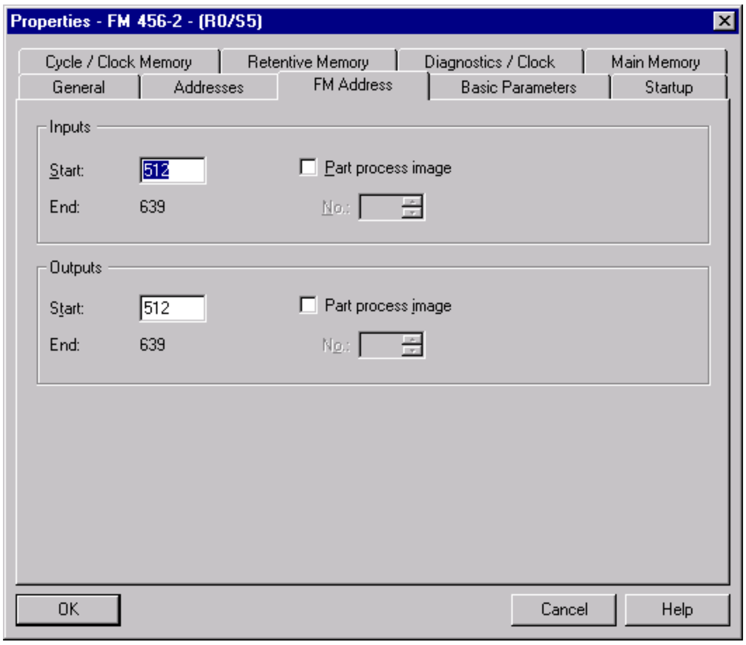
NOTE

Relative to the example project supplied, the address 512 is entered as the starting address for the inputs and outputs.

From the point of view of the S7-CPU, every M7-FM has its own address space. Therefore a second M7-FM would start, for example, at address 640.

Properties: FM address

Then select "FM address" in the properties dialog. If it is not already suggested as the default, enter 512 as the starting address for the inputs to and outputs from the M7-FM.

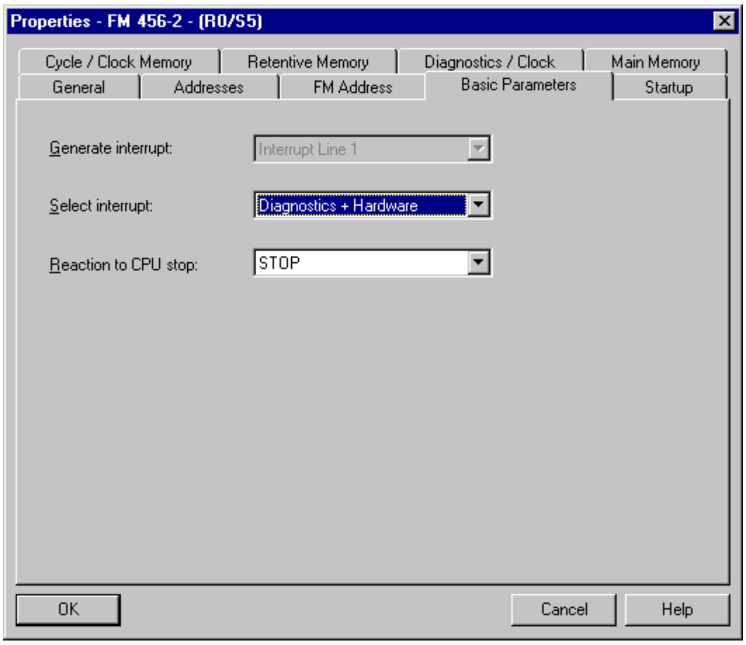


NOTICE

Irrespective of the S7-CPU address and the number of M7-FMs used, **512** must **always** be entered here as the starting address for the M7-FM.

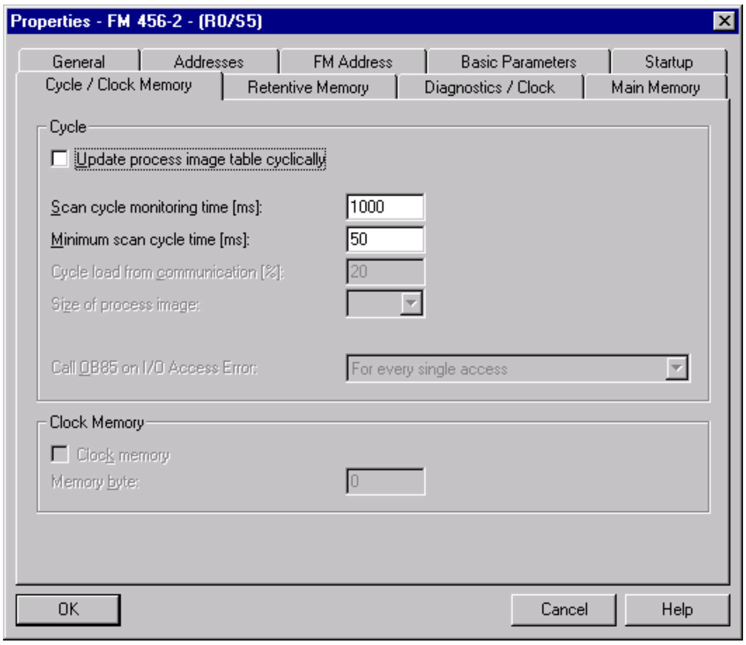
Properties: basic parameters

Then select "Basic parameters" in the properties dialog and set the alarm selection to "Diagnosis + Process". The M7 reaction to an S7-CPU-STOP should also be parameterized to STOP.



Properties: Cycle / clock sensor

Finally, select "Cycle / clock sensor" in the properties menu and set the minimum cycle time to 50 ms.



NOTE

The minimum cycle time is the term given to the time interval within the application during which the background task (FREE-CYCLE) is called.

The default setting for this minimum cycle time is 100 ms. For the M7-FM, this means that, even if the background processing only takes, for example, 5 ms, it waits the remaining time, in this case 95 ms, before beginning the next background cycle. Under some circumstances, this can mean that, for example, control tasks, which basically take several background cycles to process, can run for a long time before an acknowledgement. In order to reduce the processing time while protecting the system resources of the M7-FM, set the minimum cycle time to 20 ms.

The result of setting an even shorter minimum cycle time is that entries are made into the diagnostics buffer unnecessarily often. Longer minimum cycle times generally worsen S7-M7 communication.

3.2.3 Installing and Configuring with SIMOLINK

3.2.3.1 General description

Introduction

SIMOLINK is a high-performance optical fiber ring to connect drives to one another or to connect drives to a control system, e.g. SIMATIC M7.

SIMOLINK allows set / actual values and control / return signals to be transferred at 11 Mbit/s. The sampling intervals for all control circuits involves are synchronized with one another with quartz accuracy and jitter-free using special "Sync telegrams".

SIMOLINK, one of three bus systems

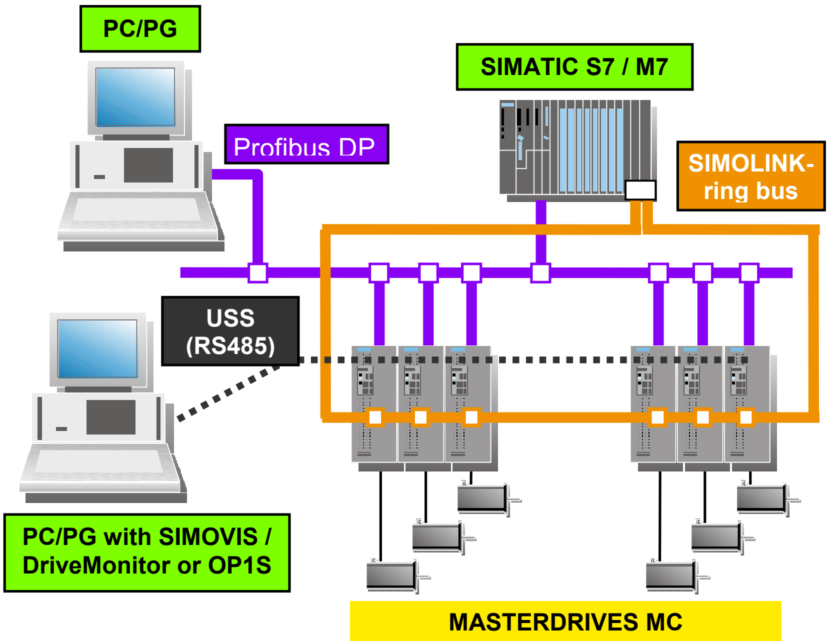


Fig. 3-1 Possible bus systems on the drives

SIMOLINK module

The SIMOLINK module is designed as an IF submodule. It can be operated in the slots for interface modules on an M7-FM or on an expansion board in the following SIMATIC M7-400 automation systems:

- SIMATIC S7-400 with FM456-2
- Expansion board EXM478

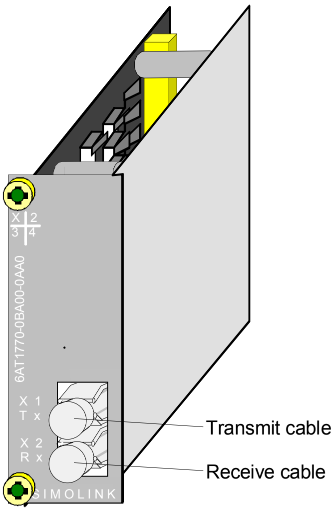


Fig. 3-2 SIMOLINK module

Further Information

The following only gives information about installation of the SIMOLINK object manager and about configuration using STEP7 HW-Config.

However, if you would like to know more about SIMOLINK, please refer to the MASTERDRIVES Motion Control Compendium.

3.2.3.2 Installing the SIMOLINK Object Manager

Hardware Requirements for SIMOLINK Communication

The following components are required for SIMOLINK communication:

- ♦ SIMATIC S7 PLC with an approved CPU
- ♦ M7-FM
- ♦ SIMOLINK IF module
- ♦ MASTERDRIVES MC with SIMOLINK board
- ♦ Fiber optic cable for setting up the ring bus (connection of SIMOLINK modules)
- ♦ "SIMOLINK" Object Manager, which enables the integration of the SIMOLINK Module after installation in the S7 hardware configuration
- ♦ Approx. 2 MB free capacity for the "SIMOLINK" Object Manager and approx. 16 MB free capacity for the GMC-Basic Standard Software on your hard disk
- ♦ Programming device/PC with STEP 7, V5 or higher, including service pack 2

Preparation

Before you start the setup program, please close all applications (e.g. STEP 7, MS-Word etc.) - you will need to reboot your programming device/PC after installing, so that all the system variables are entered completely.

Installation

Insert the CD and start the Setup program in the "SIMOLINK" directory. Important operating tips are given during the setup process. The drive on which you are installing the software is automatically detected on the basis of the STEP 7 version installed.

The object manager for parameterizing the SIMOLINK module in STEP 7 / HW-Config is installed.

CAUTION

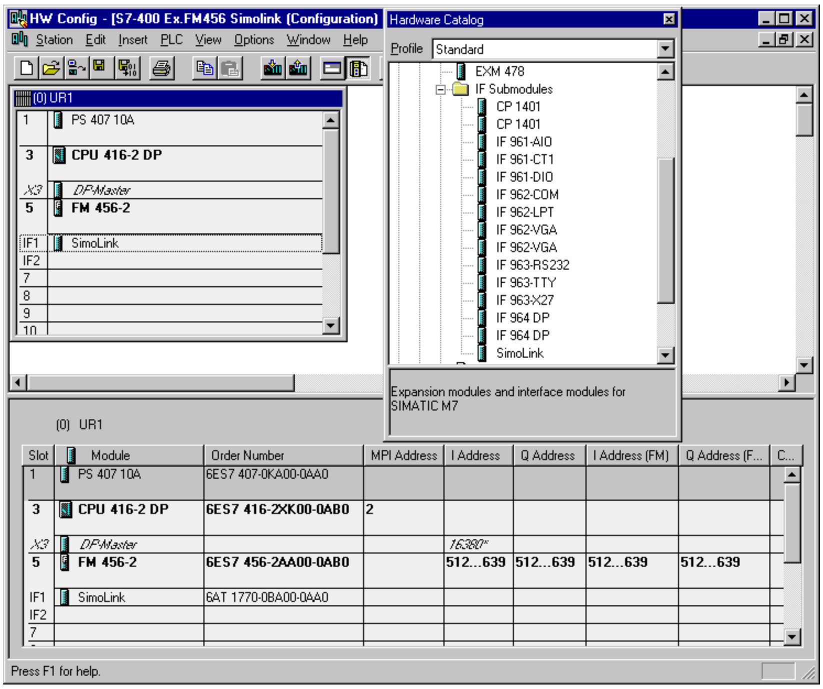
You should not use MS-Windows tools, such as the Explorer, to move STEP 7 files and folders or to modify the STEP 7 data in the MS Windows registry. Such modifications may cause a program malfunction.

3.2.3.3 Configuring the SIMOLINK Connection

Generating a SIMOLINK bus system

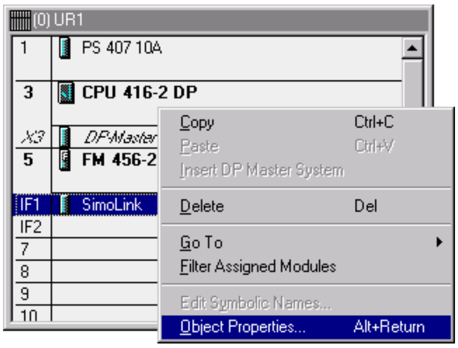
The hardware configuration can only be carried out as far as the SIMOLINK module. The SIMOLINK bus cannot be configured any further using STEP 7 HW-Config.

Select the insertion slot IF1 and double click on the SIMOLINK module in the hardware catalog to place the module in the selected slot.

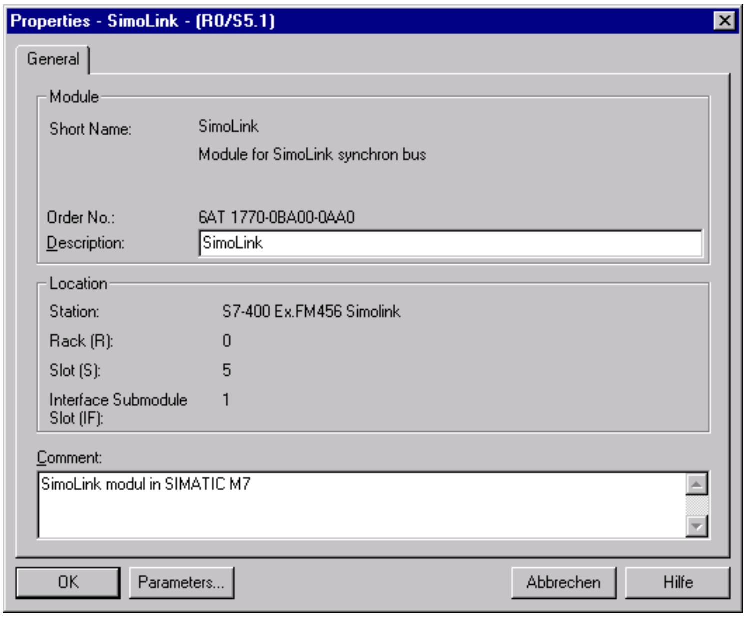


Defining the Properties

To configure the module itself, double click on the installation position of the SIMOLINK module or select the installation position, right click and select object properties.

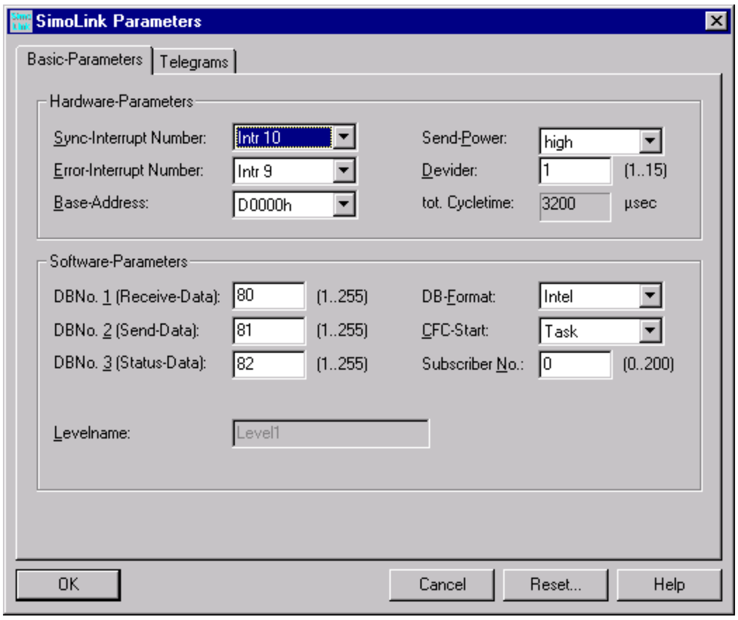


Then click on the "Parameters" button in the properties screen form.



SIMOLINK Parameters: basic settings

The "basic settings" tab is used for setting general parameters for the SIMOLINK IF modules.



CAUTION

To guarantee that the application runs correctly, the default settings for the hardware and software parameters should be used.

Hardware parameters

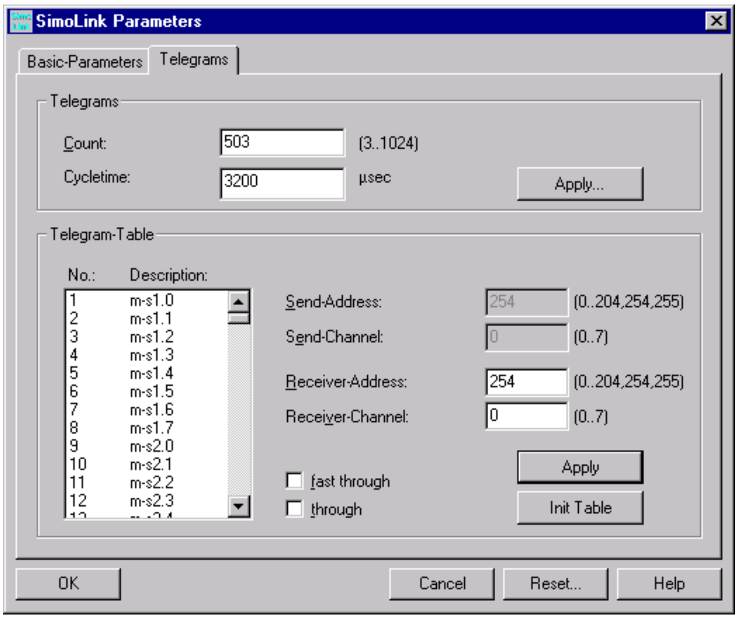
Parameters	Description	Default
Sync-Interrupt Number	Enter the synchronous interrupt number. The synchronous interrupt can be set to 9, 10, 11 or 12.	Intr 10
Fault Interrupt Number	Enter the fault interrupt number. The fault interrupt can be set to 9, 10, 11 or 12.	Intr 9
Address position	Enter the address position of the SIMOLINK board. The basic address of the dual port RAM range of the SIMOLINK-IF module can be set to the addresses D 0000h and E 0000h.	D 0000h
Send performance	Select the light intensity of the send laser (off, low, medium, high).	high
Divider	Enter the number of telegram cycles before a sync. interrupt should be triggered. This increases the overall run time by the divider set here (Value range: 1 to 15).	1
Total cycle time	This displays the total cycle time in µs. It is calculated by: Telegram tap cycle time * dividers The field is updated when you leave the "Divider" field. The same time must be entered for the MASTERDRIVES MC parameter P746.	

Software parameters

Parameters	Description	Default
DBNo. 1 (receive data)	Enter the number of the data block to be saved by the driver for receive data (Value range: 1 to 255).	80
DBNo. 2 (send data)	Enter the number of the data block to be saved by the driver for send data (Value range: 1 to 255).	81
DBNo. 3 (status data)	Enter the number of the data block to be saved by the driver for status data (Value range: 1 to 255).	82
DB format	Set the format in which the driver writes the data to the status data / send data block and reads the data from the status data / receive data block (SIMATIC, Intel).	Intel
CFC start	Select the level at which the CFC alarm level should be started. Options: Task: Start the CFC alarm level from the SIMOLINK task. To do this, the name of the start level must be configured in the SIMOLINK task. Driver: Start the CFC alarm level from the driver. To do this, enter the name of the start level in the "Start level name" field.	Task
Node no.	Enter the node number occupied by the SIMOLINK module on the sync-bus: 0: SIMOLINK module is Master 1 to 200: SIMOLINK module is Slave	0
Start level name	Enter the name of the CFC start level. This field is only accessible if the parameter ‘Driver’ is selected in ‘CFC start’ (Character string with max. 15 characters).	none

SIMOLINK parameters: telegrams

The "Telegrams" tab serves to set and parameterize all possible SIMOLINK telegrams.



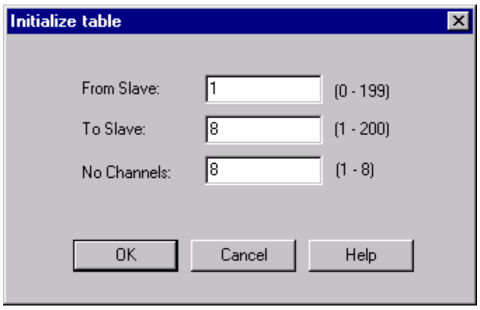
For the bus cycle time, enter the value 3200. When this is accepted, the number of telegrams are adapted to the cycle time.

Telegrams

Parameters	Description	Default
Number	This displays the number of telegrams to be processed. It corresponds to the length of the telegram table or the number of telegrams to be processed (Value range: 3 to 1024). The field is updated when you press the "Enter" or "Apply" button after making an entry in the "Cycle time" field. At the same time, the length of the telegram table is adjusted to the length displayed in the "Number" field.	503
Cycle time	This is used to parameterize the cycle time for the telegram table. It is calculated by: Number of telegrams * 6.36 µs	3200

SIMOLINK parameters Telegram table

Click on the ‘Pre-Assign’ button to configure your number of slaves and the number of channels used in the telegram table.



Parameter	Description	Default
From slave To slave	The parameters "From slave" and "To slave" describe the number of slaves (example with 8 MCB axes: From slave 1 to slave 8; example with 16 MCB axes: From slave 1 to slave 16).	1 1
Number of channels	Here, you configure the number of channels used	1 with MCB: 8

NOTE

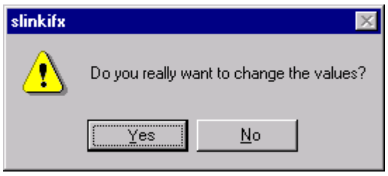
Bearing in mind the fact that the SIMOLINK board runs synchronously with the MASTERDRIVES MC control task, there is a fixed relationship between the parameters "Divider", "Number" and "Cycle time".

The time calculated for the cycle time, which is rounded off to a valid "Time slot", must correspond to MASTERDRIVES MC parameter P746 (SLB cycle time). Set the "Time slot" accordingly in the MASTERDRIVES MC.

NOTICE

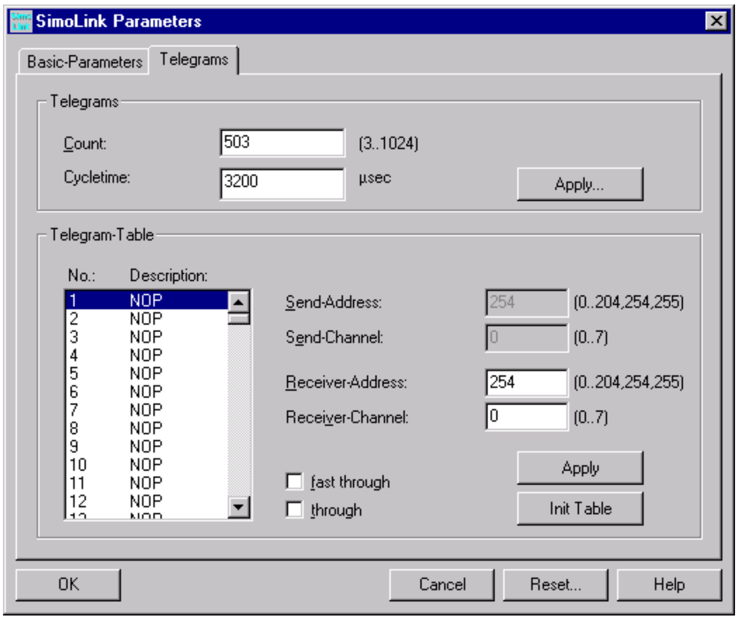
The parameterized cycle time must be a multiple of 0.1 ms.

To conclude, press OK (in ‘Pre-assign table’ dialog) and confirm again by clicking YES.



Telegram table

Once this action has been confirmed, the settings are saved for the telegram currently being processed and the display is updated in the "No.: / Description" selection list.



Parameter	Description	Default
No./description	This selection list displays all the telegrams. Select a telegram to display or amend its settings in the "Telegram table" box. The "Description" column gives a brief description of the meaning of the telegram: NOP NOP telegram (full telegram) m-s1.0 Telegram from master to slave 1, channel 0 s1.2-s3.4 Telegram from slave 1, channel 2 to slave 3, channel 4 (only possible with "Forward" or "Forward immediately" settings) SYNC Synchronous telegram The last two telegrams have to be SYNC telegrams.	-
Sender address	Enter the sender address. This field is only active if one of the options "Forward" or "Forward immediately" is activated. 0 Bus master 1 to 200 Slave number 201 to 204 Special telegrams, defined by master (not used by the technology functions) 254 NOP telegram 255 SYNC telegram	-

Parameter	Description	Default
Sender channel	Enter the channel number of the sender. This field is only active if one of the options "Forward" or "Forward immediately" is activated. (Value range 0 to 7).	-
Receiver address	Enter the receiver address (For values, see sender address).	254
Receiver channel	Enter the channel number of the receiver (Value range 0 to 7).	0
Forward / forward immediately	Select one of these options to enable direct communication between two slaves. Forward immediately: Valid data are taken from the current receive register of the Simolink module Forward: Valid data are taken from the "Shadow receive register". You cannot activate both options at the same time.	Not activated with M7!

NOTE

If you change a telegram and then select another telegram from the list without applying the settings, the changes are rejected. In order to optimize the time response of the driver, you should use directly sequential receiver addresses when parameterizing the telegram table. This is guaranteed if you pre-assign values to the table.

The last two telegrams must be SYNC telegrams. The identifiers are set accordingly by HW-Config. Do not alter these identifiers under any circumstances.

Finally, all that remains is to save the S7 hardware configuration and load it into the S7/M7-FM.

Establish the SIMOLINK connection between the IF module in the M7-FM and the MASTERDRIVES MC.

If the S7-CPU and the M7-FM are in RUN, the flashing of all 3 LEDs on the SIMOLINK board in the MASTERDRIVE MC indicates a correctly functioning bus connection.

Further Information

Further information on SIMOLINK communication is given in the MASTERDRIVES compendium.

3.2.3.4 Configuration and parameterization of the MASTERDRIVES MC converter (MCB)

Introduction

You must load a standard interconnection to the converter to ensure that the MASTERDRIVES MC is correctly controlled by the technology on the M7-FM. The script file or the download file for the standard interconnections is contained on the CD of GMC-BASIC. It is loaded into the converter with the SIMOVIS/DriveMonitor tool.

Software requirements

To be able to load the standard interconnection, you must have installed the SIMOVIS/DriveMonitor tool on your PG/PC. SIMOVIS/DriveMonitor is not included in the scope of supply of GMC-BASIC.

- ♦ PG/PC with SIMOVIS/DriveMonitor >=V5.3

Hardware requirements

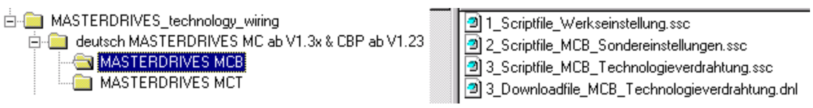
- ♦ MASTERDRIVES MC from V1.3 upwards with SIMOLINK board
- ♦ Connecting cable for PG/PC – MASTERDRIVES MC (USS)

Installation of standard interconnection

The standard interconnection is stored on CD "GMC-BASIC" as a script file and download file. In addition, two script files are provided as a basis for carrying out factory setting and for special settings. You must, however, adapt these to the components you are using (e.g. drive) or to your configuration (e.g. transceiver address).

Copy the script files/download file on your PG/PC into a directory which you first create for your drive (one directory per drive).

Contained on the CD "GMC-BASIC":



Loading inter-connection to the MASTERDRIVES MC

The standard interconnection is stored in file "3_Scriptfile_MCB_ Technologieverdrahtung.ssc" or "3_Downloadfile_MCB_ Technologieverdrahtung.dnl". You must **not** make any changes in these files under any circumstances! (See also Section 8.6).

The script file or download file is transferred with SIMOVIS/DriveMonitor. In the script file for special settings, you must parameterize the transceiver address and the number of nodes (the dispatcher function, in this case, is performed by the SIMOLINK IF module in the M7-FM).

Extract from the script file "2_Scriptfile_MCB_Sondereinstellungen.ssc":

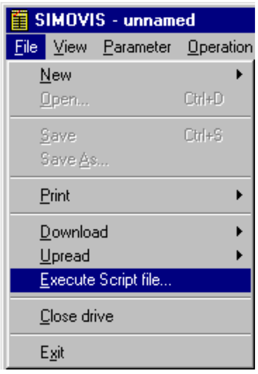
```
REM P740: Node address
REM     Master = 0, Slave1 =  1 etc.
        WRITEE      740     1     ????

REM P743: Number of nodes in the SIMOLINK ring incl. master
        WRITEE      743     1     ????
```

The node address is to be issued according to the following scheme:

Axis number (Simolink slave No.)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Node address (transceiver address)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

After you have adapted the script files to your equipment for factory setting and special settings, they are also loaded into the converter with the help of SIMOVIS/DriveMonitor. The script file can be edited with, e.g., the NOTEPAD.

Loading the interconnection to the converter:



Dialog prompts will tell you how to proceed further.

NOTE

Transfer of the script file in the SIMOVIS/DriveMonitor menu "Parameter-guided start-up of technology functions" is not supported at the present time.

3.2.4 Configuration using Profibus-DP

3.2.4.1 General description

Introduction

The IM178 module is connected to the M7-FM by means of a synchronous PROFIBUS DP link. I. e., the technology (closed-loop position controller) runs centrally on an M7-FM and the control circuit is closed by means of a synchronous Profibus DP.

Pulse synchronism (equidistant pulse signal on the bus system) ensures that the actual position values can be detected and evaluated by the M7-FM at exactly the same time. The setpoints which are output can also come into effect in the axes at exactly the same time. Actual-value recording and setpoint specification thus take place synchronously with the position controller (technology on the M7-FM).

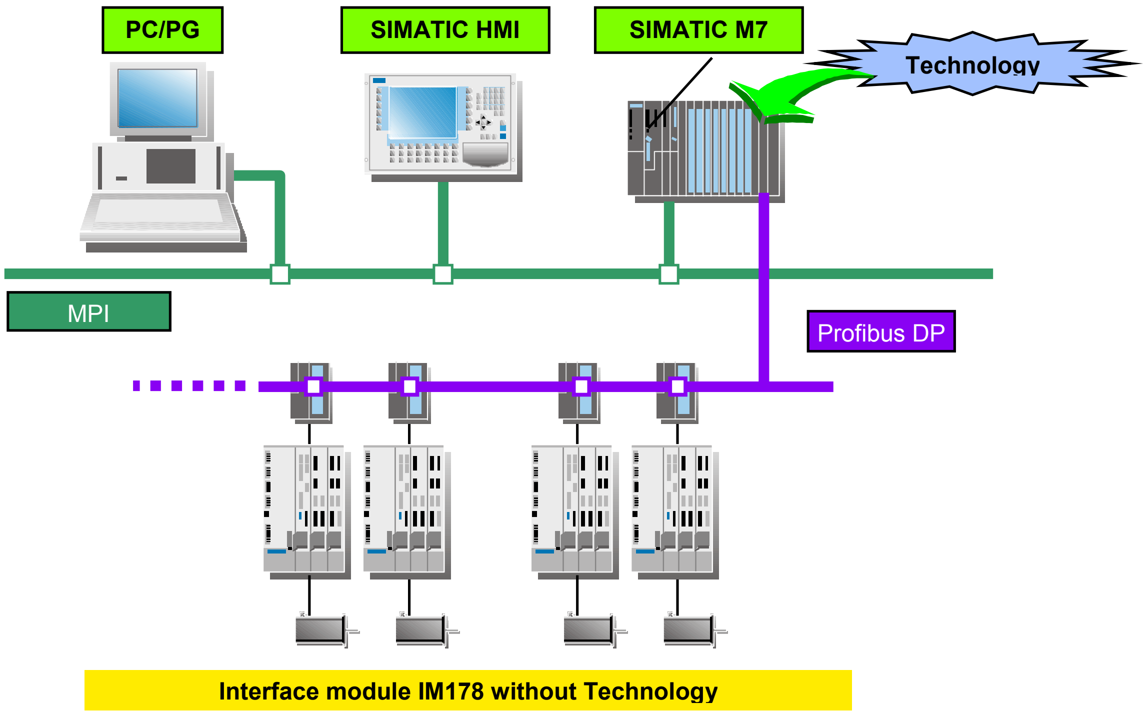


Fig. 3-3

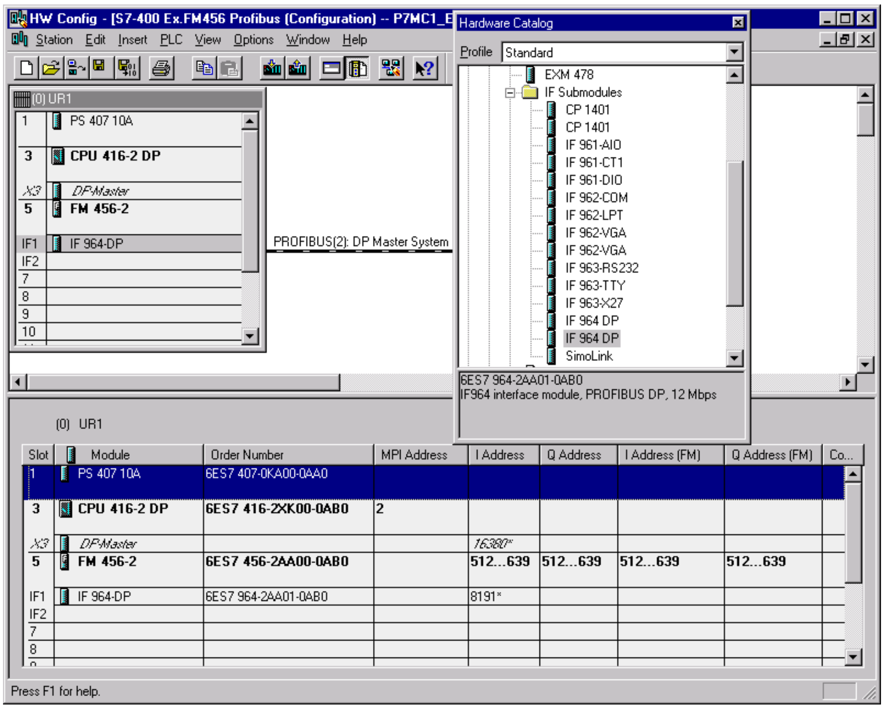
3.2.4.2 Configuration with IM178 interface module

Hardware Requirements IM178

- SIMATIC S7 PLC with an approved CPU
- M7-FM
- Profibus-DP IF module
- IM178 interface module
- Profibus connecting cable
- Approx. 16 MB free capacity for the GMC Basic Standard Software on your hard disk
- Programming device/PC with STEP 7, V5.0 or higher including Service pack 2

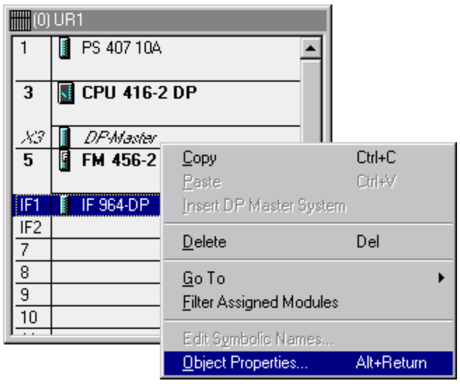
Creating a Profibus system

Select the IF1 insertion position and double click on the IF 964-DP module in the hardware catalogue to place the module in the selected slot.

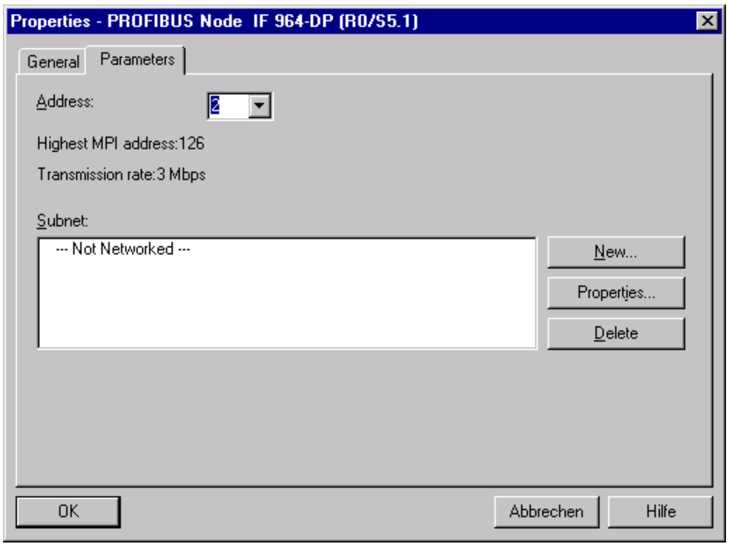


Defining the Properties

To configure the IF964-DP module itself, simply double click on its installation position or select the installation position, right click and select the object properties.

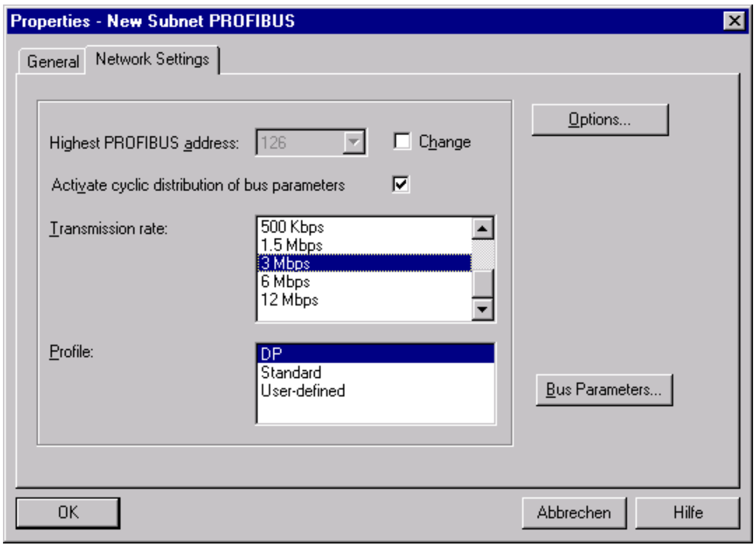


In order to operate an IM178 module, you need to define a new Profibus subnet by pressing the "New…" button in the "PROFIBUS IF964-DP interface properties" dialog box.



A Profibus interface is automatically configured with the properties Profile = DP and a baud rate of 1.5 Mbit/s.

In order to process up to eight M7 axes in an equidistance time of max. 2 ms, select 3 Mbit/s as the baud rate.



You can access the specific properties of this Profibus-DP communication by pressing the "Options" button.

The following table shows the time which the Profibus-DP needs for one bus cycle, depending on the baud rate and the number of IM178 modules. The cycle time of the application cannot be smaller than this time or the default time specified for the respective application.

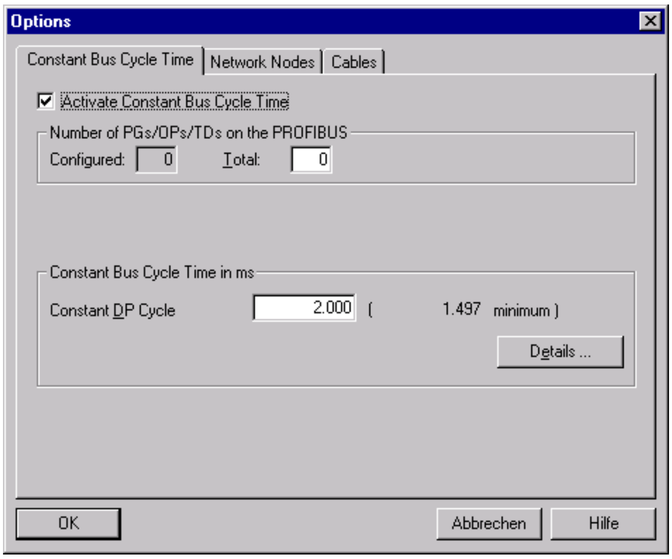
Baud rate	**Number of IM178 modules**				
	2	**4**	**8**	**12**	**16**
1,5 Mbit/s	2.234	3.694	6.614	9.534	11.574
3 Mbit/s	0.907	**1.497**	2.677	3.857	4.172
6 Mbit/s	0.628	1.022	1.809	2.597	3.019
12 Mbit/s	0.516	0.841	1.491	2.141	2.525

Table 3-1 Equidistance time [ms] depending on the number of axes and the baud rate (extract from HW-Config)

Properties: Options

In the "Equidistance" tab in the "Options" dialog, check the box to activate the equidistant bus cycles for Profibus.

Enter 2.000 (2 ms) as the equidistance time. This corresponds to the basic sampling time of the GMC application for eight axes.



Once you have entered this, apply the configuration by pressing the OK button three times. You will then be back to the hardware configuration dialog. The IF module displays one external connection (DP master system(1)). The interface modules IM178 can then be connected.

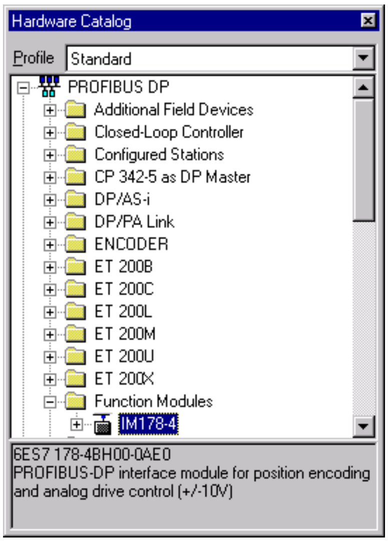
NOTE

Any properties not mentioned should keep their default settings. The baud rate can be increased manually to up to 12 Mbit/s.

NOTICE

Using the information in Table 3-1, select the equidistance time such that you always round the value to exactly 0.1 ms. You cannot use equidistance times <2.0 ms with standard GMC applications.

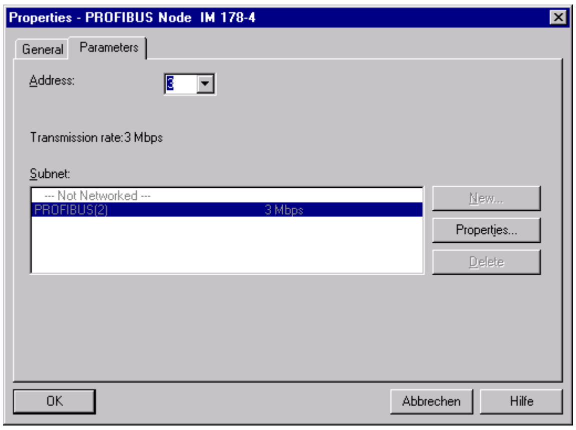
Inserting an IM178 interface module



- ♦ Select the Profibus string
- ♦ Open the hardware catalog
- ♦ In the hardware catalog, select PROFIBUS-DP → Function blocks → IM178-4
- ♦ Double click on *IM178-4*

Defining properties for Profibus IM178 interface

Enter 3 as the Profibus node address for the first IM178 interface module (see Table 3-2).



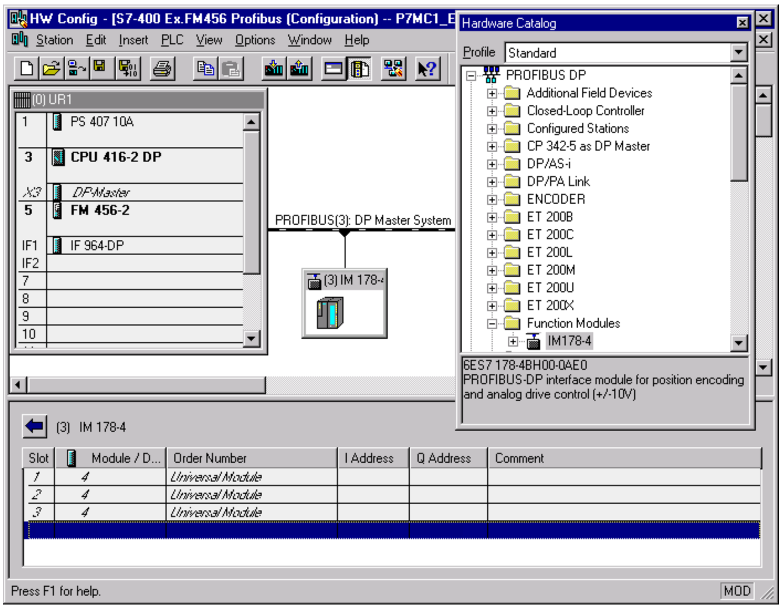
A maximum of 16 IM178 modules (i.e. 32 axes) can be operated with the GMC central application.

In order to configure the IM178 interface module, all you need at this stage is the Profibus address. All the other properties were set when you set the properties for the IF module IF964-DP.

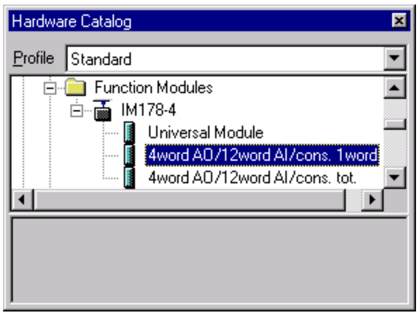
Press OK to confirm the changes and leave the properties dialog.

Creating a hardware configuration

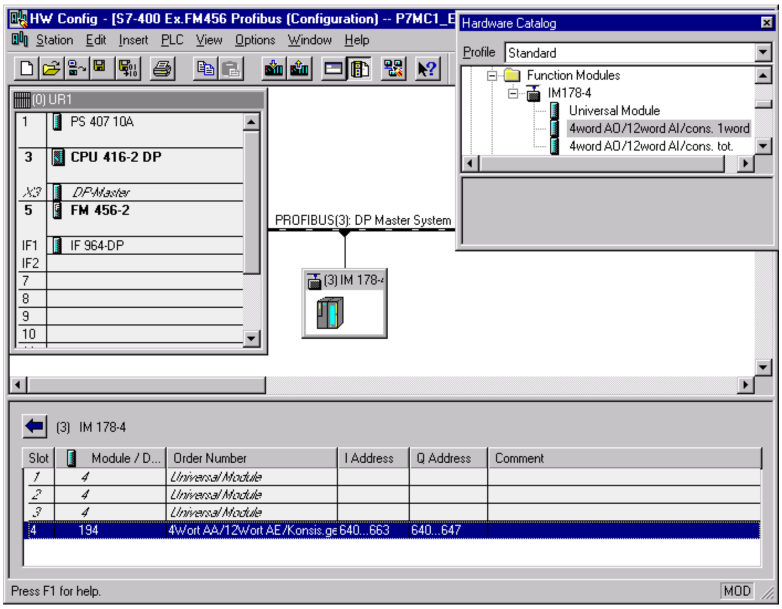
Your HW Config. will display the following when you have configured a IM178 (in this case, the first).



IM178 configuration



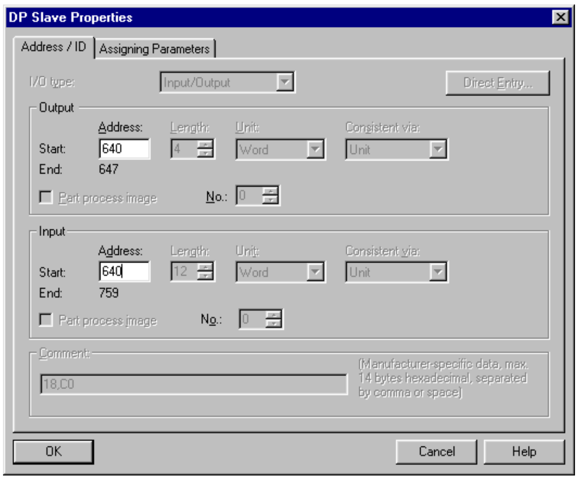
- Select slot 4 of the IM178
- Open the hardware catalog
- In the hardware catalog, select PROFIBUS-DP → Function blocks → IM178-4 → 4Word AA/12Word AE/Consis.1 Word
- Double click on *4Word AA/12Word AE/Consis.1 Word*



The above screen capture shows the results of inserting an IM178 in the Profibus system. All you need to do now is set the parameterization data for the IM178 itself.

Properties – Parameterize DP slave - Parameterization

- Select slot 4 of the IM178
- Double click on it. The dialog box "Properties – DP-Slave – Address / Identifier" then pops up



NOTICE

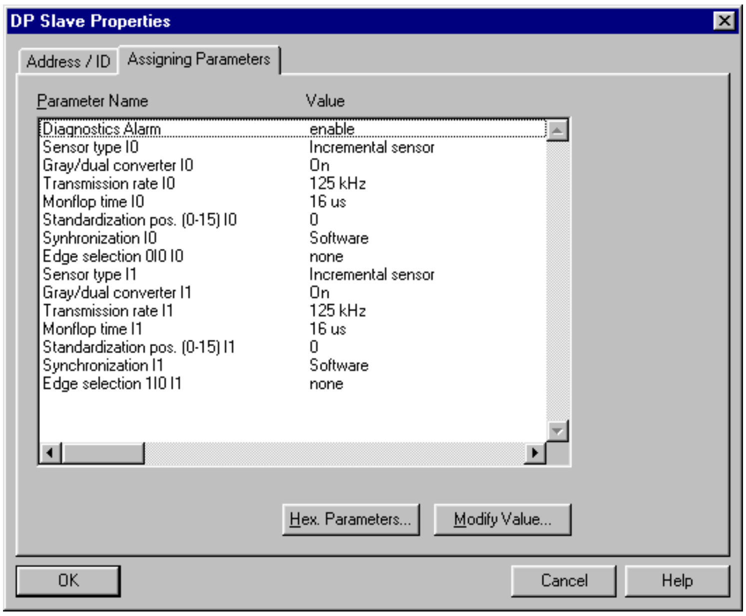
When parameterizing the starting address, take care to ensure that the starting address for the outputs is the same as that for the inputs.

M7-FM with IM178

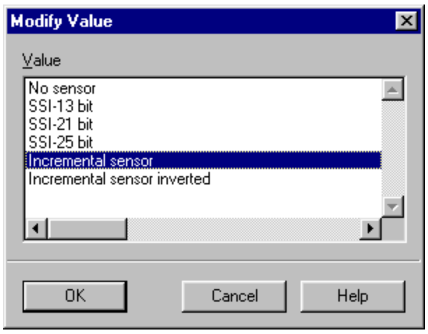
<table>
<tr><th colspan="3">Application...</th><th>IM178 the ...</th><th>Profibus address</th><th colspan="2">Local M7 axis number</th></tr>
<tr><td rowspan="16">32 axes</td><td rowspan="8">16 xes</td><td rowspan="4">8 axes</td><td>1.</td><td>3</td><td>1</td><td>2</td></tr>
<tr><td>2.</td><td>4</td><td>3</td><td>4</td></tr>
<tr><td>3.</td><td>5</td><td>5</td><td>6</td></tr>
<tr><td>4.</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td rowspan="12"></td><td>5.</td><td>7</td><td>9</td><td>10</td></tr>
<tr><td>6.</td><td>8</td><td>11</td><td>12</td></tr>
<tr><td>7.</td><td>9</td><td>13</td><td>14</td></tr>
<tr><td>8.</td><td>10</td><td>15</td><td>16</td></tr>
<tr><td rowspan="8"></td><td>9.</td><td>11</td><td>17</td><td>18</td></tr>
<tr><td>10.</td><td>12</td><td>19</td><td>20</td></tr>
<tr><td>11.</td><td>13</td><td>21</td><td>22</td></tr>
<tr><td>12.</td><td>14</td><td>23</td><td>24</td></tr>
<tr><td>13.</td><td>15</td><td>25</td><td>26</td></tr>
<tr><td>14.</td><td>16</td><td>27</td><td>28</td></tr>
<tr><td>15.</td><td>17</td><td>29</td><td>30</td></tr>
<tr><td>16.</td><td>18</td><td>31</td><td>32</td></tr>
</table>

Table 3-2 Default specification of the configuration for M7-FM with IM178 on the M7-FM

To parameterize the two channels, change to the "Parameterization" tab.



To change the parameter value, select the parameter name and then double click on it or click on the "Change value …" button. Example of changing the encoder type:



The parameter data for the channels should be allocated as follows:

Parameter name	**Value**
Diagnosis alarm	activate
Sensor type	user-defined
Gray / dual converter	on
Baud rate	user-defined
Monoflop time	user-defined
Position standardization	user-defined
Synchronization (for incremental sensor only)	Digital input I1 and zero mark
Transition selection	Rising edge

Repeat these stages for each additional IM178.

Finally, all that remains is to save the S7 hardware configuration and load it into your controller.

NOTES

Bearing in mind that you need to configure a Profibus address and a peripheral address for the IM178, it remains an M7-FM axis from the point of view of the S7-CPU.
This means that in the S7-CPU data block "GMC_DB_ORG", you need to parameterize in the "Axis descriptions" section the axis type = 1 (M7 axis), i.e. the M7 number which belongs to the axis and the local axis number for the M7.

The local axis number for the M7 does not need to be identical to the global axis number (also see Chap. 5.1).

3.3 Motion Control for MASTERDRIVES MC

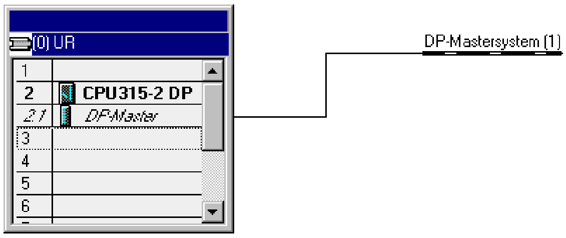
3.3.1 Configuring the PROFIBUS Connection between SIMATIC S7 and MASTERDRIVES MC

Requirements for PROFIBUS Communication

The following components are required for PROFIBUS communication:

- ♦ S7-CPU with PROFIBUS interface
- ♦ MASTERDRIVES MC with PROFIBUS module
- ♦ "Drives" object manager, which enables the integration of the MASTERDRIVES MC after installation in the S7 hardware configuration
- ♦ PROFIBUS connecting cable between the S7-CPU and the MASTERDRIVES MC

Creating a PROFIBUS System



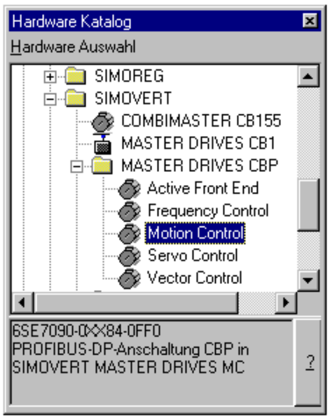
If you install an S7-CPU with a PROFIBUS interface in the S7 hardware configuration, a PROFIBUS system is automatically created with the properties Profile = DP and Baud Rate = 1.5 Mbit/s.

An external connection is also displayed ("DP Master System (1)"). The MASTERDRIVES MC units can then be connected.

NOTE

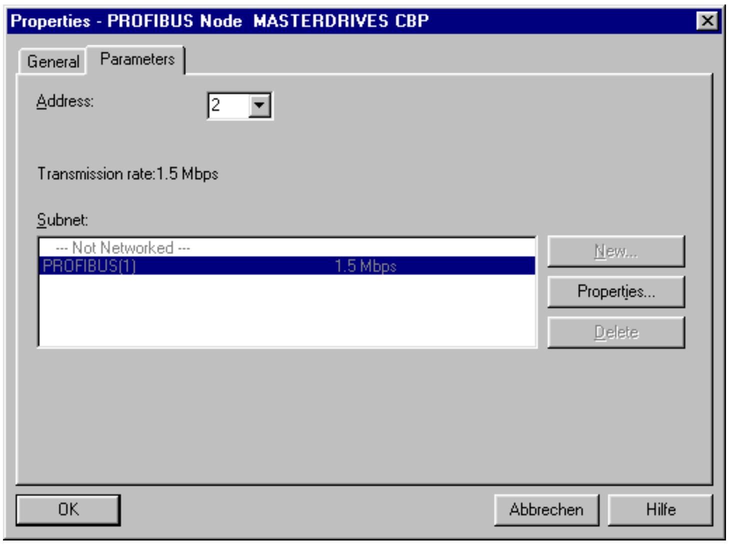
The baud rate can be increased manually up to 12 Mbit/s.

Inserting a MASTERDRIVES MC

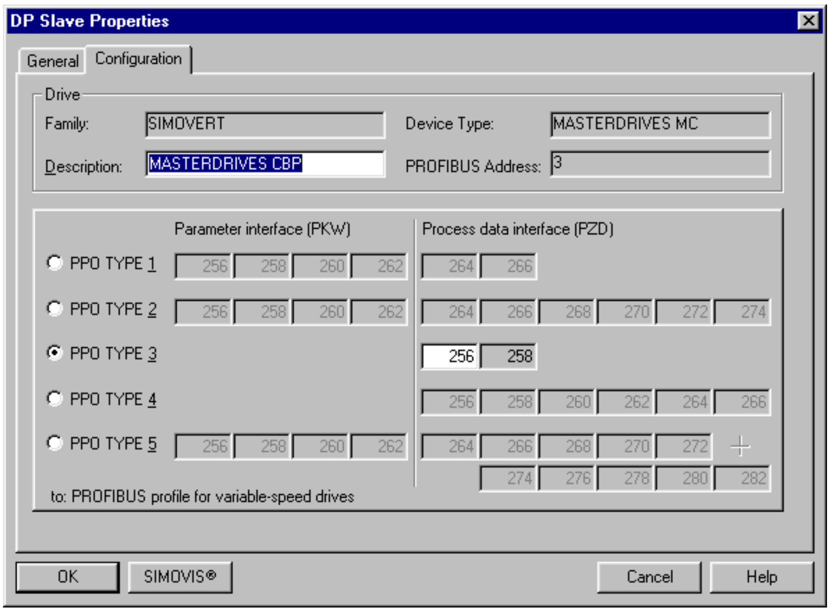


- Open the hardware catalog
- Select the PROFIBUS line
- In the hardware catalog, select PROFIBUS-DP → SIMOVERT → MASTERDRIVES CBP
- Double-click *Motion Control*

Defining the Properties



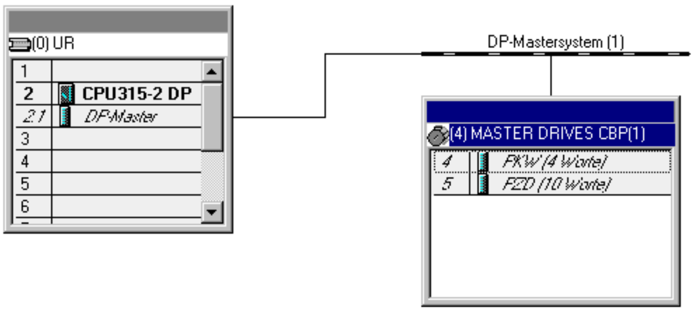
The next free PROFIBUS address is automatically selected by the system. Accept the entry with OK



Select PPO Type 5. The next free I/O addresses for the PKW component and the PZD component are initialized by the system. Accept the entries with OK.

NOTE

When you create the axis descriptions in data block GMC_DB_ORG, you must enter the parameters defined here in "PROFIBUS address", "PKW component for MCT" and "PZD component for MCT".



You will now see a MASTERDRIVES MC inserted in the PROFIBUS system. Repeat the steps above for each further MASTERDRIVES MC. Finally, save the S7 hardware configuration and load it into the S7-CPU.

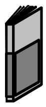
Connecting the MASTERDRIVES MC

Enter the PROFIBUS address on the MASTERDRIVES MC as follows (e.g.: PMU):

- ♦ Set parameter P60 to 4 (module settings).
- ♦ Set parameter P918.1 = Profibus address.
- ♦ Set parameter P60 to 1 ("Parameter" menu).

Establish the PROFIBUS connection between the S7-CPU and the MASTERDRIVES MC.

If the S7-CPU is in RUN mode, the flashing of all 3 LEDs on the PROFIBUS module indicates that the MASTERDRIVES MC bus connection is operational.



Further Information

Further information about PROFIBUS communication is provided in the MASTERDRIVES Compendium.

3.3.2 Configuration and parameterization of the MASTERDRIVES MC converter (MCT)

Introduction

The GMC-BASIC standard software only works together with the F01 technology option on the MASTERDRIVES MC. For this, you have to enable the F01 technology option. Successful enabling can be checked by reading parameter n978 (Parameter n978=1: Enabled). A standard interconnection in the MASTERDRIVES MC is also a prerequisite for the technology option.

When you install SIMOVIS/DriveMonitor, the standard interconnection is set up in the form of a script file or download file for operation with the technology option. Additional script files are also installed, e.g. for interconnecting the optional area. The script files are loaded into the converter with the SIMOVIS/DriveMonitor tool.

Software requirements

To be able to load the standard interconnection, you must have installed the SIMOVIS/DriveMonitor tool on your PG/PC. SIMOVIS/DriveMonitor is not included in the scope of supply of Laden GMC-BASIC.

- ♦ PG/PC with SIMOVIS/DriveMonitor >=V5.3

Hardware requirements

- ♦ MASTERDRIVES MC from V1.2 upwards with Profibus board
- ♦ Connecting cable for PG/PC – MASTERDRIVES MC (USS)

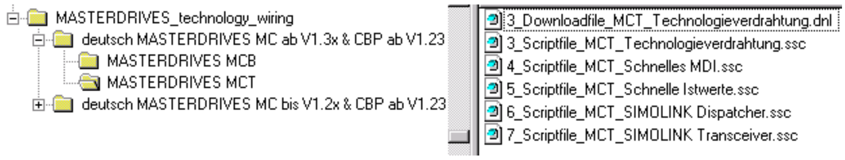
Installation of standard interconnection

When you install SIMOVIS/DriveMonitor, the standard interconnection is installed as a script file and as a download file. The additional script files described special settings. You must adapt these, however, to the components you use (e.g. driver) or to your configuration (e.g. transceiver address) (see Section 3.4).

NOTE

All the script files and a download file are also stored on the CD "GMC-BASIC". These are identical with those which SIMOVIS/DriveMonitor has installed on your computer.

Contained on the CD "GMC-Basic":

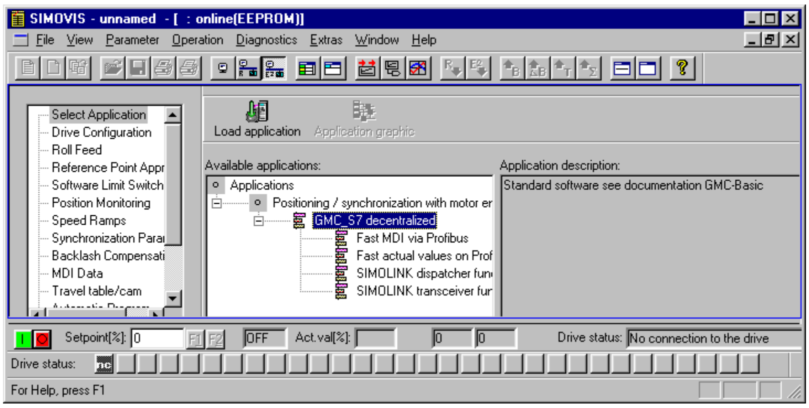


Loading interconnection to the MASTERDRIVES MC

The standard interconnection is stored in file "3_Scriptfile_MCT_ Technologieverdrahtung.ssc" or "3_Downloadfile_MCT_ Technologieverdrahtung.dnl". Changes must **not** be made in these files under any circumstances!

Use the optional area for transferring the MID set, 0, and for reading actual values. Load the script files "4_Scriptfile_MCT_Schnelles MDI.ssc" and "5_Scriptfile_MCT_Schnelle Istwerte.ssc" in this way.

The script files or download file are transferred with SIMOVIS/DriveMonitor. A "script-file browser" is displayed in the middle window by means of "Select application". Select, for example, GMC_S7_decentralized and then press the Load Application button to load the standard interconnection to the converter. For further instructions, please refer to the SIMOVIS/DriveMonitor description.



3.4 Configuring the SIMOLINK Connection between the MASTERDRIVES MC Units

Application

A SIMOLINK connection is only necessary if high-speed data exchange is required between the individual MASTERDRIVES MC units, e.g. for the synchronization technology function.

The SIMOLINK connections between the individual MASTERDIVES MC (MCT) units are configured by modifying the appropriate parameter settings on the actual units.

Further Information

A simple example is provided below, in order to describe the basic procedure. You will find a detailed description of SIMOLINK communication in the "MASTERDRIVES MC Compendium".

In a SIMOLINK ring, one station performs the master function and is referred to below as the dispatcher. All other stations are slaves, and are referred to as transceivers.

Example

In the following example, the synchronization technology function is used with 2 axes. Axis 1 is the dispatcher and transmits the position setpoint to the SIMOLINK. Axis 2 is the transceiver. It receives the position setpoint of axis 1 from the SIMOLINK and transmits it to the synchronization system. Consequently, axis 2 follows the movements of axis 1.

Settings on the dispatcher (axis 1):

- ♦ Define axis 1 as dispatcher
  Station address = 0 P740 = 0
- ♦ Define the number of channels
  1 channel for position setpoint P745 = 1
- ♦ Define the SIMOLINK cycle time
  Technology cycle = 3.2 ms P746 = 3.2
- ♦ Assign position setpoint to channel 0
  P751.1 = KK310 and P751.2 = KK310

Settings on the transceiver (axis 2):

- ♦ Define axis 2 as transceiver
  Station address = 1 P740 = 1
- ♦ Define read address, receive data
  from station 0, channel 0 P749 = 0.0
- ♦ Assign position setpoint from SIMOLINK channel 0 to synchronization setpoint control
  U600.2 = KK7031

All the necessary parameters are now configured. When the SIMOLINK connection is established between the dispatcher and the transceiver, all three LEDs should flash on the SIMOLINK modules.

All that remains to do is to set the synchronization parameters in axis 2 (setpoint control, etc.) and activate synchronization mode.

4 Data Structure

General Information

The data requests are implemented using tasks. If a task is transferred, e.g. for reading the machine data of an axis, the task is handled independently by the "GMC-BASIC" standard software.

The segmentation of larger volumes of data is handled by the technology. This makes it very easy to transfer large amounts of data to or from the technology. The user program is relieved of the task of organizing the data communication, because the technology stores and fetches the data from the location defined in the task.

Advantages of Data Exchange

This type of data exchange provides the following advantages:

- Data exchange with the technology is simple, whether the target system is M7-FM or MASTERDRIVES MCT
- The simple structures make the user program shorter. This releases memory capacity for other purposes
- Rapid commissioning and fault localization are facilitated by simple structures and clearly defined interfaces
- Short development time for the actuation software for the GMC axes
- Easy, high-speed access to information in the event of a servicing assignment (version numbers, release levels, etc.)
- Secure data exchange through standard verification routines

Principle of Data Exchange

Whereas function blocks are used to implement the necessary runtime program, the data blocks represent the application interface.

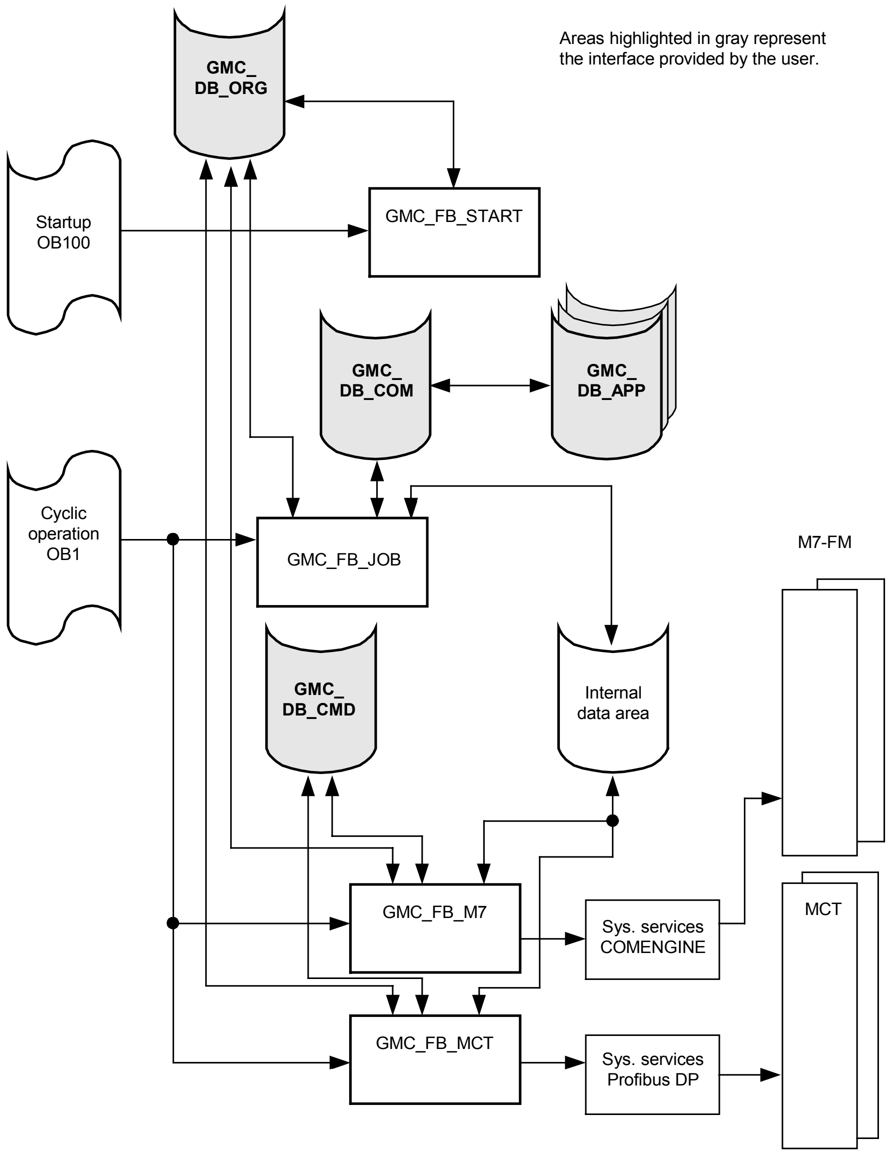


Fig. 4-1 Overview of Data Structure

Description of the Operating Principle

The organization of the system is defined in data block GMC_DB_ORG and initialized by startup block GMC_FB_START. The GMC_FB_JOB block is responsible for data communication between S7 and M7/MCT. This block is controlled by GMC_DB_COM. The necessary technology data are stored in the GMC_DB_APPs. GMC_DB_CMD represents the centralized application interface for controlling the technology. Blocks GMC_FB_MCT and GMC_FB_M7 implement the connection between the S7 system and the MCT or M7 target system using the corresponding communication services.

5 Parameter and Application Interface: Data Blocks

Contents

In this chapter you will find a description of the parameter and application interface (data blocks) of the "GMC-BASIC" standard software.

Application Interface

The application interface comprises a range of data blocks which provide all the functions required for the operation of the technology.

In order to define the capacity of your application, you must configure data block GMC_DB_ORG once after installing your S7 project.

Data block GMC_DB_COM provides the functions required for data exchange between your user program on the SIMATIC S7 system and the technology.

You can implement your own individual data management system in the GMC_DB_APP task blocks.

The GMC_DB_CMD data block allows you to operate each of your axes centrally, independent of the target system used for the technology (M7/MCT) and independent of the total number of axes.

Further Information

This chapter describes only those areas necessary to get you started quickly, since the system is largely self-configuring.
However, if you are not satisfied with the standard configuration of the system (e.g. if you want to change the data block number), or if you need additional system diagnostics, you will find a detailed description of the software in the chapter entitled "Expert Knowledge/Tips".

5.1 Organization Data Block GMC_DB_ORG (DB100)

Task of the Data Block

Data block GMC_DB_ORG (**org**anization) is required for the organization of the total system, and is subdivided into different sections. It contains the internal information required for the management of the "GMC-BASIC" standard software, the configuration that you enter in order to determine the capacity of your application, and the areas for displaying system states and version numbers.

Your task is to configure GMC_DB_ORG. Once you have done this, you will normally never need to concern yourself with the contents of GMC_DB_ORG again.

GMC_DB_ORG is supplied with the S7 software, and is configured in accordance with the programming examples.

Structure

GMC_DB_ORG is subdivided into different sections. Most of the sections are configured automatically by the system. In certain sections, you need to enter data in accordance with the type and number of axes used in your system.

DBW0	For internal use	
	"General parameters" section	X_general
	"MCT description" section	X_mct
	"M7 description" section Configured by the user	X_m7
	"Displays" section Outputs	X_indication
	"Software versions" section Outputs	X_version
	"Axis description" section Configured by the user	X_axes

Only the sections highlighted in gray are relevant in this chapter.

Further Information

The data relevant to the GMC-OP-OAM standard software are also included in the "general parameters" section. You will find more information in the "GMC-OP-OAM Standard Software" tab.

M7 Description Section

Up to 4 M7-FMs can be operated in an S7-400 system with the "GMC-BASIC" standard software.

You only need to configure this section if you want to operate the technology centrally on one or more M7-FMs.

1. M7-FM

DBWw	I/O address of M7-FM	DEC	X_m7_1.i_addr
DBWw+2	Internal configuration data		

2. M7-FM

DBWx	I/O address of M7-FM	DEC	X_m7_2.i_addr
DBWx+2	Internal configuration data		

3. M7-FM

DBWy	I/O address of M7-FM	DEC	X_m7_3.i_addr
DBWy+2	Internal configuration data		

4. M7-FM

DBWz	I/O address of M7-FM	DEC	X_m7_4.i_addr
DBWz+2	Internal configuration data		

A unique I/O address must be assigned to every I/O module in a SIMATIC S7 system, including the M7-FM. You define the I/O address of the M7-FM with the STEP7 "hardware configuration" tool during configuration of the total system. This address must be specified as "I/O address of M7-FM" in the "M7 description" section.

The M7-FMs must be configured contiguously from 1 to 4 in this section (referred to in the text as the M7 number). Specifying "0" for the I/O address signifies "M7-FM does not exist".

Axis Description Section

This is the centralized section in which you define the capacity of your application.

The "axis description" section contains the parameters required for an axis and is repeated contiguously according to the total number of axes in your system.

Address	Content	Format	Symbol
DBWx	Total number of axes	DEC	X_axes. i_number_axis
Axis 1			
DBWx+2	Type of axis: 0 = axis does not exist; 1 = M7 axis; 2 = MCT axis 3 = M7/MCB axis	DEC	X_axes.{X_axis1}. i_axis_type
DBWx+4	Assigned internally	DEC	
DBWx+6	M7 number with which axis is associated	DEC	X_axes.{X_axis1}. i_m7_no
DBWx+8	Local axis number for M7	DEC	X_axes.{X_axis1}. i_log_axis_no
DBWx+10	PROFIBUS address of MCT	DEC	X_axes.{X_axis1}. i_profibus_addr
DBWx+12	I/O address of parameter interface (PKW) for MCT/B	DEC	X_axes.{X_axis1}. i_ppkw
DBWx+14	I/O address of process data interface (PZD) for MCT/B	DEC	X_axes.{X_axis1}. i_ppzd
DBWx+16	Reserved		X_axes.{X_axis1}. i_res7
Axis 2			
DBWx+18	Type of axis: 0 = axis does not exist; 1 = M7 axis; 2 = MCT axis 3 = M7/MCB axis	DEC	X_axes.{X_axis2}. i_axis_type
	⋮		
Last axis			
DBWx+n	Type of axis: 0 = axis does not exist; 1 = M7 axis; 2 = MCT axis 3 = M7/MCB axis	DEC	X_axes.{X_axisn}. i_axis_type

The standard setting of GMC_DB_ORG when supplied (see example programs) is pre-configured for 1 (example for MCT) or 8 (example for M7-FM) axes.

Total Number of Axes

The "total number of axes" determines the number of axis descriptions in GMC_DB_ORG.

The axis number, which is subsequently required as an address for communication specifically with this axis, is derived from the number of the axis description. That means the xth axis description assigns the absolute axis number x to the described axis.

NOTE

UDT 1009 ("GMC_UDT_DB_ORG_AXIS") is also supplied with the software. You can use it to add structured axis descriptions.

At least the same number of axis descriptions as the "total number of axes" must be entered.

Type of Axis

The axes must be configured contiguously from 1 to the "total number of axes". Gaps can only be inserted by specifying "axis does not exist", although this non-existent axis is also counted in order to generate the "total number of axes".

You can assign any axis number to the M7 or MCT target system. That means if you have a certain number of axes in your system, you can decide, independent of the target system, which axis has which absolute axis number.

M7 axis means that the technology is installed centrally on an M7-FM. MCT axis means that the technology is installed decentrally on a MASTERDRIVES MC. M7/MCB axis means that the technology is installed centrally on an M7-FM, but a PROFIBUS connection also exists to the MASTERDRIVES MC, and that this connection can be used to communicate (HA2000 task) with the basic functionality of the MCB (no technology functions).

M7 Number with which the Axis is Associated

The "M7 description" section mentioned that up to 4 M7-FMs can be used simultaneously. For the axis description, you now need to specify the M7 number with which the axis is associated.

Local Axis Number for M7

Unlike the MCT, more than one axis can be configured on an M7-FM. In order to establish an association between the global axis number (always referred to as the axis number in the text) and the local axis number on the M7-FM, the "local axis number for M7" must be specified in the axis description. The local axis numbers are numbered consecutively from 1 to n on the M7-FM (according to the axis expansion of the M7-FM).

It is worth mentioning again at this point that only the global axis number is used to specify the source/destination for communication with the axes, independent of the M7 or MCT target system.

PROFIBUS Address of MCT/B

If the technology is installed decentrally on the MASTERDRIVES MCT, it is connected to the SIMATIC S7 system via PROFIBUS DP. Each of the PROFIBUS stations requires a unique address and must be specified in the axis description as "PROFIBUS address of MCT". You define this PROFIBUS address when configuring the total system with the STEP7 "hardware configuration" tool (see Chapter 3).

I/O Address of Parameter Interface (PKW) and I/O Address of Process Data Interface (PZD) for MCT/B

You also configure the I/O start addresses for PROFIBUS-DP using the STEP7 "hardware configuration" tool (see Chapter 3).

These I/O start addresses, which are defined separately for the parameter interface (PKW) and the process data interface (PZD), must be entered for each MCT/B axis in the axis description.

NOTE

If you specify "M7/MCB axis" as the "type of axis", the M7 and MCB parameters must be filled in completely.
If you specify "axis does not exist" as the "type of axis", you do not need to enter any further details in the axis description.
In all other cases, the axis description must be configured according to the "type of axis" selected.

5.2 Communication Data Block GMC_DB_COM (DB118)

Task of the Data Block

GMC_DB_COM (**com**munication) represents the application interface for data communication between the user (S7) and the technology. The term user can refer to you or to an HMI application (OP25/OP37, etc.). Irrespective of whether the target system is an M7-FM or a MASTERDRIVES MCT, or whether your capacity is large or small in terms of the number of axes, all data requests are handled centrally via the GMC_DB_COM data block.

Eight "task boxes" are provided in GMC_DB_COM to help coordinate different users during communication with the technology.

You only need to set up the GMC_DB_COM data block if you want to implement data communication with the technology on your S7 system. If data input/output takes place exclusively via standard user interfaces, you do not need to familiarize yourself with the features of GMC_DB_COM.

Each task box consists of an input area and an output area. You use the input area to specify and start a data communication task. The output area represents the current status of the task and signals the end of the data transfer or indicates any errors that occur by specifying the error number.

GMC_DB_COM is supplied with the software. If it is not found when the S7-CPU is started, it is set up automatically.

NOTE

If you attach importance to the use of symbolic names for the data in GMC_DB_COM, you should use the version supplied instead of letting the system set up the data block automatically.

CAUTION

The GMC-OP-OAM standard software uses application box 1. This application box is therefore not available for your application.

Structure

GMC_DB_COM is subdivided into nine sections. Sections 1 to 8 correspond to the 8 task boxes. Section 9 represents a copy of the last task entered in one of the 8 task boxes.

Section 9 is therefore only used in order to log the assignment of task boxes 1 to 8 (e.g. for troubleshooting).

Task box 1
Input box

DBW0	DB number GMC_DB_APP1	DEC	X_app1.I_db_no
DBW2	DBW number GMC_DB_APP1	DEC	X_app1.I_dbw_no

Output box

DBW4	DBW number of subtask on error	DEC	X_app1. I_dbw_no_job
DBW6	Error number 1	HEX	X_app1.I_fault1
DBW8	Error number 2	HEX	X_app1.I_fault2

Task status

DBB10	Reserved	BIN	X_app1.Y_res
DBX11.0	No task running	BIN	X_app1.B_no_job
DBX11.1	Reserved	BIN	X_app1.B_res1
DBX11.2	Task running	BIN	X_app1.B_job_activ
DBX11.3	Task terminated without error	BIN	X_app1.B_job_ready
DBX11.4	Task terminated with error	BIN	X_app1.B_job_false
DBX11.5	Reserved	BIN	X_app1.B_res5
DBX11.6	Reserved	BIN	X_app1.B_res6
DBX11.7	Reserved	BIN	X_app1.B_res7

2nd task box

DBW12	DB number GMC_DB_APP1	DEC	X_app2.I_db_no

9th task box (not active box, just copy)

DBW96	DB number GMC_DB_APP1	DEC	X_app9.I_db_no

Coordination of the Task Boxes

The 8 task boxes have equal priority. If several tasks are active simultaneously, they are processed consecutively, i.e. from task box 1 to task box 8. While any task is being processed, all other queued tasks must wait for their turn.

NOTICE

If you allow different and mutually independent programs to use individual task boxes simultaneously, box use cannot be coordinated simply through box assignment. In other words, the function which has initiated a task first will not necessarily receive the task acknowledgement as the acknowledgement cancels the box assignment. Another program may therefore have started using the box before the first function has had the chance to evaluate the acknowledgement.
In such cases it is essential to "lock" the assignment using a semaphore or equivalent mechanism.

NOTE

The number of task boxes and their assignment are not subject to any rules. However, you should be aware that "other users", such as human-machine interfaces, need at least one free task box for communication. It has even proved useful to keep one task box free for servicing purposes. You do not need make changes to the PLC program in order to keep one task box free.

Task Box Input Area

The pointer to the data block containing the task (GMC_DB_APP1) must be entered in the input area of the task box. The data block number must be entered in "DB number GMC_DB_APP1" and the data word number must be entered in "DBW number GMC_DB_APP1". Since entering the DB number also initiates the task, it must be entered after or at the same time as the DBW number. At the end of task processing, the input area is automatically deleted to allow a new task to be started subsequently in this task box. A deleted pointer thus indicates that the task box is free.

CAUTION

The pointer must be entered dynamically (once), otherwise the task is initiated repeatedly.

NOTE

Before a task is started by entering a pointer, the output area should be deleted dynamically in order to facilitate management of the task checkback signals by the user program (otherwise a signal edge evaluation is required in order to determine whether the status of the output area refers to the last task or the current one).
The pointer in the 9th task box is not deleted, since this task box only contains a copy of the last task which was processed.

Task Box Output Area

The output area of the task box is used for task checkback signals and fault diagnostics.

If an error occurs during processing of the task, the following diagnostic information is available:

- DBW number of the subtask on error:
  The task can consist of several subtasks. While several subtasks are being processed, the DBW number of the current subtask is always entered in GMC_DB_APP1. The entry is deleted when the tasks are terminated successfully.
  If an error occurs, the DBW number of the faulted subtask remains entered, in order to facilitate diagnostics.
- Error number 1:
  Error number 1 is the actual error message and provides information about the reason for the occurrence of the error. In the "Error Descriptions" documentation you will find a list of all error numbers complete with the error causes and remedies.
- Error number 2:
  Error number 2 is used in many error messages to specify the cause of the error more accurately.
- Task status
  The task status indicates the current status of the task as follows:
  - No task running:
    This bit is "0" while no task is running
  - Task running:
    If the task is enabled for processing, this bit is set to "1" for the complete duration of task processing.
  - Task terminated without error:
    If the task has been completed successfully, this bit is set to "1". This should be the actual result of a task.
  - Task terminated with error:
    If an error occurred during task processing, this bit is set to "1". The information in "DBW_number of subtask on error", "error number 1" and "error number 2" is available for further diagnosis of the error.

NOTE

The task status is retained until the next time the task is initiated.

5.3 Task Data Block GMC_DB_APP

Task of the Data Block

While a pointer to GMC_DB_APP is set in GMC_DB_COM in order to start a task, the GMC_DB_APP data block contains the actual task description.

You only need to create the GMC_DB_APP data blocks if you want to implement data communication with the technology on your S7 system. If data input/output takes place exclusively via standard user interfaces, you do not need to familiarize yourself with the creation of task descriptions.

The number of GMC_DB_APPs you need depends on the scope and individual organization of the tasks. You can configure several tasks in one GMC_DB_APP or create a separate GMC_DB_APP for each task.

Structure

A task description consists mainly of a task header, describing the type of data and direction of transfer, and the actual task data. There are a large number of different tasks, although the basic features are always the same.

To enable several tasks to be processed in succession on a single task initiation, the task description includes a preamble which contains only the number of subtasks.

The contents of the task header are defined to a large extent. The syntax for all possible tasks is described in the "Task Description". When you create a task, all you need to do is to enter the task header in accordance with the description, define the "DB no." and "DBW no." pointers, specify the axis number and data direction, provide an adequately sized memory area for the task data, and enter the task data.

NOTE

Areas highlighted in dark gray are defined by the user. Areas highlighted in light gray are used to set the communication options. Areas which are not highlighted have a permanent setting and may not be modified.

Pointer for GMC_DB_COM

Data block GMC_DB_APP1
Preamble for management of multiple subtasks

DBWx-2	Number of subtasks	DEC

Data block GMC_DB_APP1:
General task header

DBWx	Main task number	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer	DEC
DBWx+6	Data type	DEC
DBWx+8	Number of elements	DEC
DBWx+10	DB no. GMC_DB_APP2	DEC
DBWx+12	DBW no. GMC_DB_APP2	DEC

or

Data block GMC_DB_APP2:
Technological task header (task-dependent)

DBWy	Number of DWs								DEC
DBBy+2	E	A	0	0	0	0	0	0	BIN
DBBy+3	Task identifier								DEC
DBWy+4	DB no. GMC_DB_APP3								DEC
DBWy+6	DBW no. GMC_DB_APP3								DEC
DBWy+8	Extension 1 (Z1)								DEC
DBWy+10	Extension 2 (Z2)								DEC
DBWy+12	Extension 3 (Z3)								DEC
DBWy+14	Extension 4 (Z4)								DEC
DBWy+16	Extension 5 (Z5)								DEC
DBWy+18	Extension 6 (Z6)								DEC

Data block GMC_DB_APP2 or GMC_DB_APP3:
Task data

DBWz	Task data

Number of Subtasks The number of subtasks defines the number of task descriptions (subtasks) which follow. This feature allows you to implement a task list in which all tasks are processed serially on a single task initiation (see GMC_DB_COM).

For example, if you need to transfer data to various axes after switching on the control, you can create an appropriate task list and execute it with a single task initiation.

5.3.1 General Task Header

Task The task header can be short or long, depending on the complexity of the task.

All tasks have at least the general task header in GMC_DB_APP1 (DBWx to DBWx+8).

The task data for a single task are stored in GMC_DB_APP2. In order to address this memory area, the pointer to GMC_DB_APP2 (DBWx+10 and DBWx+12) is added to the general task header in GMC_DB_APP1.

Components of the Task Header

The general task header comprises:

- ♦ Main task number (DBWx):
  The main task number is defined for each task. For single tasks, the main task number also defines the direction of data transfer as follows:
  - Positive main task number → input (to technology)
  - Negative main task number → output (from technology)
- ♦ Absolute axis number (DBWx+2):
  When the "axis descriptions" section was configured in data block GMC_DB_ORG, an absolute axis number was assigned to all axes. This axis number is now required as the source/destination parameter for task processing.
- ♦ Element pointer (DBWx+4):
  The "element pointer" specifies the element number at which the data area in GMC_DB_APP2 begins.
  The definition of the element pointer is only variable in the case of single tasks. In all other cases, the element pointer is permanently defined as "1".
- ♦ Data type (DBWx+6):
  The "data type" defines the size of an element (Word, DWord, etc).
  The data type is defined according to the task selected.
- ♦ Number of elements (DBWx+8):
  "Number of elements" specifies the length of the data area of type "Data type" in GMC_DB_APP2.
  It is only possible to specify a variable number of elements for single tasks. In all other cases, the number of elements is defined according to the selected task. DB no. GMC_DB_APP2 and DBW no. GMC_DB_APP2:
  Pointers to GMC_DB_APP2 (task data or technological task header)
- ♦ DB no. GMC_DB_APP2 and DBW no. GMC_DB_APP2:
  Pointers to GMC_DB_APP2 (task data or technological task header)

Example of a Single Task

Let's assume that the data for a technological function comprise 7 elements of the data type "word", and that you want to transfer elements 3 to 6 from your S7 application to the technology (input).

You need to define the following parameters:

- The necessary main task number
- The axis (axis number) to which the data are to be transferred
- Element pointer = 3, because element 3 is the first element to be transferred
- Data type = 2, because the elements are one word in size
- Number of elements = 4, because you want to transfer 4 elements (3, 4, 5 and 6)
- The DB no. and the DBW no. as pointers to the data of the 4 elements, and finally the data of the 4 elements

Data block GMC_DB_APP1:

DBWx	Main task number n	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **3**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **4**	DEC
DBWx+10	DB no. DB-APP2	DEC
DBWx+12	DBW no. DB-APP2	DEC

Data block GMC_DB_APP2:

DBWy	Value for element 3
DBWy+2	Value for element 4
DBWy+4	Value for element 5
DBWy+6	Value for element 6

If you want to request data from the technology (output), simply specify the main task number with a negative sign (-n). The other previously initialized parameters remain valid.

5.3.2 Technological Task Header

Task

More complex tasks require the addition of a technological task header, which must be entered in data area GMC_DB_APP2. The length of the technological task header depends on the task in question.

Components of the Task Header

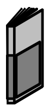
The technological task header comprises:

- Number of DWs (number of data words) (DBWy):
  The "number of DWs" is usually defined as "0" for all tasks with a technological task header. If a value greater than "0" is entered, the data input can be accelerated if certain special conditions are met (see below).
- I or Q and the task identifier of the technology (DBBy+2 and DBBy+3):
  I or Q defines the direction of data communication (I → input to technology; Q → output from technology). You can choose between I or Q for some tasks; in other cases, this parameter is fixed. The task identifier is also fixed and is used to indicate the meaning of the data to the technology.
- DB number GMC_DB_APP3 and DBW number GMC_DB_APP3 (DBWy+4 and DBWy+6):
  Pointers to the task data in data area GMC_DB_APP3.
- Extensions 1 to 6 (DBWy+8 to DBWy+18):
  These parameters are used to distinguish more finely between the meanings of a task. Part or all of these extensions are required, depending on the task. The contents are partly fixed and partly variable.

5.3.3 Task Data

Task

This is the actual data area from which the data are transferred to the technology (input) or from the technology (output). The length and assignment of task data differ according to the type of data (or task).



Further Information

This is the actual data area from which the data are transferred to the technology (input) or from the technology (output). The length and assignment of task data differ according to the type of data (or task).

5.3.4 Special Features of Task Management

General Information

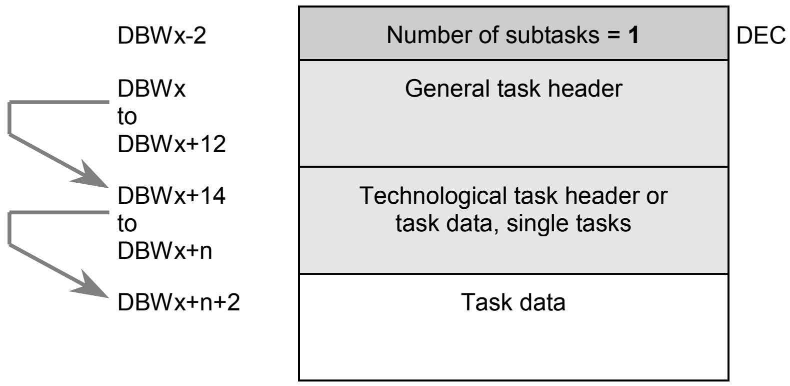
The individual areas represented in the task description are stored in different data blocks **exclusively** for the purposes of greater clarity (GMC_DB_APP1/2 and 3). The areas in the data blocks can be represented as described below, depending on the following conditions.

Only One Subtask

If only one subtask is formulated in a task ("number of subtasks" = 1), the general task header, the technological task header and the task data can be arranged contiguously in the same data block. The pointers (DB number, DBW number) must be defined correctly, however.

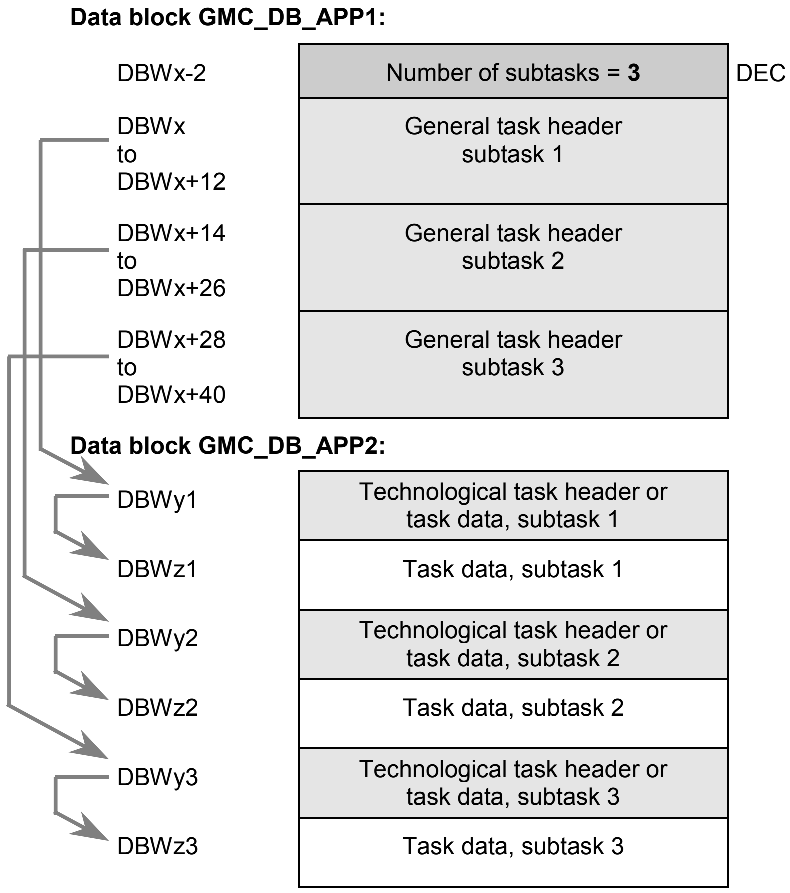
Example:

Data block GMC_DB_APP1:



Several Subtasks

If several subtasks are formulated in one task, the general task headers **must** be arranged contiguously. The pointers to the technological task headers and task data are arbitrary.



On the other hand, you can branch technological task headers and task data in any way you wish.

For example, you can store the technological task headers 1 to 3 in different data blocks, while storing the task data of subtasks 1 to 3 in other data blocks again.
Distribute task headers and data in the most suitable way for your application.

5.4 Control and Checkback Signals Data Block GMC_DB_CMD (DB117)

Task of the Data Block

Data block GMC_DB_CMD (**com**mand) represents the centralized application interface for the control and checkback signals exchanged cyclically between the S7 system and the technology. All control and checkback signals are handled centrally by the GMC_DB_CMD data block, whether the target system of the technology is an M7-FM or a MASTERDRIVES MCT, or whether your capacity is large or small in terms of the number of axes.

GMC_DB_CMD is not supplied with the software. When the S7-CPU is restarted, GMC_DB_CMD is automatically generated with the correct length (depending on the total number of axes specified in GMC_DB_ORG).

NOTE

If you attach importance to the use of symbolic names for the data in GMC_DB_CMD, you should create the data block yourself using UDT 1061 ("GMC_UDT_DB_CMD_AXIS") and UDT 1062 ("GMC_UDT_DB_CMD_AXIS_CONT"), which are supplied with the software, in accordance with the total number of axes you have configured.

The UDT 1062 is required from Version 2.0 of GMC-Basic. It describes the old value of the control signals for an axis.

Structure

GMC_DB_CMD contains a series of sections containing the control and checkback signals for each axis from 1 to the total number of axes.

Section for axis 1

Address	Content	Signals
DBW0 to DBW24	Reserved	
DBW26 to DBW44	Control signals for axis 1	axis_1.IN_1 to axis_1.IN_11
DBW46 to DBW52	Reserved	
DBW54 to DBW72	Checkback signals for axis 1	axis_1.OUT_1 to axis_1.OUT_11
DBW74 to DBW78	Reserved	

Section for axis 2

Address	Content	Signals
DBW80 to DBW104	Reserved	
DBW106 to DBW124	Control signals for axis 2	axis_2.IN_1 to axis_2.IN_11
DBW126 to DBW132	Reserved	
DBW134 to DBW152	Checkback signals for axis 2	axis_2.OUT_1 to axis_2.OUT_11
DBW154 to DBW158	Reserved	

Section for axis 3

Address	Content	Signals
DBW160 to DBW184	Reserved	

NOTE

From Version 2.0 of GMC-Basic, the control signals are checked for changes and only transferred in the event that the technology has changed. This means that after the last defined axis, a data range 20 bytes long is required for the **old control signal values** for each axis.

Control and Checkback Signals

The 20 bytes for the control signals are available to the user for controlling the technology. The 20 bytes of checkback signals provide the user with information about the current machining status of the technology.

The control and checkback signals are exchanged cyclically between the SIMATIC S7 system and the technology on the M7-FM or MASTERDRIVES MCT.

Further Information

A detailed description of the assignment and function of the control and checkback signals is provided in the central documentation entitled "Function Description – Positioning and Synchronization".

6 Function Blocks: Calling and Parameter Assignment

Contents

In this chapter you will find all the information about calling and assigning the parameters of function blocks.

Task of the Function Blocks

While the data blocks represent the application interface, the function blocks perform the actual management of communication between the application interface and the technology. They handle the tasks and ensure that the control and checkback signals are exchanged cyclically. They set the system to a defined initial state on startup and monitor all operator control functions for errors.

All function block calls are already configured and parameterized. All you have to do is copy the example program into your project (see the chapter entitled "Program/Configuration Examples"), configure the GMC_DB_ORG data block and load the project into the S7-CPU. Once you have entered the hardware configuration correctly, loaded the technology software (if you use an M7-FM), and established the necessary connections (PROFIBUS, SIMOLINK), your system is ready to run

6.1 Function Block GMC_FB_START (FB127)

Task

Function block GMC_FB_START is required for the system **start**up. It checks the entries in GMC_DB_ORG and sets the system to a defined initial state. It also sets up all the necessary data blocks with the required length.

This function block must be called up and parameterized once in OB100 (startup). It is not multi-instance-capable and requires a separate instance data block (GMC_IDB_START; default: DB127).

NOTE

You must configure GMC_DB_ORG correctly before you start up the system.

Block

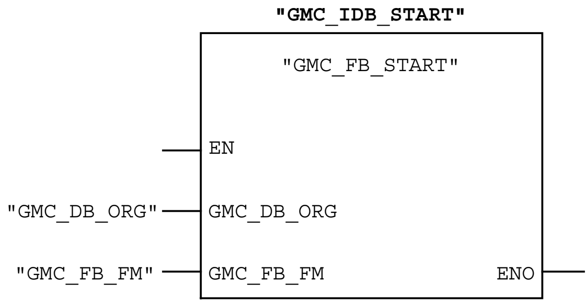


Fig. 6-1

Explanation of Input and Output Parameters

The following table describes the input and output parameters of function block "GMC_FB_START".

Name	Type	Data Type	Description	Default
GMC_DB_ORG	In	BLOCK_DB	Data block GMC-DB_ORG	DB100
GMC_FB_FM (only has to be configured if an M7-FM is used)	In	BLOCK_FB	Function block GMC_FB_FM (ComEngine)	FB124

6.2 Function Block GMC_FB_JOB (FB126)

Task

Function block GMC_FB_JOB is the centralized block for task management. It scans data block GMC_DB_COM continuously for tasks. It automatically establishes the connection to the appropriate target system (M7 or MCT) for the processing of queued tasks.

This function block must be called up and parameterized once during cyclic operation (e.g. in OB1). It is not multi-instance-capable and requires a separate instance data block (GMC_IDB_JOB; default: DB126).

Block

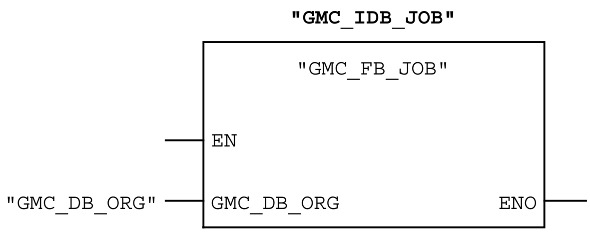


Fig. 6-2

Explanation of Input and Output Parameters

The following table describes the input and output parameters of function block "GMC_FB_JOB".

Name	Type	Data Type	Description	Default
GMC_DB_ORG	In	BLOCK_DB	Data block GMC-DB_ORG	DB100

6.3 Function Block GMC_FB_MCT (FB125)

Task

Function block GMC_FB_MCT is required for communication with the MASTERDRIVE MCT via PROFIBUS-DP. It uses internal services in the form of functions (FC) to handle the data transfer across the PROFIBUS-DP interface. This function block must be called up and parameterized once during cyclic operation (e.g. in OB1). It is not multi-instance-capable and requires a separate instance data block (GMC_IDB_MCT; default: DB125).

Block

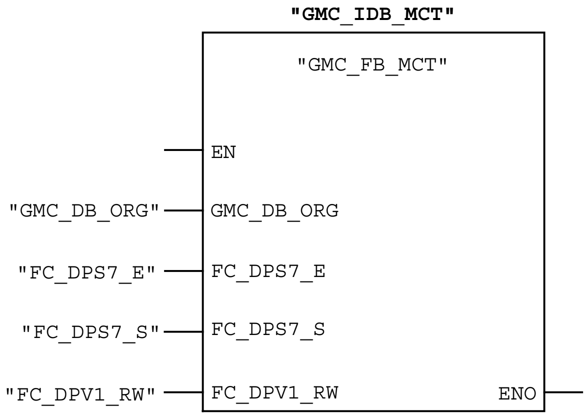


Fig. 6-3

Explanation of Input and Output Parameters

The following table describes the input and output parameters of function block "GMC_FB_MCT".

Name	Type	Data Type	Description	Default
GMC_DB_ORG	In	BLOCK_DB	Data block GMC_DB_ORG	DB100
FC_DPS7_E	In	BLOCK_FC	Function FC_DPS7_E	FC22
FC_DPS7_S	In	BLOCK_FC	Function FC_DPS7_S	FC21
FC_DPV1_RW	In	BLOCK_FC	Function FC_DPV1_RW	FC20

6.4 Function Block GMC_FB_M7 (FB121)

Task

Function block GMC_FB_M7 is required for communication with the M7-FM(s). It uses internal services in the form of a function block (GMC_FB_FM) for communication with the M7-FM. The function block is designed to allow the operation of up to 4 M7-FMs.

This function block must be called up and parameterized once in the startup cycle (OB100), once in the process interrupt (OB40) cycle and once during cyclic operation (e.g. in OB1). It is not multi-instance-capable and requires a separate instance data block (GMC_IDB_M7; default: DB121).

Block

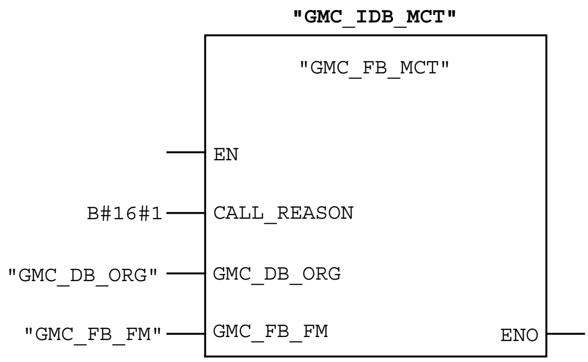


Fig. 6-4

Explanation of Input and Output Parameters

The following table describes the input and output parameters of function block "GMC_FB_M7".

Name	Type	Data Type	Description	Default
CALL_REASON	In	BYTE	Specifies the number of the OB in which GMC_FB_M7 is called B#16#1 → OB1 B#16#64 → OB100 B#16#28 → OB40	
GMC_DB_ORG	In	BLOCK_DB	Data block GMC_DB_ORG	DB100
GMC_FB_FM	In	BLOCK_FB	Function block GMC_FB_FM (COMENGINE)	FB124

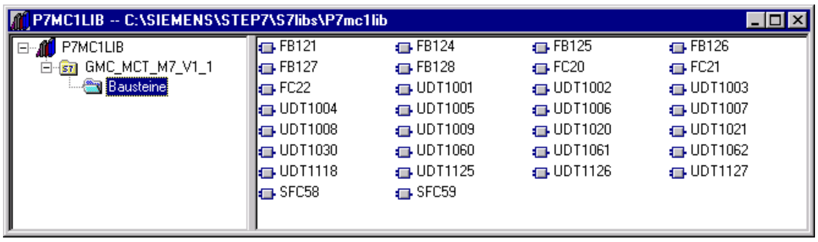
7 Program/Configuration Examples

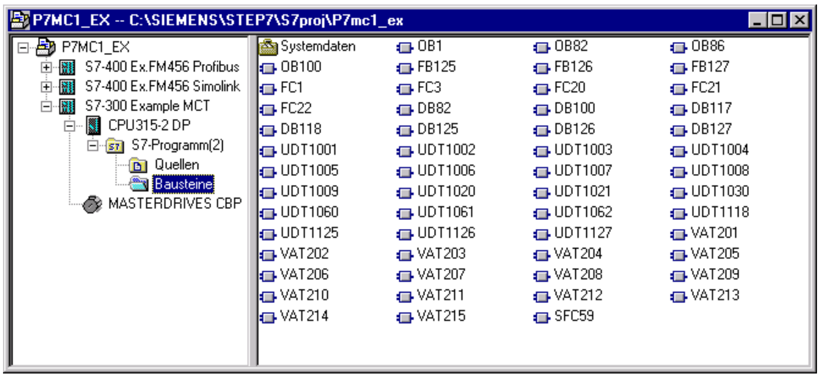
Contents

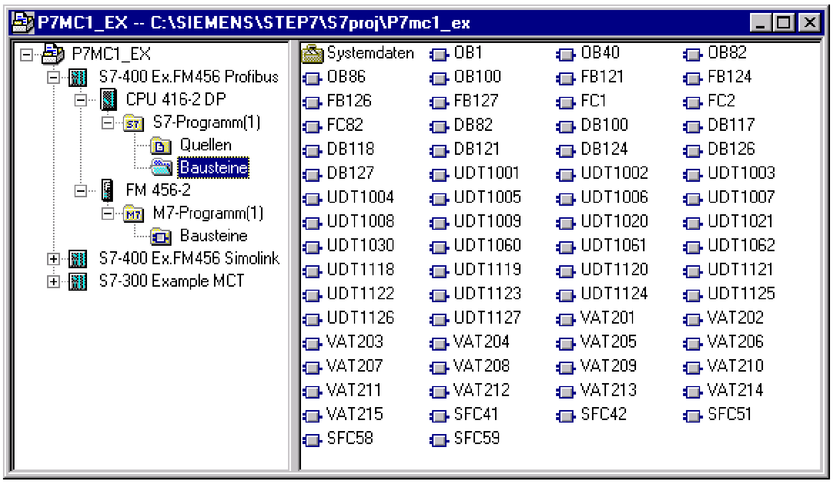
In this chapter you will find all the information you need in order to integrate the "GMC-BASIC" standard software.

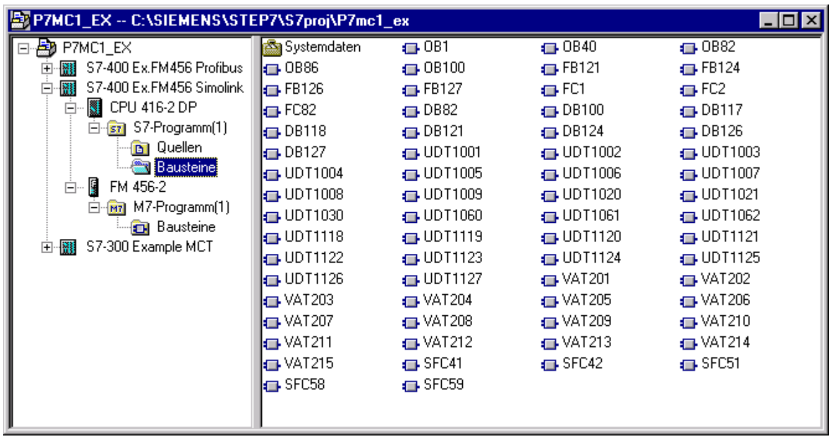
7.1 Introduction

When you have installed the "GMC-BASIC" standard software from the SETUP program, call up the S7 file manager. A library named "P7MC1LIB" and a project named "P7MC1_EX" will have been installed.









The "P7MC1LIB" library contains all the blocks required for the operation of the "GMC-BASIC" standard software:

The "P7MC1_EX" project contains three example programs:

- Example program "S7-400 Ex.FM456 Profibus" contains all the standard blocks from the library required for centralized operation of the technology on an M7-FM. The drives are addressed via a Profibus-DP connection.
- Example program "S7-400 Ex.FM456 SIMOLINK" contains all the standard blocks from the library required for centralized operation of the technology on an M7-FM. The drives are addressed via a SIMOLINK connection.
- Example program "S7-300 Example MCT" contains all the standard blocks from the library required for distributed operation of the technology on a MASTERDRIVES MC. The drives are addressed via a Profibus-DP connection.

The example programs include all the necessary function block calls and function calls. The necessary data blocks are also generated and initialized in the programs.

Axis descriptions are already entered in organization data block GMC_DB_ORG (one axis description for the MCT example or eight axis descriptions for the M7 example).

The UDTs (user-defined data types) in the library and in the project are used for internal organization of the data blocks and provide you with a structure to help you add any parameters you may need (e.g. for the addition of further axis descriptions to GMC_DB_ORG).

7.2 Creating the User Project

General Information

There are various ways to generate your individual user project. The easiest way is to use an example program that suits your needs. You then only need to modify or expand the axis descriptions that already exist. You can naturally use any of the blocks in the library. In this case, however, you need to know which blocks are needed (if you don't load all of the blocks), and you have to program all the calls and function block parameters yourself.

Step 1: Create a New Project

The first step is to create a new project or open an existing one. Create your S7 hardware configuration according to the capacity of your system in terms of M7-FM or MASTERDRIVES MC units (see the chapter entitled "Installation").

NOTE

You can also start a new project by opening the example project and saving it under a new name. Then delete the program examples you do not require and adjust the hardware configuration accordingly.

Step 2: Copy the Required Data into Your New Project

The next step is to copy the blocks you need from the library, or copy all the blocks from the example program of your choice into your S7 project. You will also need to copy the symbol table from the example project into your own project, in order to access the symbolic names.

NOTE

If you use the blocks from the library, you must program all the calls and the parameters for the function blocks yourself. You can follow the procedures used in the example programs or in the example calls described below.

Overlapping of block numbers:
If there is an overlap between the block numbers of the GMC-BASIC standard software and your own block numbers, you should consider renaming your own blocks. If this is not possible, please refer to the chapter entitled "Expert Knowledge/Tips". The information described in this chapter includes options for modifying the organization of the "GMC-BASIC" standard software.

Combination of MCT and M7-FM:
If you need both of the programming examples for M7 and MCT for your application, first copy the programming example for MCT into your project. Then copy the programming example for M7-FM, overwriting all the blocks. Then add the statement "Call FC3" to network 3 of OB1. Follow the procedure in the following example call.

Step 3:
Define the M7-FM Address in GMC_DB_ORG if Using an M7-FM

If you are not using an M7-FM, you can skip this step.

If you use one or more M7-FMs, you must enter the I/O address configured in the "hardware configuration" for each M7-FM in the "M7 description" section (see the chapter entitled "Data Blocks").

I/O address of M7-FM	DEC	X_m7_**n**.i_addr

n matches the M7 number

The address 512 is saved for the first M7-FM.

Step 4:
Modify the Axis Descriptions

GMC_DB_ORG already contains one or eight axis descriptions, depending on the programming example.

Open DB100 with the DB editor and go to the end of the block.

=64.0		END_STRUCT		
+364.0	X_axes	STRUCT		Section description for axes
+0.0	i_number_axes	INT	1	Number of axes in GMC_DB_ORG
+2.0	X_axis1	GMC_UDT_DB_ORG_AXIS		Section of 1st axis
=18.0		END_STRUCT		
=382.0		END_STRUCT		

In the figure, 1 is entered as the number of axes (i_number_axes). This is followed by an axis description named X_axis1.

Example:
You want to create the structures for 3 axis descriptions and allocate the names "AXIS_1", "AXIS_2" and "AXIS_3".

+364.0	X_axes	STRUCT		Section description for axes
+0.0	i_number_axes	INT	1	Number of axes in GMC_DB_ORG
+2.0	AXIS_1	GMC_UDT_DB_ORG_AXIS		Section of 1st axis
+18.0	AXIS_2	GMC_UDT_DB_ORG_AXIS		Section of 2nd axis
+34.0	AXIS_3	GMC_UDT_DB_ORG_AXIS		Section of 3rd axis
=50.0		END_STRUCT		
=414.0		END_STRUCT		

You can do this by renaming the existing axis 1 from "X_axis1" to "AXIS_1". Then insert the structures for 2 further axis descriptions by specifying data type "GMC_UDT_DB_ORG_AXIS" (UDT 1009) for names "AXIS_2" and "AXIS_3". You can use any names, of course.

Now change to the data view for the data block, in order to fill in the axis descriptions.

NOTE

Do not use the command "Edit -> Initialize Data Block", otherwise the initial values of the axis description (always 0) will overwrite the data. For this reason, you should always enter the values in the data view.

364.0	X_axes.i_number_axes	INT	3	3	Number of axes in GMC_DB_ORG
366.0	X_axes.AXIS_1.i_axis_type	INT	0	1	<1> = M7, <2> = MCT, <3> = M7/MCT
368.0	X_axes.AXIS_1.i_dbw_no_cmd	INT	0	0	Pointer of the commands
370.0	X_axes.AXIS_1.i_m7_no	INT	0	1	Number of the M7 (1..4)
372.0	X_axes.AXIS_1.i_log_axis_no	INT	0	1	Logical axis number 1..n
374.0	X_axes.AXIS_1.i_profibus_addr	INT	0	0	MCT PROFIBUS address
376.0	X_axes.AXIS_1.i_ppkw	INT	0	0	I/O area, PKW address of the MCT
378.0	X_axes.AXIS_1.i_ppzd	INT	0	0	I/O area, PZD address of the MCT
380.0	X_axes.AXIS_1.res7	INT	0	0	
382.0	X_axes.AXIS_2.i_axis_type	INT	0	2	<1> = M7, <2> = MCT, <3> = M7/MCT
384.0	X_axes.AXIS_2.i_dbw_no_cmd	INT	0	0	Pointer of the commands
386.0	X_axes.AXIS_2.i_m7_no	INT	0	0	Number of the M7 (1..4)
388.0	X_axes.AXIS_2.i_log_axis_no	INT	0	0	Logical axis number 1..n
390.0	X_axes.AXIS_2.i_profibus_addr	INT	0	3	MCT PROFIBUS address
392.0	X_axes.AXIS_2.i_ppkw	INT	0	256	I/O area, PKW address of the MCT
394.0	X_axes.AXIS_2.i_ppzd	INT	0	264	I/O area, PZD address of the MCT
396.0	X_axes.AXIS_2.res7	INT	0	0	
398.0	X_axes.AXIS_3.i_axis_type	INT	0	2	<1> = M7, <2> = MCT, <3> = M7/MCT
400.0	X_axes.AXIS_3.i_dbw_no_cmd	INT	0	0	Pointer of the commands

Now complete the axis descriptions by entering the data for the selected axis type. Don't forget to enter "3" for the number of axes.

Further Information

You will find further information about the axis descriptions in "Axis Description Section" in Section 5.1.

NOTE

You don't need to enter the DBW pointer in data block GMC_DB_CMD (...i_dbw_no_cmd). GMC_FB_START automatically enters the pointer correctly for all axes when the S7-CPU is started up.

Step 5:
Only if Necessary

Data block GMC_DB_CMD contains the control and checkback signals for all axes and is automatically generated with the correct length when the S7-CPU is started up. GMC_DB_CMD can also be created manually since no symbol structures are created when it is generated automatically.

To do this, create a new data block with number 117 in your S7 project, and use the DB editor to insert the required structures with data type GMC_UDT_DB_CMD_AXIS (UDT 1061).

0.0		STRUCT		
+0.0	AXIS_1	GMC_UDT_DB_CMD_AXIS		1st axis
+80.0	AXIS_2	GMC_UDT_DB_CMD_AXIS		2nd axis
+160.0	AXIS_3	GMC_UDT_DB_CMD_AXIS		3rd axis
=240.0		END_STRUCT		

NOTE

From GMC-Basic Version 2.0, the procedure is as follows: save a new data block as number 117 in your S7 project and add the necessary structures using the DB editor and data types:
GMC_UDT_DB_CMD_AXIS (UDT 1061) and
GMC_UDT_DB_CMD_AXIS_CONT (UDT 1062).

0.0	stat		STRUCT		
+0.0	stat	AXIS_1	"GMC_UDT_DB_CMD_AXIS"		1st axis
+80.0	stat	AXIS_2	"GMC_UDT_DB_CMD_AXIS"		2nd axis
+160.0	stat	AXIS_3	"GMC_UDT_DB_CMD_AXIS"		3rd axis
+240.0	stat	V_AXIS_1	"GMC_UDT_DB_CMD_AXIS_CONT"		1st axis, verify of the in- output signals
+260.0	stat	V_AXIS_2	"GMC_UDT_DB_CMD_AXIS_CONT"		2nd axis, verify of the in- output signals
+280.0	stat	V_AXIS_3	"GMC_UDT_DB_CMD_AXIS_CONT"		3rd axis, verify of the in- output signals
=300.0	stat		END_STRUCT		

Step 6

When you have created your project correctly, you can load it into the S7-CPU with the CPU in stop mode.

Step 7

Switch the S7-CPU to RUN mode. When the S7-CPU starts up, the parameters in GMC_DB_ORG are checked and all required data blocks are set up automatically.

If all the parameters are correct, the S7-CPU will switch to RUN. If this is not the case, you should read out the S7 diagnostic buffer. If a block of the "GMC-BASIC" standard software is responsible for the error, please refer to the chapter entitled "Expert Knowledge/Tips". The information in this chapter includes a description of troubleshooting procedures in the event of problems with "GMC-BASIC".

NOTE

If you have made changes to your project, you must switch the S7-CPU to stop in order to load the changes.

If the number of axes (axis descriptions) in GMC_DB_ORG is increased, you also need to add the new axes to GMC_DB_CMD (except if GMC_DB_CMD is to be set up automatically). GMC_DB_PARA must then be deleted so that it can be set up again.

Example calls:

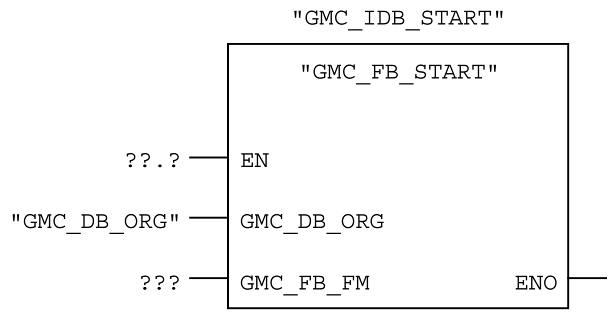
The example calls correspond to the example projects supplied with the software.

1 Startup

1.1 Use of the technology only on MASTERDRIVES MCT

OB100 : Start-up

Network 1 : Call GMC_FB_START



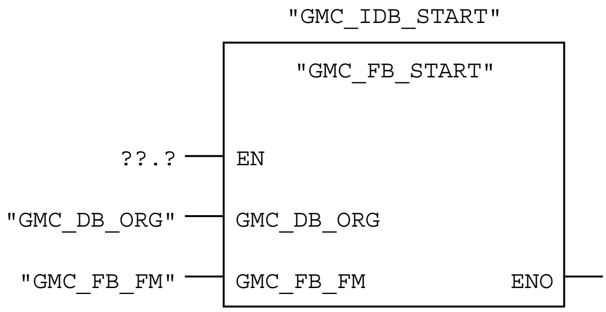
Symbol information:

FB127	GMC_FB_START	GMC FB startup block
DB127	GMC_IDB_START	instance DB for GMC_FB_START
DB100	GMC_DB_ORG	axes assignment

1.2 Use of the technology on the M7-FM or hybrid configuration with MASTERDRIVES MCT and M7-FM

OB100 : Start-up

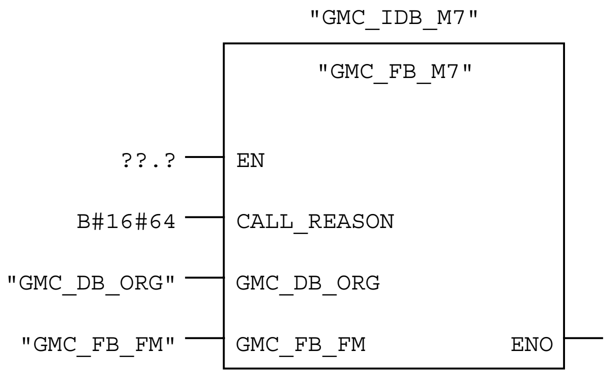
Network 1 : Call GMC_FB_START



Symbol information:

FB127	GMC_FB_START	GMC FB startup block
DB127	GMC_IDB_START	instance DB for GMC_FB_START
DB100	GMC_DB_ORG	axes assignment
FB124	GMC_FB_FM	GMC FB for communication with M7

Network 2 : Call GMC_FB_M7



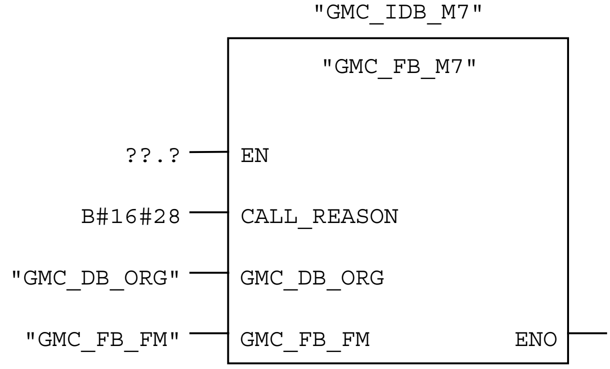
Symbol information:

FB121	GMC_FB_M7	block shell for FB GMC_FB_FM
DB121	GMC_IDB_M7	instance DB for FB GMC_FB_M7
DB100	GMC_DB_ORG	axes assignment
FB124	GMC_FB_FM	GMC FB for communication with M7

2. Process interrupt for technology on M7-FM

OB40 : Process interrupt

Network 1 : Call GMC_FB_M7



Symbol information:

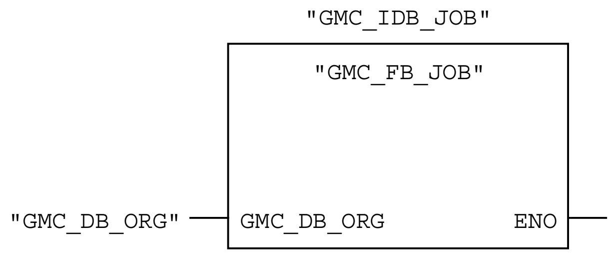
FB121	GMC_FB_M7	block shell for FB GMC_FB_FM
DB121	GMC_IDB_M7	instance DB for FB GMC_FB_M7
DB100	GMC_DB_ORG	axes assignment
FB124	GMC_FB_FM	GMC FB for communication with M7

3. Cyclic operation

3.1 GMC_FB_JOB call

FC1 :FC_CALL_FB_JOB

Network 1 : Call GMC_FB_JOB



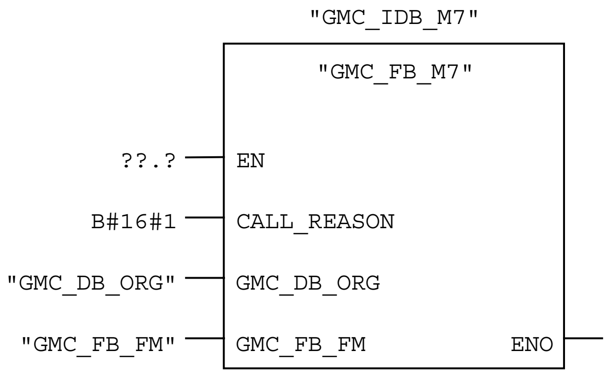
Symbol information:

FB126	GMC_FB_JOB	GMC FB for order management
DB126	GMC_IDB_JOB	instance DB for GMC_FB_JOB
DB100	GMC_DB_ORG	axes assignment

3.2 GMC_FB_M7 call for use of technology on M7-FM

FC2 :FC_CALL_FB_M7

Network 1 : Call GMC_FB_M7



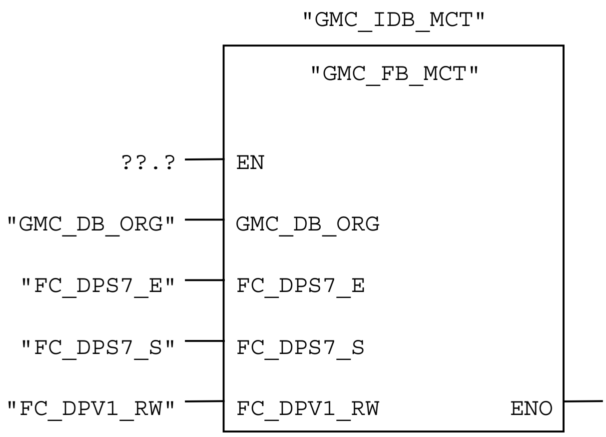
Symbol information:

FB121	GMC_FB_M7	block shell for FB GMC_FB_FM
DB121	GMC_IDB_M7	instance DB for FB GMC_FB_M7
DB100	GMC_DB_ORG	axes assignment
FB124	GMC_FB_FM	GMC FB for communication with M7

3.3 GMC_FB_MCT call for use of technology on MASTERDRIVES MCT

FC3 : FC_CALL_FB_MCT

Network 1 : Call GMC_FB_MCT



Symbol information:

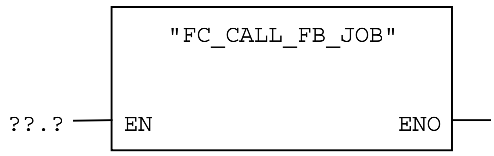
FB125	GMC_FB_MCT	GMC FB for MCT coupling
DB125	GMC_IDB_MCT	instance DB for GMC_FB_MCT
DB100	GMC_DB_ORG	axes assignment
FC22	FC_DPS7_E	receive extended control signals from MCT
FC21	FC_DPS7_S	send extended control signals to MCT
FC20	FC_DPV1_RW	DPV1 services

3.4 Block calls in OB1

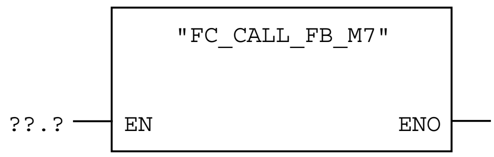
OB1 is used exclusively as a launching pad for the cyclic component of the "GMC-BASIC" standard software.

The call in network 2 is only required for the M7-FM target system. The call in network 3 is only required for the MCT target system.

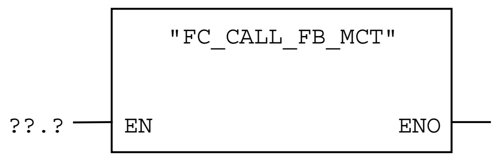
```
OB1 :
Network 1 : Call FC_CALL_FB_JOB
```




```
Network 2 : Call FC_CALL_FB_M7
```




```
Network 3 : Call FC_CALL_FB_MCT
```



4. Evaluating diagnostics alarms in the S7-CPU

If a diagnostics alarm is triggered, data set DS 1 is read by the appropriate module. Once read, the data set is saved in the data block DB_IO_POINT_FAULT (DB82). In the example, each diagnostics alarm overwrites the data of the previous alarm. The variables "DB_IO_POINT_FAULT".retval and "DB_IO_POINT_FAULT".busy display the status after the status of the data set has been read.

At the same time in the variable "DB_IO_POINT_FAULT".diag_counter_all the number of all diagnostics alarms and in the variable "DB_IO_POINT_FAULT".diag_counter_m7 the number of all diagnostics alarms passed on from the M7-FM to the S7-CPU are displayed.
If a diagnostics alarm which has been passed on is detected, data set DS 5 is also read. The actual diagnostic data of the triggering module are then derived from this. The read data set DS 5 is also stored in the data block DB_IO_POINT_FAULT (DB82).

If it is not possible to read data set DS 5 in the diagnostics OB for resource reasons, the variable "DB_IO_POINT_FAULT".dr5_busy is triggered. In order to still be able to reach the information of the diagnostics data set DS 5, you can read it using function FC82 "FC_read_dr5_cycle" in the OB1 cycle. If the data set DS 5 was able to be read without any errors, the variable DB_IO_POINT_FAULT".dr5_busy is reset.

The current start information for the OB82 is stored from DBD22 to DBB41 in data block DB_IO_POINT_FAULT (DB82).

NOTE

When a diagnostics event has occurred in the M7-FM due to a diagnostics alarm of a peripheral module and then the M7-FM triggers an alarm to the S7-CPU, this alarm is referred to as a passed-on diagnostics alarm.

Reason: The actual diagnostic data are lost in data set DS 1 when a diagnostics alarm is passed on from the M7-FM to the S7-CPU.

OB82 : "I/O Point Fault"

Comment:

Network 1: OB82 diagnosis-data save to DB_IO_POINT_FAULT

Comment:

```
      OPN   "DB_IO_POINT_FAULT"         DB82              -- DB_IO_POINT_FAULT
      L     LD     0
      T     DBD   52
      L     LD     4
      T     DBD   56
      L     LD     8
      T     DBD   60
      L     LD    12
      T     DBD   64
      L     LD    16
      T     DBD   68
```

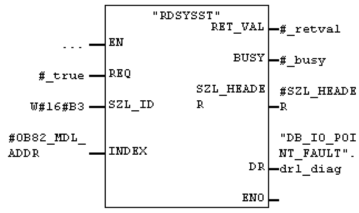
Network 2: Counter of all interrupts

Comment:

```
      L     "DB_IO_POINT_FAULT".diag_counter_all    DB82.DBW72         --
      +     1
      T     "DB_IO_POINT_FAULT".diag_counter_all    DB82.DBW72         --
//
      AN    #_true
      S     #_true
```

Network 3: Read DR1

Comment:



Symbol Information:

SFC51	RDSYSST	Read System Status
P#DB82.DBX36.0	"DB_IO_POINT_FAULT".dr1_diag	

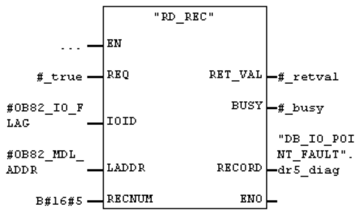
Network 4: Read Task-ID

When task-id like 0x00, then read record-number 5 else end of OB82

```
      L     "DB_IO_POINT_FAULT".dr1_diag.dr1_byte4    DB82.DBB40         --  Task-ID
      L     B#16#0
      <>I
      BEC
      L     "DB_IO_POINT_FAULT".diag_counter_m7   DB82.DBW34 -- Read record number 1 from m7:
      +     1
      T     "DB_IO_POINT_FAULT".diag_counter_m7   DB82.DBW34 -- Read record number 1 from m7:
//
      AN    #_true
      S     #_true
```

Network 5: Read DR5

Addon for diagnosis m7

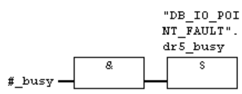


Symbol Information:

SFC59	RD_REC	Read Data Record
P#DB82.DBX0.0	"DB_IO_POINT_FAULT".dr5_diag	

Network 6: Read DR5 - was it o.k. ?

Addon for diagnosis m7



Symbol Information:

DB82.DBX32.0	"DB_IO_POINT_FAULT".dr5_busy

To read data set DS 5 in the cycle the function FC82 "FC_read_dr5_cycle" is available. For this purpose you have to enter the function call in the cyclic programming level.

FC82 : Read dr5 in the cycle

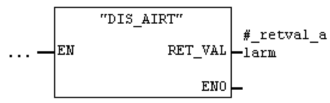
Netzwerk 1:

Kommentar:

```
      UN    "DB_IO_POINT_FAULT".dr5_busy    DB82.DBX32.0       --
      BEB
      S     #_true
//
```

Netzwerk 2 : Disable alarm

Addon for diagnosis m

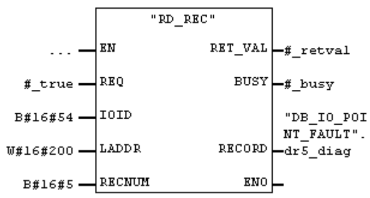


Symbolinformation:

SFC41	DIS_AIRT	Disable Alarm Interrupts

Netzwerk 3 : Read DR5

Addon for diagnosis m7

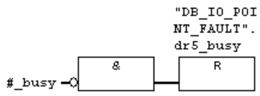


Symbolinformation:

SFC59	RD_REC	Read Data Record
P#DB82.DBX0.0	"DB_IO_POINT_FAULT".dr5_diag	

Netzwerk 4 : Read DR5 - was it o.k. ?

Addon for diagnosis m7

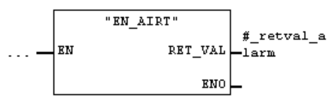


Symbolinformation:

DB82.DBX32.0	"DB_IO_POINT_FAULT".dr5_busy

Netzwerk 5 : Enable alarm

Addon for diagnosis m7



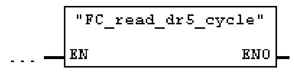
Symbolinformation:

SFC42	EN_AIRT	Enable Higher Priority Interrupts and Asynchronous Errors

Integrating the function call in the OB1, e.g. in network 3.

Netzwerk 3 : Titel:

Kommentar:



Symbolinformation:

FC82	FC_read_dr5_cycle	Read DR5 in the cycle

8 Expert Knowledge / Tips SIMATIC S7

Contents

In this chapter you will find various tips for problems that may arise in exceptional circumstances.

8.1 Diagnostics – Evaluating the "Displays" Section in GMC_DB_ORG

GMC_DB_ORG includes the "displays" section, in which the various blocks of the "GMC-BASIC" standard software enter diagnostic information in the event of an error.

You will only need to examine the "display" section if the error messages generated during operation of the technology do not help you solve the problem.

You will find a central description of all the error messages in "Appendix A: Error Messages".

Displays of DPV1 Service

			X_indication.
DBWx	**Display 1**: Messages of SFC5	Hex	X_dpv1.i_indication_ gadr_lgc
DBWx+2	**Display 2**: Messages of SFC58	Hex	X_dpv1.i_indication_ wr_rec
DBWx+4	**Display 3**: Messages of SFC59	Hex	X_dpv1.i_indication_ rd_rec
DBWx+6		Hex	X_dpv1.i_res4

Display 1: You will find the messages of SFC5 "GADR_LGC" under main error number 4003.

Display 2: You will find the messages of SFC58 "WR_REC" under main error number 4004.

Display 3: You will find the messages of SFC59 "RD_REC" under main error number 4005.

Displays of DVA_S7-Service

			X_indication.
DBWx	**Display 1**: Number of messages which occurred	DEC	X_dva_s7.i_number
DBWx+2	**Display 2**: Absolute axis number	DEC	X_dva_s7.i_achs_no
DBBx+4	**Display 3a**: Receive parameter error	Hex	X_dva_s7.i_pafe_r
DBBx+5	**Display 3b**: Send parameter error	Hex	X_dva_s7.i_pafe_s
DBWx+6			X_dva_s7.i_res4

Display 1: Indicates the number of messages which occurred
Display 2: Indicates the absolute axis number
Display 3: Indicates the error code → a for receive, b for send.

Error Codes	Meaning
1	Illegal address type
2	Incorrect PPO type
90	No module is configured for the specified axis.
93	The I/O address of the PZD component for this axis does not address a DP module from which consistent data can be read.
A0	The selected axis is incorrect (send).
A1	The selected axis is incorrect (receive).
B1	The length of the PZD component is not set correctly.

Displays of GMC_FB_START

GMC_FB_START checks data block GMC_DB_ORG during the startup procedure, and sets up the necessary data structure on the basis of the parameters contained in GMC_DB_ORG. If errors occur during this process, the error codes are entered in the displays.

			X_indication.
DBWx	**Display 1**	Hex	X_fb_start.i_res1
DBWx+2	**Display 2**	DEC	X_fb_start.i_res2
DBWx+4	**Display 3**	DEC	X_fb_start.i_res3
DBWx+6	**Display 4**	Hex	X_fb_start.i_res4

Displays 1 to **4**:
The meaning of the displays is described centrally in "Appendix A: Error Messages"; the number entered in **display 1** can be used as a search criterion.

Displays of GMC_FB_JOB

			X_indication.
DBWx	**Display 1**: DB number	DEC	X_fb_job.i_res1
DBWx+2	**Display 2**: DBW number	DEC	X_fb_job.i_res2
DBWx+4	**Display 3**: Error number 1	Hex	X_fb_job.i_res3
DBWx+6	**Display 4**: Error number 2	Hex	X_fb_job.i_res4

Display 1: Number of the data block containing the invalid task

Display 2: Number of the data block word at which the invalid task starts

Display 3 and **4**: The meaning of the displays is described centrally in "Appendix A: Error Messages"; the number entered in **display 3** can be used as a search criterion.

Displays of GMC_FB_PICTMAN

			X_indication.
DBWx	**Display 1**: Error message	Hex	X_fb_pictman.i_res1
DBWx+2			X_fb_pictman.i_res2
DBWx+4			X_fb_pictman.i_res3
DBWx+6			X_fb_pictman.i_res4

Display 1: The meaning of the display is described centrally in "Appendix A: Error Messages"; the number entered in **display 1** can be used as a search criterion.

Displays of GMC_FB_PICTDATA

		X_indication.
DBWx		X_fb_pictdata.i_res1
DBWx+2		X_fb_pictdata.i_res2
DBWx+4		X_fb_pictdata.i_res3
DBWx+6		X_fb_pictdata.i_res4

This function block does not currently generate any displays.

Displays of GMC_FB_M7

			X_indication.
DBWx	**Display 1**: M7 number (OB1,35,100)	DEC	X_fb_m7.i_res1
DBWx+2	**Display 2**: Error code of FB_FM	HEX	X_fb_m7.i_res2
DBWx+4	**Display 3**: M7 number (OB40)	DEC	X_fb_m7.i_res3
DBWx+6	**Display 4**: Error code of FB_FM	HEX	X_fb_m7.i_res4

Display 1: Displays the M7 number at which an error occurred during execution of organization blocks OB1, OB35 or OB100.

Display 2: The meaning of the display is described centrally in "Appendix A: Error Messages"; the number entered in **display 2** can be used as a search criterion.

Display 3: Displays the M7 number at which an error occurred during execution of organization blocks OB40.

Display 4: The meaning of the display is described centrally in "Appendix A: Error Messages"; the number entered in **display 4** can be used as a search criterion.

Displays of GMC_FB_MCT

			X_indication.
DBWx	**Display 1**: Main task number	DEC	X_fb_mct.i_res1
DBWx+2	**Display 2**: Absolute axis number	DEC	X_fb_mct.i_res2
DBWx+4	**Display 3**: Error number 1	HEX	X_fb_mct.i_res3
DBWx+6	**Display 4**: Error number 2	HEX	X_fb_mct.i_res4

Display 1: Indicates the main task number at which the error occurred.

Display 2: Indicates the axis number on which the error occurred.

Display 3 and **4**: The meaning of the displays is described centrally in "Appendix A: Error Messages"; the number entered in **display 3** can be used as a search criterion.

8.2 Fault diagnosis – Evaluation of the data sets DS1 and DS5 in the OB82 organization block

General

The diagnostics alarm can be triggered by any module capable of diagnosis. The OB82 start information (20 bytes of local data) and the DS1 data set are provided for this.

NOTE

In the project example provided, the diagnostics organization block, OB82, is fully programmed. For visualization purposes, the VAT213 to VAT215 variable tables are provided and also the DB82 data block.

Fault diagnosis, triggered by the GMC application

When the GMC application is used on the M7-FM, it can be necessary to read data set DS5 in addition to data set DS1. This is always the case if the diagnostics event has occurred due to a diagnostics alarm of a peripheral module assigned to the M7-FM and the M7-FM only "passes on" the diagnostics event to the S7-CPU.

This means that the diagnostics message in data set DS1 says that the M7-FM has triggered a diagnostics alarm. The actual cause, why, is lost. The relevant diagnostics data, therefore, cannot be given along with the DS1 data set.

If such a diagnostics alarm is received, the DS5 data set in the OB82 must also be read. This contains the actual diagnostics data.

NOTE

If you receive a diagnostics alarm as described above, this is not a diagnosis due to a fault status of the M7-FM!

Data-set structure of DS1

After reading the DS1 data set, check byte 4, task identification, to see if it has the value 0. If the task value byte has the value 0, you must read the DS5 data set immediately.

Task value 0 means that the alarm is a "passed-on" alarm.

Byte offset	**Description**	**Contents**
0	Bit 0: Module fault/OK (group fault ID) Bit 1: Internal fault Bit 2: External fault Bit 3: Channel fault occurred Bit 4: No external auxiliary voltage Bit 5: No front plug Bit 6: Module not parameterized Bit 7: Wrong parameters in module	0x05
1	Bit 0 to Bit 3:Module class (CPU, FM, CP, IM, SM, ...) Bit 4: Channel information available Bit 5: User information available (Information from Byte 4 onwards) Bit 6: Diagnostics alarm by representative??? Bit 7: Reserve (initialized with 0)	0x22
2	Bit 0: User module incorrect/missing Bit 1: Communication fault Bit 2: Operating status RUN/STOP (0 = RUN, 1 = STOP) Bit 3: Time monitor triggered (watch dog) Bit 4: Supply voltage inside module failed Bit 5: Battery empty (BFS) Bit 6: Whole back-up failed Bit 7: Reserve (initialized with 0)	0x00
3	Bit 0: Rack failure (detected by IM) Bit 1: Processor failure Bit 2: EPROM failure Bit 3: RAM failure Bit 4: ADU/DAU failure Bit 5: Fuse failure Bit 6: Process alarm lost Bit 7: Reserve (intialized with 0)	0x00
4	Task identification	**0x00**
5	Cause: 0: General fault 1: Fault during run-up of the application 2: Fault during initialization of the application 3: Signal handler 4: Watchdog	0x00

Byte offset	Description	Contents
6	Details regarding causes 1 and 2: 0x10: General RMOS fault 0x11: Incorrect command-line parameter 0x12: Generation of a task unsuccessful 0x13: Generation of a FLAG group unsuccessful 0x14: Fault occurred during start of a task 0x20: Fault during access to data block 0x21: Internal fault 1 0x22: Internal fault 2 0x23: Internal fault 3 0x24: Internal fault 4 0x25: Internal fault 5 0x26: Internal fault 6 0x27: Internal fault 7 0x28: Internal fault 8 0x29: Internal fault 9 Details regarding cause 3: IR identification	0xxx
7	Internal additional information 1	0x00
8	Internal additional information 2	0x00
9 to 12	Detail regarding cause 4: The time value which the monitoring function has caused to trigger the watchdog has been placed in byte 4.	xxxx
13 to 15	Reserve	0000

Structure of data set DS5

In order to interpret DS5, consult the corresponding manual of the module used.

Byte offset	Description
0	Diagnostics address of the module triggering the diagnostics alarm
2..61	Max. 30 bytes for diagnostics information (for structure, see manual of the module, e.g. SM335, IM178, ET200M etc.) SM modules: 16 bytes (corresponds to DS1 of the SM module) DP slaves: 29 bytes for DP standard diagnosis

8.3 Changing the Block Numbers – Assigning the "General Parameters", "MCT Description" and "M7 Description" Sections in GMC_DB_ORG

Since the SIMATIC S7 system provides only a restricted range of numbers for blocks, the block numbers may overlap if several software packages use the same S7-CPU.

For this reason, it is possible to configure the block numbers of the "GMC-BASIC" standard software.

Function blocks:
The numbers of the function blocks can be changed as required. The parameters of the function blocks are used to configure blocks which are called up internally.

Functions:
The numbers of the functions can be changed as required. Please note that you can configure the function blocks which use these functions internally.

Data blocks:
All data blocks required by the system are specified in GMC_DB_ORG. If you change the data block numbers, you must change the corresponding entries in GMC_DB_ORG.

If you specify the value 0 in GMC_DB_ORG for the data block numbers, the data blocks required on the basis of an axis configuration (valid axis description) are initialized automatically with default settings.

Function block GMC_FB_START enters the default DB number in the corresponding section in the GMC_DB_ORG data block (see also Section 8.3).

NOTE

You should only modify GMC_DB_ORG when the S7-CPU is in STOP mode. You should then switch the S7-CPU to RUN. This action activates GMC_FB_START, which verifies GMC_DB_ORG and sets up all the necessary data blocks.

Delete data blocks GMC_DB_PARA and GMC_DB_CMD when you change the "total number of axes", because the length of the data blocks will change.

In the description below, the default block numbers are specified in parentheses (state at time of supply).

"General Parameters" Section

The "general parameters" section contains the data block parameters for the application interface and the optional "GMC-OP-OAM" standard software (e.g. user interface with SIMATIC HMI OP25/27).

GMC_DB_PICT_POINTER			X_general.
DBWx	DB number (**-1,120**)	DEC	X_gmc_pict_pointer. i_db_no
DBWx+2	DBW number (0, variable)	DEC	X_gmc_pict_pointer. i_dbw_no
DBWx+4	Reserved (0)	DEC	X_gmc_pict_pointer. i_res1
GMC_DB_ORDER			
DBWx+6	DB number (**-1,119**)	DEC	X_gmc_order. i_db_no
DBWx+8	DBW number (0)	DEC	X_gmc_order. i_dbw_no
DBWx+10	Reserved (0)	DEC	X_gmc_order.i_res1
GMC_DB_COM			
DBWx+12	DB number (**118**)	DEC	X_gmc_com. i_db_no
DBWx+14	DBW number (0, variable)	DEC	X_gmc_com. i_dbw_no
DBWx+16	Reserved (0)	DEC	X_gmc_com.i_res1
GMC_DB_CMD			
DBWx+18	DB number (**117**)	DEC	X_gmc_cmd. i_db_no
DBWx+20	DBW number (0)	DEC	X_gmc_cmd. i_dbw_no
DBWx+22	Reserved (0)	DEC	X_gmc_cmd.i_res1
GMC_DB_PARA			
DBWx+24	DB number (**116**)	DEC	X_gmc_para. i_db_no
DBWx+26	DBW number (0)	DEC	X_gmc_para. i_dbw_no
DBWx+28	Reserved (0)	DEC	X_gmc_para. i_res1
GMC_DB_PARAM			
DBWx+30	DB number (**50**)	DEC	X_GMC_PARAM. i_db_no
DBWx+32	DBW number (0)	DEC	X_GMC_PARAM. i_dbw_no
DBWx+34	Reserved (0)	DEC	X_GMC_PARAM. i_res1

Data blocks GMC_DB_PICT_POINTER and GMC_DB_ORDER are only relevant if the "GMC-OP-OAM" standard software is used. Otherwise the DB number entries are indicated by "-1" as non-existent.

The numbers of data blocks GMC_DB_COM, GMC_DB_CMD (application interface), GMC_DB_PARA and GMC_DB_PARAM can be changed as required.

Data block GMC_DB_PARA is required for the parameter interface between S7 and M7/MC. Data block GMC_DB_PARAM is for a configuration tool. The Data block is set up by the function block GMC_FB_START with a length of 1100 Bytes in the CPU.

"MCT Description" Section

The "MCT description" section contains all the data blocks which are required internally if the target system for the technology is a MASTERDRIVES MCT.

GMC_DB_DPV1

DBWx+6	DB number (**115**)	DEC	X_gmc_dpv1. i_db_no
DBWx+8	DBW number (0, variable)	DEC	X_gmc_dpv1. i_dbw_no
DBWx+10	Reserved (0)	DEC	X_gmc_dpv1. i_res1

GMC_DB_AWP1_MCT

DBWx+12	DB number (**114**)	DEC	X_gmc_awp_1. i_db_no
DBWx+14	DBW number (0, variable)	DEC	X_gmc_awp_1. i_dbw_no
DBWx+16	Reserved (0)	DEC	X_gmc_awp_1. i_res1

GMC_DB_AWP2_MCT

DBWx+18	DB number (**-1** = not required)	DEC	X_gmc_awp_2. i_db_no
DBWx+20	DBW number (0, variable)	DEC	X_gmc_awp_2. i_dbw_no
DBWx+22	Reserved (0)	DEC	X_gmc_awp_2. i_db_res1

Data block GMC_DB_DPV1 is required for the PROFIBUS DPV1 service.

Data block GMC_DB_AWP1_MCT is the task interface to the MCT and receives data from GMC_FB_JOB.

Data block GMC_DB_AWP2_MCT is a second task interface to the MCT which can be used directly by the user (under development).

The numbers of data blocks GMC_DB_DPV1, GMC_DB_AWP1 and GMC_DB_AWP2 (if needed) can be changed as required.

"M7 Description" Section

The "M7 description" section contains all the data blocks which are required internally if the target system for the technology is one or up to four M7-FMs.

DBWx	I/O address of M7-FM (512,0,0,0)	DEC	X_m7_{n}. i_addr

GMC_DB_FMPA_{n}

DBWx+2	DB number (**112,-1,-1,-1**)	DEC	X_gmc_fmpa. i_db_no
DBWx+4	DBW number (0)	DEC	X_gmc_fmpa. i_dbw_no
DBWx+6	Reserved (0)	DEC	X_gmc_fmpa.i_res1

GMC_DB_IN_{n}

DBWx+8	DB number (**111,-1,-1,-1**)	DEC	X_gmc_in.i_db_no
DBWx+10	DBW number (0)	DEC	X_gmc_in.i_dbw_no
DBWx+12	Reserved (0)	DEC	X_gmc_in.i_res1

GMC_DB_OUT_{n}

DBWx+14	DB number (**110,-1,-1,-1**)	DEC	X_gmc_out.i_db_no
DBWx+16	DBW number (0)	DEC	X_gmc_out. i_dbw_no
DBWx+18	Reserved (0)	DEC	X_gmc_out.i_res1

GMC_DB_AWP1_M7_{n}

DBWx+20	DB number (**109,-1,-1,-1**)	DEC	X_gmc_awp_1. i_db_no
DBWx+22	DBW number (0)	DEC	X_gmc_awp_1. i_dbw_no
DBWx+24	Reserved (0)	DEC	X_gmc_awp_1. i_res1

GMC_DB_AWP2_M7_{n}

DBWx+26	DB number (**-1,-1,-1,-1**)	DEC	X_gmc_awp_2. i_db_no
DBWx+28	DBW number (0)	DEC	X_gmc_awp_2. i_dbw_no
DBWx+30	Reserved (0)	DEC	X_gmc_awp_2. i_res1

GMC_DB_IDB_FM_{n}

DBWx+32	DB number (**124,-1,-1,-1**)	DEC	X_gmc_idb_fm. i_db_no
DBWx+34	DBW number (0)	DEC	X_gmc_idb_fm. i_dbw_no
DBWx+36	Reserved (0)	DEC	X_gmc_idb_fm. i_res1

x = (*n*-1) ∗ 38

n M7 number 1 to 4

Data block GMC_DB_FMPA_{*n*} contains configuration information for function block GMC_FB_FM.

Data block GMC_DB_IN_{*n*} contains the control signals from S7 to M7.

Data block GMC_DB_OUT_{*n*} contains the checkback signals from M7 to S7.

Data block GMC_DB_AWP1_M7_{*n*} is the task interface to the M7-FM and receives data from GMC_FB_JOB.

Data block GMC_DB_AWP2_M7_{*n*} is a second task interface to the M7-FM which can be used directly by the user (under development).

Instance data block GMC_IDB_FB_{*n*} is required by GMC_FB_FM.

The numbers of the data blocks for internal M7 communication can be changed as required.

8.4 Data Structure "GMC-BASIC" – Default Assignment of Data Block Numbers

DB Number	For Data Block....
50	GMC_DB_PARAM
82	GMC_IDB_FM_4
83	GMC_DB_AWP2_M7_4
84	GMC_DB_AWP1_M7_4
85	GMC_DB_OUT_4
86	GMC_DB_IN_4
87	GMC_DB_FMPA_4
88	GMC_IDB_FM_3
89	GMC_DB_AWP2_M7_3
90	GMC_DB_AWP1_M7_3
91	GMC_DB_OUT_3
92	GMC_DB_IN_3
93	GMC_DB_FMPA_3
94	GMC_IDB_FM_2
95	GMC_DB_AWP2_M7_2
96	GMC_DB_AWP1_M7_2
97	GMC_DB_OUT_2
98	GMC_DB_IN_2
99	GMC_DB_FMPA_2
108	GMC_DB_AWP2_M7_1
109	GMC_DB_AWP1_M7_1
110	GMC_DB_OUT_1
111	GMC_DB_IN_1
112	GMC_DB_FMPA_1
113	GMC_DB_AWP2_MCT
114	GMC_DB_AWP1_MCT
115	GMC_DB_DPV1
116	GMC_DB_PARA
117	GMC_DB_CMD
118	GMC_DB_COM
119	GMC_DB_ORDER
120	GMC_DB_PICT_POINTER
124	GMC_IDB_FM_1

8.5 Data Structure "GMC-BASIC" – Individual Configurations, Optimum Memory Utilization of S7-CPU

```
S7 Program
  OB1                                              only if task management required
    FC1
      FB126 (GMC_FB_JOB) , DB126 (GMC_IDB_JOB)
        DB100 (GMC_DB_ORG)
        DB118 (GMC_DB_COM)
                                                                  for M7-FM only
    FC2
      FB121 (GMC_FB_M7) , DB121 (GMC_IDB_M7)
        DB100 (GMC_DB_ORG)
        DB117 (GMC_DB_CMD)
        FB124 (GMC_FB_FM) , DB124 (GMC_IDB_FM1)
          DB112 (GMC_DB_FMPA_1)
          DB111 (GMC_DB_IN_1)
          DB110 (GMC_DB_OUT_1)
          DB109 (GMC_DB_AWP1_M7_1)
          .......... (GMC_DB_AWP2_M7_1)
                                                                  for MCT only
    FC3
      FB125 (GMC_FB_MCT) , DB125 (GMC_IDB_MCT)
        DB100 (GMC_DB_ORG)
        FC22 (FC_DPS7_E)
        FC21 (FC_DPS7_S)
        FC20 (FC_DPV1_RW)
        DB115 (GMC_DB_DPV1)
        DB116 (GMC_DB_PARA)
        DB114 (GMC_DB_AWP1_MCT)
```

The graphic shows the entire structure of the OB1 when you use SIMATIC Motion Control and MASTERDRIVES MC with technology option F01.

8.6 Using GMC_FB_START_MINI

General Information

Function block GMC_FB_START is required for the system startup. It checks the entries in GMC_DB_ORG and sets the system to a defined initial state. It also sets up all the necessary data blocks with the required length.

This function block must be called up and parameterized once in OB100 (startup). It is not multi-instance-capable and requires a separate instance data block.

NOTE

You must configure GMC_DB_ORG correctly before you start up the system.

When the system has been set up and verified once in the PLC by GMC_FB_START, this function block can be replaced by **GMC_FB_START_MINI**. This will save you approximately 4 KB of code in the PLC.

Requirements

You have created a new S7 project and set up your hardware configuration for your system capacity in terms of M7-FM or MASTERDRIVES MC units.

You have copied all the necessary blocks from the library or you have copied all the blocks from the example program of your choice into your S7 project.

The resulting S7 project is referred to below as the "Master_Project".

Step 1

The first step is to configure GMC_DB_ORG in accordance with your hardware configuration.

Step 2

When you have created your S7 project correctly, you can load it into the S7-CPU with the CPU in STOP mode.

Step 3

Switch the S7-CPU to RUN mode. When the S7-CPU starts up, the parameters in GMC_DB_ORG are checked by GMC_FB_START and all the necessary data blocks are set up automatically.

If all the parameters are correct, the S7-CPU will switch to RUN. If this is not the case, you should read out the S7 diagnostic buffer. If a block of the "GMC-BASIC" standard software is responsible for the error, please refer to the chapter entitled "Expert Knowledge/Tips". The information in this chapter includes a description of troubleshooting procedures in the event of problems with "GMC-BASIC".

Step 4

Switch the S7-CPU to STOP mode.

Step 5

Save your S7 project under a different name. You now have a copy of your S7 project.

Step 6

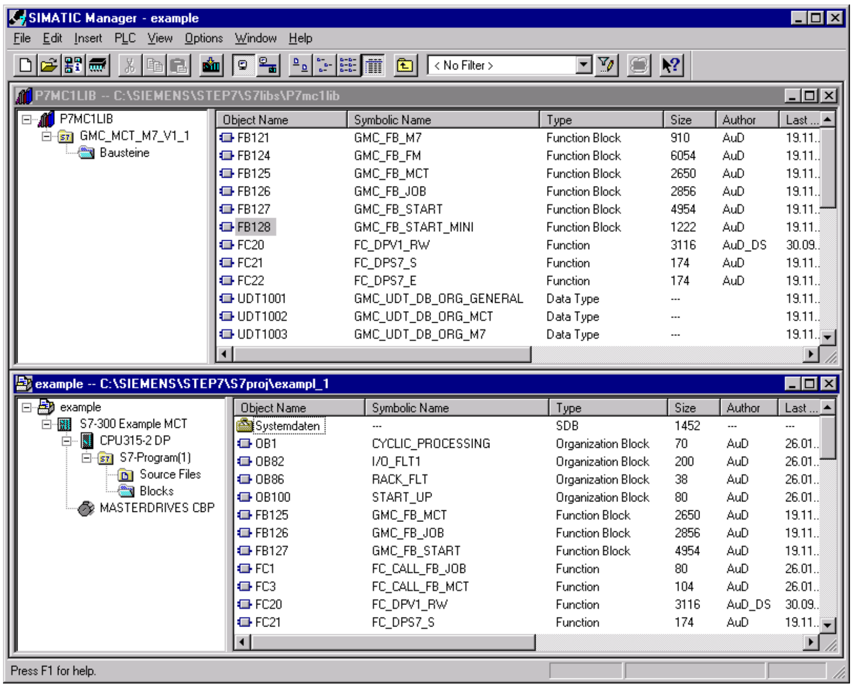
Copy all the data blocks belonging to the GMC_Project from the S7-CPU into your S7 project.

The following data blocks belong to the GMC_Software:

Symbol	**Default DB Number**	**M7**	**MCT**	**OAM**
GMC_DB_ORG	DB100	X	X	
GMC_DB_AWP2_M7_1	DB108	X		
GMC_DB_AWP1_M7_1	DB109	X		
GMC_DB_OUT_1	DB110	X		
GMC_DB_IN_1	DB111	X		
GMC_DB_FMPA_1	DB112	X		
GMC_DB_AWP2_MCT	DB113		X	
GMC_DB_AWP1_MCT	DB114		X	
GMC_DB_DPV1	DB115		X	
GMC_DB_PARA	DB116	X	X	
GMC_DB_CMD	DB117	X	X	
GMC_DB_COM	DB118	X	X	
GMC_DB_ORDER	DB119			X
GMC_DB_PICT_POINTER	DB120			X
GMC_IDB_M7	DB121	X		
GMC_IDB_PICTMAN_DATA	DB122			X
GMC_IDB_FM_1	DB124	X		
GMC_IDB_MCT	DB125		X	
GMC_IDB_JOB	DB126	X	X	
GMC_IDB_START	DB127	X	X	

Step 7

Open the library "P7MC1LIB". Copy GMC_FB_START_MINI (FB128) into your S7 project.



Step 8

Delete function block GMC_FB_START (FB127) from your S7 project. Change the function block number of GMC_FB_START_MINI from FB128 to FB127.

NOTE

The S7 project which you have now created is matched to your system capacity in terms of the number of M7-FM or MASTERDRIVES MC units.

If you want to add or change an M7-FM or MASTERDRIVES MC, repeat steps 1 to 8 with the master project.

8.7 Description of SIMOLINK telegrams

General

All the stations only send their data in the form of the telegrams allocated to them by the address. Each telegram has a length of 4 bytes. The SIMOLINK makes 8 channels available for each slave for data transfer (same address and channel no. 0 to 7). This means that a channel number is allocated to each 4-byte value, and therefore also a telegram on the bus.

Telegrams from M7-FM to MASTER-DRIVES MC (MCB)

When using the M7 technology functions, the channels are allocated as described in the following.

Channel	Byte	7	6	5	4	3	2	1	0
0	0	RES	RES	A6	A5	A4	A3	A2	A1
	1	RES	CP-	CP+	RES	EN_ MV	EN_ REF	EN_ CON	RES
0	2	ACK_ F	RES	RES	RES	ENC	OFF3	OFF2	OFF1
	3	RES	RES	RES	RES	RES	RES	RES	RES
1	DD	Set value for position [P190]							
2	DD	Set value for speed [P212]							
3	DD	Pre-control for speed [P209]							
4	DD	Correction value [P174]							
5	DD	Pre-control for acceleration [available soon]							
6	DD	RES							
7	DD	RES							

NOTE

The assignment of the channels on the MASTERDRIVES MC is configured with the standard interconnections as indicated in the script file "3_Scriptfile_MCB_Technologieverdrahtung.ssc" or in the dowload file "3_Downloadfile_MCB_Technologieverdrahtung.dnl" (see Section 3).

OFF1	0 = OFF1, Stop from acceleration rate limiter, then pulse inhibitor 1 = ON, Operating condition (transition operated)	(P554)
OFF2	0 = OFF2 Pulse inhibitor, motor coasts 1 = ON, Operating condition	(P555)
OFF3	0 = OFF3 Rapid stop 1 = ON, Operating condition	(P558)
ENC	Enable controller 1 = Enable inverter, pulse activation 0 = Pulse inhibitor	(P561)
ACK_F	**Ack**nowledgement **f**ault 0 -> 1 Transition: Acknowledge	(P565)
EN_CON	**En**able position **con**trol 0 = Enable position control 1 = Activate speed control	(P213)
EN_REF	**En**able **ref**erence point detection	(P177)
EN_MV	**En**able **m**easuring **v**alue memory (Enable position measured value)	(P179)
CP+	Correction value add to position value (Add correction value to actual position value)	(P175.01)
CP-	Correction subtract to position value (Subtract correction value from actual position value)	(P175.02)
A1..6	Control digital outputs 1..6	

NOTE

You parameterize the digital outputs (technology outputs) by means of machine data 47 and 48.

Assignment:	Technology	MASTERDRIVES MC
	Digital output 1	A1
	Digital output 2	A2
	Digital output 3	A3

Telegrams from MASTERDRIVES MC (MCB) to M7-FM

Channel	Byte	7	6	5	4	3	2	1	0
0	0	RES	RES	E6	E5	E4	E3	E2	E1
	1	RES	RES	RES	RES	RES	MV_V	ARFD	AV_V
0	2	RES	WA RN	OFF3	OFF2	FAU LT	IOP	RDY	RTS
	3	RES	RES	RES	RES	OTM	OTC	OLC	S MAX
1	DD	Actual position value [KK120]							
2	DD	Measured value for position [KK122]							
3	DD	Actual speed value (optional) [KK151]							
4	DD								
5	DD	Fault number / warning number [K250]							
6	DD	RES							
7	DD	RES							

SMAX	**Max**imum rotational **s**peed (maximum speed exceeded)	[B136] [2076.09]
OLC	**O**ver**l**oad **c**onverter	[B144] [2076.10]
OTC	**O**ver**t**emperature in **c**onverter	[B148] [2076.11]
OTM	**O**ver**t**emperature in **m**otor	[B150] [2076.12]
RTS	**R**eady **t**o **s**tart	[B100] [2076.01]
RDY	**R**ea**dy**	[B102] [2076.02]
IOP	**I**n **op**eration	[B104] [2076.03]
FAULT		[B106] [2076.04]
OFF2	OFF2 not active	[B105] [2076.05]
OFF3	OFF3 not active	[B110] [2076.06]
WARN	**Warn**ing	[B114] [2076.07]
AV_V	**A**ctual **V**alues **V**alid	[B070] [2078.09]
ARFD	**A**xis **R**e**f**erence**d**	[B210] [2078.10]
MV_V	**M**easuring **v**alue is **v**alid	[B212] [2078.11]
E1..6	Read digital inputs 1..6	[B016, B018, B020] [2078.01 to 2078.06]

NOTE

You parameterize the digital outputs (technology outputs) by means of machine data 45 and 46.

Assignment:	Technology	MASTERDRIVES MC
	Digital input 4	E4
	Digital input 5	E5
	Digital input 6	E6

8.8 Technical Specifications of GMC-BASIC Standard Software V2.0

Memory Requirements and Processing Time

The block lengths are specified in number of bytes. For further run-time information and information which is more up-to-date, consult the read-me file on the CD of GMS-BASIC.

FB Name	FB_START	FB_START_MINI
Programming language	STL	STL
Length of block (no. of bytes)	(6.402) 6.4 kb	(2.348) 2.3 kb
RAM required (no. of bytes)	(4.954) 4.9 kb	(1.222) 1.2 kb
Local data	226	226
Nesting depth	1	1
System functions called	CREAT_DB BLKMOV	
Data area assignment	IDB_STRT = 428	

FB Name	FB_JOB	DB_ORG
Programming language	STL	STL
Length of block (no. of bytes)	(3.416) 3.4 kb	-
RAM required (no. of bytes)	(2.856) 2.8 kb	-
Local data	34	-
Nesting depth	1	-
System functions called	BLKMOV	
Data area assignment	IDB_JOB = 138 DB_COM = 144 (for 8 boxes)	DB_ORG = 408 + 16 per axis
Processing time in ms		-
• CPU 315-2DP		
Idle	< 0.8	
Processing of task	up to 1.0 (type < 0.8)	
• CPU 413-2DP		
Idle	< 0.2	
Processing of task	< 0.3	
• CPU 414-2DP		
Idle	< 0.08	
Processing of task	< 0.1	
• CPU 416-2DP		
Idle	< 0.03	
Processing of task	< 0.04	

FB-Name	**FB_MCT**	**FC_DPS7-E (Ersatz)**
Programming language	STL	STL
Length of block (no. of bytes)	(3.386) 3.4 kb	(250) 0.25 kb
RAM required (no. of bytes)	(2.650) 2.6 kb	(174) 0.2 kb
Local data	62	20
Nesting depth	1	1
System functions called	BLKMOV WR_REC RD_REC	DPRD_DAT
Data area assignment	IDB_MCT = 232 DB_AWP1 = 276 DB_AWP2 = 276 DB_CMD = 36 + 100 per axis	DB_PARA = 44 + 12 per axis
Processing time in ms		Processing time per axis
• CPU 315-2DP (one MCT axis)		< 0.75
idle, cyclic service	0.9 to 1.2	
Cyclic service and DPV1 service	2.0 to 6.6	
• CPU 413-2DP (one MCT axis)		< 0.5
idle, cyclic service	< 4.0	
Cyclic service and DPV1 service	< 9.0	
• CPU 414-2DP (one MCT axis)		
idle, cyclic service	< 2.0	
Cyclic service and DPV1 service	< 4.5	
• CPU 416-2DP (one MCT axis)		
idle, cyclic service	< 1.0	
Cyclic service and DPV1 service	< 2.3	

FB Name	FC_DPS7-S (Substitute)	FC_DPV1_RW
Programming language	STL	STL
Length of block (no. of bytes)	(256) 0.26 kb	(3.444) 3.4 kb
RAM required (no. of bytes)	(174) 0.2 kb	(3.116) 3 kb
Local data	20	104
Nesting depth	1	1
System functions called	DPWR_DAT	GADR_LGC BLKMOV WR_REC RD_REC
Data area assignment		DB_DPV1 = 1.332
Processing time in ms • CPU 315-2DP • CPU 413-2DP • CPU 414-2DP • CPU 416-2DP	Processing time per axis: < 0.75 < 0.5	Processing time included in FB_MCT

FB Name	FB_M7	FB_FM
Programming language	STL	STL
Length of block (no. of bytes)	(1.294) 1.3 kb	(6.086) 6.1 kb
RAM required (no. of bytes)	(910) 0.9 kb	(6.122) 6.1 kb
Local data	42	104
Nesting depth	1	1
System functions called	BLKMOV WR_REC RD_REC	BLKMOV WR_REC RD_REC
Data area assignment	IDB_M7 = 188 DB_PARA = 44 + 12 per axis	IDB_FM = 352 DB_FMPA = 124 + 36 per axis DB_AWP1 = 276 DB_AWP2 = 276 DB_IN = 276 DB_OUT = 276
Processing time in ms • CPU xxx (one axis with FB_FM)	xxx	In OB1: Processing time included in FB_M7. In OB40: xxx

8.9 General Information on Response Time: PROFIBUS-DP

The information given is applicable for the CPU313-2DP. The Profibus was operated at 1.5 MBaud.

The control and checkback signals are read and written with SFCs DPRD_DAT and DPWR_DAT. The processing time of the SFCs is approximately 200 µs for 20 bytes.

The time measured between enabling of the start signal on the CPU and the arrival of the checkback signal from the technology yielded a processing time of < 10 ms. The technology processes the control and checkback signals in a period of 3.2 ms.

A processing time measurement of the actual value via the PZD channel showed that a new actual value is available in the CPU approximately every 7 ms to 8 ms.

The following parameters are included in the calculation of the total time response for the MCT interface:

- PROFIBUS baud rate
- Number of stations
- Total cycle time of CPU
- All user input tasks which use the DPV1 service require at least 4 CPU cycles (best case).
- The technological acknowledgement depends on the reaction time of the technology, which itself depends on the basic sensor time set for the technology processing (basic sensor time for MCT = 3.2 ms).

The following parameters are included in the calculation of the total time response for the M7-FM interface:

- Number of M7-FM modules -> number of axes
- Total cycle time of CPU
- Single user input tasks require at least 2 CPU cycles;
(blocked user input tasks require at least 4 CPU cycles) (best case)
- The technological acknowledgement depends on the reaction time of the technology, which itself depends on the basic sensor time set for the technology processing (basic sensor time: for MCB via SIMOLINK = 3.2 ms, for IM178 via Profibus, the configured equidistance time is significant, with 3 Mbit/s for up to 8 axes and 6 Mbit/s=2 ms for up to 16 axes).

NOTE

For data exchange with the technology, the flat control tasks only require the "general task header", while blocked control tasks required the "technological task header" in addition to the "general task header".

8.10 Typical Plant Configurations

Estimated Memory for Typical Applications

	1 MCT axis	(1+n) MCT axes
FC_DPS7-S	174	174
FC_DPS7-E	174	174
FC_DPV1_RW	3116	3116
FB_FM	0	0
FB_M7	0	0
FB_START_MINI	1222	1222
FB_MCT	2650	2650
FB_JOB	2856	2856
Total FB:	10192	10192
DB_ORG	424	424**+n*16**
DB_CMD	136	136**+n*100**
DB_COM	144	144
DB_AWP1_MCT	276	276
DB_AWP1_M7	0	0
DB_OUT1 (M7)	0	0
DB_IN1 (M7)	0	0
DB_FMPA	0	0
DB_PARA	56	56**+n*12**
DB_DPV1	1322	1332
IDB for FB_START	428	428
IDB for FB_M7	0	0
IDB for FB_FM	0	0
IDB for FB_MCT	232	232
IDB for FB_JOB	138	138
Total DB:	3156	3156**+n*(16+100+12)**
Overall Total:	**13348**	**13348+n*128**

n describes the number of additional axes!

	1 M7-FM (k axes)	1 M7-FM (8 axes)	(1+m) M7-FM (per k_m axes)	1 M7-FM (k axes) and n MCT axes
FC_DPS7-S	0	0	0	174
FC_DPS7-E	0	0	0	174
FC_DPV1_RW	0	0	0	3116
FB_FM	6122	6122	6122	6122
FB_M7	910	910	910	910
FB_START_MINI	1222	1222	1222	1222
FB_MCT	0	0	0	2650
FB_JOB	2856	2856	2856	2856
Total FB:	11110	11110	11110	17224
DB_ORG	408+k*16	536	408**+g*16**	408+(**k+n)*16**
DB_CMD	36+k*100	836	36+**g*100**	36**+(k+n)*100**
DB_COM	144	144	144	144
DB_AWP1_MCT	0	0	0	276
DB_AWP2_MCT	0	0	0	0
DB_AWP1_M7	276	276	276**+m*276**	276
DB_AWP2_M7	0	0	0	0
DB_OUT1 (M7)	276	276	276**+m*276**	276
DB_IN1 (M7)	276	276	276**+m*276**	276
DB_FMPA	124+k*36	412	**m*124+g*36**	124+**k*36**
DB_PARA	44+k*12	140	44**+g*12**	44**+k*12**
DB_DPV1	0	0	0	1332
IDB for FB_START	428	428	428	428
IDB for FB_M7	188	188	188	188
IDB for FB_FM	352	352	352**+m*352**	352
IDB for FB_MCT	0	0	0	232
IDB for FB_JOB	138	138	138	138
Total DB:	2690+k*164	4002	2690**+ g*164+ m*1166**	4530**+ k*164+n*116**
Overall Total:	13800+ k*164	15112	13800**+ g*164+ m*1166**	21754**+ k*164+n*116**

g describes the total number of axes (M7 and MCT)
k describes the number of axes on a M7-FM
m describes the additional M7-FM modules.
n describes the number of additional MCT axes.

8.11 Typical Processing Times on the S7-CPU: CPU315-2DP

Distributed Solution with S7-300, CPU315-2DP

	1 MCT axis	(1+n) MCT axes
FB_MCT	4.2	4.2**+n*(0.75+0.75)**
FB_JOB	0.8	0.8
Total FB:	5.0	5.0**+n*(0.75+0.75)**

Centralized Solution with S7-400, CPU416-2DP(avail-able soon development)

	1 M7-FM (8 axes)	(1+m) M7-FM (per 8 axes)
FB_M7		
FB_JOB		
Total FB:		

	1 M7-FM (8 axes)	(1+m) M7-FM (per 8 axes)
FB_FM **in OB40**		
Total FB:		

m describes the additional M7-FM modules
n describes the number of additional axes

9 Expert Knowledge / Tips SIMATIC M7

Contents

In this chapter, you will find ...

9.1 Software Structure of the M7

9.1.1 Memory Card

Memory Card

All the files and programs you need to start the system can be found on the memory card supplied.

The following is a breakdown of the contents of the memory card:

- the real time operation system M7 – RMOS32
- a hardware configuration (system data blocks [SDB's])
- the GMC application
- the application data / parameters (data blocks [DB's])
- batch files to start the application

NOTE

The HW configuration on the flash represents the structure in which the memory module supplied to you was tested. You must replace this with your own HW configuration.

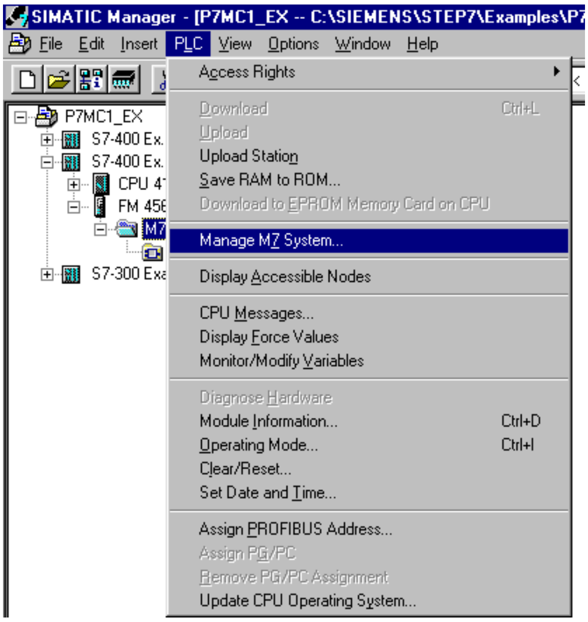
Read Contents of Memory Card

You can use 'Explorer' to look at the data structure on the memory card. To do this, proceed as follows:

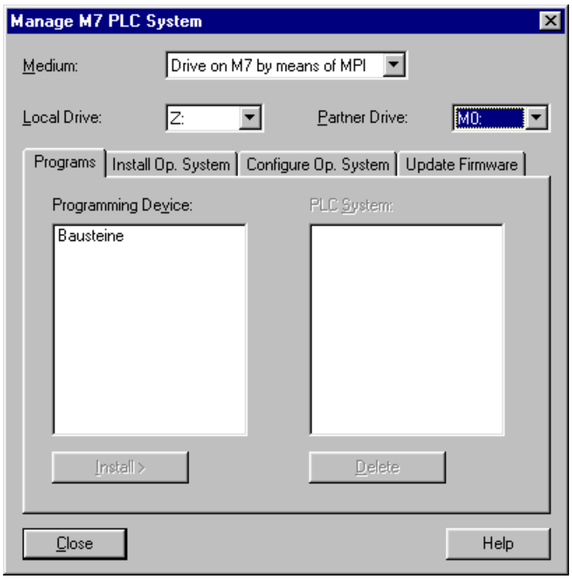
- You have a PG / PC with a memory card slot. This is given its own drive letter in Explorer

OR

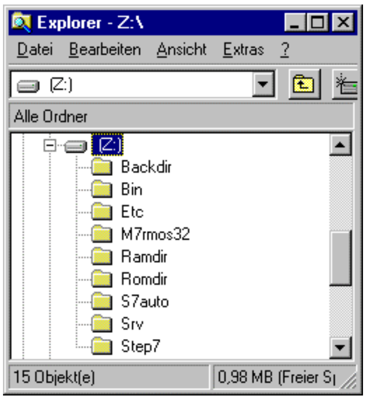
- You create an MPI connection to the M7-FM.
  i.e. in your S7/M7 project, first select the M7 program container



Go to target system – M7 – administer target system, and the dialog box below will appear. Complete the dialog as shown below. As long as this dialog is active, the content of the flash is shown in Explorer under the drive letter Z.



Go to Explorer.



View of container structure.

NOTE

The option of viewing the contents of the memory module using M7 - Administer target system is no longer available in Windows NT. The options are "grayed out" in the menus.

WARNING

Do not change any entries in files on the memory card using the GMC application, and do not delete or move any files.

Application Parameters

All the parameters for the application are stored in data blocks. If new parameters are transferred to the application, they are copied to the appropriate data block on the M7.

9.1.2 Memory Model of the M7

Memory Structure

There are five different memory areas within the M7 memory model, and it is the interaction between these areas which enables parameters to be backed up and stored, even when the M7 is switched off. They are as follows:

- ROMDIR
- RAMDIR
- BACKDIR
- SRAM
- WORKING MEMORY

The structure of the various memory areas is organized such that data (DBs or SDBs) can be available in multiple form throughout. The remanence is achieved by means of a set copying sequence from the existing memory areas to the target area, i.e. the RAM.

ROMDIR

The ROMDIR directory (fixed memory) stores the parameters / tables supplied with the M7 application.

RAMDIR

The RAMDIR (permanent write memory) stores all the data you have saved using SIMATIC-MANAGER. For example, the hardware configuration for your user project or your M7 user DBs.

BACKDIR

The BACKDIR directory (backup memory) stores the parameters changed by the running of the application.

SRAM

This buffered RAM area is not a part of the MEMORY-CARD. It is also not used for data backup!

This enables a complete change of parameterization simply by pulling and plugging the memory card, e.g. if the hardware changes.

Working Memory

The working memory is an internal RAM memory. During the system startup phase, all the relevant data is copied to RAM. The application works exclusively with the data in the RAM memory. This data can be viewed using SIMATIC-MANAGER.

9.1.3 Memory Access on M7 System Startup

When the M7 system starts up, the memory areas detailed above are processed in the following order:

POWER ON / New Start

1. Delete all data blocks in working memory
2. Copy DBs from ROMDIR to working memory
3. Copy DBs from RAMDIR to working memory
4. Copy DBs from BACKDIR to working memory

During the copying process, any DBs with the same name are automatically overwritten.

9.1.4 Data Backup for the M7 Application

Default Status

When the system is supplied, the parameters and tables are stored in ROMDIR. The copying sequence described above for the data from the various memory areas guarantees that it is always the most up to date status which is loaded into the working memory for you to continue working.

Overall Reset

You can reapply the initial defaults settings at any time using the "overall reset" function on the M7-FM.

The overall reset function first deletes all the S7 objects in the working memory (buffered and not buffered), in BACKDIR and in RAMDIR. Then all the S7 objects in ROMDIR (fixed memory) are copied to the working memory and activated.

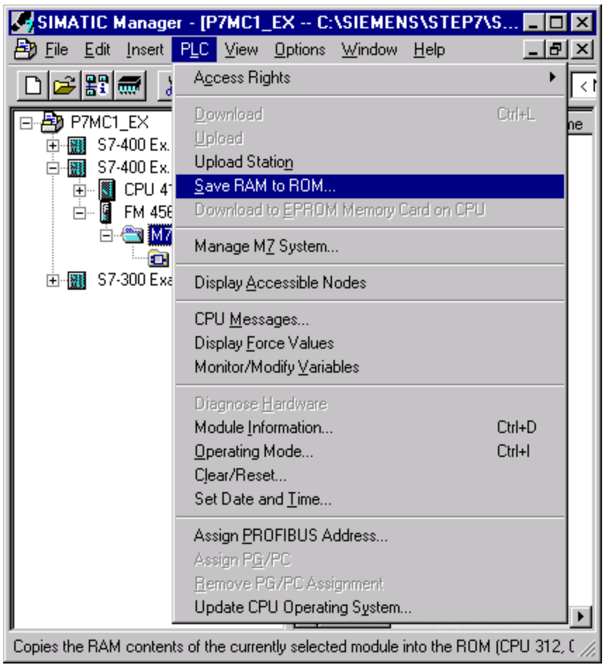
NOTE

Before a general reset, the M7-FM and the S7-CPU must be set to STOP. It is possible that the system will have to be rebooted after a general reset.

The general reset also deletes the application’s hardware configuration. This means that you will have to reload your HW configuration into the M7-FM using the SIMATIC MANAGER before restarting the system.

The default setting is that the hardware configuration, which is stored in system data blocks, is copied to the RAMDIR of the M7. In order that your system will start up again without you having to transfer the hardware configuration again, you need to copy the system data blocks to ROMDIR.

SIMATIC Manager supports the copying of data from RAMDIR to ROMDIR. To do this, select the M7 program container and select the "Copy RAM to ROM" function in the Target system menu.



Then you can run an overall reset on the M7 without having to transfer the hardware configuration again.

Data Backup

Data is backed up in the BACKDIR directory. Saving parameters and tables to the memory card of the M7 is activated in the default status and after the system is switched on (after power ON).

This mechanism guarantees that all the parameters changed during operation are stored and buffered, which enables you to return to the parameterization status you had before switching the system off when you switch the M7 back on again.

NOTE

Data stored in the BACKDIR directory are not accessible to the SIMATIC Manager in ONLINE PROJECT mode. They can only be modified by GMC operation commands.

Access to the memory card is displayed by a flashing "SD" LED on the M7.

NOTE

Saving parameters and tables to the memory card of the M7 affects the communication speed. Any new task is only accepted once the data from the previous task have been saved completely. Tasks triggered in the interim period are terminated with the error message "2251-M7: Data transfer not possible because data backup is still active". In this case, you have to trigger the task again. Depending on the data volume and the capacity of the M7, the times are up to three seconds.

CAUTION

The main order "Write data to EEPROM or ROM" allows you to decide while the system is running whether the next data should only be written to the working memory or whether it should also be written to BACKDIR (EEPROM). If "Write data to RAM" is set, the parameters are only stored in the working memory of the M7 and are lost if the system is rebooted or the power switched off.

9.2 Communication between S7 and M7

9.2.1 Overview of Communications

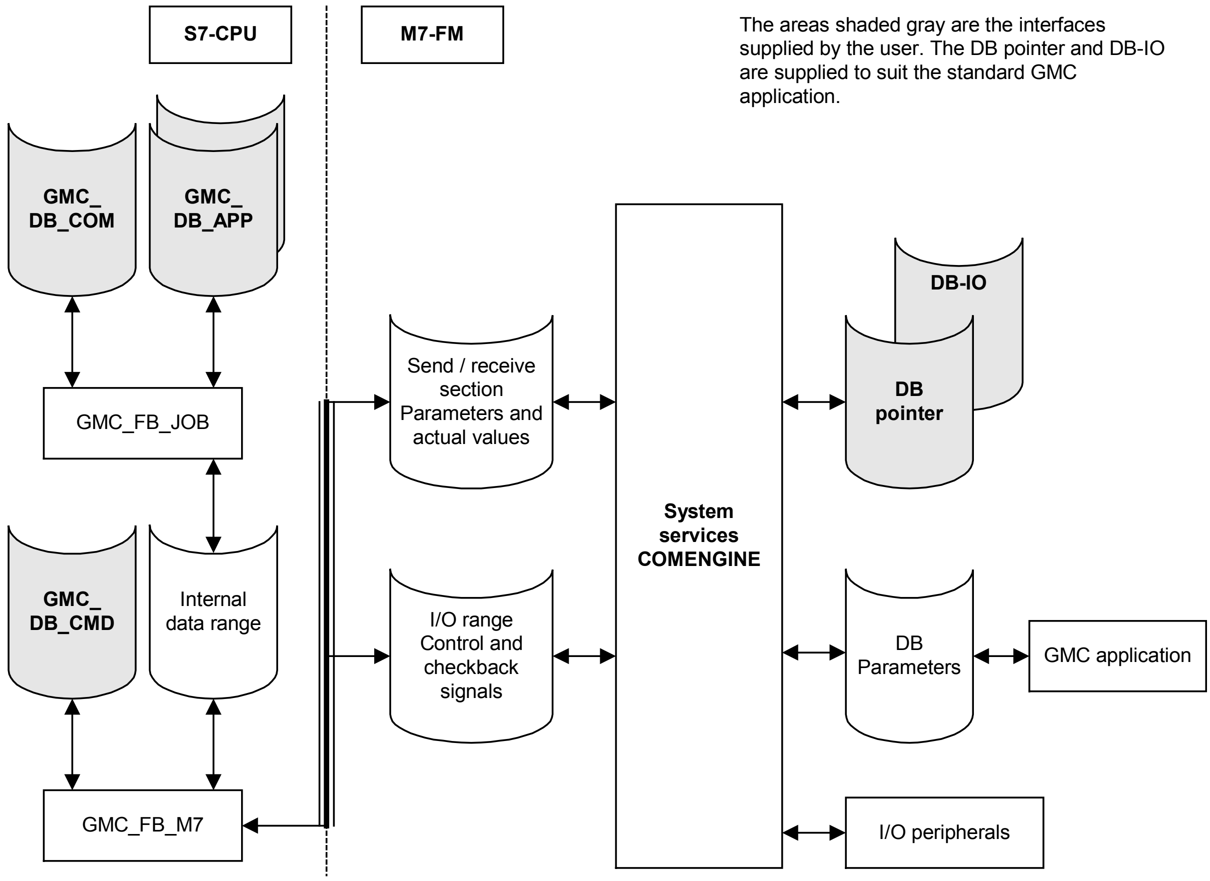


Fig. 9-1 Communication routes within the GMC – M7 application

ComEngine

The ComEngine processes communications between the controller program on the S7-CPU and the application on the M7-FM.

The ComEngine transfers the following data between M7 and S7:

- User data (parameters and actual values)
- Control signals
- Checkback signals

The ComEngine is also responsible for the following tasks:

- Data backup on the memory card
- Communication between the application and the drive
- Executing all peripheral modules and interfaces (including Simolink and Profibus-DP)

Data Blocks

The communications and data interfaces are precisely defined within the GMC application. These definitions are set within the central GMC_DB_ZEIGER and GMC_DB_IO data blocks.

All the other data blocks in the application are defined within these two data blocks.

NOTE

The data blocks are saved on an application-specific basis (8 axis or16(+) axis application, additional IF modules). You must not make any changes to the data blocks!

The standard M7 application provides the entire modules structure (fixed configuration 8/16(+) axes). If necessary, you can adapt the configuration. To do this, you need an Excel-based configuration tool, which is not supplied as standard, but is available on request.

9.3 M7 Security Functions

General

Under certain circumstances, fatal damage can be caused to an automation system if the controller does not behave in the defined manner because of an error or a defect. Within the GMC application, the M7 security function ensures that errors or error conditions are recognized and evaluated. The error is passed on to the superordinate controller and the M7 and its peripherals is brought into a defined condition.

NOTE

Error conditions in the operating system or the M7 driver are not recognized by the M7 security functions.

Component Parts

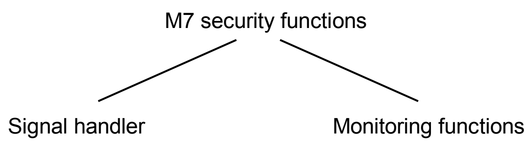


Fig. 9-2 Component parts of the M7 security functions

In terms of software, the signal handler and the life-sign monitoring system are component parts of the GMC application.

Start Monitoring Systems

In the GMC application, the security function is enabled as standard.

System Start

The different phases of the system startup mean that it is not necessarily possible to report errors which occur during startup using the security functions. Therefore, the means available at any one point are used to report errors. A description of this can be found in the chapter on "Diagnosis messages on system startup".

Signal Handler

Exception conditions, such as division by 0, addressing errors or faulty floating point operations, trigger a processor interrupt. This incoming interrupt is received and processed by the signal handler.

Diagnostic Buffer

An appropriate entry is made in the diagnostic buffer of the M7 which then goes into a "STOP" status. You can read the diagnostic buffer using the "Administer target system – Diagnostic messages" function.

Diagnostic Alarm

A diagnostic alarm is triggered to inform the superordinate controller (S7-CPU) about the status of the M7-FM.

Monitoring Functions

The application tasks are controlled by a life-sign monitoring system.

The aim of the life-sign monitoring system is to receive error conditions which cannot be processed by the signal handler alone (e.g. endless loops or runtime errors within a task). In the event of an error, an appropriate entry is made in the diagnostic buffer of the M7. If a serious runtime error occurs, the M7 is then brought into a "STOP" status. A diagnostic alarm is triggered to inform the superordinate controller (S7-CPU) about the status of the M7-FM.

NOTE

Within the application, the low-priority background task is not started in a fixed pattern. The start frequency depends on the extent to which the system is working to capacity. Retriggering must take place once within the set monitoring time if the monitoring system is not to respond.

The control task has to be started in a set pattern (equidistant). In this case, the task monitoring system is in two stages:

- The system checks whether the task is retriggered at the correct frequency within the basic monitoring cycle.
- At the controller level, the system checks whether the starts of the task are equidistant. To do this, it monitors the time between the individual task starts (assessed with a tolerance time, Δt).

NOTE

The life-sign monitoring system works on the basis of default values. These values are adapted to the standard GMC application.

Monitoring Events

The life-sign monitoring system makes a distinction between the following monitoring events:

- Event 0: Time error
- Event 1: Time over monitoring time + tolerance time
- Event 2: Time under monitoring time – tolerance time
- Event 3: Task retriggered too seldom

NOTE

The tolerance time is defined as follows:

$$\Delta t = \frac{\text{overall cycle time}}{2}$$

Representation in Timeline Diagrams

Tasks which run in a set pattern are normally high-priority control tasks which require fixed scanning times from their software controller.

The time diagrams are based on the following assumptions:

- Basic monitor cycle = 16 ms
- Task monitoring time = 3.2 ms +/- 1.6 ms

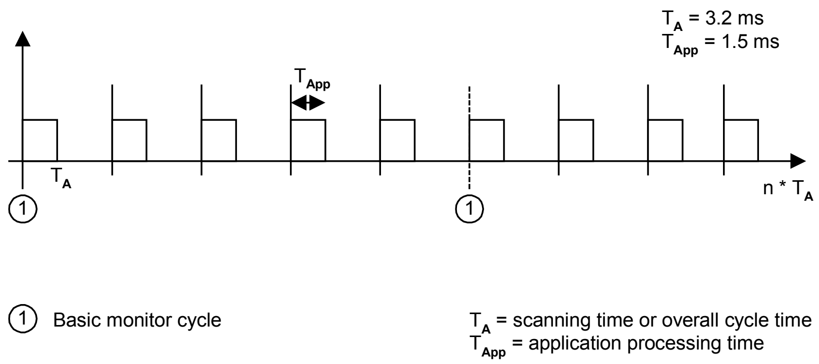
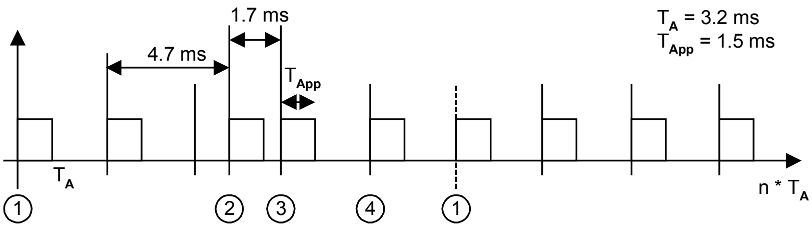


Fig. 9-3 Timeline diagram, error-free tasks in a fixed pattern

The task triggers a life-sign on each rising edge. From the point of view of the monitor, the task must be retriggered at least four times during two basic task intervals, bearing in mind the tolerance time.

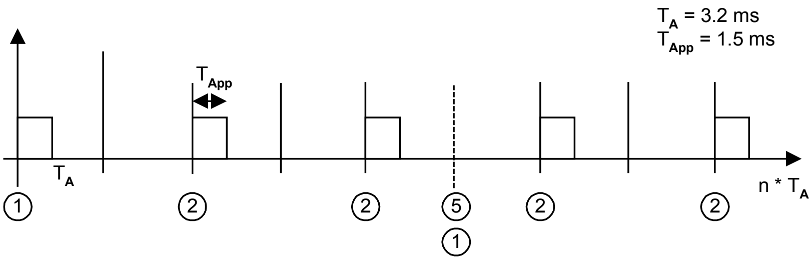


(1) Basic monitor cycle

T_A = scanning time or overall cycle time
T_{App} = application processing time

Report of...
(2) Event 1: Time over monitoring time + tolerance time
(3) Event 2: Time under monitoring time - tolerance time
(4) Event 0: Time error

Fig. 9-4 Timeline diagram, tasks with errors in fixed pattern



(1) Basic monitor cycle

T_A = scanning time or overall cycle time
T_{App} = application processing time

Report of...
(2) Event 1: Time over monitoring time + tolerance time
(5) Event 3: Task retriggered too seldom

Fig. 9-5 Time-line diagram, free-running task with errors

NOTE

The monitoring time, or watchdog, is always activated if a task is not retriggered within an appropriate period of time.

9.4 Diagnostic Signals

General

If the security function detects that a monitored task has not been triggered or receives an interrupt signal (IR signal), this leads to the following actions, depending on the monitoring event.

9.4.1 Diagnostic Signals on Receipt of an Interrupt Signal

Operating Principle

When a diagnostic signal is logged by the monitor program or the GMC application, the procedure is as follows:

- the registry contents of the task triggering the interrupt are entered into the diagnostic buffer of the M7 as detailed information, as is the signal type,
- a diagnostic alarm is triggered to the superordinate S7 controller and
- then the M7 is brought to a STOP.

Diagnostic Entry on IR Signals

An event ID is stored, and the task ID of the signal triggering the task is stored in additional information 1 of the diagnostic buffer.

The event ID consists of three hexadecimal 16 bit parameters, and is structured as follows:

♦ Event code	16#Axxx	
♦ Direction	16#x1xx :	incoming signal
♦ Event	16#xx10 :	signal recognized

As the task is always terminated if an IR signal is received, the registers of the terminated task are also entered as additional information 2 and 3, so you can trace the cause of the software error.

This means there are 16 diagnostic entries for one signal, which are allocated as follows (last entry first):

ZI1	**ZI2/3**
Task-ID	Signal and type
Task-ID	PSW
Task-ID	CS
Task-ID	EIP
Task-ID	Error code
Task-ID	FS
Task-ID	ES
Task-ID	DS
Task-ID	BP
Task-ID	DI
Task-ID	SI
Task-ID	EAX
Task-ID	EBX
Task-ID	ECX
Task-ID	EDX

"STOP M7" Command from IR Signals

The parameterization / programming within the GMC application is such that the M7 goes to a STOP status when an IR signal is received. This results in an additional diagnostic entry.

In this case, the event ID is a single hexadecimal 16 bit parameter:

Event code 16#B3xx where xx indicates the IR signal triggered.

List of IR Signals

In general, the following 7 interrupt signals can occur. There is an additional specification for each interrupt signal.

- 0:= SIGFPE (Floating Point Error)
  - 0:= - Division by zero
  - 4:= - Overflow
  - 7:= - Device unavailable
  - 16:= - by NDP 80x87
- 1:= SIGILL (Illegal Opcode)
  - 6:= - Illegal Opcode
- 2:= SIGSEGV (Segment violation)
  - 5:= - Bound Violation
  - 9:= - Coprocessor Segment Violation
  - 10:= - TSS Error
  - 11:= - Segment does not exist
  - 13:= - General Protection Fault
  - 14:= - Page - Error
  - 17:= - Allignment Check Error
- 3:= SIGABRT (occurs when raise is called)
- 4:= SIGTERM (occurs when raise is called)
- 5:= SIGUSR1 (occurs when raise is called)
- 6:= SIGUSR2 (occurs when raise is called)

If an interrupt signal occurs, this information is passed on to the superordinate S7 controller within diagnostic data set 1. The number of the signal is entered in byte 7, and the number of the event triggering the interrupt is entered in byte 8.

Information about the evaluation of diagnostic alarms and the data transferred to the S7-CPU can be found in Chapter 8.2 of the GMC Basic Documentation.

9.4.2 Diagnostic Signals from Life-Sign Monitoring

Operating Principle

The reaction of the monitor program or the GMC application to the diagnostic signal depends on the event which has occurred.

Essentially, for all events, 0, 1, 2 and 3, the reaction is to make an appropriate entry in the diagnostic buffer of the M7.

If event 3 occurs, a diagnostic alarm to the superordinate S7 controller is also triggered, and the M7 is brought to a STOP.

Diagnostic Entry for Monitoring Signals

If the life-sign monitoring system is addressed, the event ID is stored, while the task ID of the task triggering the event and the time in hexadecimal format are stored as additional information.

The event ID consists of three hexadecimal 16 bit parameters, and is structured as follows:

Event code	16#Axxx	
Direction	16#x0xx :	Outgoing signal
	16#x1xx :	Incoming signal
Event	16#xx00 :	Outgoing time error
	16#xx01 :	Time over monitoring time + jitter
	16#xx02 :	Time under monitoring time - jitter
	16#xx03 :	Task retriggered too seldom

STOP M7” Command from Monitoring Signals

The parameterization / programming within the GMC application is such that the M7 goes to a STOP status when a monitoring signal with the event code xx03 is received.

This results in an additional diagnostic entry.

In this case, the event ID consists of two hexadecimal 16 bit parameters, and is structured as follows:

Event code	16#B1xx	
Event	16#xx03 :	Task retriggered too seldom

Diagnostic Alarm to the S7-CPU

If the M7 goes into a STOP status because of a monitoring event, then a diagnostic alarm is triggered on the S7-CPU at the same time. Information about the evaluation of this diagnostic alarm and the data transferred to the S7-CPU can be found in Chapter 8.2 of the GMC Basic Documentation.

9.5 Diagnostics using Step7

General

In addition to the security functions (Chapter 9.2) which recognize and report all errors under normal circumstances, the following diagnostic tools are also available:

- Step7: read the diagnostic signals on the M7
- Step7: variable tables (VATs)
- Step7: monitor data blocks (DBs)
- Step7: LOW – Level Debugger

Requirements

You need:

- a PG / PC with an MPI interface
- Step7 Version V5.0 with service pack 2, M7-SYS V5.00.10
- MPI connection cable (PG/PC to S7-CPU)

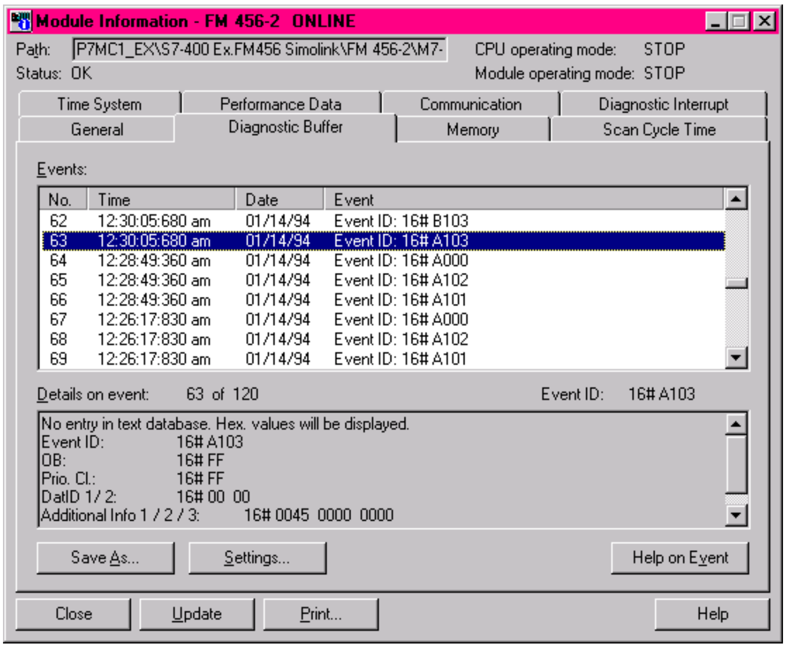
NOTE

With MASTERDRIVES MC, there is the option of monitoring the setpoints and actual values of the application while the system is running. There is also an additional tool available in the form of SIMOVIS/DriveMonitor (monitoring parameters, trace).

9.5.1 Read Diagnostic Signals on M7

Select the M7 program directory within your STEP7 project (OFFLINE). Then select "Target system" – "Administer M7 target system" – "Component status" – "Diagnostic buffer" to get to the diagnostic entries.

The following example was caused by a break in the SIMOLINK cable. This means that the controller task triggered by the Simolink interrupt is no longer being processed by your life-sign monitoring system and the monitoring function brings the M7 into a STOP status.



Example: Entries triggered by the life-sign monitoring system

Variable Tables

The example project supplied with GMC Basic (P7MC1_EX) incorporates variable tables which have already been completely filled in and commented, and these can be used for diagnostic purposes. To do this, select the block container in the M7 program. This gives you the option of viewing internal data in the M7 application while the system is running. The variable tables are saved in the block containers of the FM456-2.

Symbol	**Address**	**Comment**
Checkback M7->S7 (DB78)	VAT 78	Internal Data Interface between M7 & S7 (DB78)
Control S7->M7 (DB79)	VAT 79	Internal Data Interface between S7 & M7 (DB79)
SIMOLINK: Receive-Direc.	VAT 80	Receive Data of SIMOLINK (DB80)
SIMOLINK: Send-Direction	VAT 81	Send Data of SIMOLINK (DB81)
SIMOLINK: Status	VAT 82	Status of SIMOLINK (DB82)
DB Instance Com-Engine	VAT 83	Internal Information of the COM-Engine
GMC_DEBUG-INFO	VAT 84	Internal Debug - Information of GMC
DB_IO_Values	VAT 99	DB_IO_Values
Control/Checkback(DB11)	VAT 111	Application Data Interface between S7 & M7 (DB11)
Order Flow of COM-Engine	VAT 115	Order Flow of COM – Engine
Intern. order of COM-Eng	VAT 117	Internal Task Data of the COM-Engine

ONLINE Observation of Data Blocks

In addition to the variable tables, you also have the option of observing data blocks in ONLINE mode using the STEP7 editor. Here, too, the data is updated "cyclically".

LOW LEVEL Debugger

The LOW LEVEL debugger gives you the option of investigating system-internal processes (with the help of the hotline). However, you are very unlikely to need to use this, so it will not be dealt with here.

9.5.2 Diagnostic Alarms to the S7-CPU

Diagnostic Data

If the M7 goes into a STOP status because of a monitoring event, then a diagnostic alarm is triggered on the S7-CPU almost at the same time. 16 byte diagnostic data is transferred from the FM456-2 to the S7-CPU. The first 4 bytes are reserved for system services, and the application information is stored in bytes 5 to 16.
Please refer to Chapter 8.2 "Troubleshooting – Assessing data sets DS1 and DS2 in organization block OB82".

9.5.3 Diagnostic Signals on System Startup

Startup Phases

On the M7, system startup with the GMC application is divided into the following phases:

- System startup
- Application startup
- Application initialization

During these phases, it is not yet possible to report any errors which occur using the security functions. Therefore the resources available at any time are used to log and report any errors which occur.

System Startup

The system is started up by the operating system. Any errors detected are reported in accordance with the rules described in the M7 documentation.

Application Startup

During startup, the operation environment for the application is generated, i.e. the data on the M7 flash EPROM is copied to the M7 working memory (RAM). This is done using the operating system’s standard mechanisms. If an error occurs during this process, then:

- startup is cancelled
- a diagnostic signal is triggered and sent to the S7
- the USR2 – LED and the RUN – LED on the FM456-2 start flashing

System Initialization

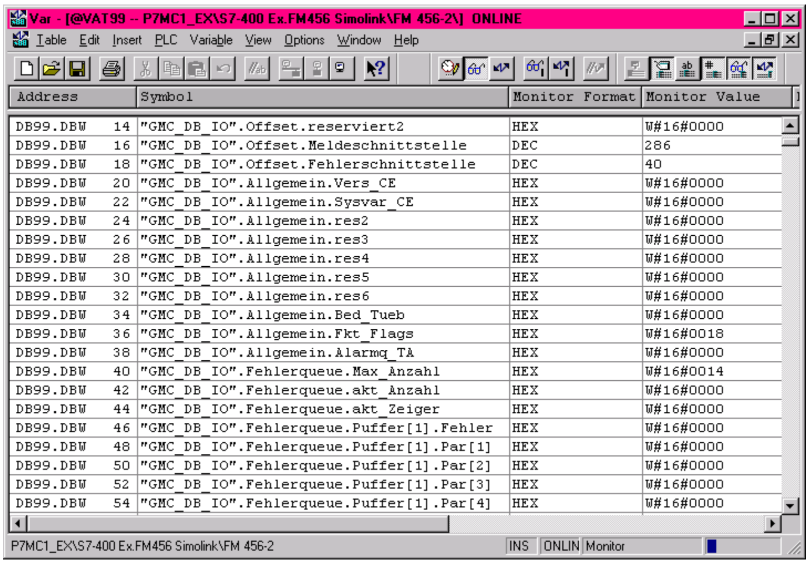
The initialization brings the application into its starting status. If an error occurs at this stage, then:

- initialization is cancelled,
- if applicable, the USR1-LED on the FM456-2 starts flashing,
- if applicable, the FM456-2 is brought to a STOP,
- if applicable, a diagnostic signal is triggered and sent to the S7,
- if applicable, additional reports are entered in the diagnostic buffer of the S7-CPU and the FM456-2.

If the USR1-LED is flashing, you can obtain information about the error within the data block GMC_DB_IO (DB99) on the FM456-2, or in the M7ERR.LOG file in the root directory of the M7 MEMORY CARD.

Variable Table 99

You can access the contents of the GMC_DB_IO data block (DB99) using the variable table VAT99, which is in the M7 block container. Under certain circumstances, the cyclical updating of the variable table may not work. In this case, you have to read the values by clicking on the "Update status values" button.



Address		Symbol	Monitor Format	Monitor Value
DB99.DBW	14	"GMC_DB_IO".Offset.reserviert2	HEX	W#16#0000
DB99.DBW	16	"GMC_DB_IO".Offset.Meldeschnittstelle	DEC	286
DB99.DBW	18	"GMC_DB_IO".Offset.Fehlerschnittstelle	DEC	40
DB99.DBW	20	"GMC_DB_IO".Allgemein.Vers_CE	HEX	W#16#0000
DB99.DBW	22	"GMC_DB_IO".Allgemein.Sysvar_CE	HEX	W#16#0000
DB99.DBW	24	"GMC_DB_IO".Allgemein.res2	HEX	W#16#0000
DB99.DBW	26	"GMC_DB_IO".Allgemein.res3	HEX	W#16#0000
DB99.DBW	28	"GMC_DB_IO".Allgemein.res4	HEX	W#16#0000
DB99.DBW	30	"GMC_DB_IO".Allgemein.res5	HEX	W#16#0000
DB99.DBW	32	"GMC_DB_IO".Allgemein.res6	HEX	W#16#0000
DB99.DBW	34	"GMC_DB_IO".Allgemein.Bed_Tueb	HEX	W#16#0000
DB99.DBW	36	"GMC_DB_IO".Allgemein.Fkt_Flags	HEX	W#16#0018
DB99.DBW	38	"GMC_DB_IO".Allgemein.Alarmq_TA	HEX	W#16#0000
DB99.DBW	40	"GMC_DB_IO".Fehlerqueue.Max_Anzahl	HEX	W#16#0014
DB99.DBW	42	"GMC_DB_IO".Fehlerqueue.akt_Anzahl	HEX	W#16#0000
DB99.DBW	44	"GMC_DB_IO".Fehlerqueue.akt_Zeiger	HEX	W#16#0000
DB99.DBW	46	"GMC_DB_IO".Fehlerqueue.Puffer[1].Fehler	HEX	W#16#0000
DB99.DBW	48	"GMC_DB_IO".Fehlerqueue.Puffer[1].Par[1]	HEX	W#16#0000
DB99.DBW	50	"GMC_DB_IO".Fehlerqueue.Puffer[1].Par[2]	HEX	W#16#0000
DB99.DBW	52	"GMC_DB_IO".Fehlerqueue.Puffer[1].Par[3]	HEX	W#16#0000
DB99.DBW	54	"GMC_DB_IO".Fehlerqueue.Puffer[1].Par[4]	HEX	W#16#0000

The above is an extract from the GMC_DB_IO data block (DB99), which is displayed using the VAT99 variable table. The hotline team can use this data to help you troubleshoot problems.

Index

I

M

O

P

R

S

T

U

SIEMENS

System Solutions

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Part 2: SIMATIC S7 Communication Functions 11.2002

Task Description

SIEMENS

System Solutions

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Part 2: SIMATIC S7 Communication Functions

Task Description

Edition 11.2002

Documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the "Remarks" column.

Status code in the "Remarks" column:

A .... New documentation.
B .... Unrevised reprint with new Order No.
C .... Revised edition with new status.

Edition	Order No.	Remarks
04.98	6AT1880-0AA00-1BA0	**A**
10.99	6AT1880-0AA00-1BC0	**C**
11.2002	6AT1880-0AA00-1BE0	**C**

Order No.: 6AT1880-0AA00-1BE0
Printed in the Federal Republic of Germany

Siemens-Aktiengesellschaft

Contents

1 Overview

General Information

"Tasks" are used for data communication between the user and the technology. The number of tasks depends on the number of different types of data.

You only need to be familiar with these tasks in order to implement data exchange between your S7 user program and the technology. If you use the standard user interfaces for input/output of technology data, you can ignore the task descriptions below.

This description is concerned only with the assignment of individual data areas for each possible task. Handling procedures and the various options available for task management are described centrally in the "GMC-BASIC Standard Software" document.

1.1 Representation of Data Words

In the graphical representation of the data words, the number of the data block byte (or word or doubleword) is shown on the left, and the data format is shown on the right.

DBDz	Lower input limit	DEC
DBDz+4	Upper input limit	DEC

The key to the format specifications is as follows:

BIN : Bitmap
DEC: Decimal number
HEX: Hexadecimal
ZEI : ASCII character

1.2 Notes on the Task Header

Structure

A task description basically consists of a task header which describes the data type and the transfer direction and the task data itself. Although there are a large number of different types of tasks, the basic mechanism is always the same.

So that more than one task can be processed in sequence after a task is triggered, there is also a leader in the task description which simply incorporates the number of partial tasks.

The allocation of the task header is largely pre-defined. All the different task formulations can be found in the “Task descriptions” documentation. Therefore, all you have to do when creating a task is to enter the task header in accordance with the description, set the "DB no." and DBW no." pointers, define the axis number and the data direction and provide enough memory for the task data or enter the task data.

NOTE

The areas marked in dark gray are set by the user. Light gray areas serve to set the transfer options. White areas have a fixed setting and cannot be changed.

Pointer for GMC_DB_COM

Data block GMC_DB_APP1
Leader for administering multiple partial tasks

DBWx-2	Number of partial tasks	DEC

Data block GMC_DB_APP1:
General task header

DBWx	Main task number	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer	DEC
DBWx+6	Data type	DEC
DBWx+8	Number of elements	DEC
DBWx+10	DB no. GMC_DB_APP2	DEC
DBWx+12	DBW no. GMC_DB_APP2	DEC

or

Data block GMC_DB_APP2:
Technological task header (task-dependent)

DBWy	Number of DW								DEC
DBBy+2	E	A	0	0	0	0	0	0	BIN
DBBy+3	Task code								DEC
DBWy+4	DB no. GMC_DB_APP3								DEC
DBWy+6	DBW no. GMC_DB_APP3								DEC
DBWy+8	Additional information 1 (Z1)								DEC
DBWy+10	Additional information 2 (Z2)								DEC
DBWy+12	Additional information 3 (Z3)								DEC
DBWy+14	Additional information 4 (Z4)								DEC
DBWy+16	Additional information 5 (Z5)								DEC
DBWy+18	Additional information 6 (Z6)								DEC

Data block GMC_DB_APP2 or GMC_DB_APP3:
Task data

DBWz	Task data

Number of Partial Tasks

The number of partial tasks determines how many task descriptions (partial tasks) are to follow. This mechanism allows you to create a task list which processes all the tasks in sequence on the basis of a single task trigger (see GMC_DB_COM).

For example, if you have to transfer data on a number of axes after switching the controller on, you can create an appropriate task list and process all the tasks with a single command.

Parts of the Task Header

A typical task header consists of the following elements:

- ♦ Main task number (DBWx):
  A main task number is defined for every task. For simple tasks, the main task number also determines the data transfer direction, as follows:
  - Positive main task number → input (to technology)
  - Negative main task number → output (from technology)
- ♦ Absolute axis number (DBWx+2):
  For the configuration in the data block GMC_DB_ORG in the “Axis descriptions” sector, all axes are allocated an absolute axis number. This axis number is now required as a source / target specification for the processing of the task.
- ♦ Element pointer (DBWx+4):
  The "Element pointer" specifies the element number at which the data range in the GMC_DB_APP2 begins.
  The entry for the element pointer is only variable for simple tasks. In all other cases, the element pointer is set to “1”.
- ♦ Data type (DBWx+6):
  The "data type" sets the size of an element
  2 = Word,
  5 = Dword.
  The data type is set in accordance with the task selected. No other data types are currently used.
- ♦ Number of elements (DBWx+8):
  The "number of elements" determines the length of the data range in GMC_DB_APP2 of a “data type” type.
  The entry for the number of elements is only variable for simple tasks. In all other cases, the number of elements is determined on the basis of the task selected. DB no. GMC_DB_APP2 and DBW no. GMC_DB_APP2:
  Pointer to the GMC_DB_APP2 (task data or technological task header)
- ♦ DB no. GMC_DB_APP2 and DBW no. GMC_DB_APP2:
  Pointer to the GMC_DB_APP2 (task data or technological task header)

Example of a Simple Task

Assuming that the data for a technological function consists of 7 elements of data type “word”, and that elements 3 to 6 need to be transferred from your S7 application to the technology (input).

The following settings have to be made:

- the main task number required
- the axis (axis number) to which you want to transfer the data
- element pointer = 3, because element 3 is the first element to be transferred
- data type = 2, because the elements are words
- number of elements = 4, because you want to transfer 4 elements (3,4,5 and 6)
- the DB no. and the DBW no. as a pointer to the data in the 4 elements, and finally the data in the 4 elements

Data block GMC_DB_APP1:

DBWx	Main task number n	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **3**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **4**	DEC
DBWx+10	DB no. DB-APP2	DEC
DBWx+12	DBW no. DB-APP2	DEC

Data block GMC_DB_APP2:

DBWy	Value for element 3
DBWy+2	Value for element 4
DBWy+4	Value for element 5
DBWy+6	Value for element 6

If you want the data to be sent from the technology (output), simply enter the main task number with a negative sign (-n). The remaining settings are still valid as described above.

NOTE: See also documentation on “GMC-Basic standard software”.

2 Machine Data and Function Parameter Tasks

Contents

In this chapter you will find a description of the tasks for the machine data.

2.1 Input/Output Machine Data

Data Block GMC_DB_APP1

Address	Content	Format
DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **7**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

Address	Content	Format
DBWy	Number of DWs = **0**	DEC
DBBy+2	**E** | **A** | 0 | 0 | 0 | 0 | 0 | 0	BIN
DBBy+3	TI = **1**	DEC
DBWy+4	DB no.	DEC
DBWy+6	DBW no.	DEC
DBWy+8	Z1 = **1**	DEC
DBWy+10	Z2 = first MD number **x**	DEC
DBWy+12	Z3 = number of MD numbers **y**	DEC

Data Block GMC_DB_APP3

Address	Content	Format
DBDz	Machine data x	DEC
DBDz+4	Machine data x+1	DEC
	⋮	
DBDz+n	Last machine data x+y	DEC

Task Description

The "input/output machine data" task can be used to input or output the machine data of the technology (MD1 to MD50). One machine data, part of the machine data or all of the machine data can be input or output by specifying extension 2 (Z2) and extension 3 (Z3). Each machine data occupies 4 bytes (1 doubleword) in the data block. After input, the machine data must be activated with the "activate machine data" task.

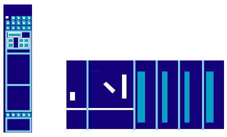
Further Information

Machine data numbers MD51 to MD70 are input or output using a single order. A description of the order can be found in the chapter on "Orders for SIMATIC Motion Control".

You will find a list and description of the machine data in the Function Description in the chapter entitled "Machine Data und Parameter of the Technology".

2.2 Input / Output Function Parameters

NOTE



Function implemented from:

MASTERDRIVES MC:	≥ V1.4x
SIMATIC Motion Control:	≥ V1.00.48

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **7**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**E**	**A**	0	0	0	0	0	0	BIN
DBBy+3	TI = **5**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **1**								DEC
DBWy+10	Z2 = first FP number **x**								DEC
DBWy+12	Z3 = number of FP numbers **y**								DEC

Data Block GMC_DB_APP3

DBDz	Function parameter x	DEC
DBDz+4	Function parameter x+1	DEC
	⋮	
DBDz+n	Last function parameter x+y	DEC

Task Description

The "Input / output function parameters" task can be used to input or output the function parameters FP1 to FP10 to or from the technology. One function parameter, some of the function parameters or all of the function parameters can be input or output by specifying extension 2 (Z2) and extension 3 (Z3). Each function parameter occupies 4 bytes (1 doubleword) in the data block.

The function parameters are effective as soon as they have been transferred. They do not need to be activated in the same way as machine data.

2.3 Activate Machine Data

Data Block GMC_DB_APP1

DBWx	Main task number = **5**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **0**	DEC
DBWx+10	**0**	DEC
DBWx+12	**0**	DEC

Task Description

When machine data are transferred to an axis, they are stored initially in a buffer. The axis continues to use the previous set of machine data.

The "activate machine data" task is used to replace the previous set of machine data with the machine data in the buffer. The advantage is that the machine data can be transferred while the axis is moving, without affecting the axis behavior. It is only possible to activate the machine data when the axis is stationary. The machine data set is subject to a plausibility check on activation. If an error is detected, an error message is output and the machine data are not activated.

2.4 Machine Data - Limit Value Output

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **7**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **1**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **2**								DEC
DBWy+10	Z2 = MD number								DEC
DBWy+12	Z3 = axis type according to MD 1								DEC

Data Block GMC_DB_APP3

DBDz	Lower input limit	DEC
DBDz+4	Upper input limit	DEC

Task Description

The "machine data - limit value output" task is used to output the upper and lower input limit of the selected machine data. The axis type parameter outputs the limits for the selected axis type. The axis type corresponds to the value in machine data 1.

Further Information

The limit values for machine data numbers MD51 to MD70 are output via a separate task. A description of the task can be found in the chapter on “Tasks for SIMATIC Motion Control".

2.5 Function Parameters - Limit Value Output

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **7**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **5**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **2**								DEC
DBWy+10	Z2 = MD number								DEC
DBWy+12	Z3 = **1**								DEC

Data Block GMC_DB_APP3

DBDz	Lower input limit	DEC
DBDz+4	Upper input limit	DEC

Task Description

With the task "Function parameters – limit value output" the upper and lower input limit of the required function parameter is output.

3 NC Program Function Tasks

Contents

In this chapter you will find a description of the tasks for the NC program functions.

3.1 Clear NC Program Memory

Data Block GMC_DB_APP1

DBWx	Main task number = **13**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **0**	DEC
DBWx+10	**0**	DEC
DBWx+12	**0**	DEC

Task Description

The "clear NC program memory" task clears the complete memory area for NC program data.

The task can only be initiated when automatic and single-block mode are not active in the axis.

If the technology is used on an M7-FM, automatic and single-block modes must be deselected in all axes to enable the NC program memory to be cleared.

3.2 Delete NC Program

Data Block GMC_DB_APP1

DBWx	Main task number = **14**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **1**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	NC program number	DEC

Task Description

The "delete NC program" task deletes the specified NC program from the memory of the technology. The NC program which is currently running cannot be deleted.

3.3 Delete NC Block

Data Block GMC_DB_APP1

DBWx	Main task number = **15**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **2**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	NC program number	DEC
DBWz+2	NC block number	DEC

Task Description

The "delete NC block" task deletes the specified NC block and all following blocks with the specified NC block number from the specified NC program. You cannot delete an NC block in an NC program which is currently running.

3.4 Output NC Program Numbers

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **7**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **2**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **4**								DEC
DBWy+10	Z2 = First NC program number **x**								DEC
DBWy+12	Z3 = Number of NC program numbers **y**								DEC

Data Block GMC_DB_APP3

DBWz	Number of free NC blocks	DEC
DBWz+2	NC program number x	DEC
	⋮	
DBWz+n	NC program number x+y	DEC
DBWz+n+2	(0) End criterion	DEC

Task Description

The "output NC program numbers" task outputs all NC program numbers sorted from first to last. The number of NC blocks which are still free is also output. If you request more NC program numbers than actually exist, the end criterion 0 is entered after the last NC program number output.

If all NC program numbers are to be output, you must specify NC program number 1 as the first program number and 200 as the number of NC program numbers.

3.5 Output NC Block Numbers

Data Block GMC_DB_APP1

Address	Content	Format
DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **6**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

Address	Content	Format
DBWy	Number of DWs = **0**	DEC
DBBy+2	**0** **1** 0 0 0 0 0 0	BIN
DBBy+3	TI = **2**	DEC
DBWy+4	DB no.	DEC
DBWy+6	DBW no.	DEC
DBWy+8	Z1 = **6**	DEC
DBWy+10	Z2 = NC program number	DEC

Data Block GMC_DB_APP3

Address	Content	Format
DBWz	First NC block number	DEC
DBWz+2	Second NC block number	DEC
⋮	⋮	
DBWz+n	Last NC block number	DEC
DBWz+n+2	(0) End criterion	DEC

Task Description

The "output NC block numbers" task outputs all existing NC block numbers from the specified NC program number. The NC block numbers are sorted in ascending order. A value of 0 is entered as the end criterion after the last NC block number.

3.6 Output Software Version

Data Block GMC_DB_APP1

DBWx	Main task number = **-7**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **4**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	V(P)1	ZEI
DBWz+2	.0	ZEI
DBWz+4	__	ZEI
DBWz+6		ZEI

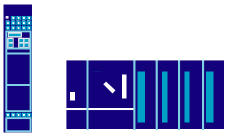
V Official status
P Pilot status
__ Space

Task Description

The "output software version " task outputs the current software version of the technology.

3.7 Software Version with Generation Date Output

NOTE



Function implemented from:
MASTERDRIVES MC: ≥ V1.4x
SIMATIC Motion Control: ≥ V1.00.48

Data Block GMC_DB_APP1

DBWx	Main task number = **-8**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **10**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	V(P)1	ZEI
DBWz+2	.0	ZEI
DBWz+4	__	ZEI
DBWz+6	__	ZEI
DBWz+8	__	ZEI
DBWz+10	19	ZEI
DBWz+12	99	ZEI
DBWz+14	.0	ZEI
DBWz+16	3.	ZEI
DBWz+18	01	ZEI

V Official status
P Pilot status
__ Space

Format: FW version, year.month.day
Example: V_1.41____2000.03.22

Task Description

The “software version with generation date output” task outputs the current software version of the technology and the generation date of the firmware version.

3.8 Output NC Program to OP

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **10**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **3**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **1**								DEC
DBWy+10	Z2 = NC program number								DEC
DBWy+12	Z3 = NC block number w								DEC
DBWy+14	Z4 = subsequent block code								DEC
DBWy+16	Z5 = number of NC blocks z								DEC
DBWy+18	Z6 = forwards = 1 / backwards = 2								DEC

Data Block GMC_DB_APP3

DBWz	NC block w
DBWz+52	NC block w+1
	⋮
DBWz+n	NC block w+z-1
DBWz+n+52	NC block w+z

Task Description

The "output NC program to OP" task outputs part of an NC program in a suitable format for the standard OP user interface.

The supplementary information in extensions Z2 to Z4 determines the start of the area to be output. Z5 determines the number of NC blocks to be output. Z6 can be used to specify whether the NC blocks are output from Z3 + Z5 (forwards) or from Z3 – Z5 (backwards). This facilitates the implementation of a simple forward/backward scroll function. If an NC block number greater than the highest NC block number in the NC program is requested, the last NC block number in the NC program is output.

NC Block Format for OP

The diagram below shows you the structure of an NC block in the format suitable for the standard OP user interface.

DBWz	Skip block "/”	ZEI
DBWz+2	Parameter set "*”	ZEI
DBWz+4	NC program number	DEC
DBWz+6	NC block number	DEC
DBWz+8	Following block identifier	DEC
DBWz+10	Axis allocation "X”	ZEI
DBWz+12	G1 function	DEC
DBWz+14	G2 function	DEC
DBWz+16	G3 function	DEC
DBWz+18	G4 function	DEC
DBWz+20	Reserved	DEC
DBWz+22	Reserved	DEC
DBDz+24	Position in [LU]	DEC
DBDz+28	Velocity in [10 LU/min]	DEC
DBWz+32	Velocity type "F” or "FN”	ZEI
DBWz+34	M1 function	DEC
DBWz+36	M2 function	DEC
DBWz+38	M3 function	DEC
DBWz+40	D function	DEC
DBWz+42	Reserved	DEC
DBWz+44	Subprogram number	DEC
DBWz+46	Loop count	DEC

For the display on the OP, you must identify NC block elements which are missing. The following format defines the allocation for "non-existent" values:

DBWz	Skip block " " (2 spaces)	ZEI
DBWz+2	Parameter set " " (2 spaces)	ZEI
DBWz+4	NC program number "0"	DEC
DBWz+6	NC block number "0"	DEC
DBWz+8	Following block identifier "0"	DEC
DBWz+10	Axis allocation "##"	ZEI
DBWz+12	G1 function "-32768"	DEC
DBWz+14	G2 function "-32768"	DEC
DBWz+16	G3 function "-32768"	DEC
DBWz+18	G4 function "-32768"	DEC
DBWz+20	Reserved	ZEI
DBWz+22	Reserved	ZEI
DBDz+24	Position in [LU] "-2147483648"	DEC
DBDz+28	Velocity in [10 LU/min] "-2147483648"	DEC
DBWz+32	Velocity type "##"	ZEI
DBWz+34	M1 function "-32768"	DEC
DBWz+36	M2 function "-32768"	DEC
DBWz+38	M3 function "-32768"	DEC
DBWz+40	D function "-32768"	DEC
DBWz+42	Reserved	DEC
DBWz+44	Subprogram number "-32768"	DEC
DBWz+46	Loop count "-1"	DEC

3.9 Input NC Block to OP

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **7**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**1**	**0**	0	0	0	0	0	0	BIN
DBBy+3	TI = **3**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **1**								DEC
DBWy+10	Z2 = program number								DEC
DBWy+12	Z3 = 0								DEC

Data Block GMC_DB_APP3

DBWz	First NC block

Task Description

The "input NC block to OP" task can be used to input an NC block in a suitable format for the standard OP user interface.

3.10 Input NC Program

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**	DEC
DBBy+2	**1** | **0** | 0 | 0 | 0 | 0 | 0 | 0	BIN
DBBy+3	TI = **3**	DEC
DBWy+4	DB no.	DEC
DBWy+6	DBW no.	DEC
DBWy+8	Z1 = **2**	DEC
DBWy+10	Z2 = NC program number	DEC
DBWy+12	Z3 = **0**	DEC
DBWy+14	Z4 = number of NC blocks	DEC

Data Block GMC_DB_APP3

DBWz bis DBWz+27	First NC block
DBWz+28 bis DBWz+55	Second NC block
	⋮
DBWz+n bis DBWz+n+27	Last NC block
DBDz+n+28	(0) End criterion

Task Description

The "input NC program" task can be used to input an NC program or parts of an NC program. You do not have to input all of the NC blocks with the same task. Further NC blocks belonging to the specified NC program number can be input in one or more subsequent tasks. The task only verifies the NC program number in the task data. If the same NC block number exists more than once, the NC blocks are overwritten. The end of an NC program is identified by the value 0 after the last NC block.

The "number of NC blocks" indicates the length of the task data to the technology. The "number of NC blocks" is the maximum number of blocks transferred. If fewer NC blocks are stored in the task data, the blocks are transferred up to the end criterion.

NC Block Format The graphic below shows you the structure of an NC block.

DBWz	NC program number								DEC
DBWz+2	NC block number								DEC
DBWz+4	Continuation identifier								DEC
DBWz+6	Reserved								
DBBz+8	**A**	**L**	0	**P**	**G4**	**G3**	**G2**	**G1**	BIN
DBBz+9	0	0	**FN**	**D**	**M3**	**M2**	**M1**	**F**	BIN
DBBz+10	Reserved								
DBBz+11	Axis allocation								DEC
DBBz+12	G1 function								DEC
DBBz+13	G2 function								DEC
DBBz+14	G3 function								DEC
DBBz+15	G4 function								DEC
DBDz+16	Position in [LU] or dwell time in [ms] or subprogram number								DEC
DBDz+20	Velocity in [10 LU/min] or subprogram loop count								DEC
DBBz+24	M1 function								DEC
DBBz+25	M2 function								DEC
DBBz+26	M3 function								DEC
DBBz+27	D function								DEC

Meaning of the identifier bits DBBz+8 and DBBz+9

Each of the identifier bits identify a valid value within the transferred or read NC block. This means that if NC blocks are input in this format, you must also set the appropriate identifier bit for the value entered.

Data word	Meaning
DBBz+8	Representation: DBBz+8 | A | L | 0 | P | G4 | G3 | G2 | G1 | BIN G1: G1 function G2: G2 function G3: G3 function G4: G4 function P: Position or dwell time L: Sub-program A: Skippable block
DBBz+9	Representation: DBBz+9 | 0 | 0 | FN | D | M3 | M2 | M1 | F | BIN F: Track speed M1: M1 function M2: M2 function M3: M3 function D: D function FN: Axis speed

Table 3-1 Meaning of the identifier bits

Further Information

The individual block elements are described centrally in the "Programming Guide" in chapter 1.2 "NC blocks".

3.11 Output NC Program

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **3**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	**Z1 = 2**								DEC
DBWy+10	Z2 = NC program number								DEC
DBWy+12	Z3 = NC block number								DEC
DBWy+14	Z4 = number of NC blocks								DEC

Data Block GMC_DB_APP3

DBWz	Up to the maximum number of NC blocks specified in "number of NC blocks"	
DBDz+n	(0) End criterion	DEC

Task Description

The "output NC program" task outputs NC blocks up to the maximum number of blocks, starting at the specified NC block number.

This task implements block-oriented output of NC blocks. If you specify an NC block number of 0, the output starts at the first NC block number. If you specify an NC block number of 10, for example, the output starts at block number 10 or, if this block does not exist, at the next higher NC block number. If no NC block with a higher number exists, the task is denied with error message "2047: NC block number does not exist". The end criterion is always entered after the last NC block.

3.12 Input NC Block

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**1**	**0**	0	0	0	0	0	0	BIN
DBBy+3	TI = **3**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **0**								DEC
DBWy+10	Z2 = NC program number								DEC
DBWy+12	Z3 = NC block number								DEC
DBWy+14	Z4 = number of NC blocks								DEC

Data Block GMC_DB_APP3

DBWz	One NC block with following blocks	
DBDz+n	(0) End criterion	DEC

Task Description

The "input NC block" task is used to input an NC block. The task can input 1 main block and 1 following block, 1 main block and several following blocks or 1 main block and all following blocks (up to 19). The end of the task data area must be identified by a 0 after the last block if the "number of NC blocks" is greater than the actual number of NC blocks in the task data. The number of NC blocks indicates the length of the data area to the technology. The maximum "number of NC blocks" is read in. If the end criterion is detected before the maximum number is reached, fewer NC blocks are transferred.

This task verifies the NC program number and the NC block number in the task data. If the NC program number and/or the NC block number in the task data differs from the data in the technological task header, the task is denied with an error message.

3.13 Output NC Block

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **3**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	**Z1 = 0**								DEC
DBWy+10	Z2 = NC program number								DEC
DBWy+12	Z3 = NC block number								DEC
DBWy+14	Z4 = number of NC blocks								DEC

Data Block GMC_DB_APP3

DBWz	One NC block with following blocks	
DBDz+n	(0) End criterion	DEC

Task Description

The "output NC block" task is used to output an NC block. The task can output 1 main block and 1 following block, 1 main block and several following blocks or 1 main block and all following blocks (up to 19). The end of the task data area is identified by a 0 after the last block. The "number of NC blocks" indicates the length of the data area to the technology. The maximum "number of NC blocks" is output. The end criterion is always entered after the last NC block.

4 MDI Data Tasks

Contents

In this chapter you will find a description of the tasks for the MDI data.

4.1 Input/Output MDI Block to OP

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **6**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**	DEC
DBBy+2	**E** **A** 0 0 0 0 0 0	BIN
DBBy+3	TI = **4**	DEC
DBWy+4	DB no.	DEC
DBWy+6	DBW no.	DEC
DBWy+8	Z1 = **2**	DEC
DBWy+10	Z2 = MDI number	DEC

Data Block GMC_DB_APP3

DBBz	G1 function	DEC
DBBz+1	G2-Funktion	DEC
DBBz+2	0	DEC
DBBz+3	0	DEC
DBDz+4	Position in [LU]	DEC
DBDz+8	Velocity in [10*LU/min]	DEC

Task Description

The "input/output MDI block to OP" task inputs or outputs a complete MDI block in a format prepared for the standard OP user interface to the selected MDI number. Each axis has 10 MDI block memories.

4.2 Input/Output MDI Block

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **6**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**E**	**A**	0	0	0	0	0	0	BIN
DBBy+3	TI = **4**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **1**								DEC
DBWy+10	Z2 = MDI number								DEC

Data Block GMC_DB_APP3

DBBz	0	0	0	**P**	0	0	**G2**	**G1**	BIN
DBBz+1	0	0	0	0	0	0	0	**F**	BIN
DBWz+2	Reserved								
DBBz+4	G1 function								DEC
DBBz+5	G2 function								DEC
DBBz+6	0								DEC
DBBz+7	0								DEC
DBDz+8	Position in [LU]								DEC
DBDz+12	Velocity in [10*LU/min]								DEC

Task Description

The "input/output MDI block" task inputs or outputs a complete MDI block to the selected MDI number. Each axis has 10 MDI block memories.

The reference bits in DBBz and DBBz+1 are used to validate the values in DBz+4 to DBDz+12, where:
P die Position
G1 1st G function
G2 2nd G function
F Velocity.

If the corresponding bits are not enabled, the associated values are transferred to the technology, but are not stored in the MDI block memory. You can therefore only transmit those items which can actually be modified.

You can program G90 (absolute dimensions) or G91 (incremental dimensions) as the G1 function (only G91 with roll feed version).

You can program an acceleration override from G30 to G39 as the G2 function. These overrides have the following effect:
G30 → 100 % of acceleration/deceleration specified in machine data
G31 → 10 % of acceleration/deceleration specified in machine data
.....
G39 → 90 % of acceleration/deceleration specified in machine data

4.3 Input/Output Roll Feed Loop Count MDI

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 16**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Loop count	DEC

Task Description

The "input/output roll feed loop count MDI" task can be used to input or output the loop count for MDI mode. You can enter values between 0 and 2 147 483 647 in the loop counter.

You can output the actual value of the MDI loop counter with the "output actual values" task.

The loop count in DBDz is retained until the loop counter reaches zero. The setpoint in DBDz is then also set to zero. To reinitialize the loop counter, you must enter a new value in DBDz.

If a new value is transferred in DBDz before the current MDI loop count reaches zero, the current MDI loop count is also reset.

5 Operating Data Tasks

Contents

In this chapter you will find a description of the tasks for the operating data.

5.1 Input/Output Setup Velocities

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 17**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer (EP) = **1** oder **2**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1** oder **2**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Velocity level 1 in [10*LU]	EP=1	DEC
DBDz+4	Velocity level 2 in [10*LU]	EP=2	DEC

Task Description

The "input/output setup velocities" task is used to input or output the velocities for setup mode.

If you enter a velocity greater than the maximum traversing velocity (MD23), the task generates an error message. The maximum traversing velocity is entered as the setup velocity on the technology.

5.2 Input/Output Block Search

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 18**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	NC program number level 0	DEC
DBWz+2	NC block number level 0	DEC
DBWz+4	NC program number level 1	DEC
DBWz+6	NC block number level 1	DEC
DBWz+8	Remaining loop count level 1	DEC
DBWz+10	NC program number level 2	DEC
DBWz+12	NC block number level 2	DEC
DBWz+14	Remaining loop count level 2	DEC

Task Description

The "input block search" task enters the point at which the NC program is to start running. When the NC program is then started, program execution begins at the specified entry point.

The "output block search" task can be used to output the current block search data subsequent to an "input block search" or "input automatic block search" task.

After the NC program has been started, the "output block search data" task returns zero.

Further Information

The detailed procedure of the block search is described in the Programming Guide in the chapter entitled "NC Program/Subprogram Execution".

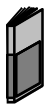
5.3 Input Automatic Block Search

Data Block GMC_DB_APP1

DBWx	Main task number = **19**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **0**	DEC
DBWx+10	**0**	DEC
DBWx+12	**0**	DEC

Task Description

The "input automatic block search" task sets the current program execution data to the last breakpoint. When the program is started again, execution begins at this breakpoint.



Further Information

The detailed procedure of the block search is described in the Programming Guide in the chapter entitled "NC Program/Subprogram Execution".

5.4 Input Teach-In

Data Block GMC_DB_APP1

DBWx	Main task number = **20**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **2**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	NC program number	DEC
DBWz+2	NC block number	DEC

Task Description

The "input teach-in" task enters the actual position value of the selected axis number in the specified NC block as a position setpoint. The NC block is specified by the NC program number and NC block number. The selected NC block must already be loaded in the technology memory.

The "teach-in" task can only be used in setup mode.

If illegal numbers or non-existent numbers are specified for the NC program numbers and/or NC block number, the task is aborted with an error message.

5.5 Input/Output Zero Offset

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 21**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer (EP) = **1** bis **6**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1** bis **6**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Zero offset G54 in [LU]	EP=1	DEC
DBDz+4	Zero offset G55 in [LU]	EP=2	DEC
DBDz+8	Zero offset G56 in [LU]	EP=3	DEC
DBDz+12	Zero offset G57 in [LU]	EP=4	DEC
DBDz+16	Zero offset G58 in [LU]	EP=5	DEC
DBDz+20	Zero offset G59 in [LU]	EP=6	DEC

Task Description

The "input/output zero offset" task inputs or outputs the zero offsets. You can use this task to input or output one, several or all zero offsets.

5.6 Input/Output Roll Feed Velocity Override

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 22**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Velocity in [10*LU]	DEC

Task Description

The "input roll feed velocity override" task can be used to override the velocity value for the roll feed. The "output roll feed velocity override" task can be used to output the velocity override value.

The velocity override is only active for one movement.

5.7 Input/Output Simulation

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 23**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **1**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	1 = on / 2 = off	DEC

Task Description

The "input simulation" task activates or deactivates the simulation for the selected axis. The "output simulation" task outputs the current simulation status.

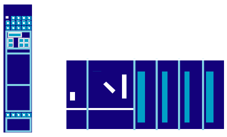
The "reset technology" control signal [RST] must be activated or the drive must be restarted (network OFF-ON) after the simulation is switched on or off, in order to activate or deactivate the simulation.

Further Information

Simulation and its effects and capabilities are described in the "Function Description" in the chapter on "Simulation".

5.8 Input / Output Write Data in EEPROM or RAM

NOTE



Function implemented from:

MASTERDRIVES MC: ≥ V1.4x
SIMATIC Motion Control: ≥ V1.0

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 37**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **1**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	1 = EEPROM / 2 = RAM	DEC

Task Description

With the “input write data in EEPROM or RAM" task you define that all following data are either written in the EEPROM memory or “only“ in the internal RAM memory (working memory)

With the “output write data in EEPROM or RAM" task the current status is output, i.e. whether the data are written in the EEPROM memory or “only“ in the RAM memory (working memory).

NOTE

After you switch the network on, the default setting is “write data to EEPROM”.
Data stored in RAM before the network was switched off is no longer available.

5.9 Input / Output Activate/Deactivate Encoder Changeover

NOTE

Function implemented from:

MASTERDRIVES MC: ≥ V1.41

SIMATIC Motion Control: --

Data Block GMC_DB_APP1

DBWx	Main task number = **40**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **1**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	1 = Activation of machine encoder 2 = Deactivation of machine encoder	DEC

Task Description

With the “input activate/deactivate encoder changeover“ task you can change over from a motor encoder to a machine encoder of an axis. An encoder changeover is only permissible for the “roll feed“ axis type (MD1=3).

Data Block GMC_DB_APP1

DBWx	Main task number = -**40**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **1**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

DBWz	1 = Machine encoder active 2 = Machine encoder inactive	DEC

Task Description

With the “output activate/deactivate encoder changeover“ task the current status of the encoder changeover is output.

6 GMC Status Data Output Tasks

Contents

In this section you will find a description of the tasks used to read out the GMC status data.

6.1 Output Actual Values

Data Block GMC_DB_APP1

DBWx	Main task number = **3**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **10**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **8**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = Identifier for GMC status data 1								DEC
DBWy+10	Z2 = Identifier for GMC status data 2								DEC
DBWy+12	Z3 = Identifier for GMC status data 3								DEC
DBWy+14	Z4 = Identifier for GMC status data 4								DEC
DBWy+16	Z5 = Identifier for GMC status data 5								DEC
DBWy+18	Z6 = Identifier for GMC status data 6								DEC

Data Block GMC_DB_APP3

DBDz	Value associated with identifier 1
DBDz+4	Value associated with identifier 2
DBDz+8	Value associated with identifier 3
DBDz+12	Value associated with identifier 4
DBDz+16	Value associated with identifier 5
DBDz+20	Value associated with identifier 6

Task Description

The "output actual values" task can output up to 6 actual values associated with the specified identifiers.

Identifiers for GMC Status Data

The table below contains a list of possible identifiers.

Ident.	MC	M7	Description
1	**V1.2**	**V1.0**	**Actual position value 1:** Actual position value 1 is the actual position value allowing for all possible measuring system shifts (zero offset, tool offset and the encoder alignment for absolute measuring systems). Example: If the axis approaches position 1000 mm with an active zero offset of 100 mm, actual position value 1 also indicates 1000 mm → reference to parameters Actual position value 1 is normally used for display. Representation: DBDz [Actual position value 1 in [LU]] DEC
2	**V1.2**	**V1.0**	**Position setpoint 1:** Position setpoint 1 represents the programmed target position. Example: If the axis approaches position 1000 mm with an active zero offset of 100 mm, position setpoint 1 also indicates 1000 mm → reference to parameters Position setpoint 1 is normally used for display. NOTE: In setup and reference point approach modes, the position setpoint (position control input) is indicated as position setpoint 1. In control mode, the actual position is indicated as position setpoint 1. Representation: DBDz [Position setpoint 1 in [LU]] DEC
3	**V1.2**	**V1.0**	**Following error:** Indicates the following error. The following error is the difference between the position setpoint and the actual position. Representation: DBDz [Following error in [LU]] DEC
4	**V1.2**	**V1.0**	**Actual position value 2:** Actual position value 2 is the actual position without measuring system shifts (exception: an encoder alignment for absolute encoders is included). Example: If the axis approaches position 1000 mm with an active zero offset of 100 mm, the actual position value indicates 2 900 mm. → reference to the mechanical system Representation: DBDz [Actual position value 2 in [LU]] DEC

Table 6-1 List of Identifiers for the GMC Status Data

Ident.	MC	M7	Description
5	**V1.2**	**V1.0**	**Position setpoint 2:** Position setpoint 2 represents the programmed target position without existing measuring system shifts. Example: If the axis approaches position 1000 mm with an active zero offset of 100 mm, the position setpoint indicates 2 900 mm → reference to mechanical system Note: In setup and reference point approach modes, the position setpoint (position control input) is indicated as position setpoint 2. In control mode, the actual position is indicated as position setpoint 2. Representation: DBDz [Position setpoint 2 in [LU]] DEC
6	**V1.2**	**V1.0**	**Traversing velocity:** Indicates the current traversing velocity allowing for the override. Representation: DBDz [Traversing velocity in [1000*LU/min]] DEC
7	**V1.2**	**V1.0**	**Servo gain factor:** Indicates the current servo gain factor. The servo gain factor is calculated from the traversing velocity / following error. Representation: DBDz [Servo gain factor in [1/min]] DEC
8	**V1.2**	**V1.0**	**Inprocess measured value:** When you use the functions "inprocess measurement", "set actual value on-the-fly" and "external block change", detection of the actual position is interrupt-driven, triggered by a digital input. The actual position is output as the inprocess measured value. Representation: DBDz [Inprocess measured value in [LU]] DEC
9	**V1.2**	**V1.0**	**Following error limit:** Indicates the current following error monitoring limit: At a standstill it indicates "following error – at standstill" (MD14); in motion it indicates "following error – in motion" (MD15). If you are using the IM178, the dynamic error calculated is displayed when the system is in motion. Representation: DBDz [Following error limit in [LU]] DEC

Table 6-1 List of Identifiers for the GMC Status Data (Continued)

<table>
<tr><th>Ident.</th><th>MC</th><th>M7</th><th>Description</th></tr>
<tr><td>10</td><td>V1.2</td><td>V1.0</td><td>Override:
Indicates the override currently selected.
Representation:
<table><tr><td>DBDz</td><td>Override in [%]</td><td>DEC</td></tr></table></td></tr>
<tr><td>11</td><td>V1.2</td><td>V1.0</td><td>Digital I/Os:
Indicates the state of the digital inputs and outputs.
Representation:
<table>
<tr><td>DBBz</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>BIN</td></tr>
<tr><td>DBB z+1</td><td>0</td><td>0</td><td>E6</td><td>E5</td><td>E4</td><td>E3</td><td>E2</td><td>E1</td><td>BIN</td></tr>
<tr><td>DBB z+2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>BIN</td></tr>
<tr><td>DBB z+3</td><td>0</td><td>0</td><td>A6</td><td>A5</td><td>A4</td><td>A3</td><td>A2</td><td>A1</td><td>BIN</td></tr>
</table></td></tr>
<tr><td>12</td><td>V1.2</td><td>V1.0</td><td>Following error:
Indicates the following error which triggered the "following error - at standstill" or "following error - in motion" signal.
Representation:
<table><tr><td>DBDz</td><td>Following error in [LU]</td><td>DEC</td></tr></table></td></tr>
<tr><td>13</td><td>V1.2</td><td>V1.0</td><td>Remaining distance:
Indicates the distance currently remaining. The remaining distance is defined as the difference between the programmed target position and the actual position value.
Representation:
<table><tr><td>DBDz</td><td>Remaining distance in [LU]</td><td>DEC</td></tr></table></td></tr>
<tr><td>14</td><td>V1.2</td><td>V1.0</td><td>Decoder error location:
Indicates the point in the NC program where the decoder detected an error.
Representation:
<table><tr><td>DBWz</td><td>NC program number</td><td>DEC</td></tr><tr><td>DBW z+2</td><td>NC block number</td><td>DEC</td></tr></table></td></tr>
<tr><td>15</td><td>V1.2</td><td>V1.0</td><td>Roll feed time ratio:
Indicates the ratio between the positioning time and the read-in enable time according to the following equation:
(Positioning time / read-in enable time) *1000
Representation:
<table><tr><td>DBDz</td><td>Roll feed time ratio</td><td>DEC</td></tr></table></td></tr>
</table>

Table 6-1 List of Identifiers for the GMC Status Data (Continued)

Ident.	MC	M7	Description
16	**V1.2**	**V1.0**	**Roll feed remaining loop count:** Indicates the current remaining loop count (number of positioning operations) if a loop counter has been set for MDI mode. Representation: DBDz [Roll feed remaining loop count] DEC
17	**V1.2**	**V1.0**	**Simulation status:** Status display of whether simulation is active or not. Representation: DBDz [1 = active 2 = not active] DEC
18	**V1.4**	**-**	**Number of free supports:** Displays the number of free cam disc supports on the MCT. Representation: DBDz [Number of free supports] DEC
19	**V1.41**	**-**	**Difference between actual value of encoder 1 and actual value of encoder 2:** Displays the difference between the actual value of encoder 1 and the actual value of encoder 2. Representation: DBDz [Actual value of encoder 1 minus actual value of encoder 2 in [LU]] DEC
20	**V1.41**	**-**	**Difference between actual value of encoder 1 and actual value of encoder 2 per positioning operation:** Displays the difference between actual value of encoder 1 and actual value of encoder 2 per positioning operation. Representation: DBDz [Actual value of encoder 1 minus actual value of encoder 2 per pos.oper. in [LU]] DEC
21	**V1.41**	**-**	**Actual value of encoder 2:** Displays the actual value of encoder 2. Representation: DBDz [Actual value of encoder 2 in [LU]] DEC
30	**-**	**V1.0**	**Raw actual value IM178:** Displays the read raw actual value of the position sensing module IM178 without actual value weighting factor Representation: DBDz [Raw actual value IM178] DEC

Table 6-1 List of Identifiers for the GMC Status Data (Continued)

Ident.	MC	M7	Description
31	-	V1.0	**Setpoint position error following error:** Displays the last setpoint position in event of error. Representation: DBDz [Setpoint position error following error in [LU]] DEC
32	-	V1.0	**Actual position error following error:** Displays the last actual position in event of error. Representation: DBDz [Actual position error following error in [LU]] DEC
33	-	V1.0	**DAC integer factor:** Displays the DAC integer factor. Representation: DBDz [DAC integer factor] DEC
34	-	V1.0	**SSI positional difference error:** Displays the number of increments between two measurements. Representation: DBDz [SSI positional difference error] DEC
35	-	V1.0	**SSI positional difference limit:** Displays the permissible number of increments between two measurements. Representation: DBDz [SSI positional difference limit] DEC
36	-	V1.0	**DAC output value:** Displays the DAC output value Representation: DBDz [DAC output value where 30440 = 10 Volt] DEC
37	-	V1.0	**Actual value IM178 with AVWF:** Displays the actual value of IM178 with actual value weighting factor. Representation: DBDz [Actual value IM178 with AVWF in [LU]] DEC

Table 6-1 List of Identifiers for the GMC Status Data (Continued)

6.2 Output Actual Value Data Collection 1 to OP

Data Block GMC_DB_APP1

Address	Content	Format
DBWx	Main task number = **3**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **5**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

Address	Content								Format
DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **7**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **1**								DEC

Data Block GMC_DB_APP3

Address	Content	Format
DBWz	Mode	DEC
DBWz+2	Override in [%]	DEC
DBWz+4	Program number level 0	DEC
DBWz+6	Program number level 1	DEC
DBWz+8	Program number level 2	DEC
DBWz+10	Block number level 0	DEC
DBWz+12	Block number level 1	DEC
DBWz+14	Block number level 2	DEC
DBWz+16	Remaining loop count level 1	DEC
DBWz+18	Remaining loop count level 2	DEC
DBWz+20	M1 function	DEC
DBWz+22	M2 function	DEC
DBWz+24	M3 function	DEC
DBWz+26	D number	DEC
DBWz+28 to DBWz+92	List of G functions max. 64 bytes (ASCII characters)	ZEI

Task Description The "output actual value data collection 1 to OP" task outputs the data shown above in a suitable format for the OP.

6.3 Output Actual Value Data Collection 2 to OP

Data Block GMC_DB_APP1

DBWx	Main task number = **3**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **5**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**	DEC
DBBy+2	**0** **1** 0 0 0 0 0 0	BIN
DBBy+3	TI = **7**	DEC
DBWy+4	DB no.	DEC
DBWy+6	DBW no.	DEC
DBWy+8	Z1 = **2**	DEC

Data Block GMC_DB_APP3

DBWz	Mode	DEC
DBWz+2	Override in [%]	DEC
DBDz+4	Setpoint [LU]	DEC
DBDz+8	Actual value [LU]	DEC
DBDz+12	Velocity [10*LU/min]	DEC
DBDz+16	Following error [LU]	DEC

Data Block GMC_DB_APP1

DBWx	Main task number = **3**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **5**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **7**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **3**								DEC

Data Block GMC_DB_APP3

DBWz	Mode	DEC
DBWz+2	Override in [%]	DEC
DBDz+4	Setpoint [LU]	DEC
DBDz+8	Actual value [LU]	DEC
DBDz+12	Velocity [10*LU/min]	DEC
DBDz+16	Following error [LU]	DEC
DBDz+20	Current dwell time [s]	DEC

Task Description

The "output actual value data collection 2 to OP" task outputs the data shown above with or without the current dwell time in a suitable format for the OP. The distinction is made by means of additional information 1.

6.4 Output Runtime Data

Data Block GMC_DB_APP1

DBWx	Main task number = **-24**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	NC program number level 0	DEC
DBWz+2	NC block number level 0	DEC
DBWz+4	NC program number level 1	DEC
DBWz+6	NC block number level 1	DEC
DBWz+8	Remaining loop count level 1	DEC
DBWz+10	NC program number level 2	DEC
DBWz+12	NC block number level 2	DEC
DBWz+14	Remaining loop count level 2	DEC

Task Description

The "output runtime data" task outputs the runtime data generated while an NC program is running.

7 Tool Offset Tasks

Contents

In this chapter you will find a description of the tasks for the tool offset.

7.1 Input/Output Tool Offset

Data Block GMC_DB_APP1

Address	Content	Format
DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **6**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

Address	Content	Format
DBWy	Number of DWs = **0**	DEC
DBBy+2	**E** | **A** | 0 | 0 | 0 | 0 | 0 | 0	BIN
DBBy+3	TI = **14**	DEC
DBWy+4	DB no.	DEC
DBWy+6	DBW no.	DEC
DBWy+8	Z1 = first tool offset number **x**	DEC
DBWy+10	Z2 = number of tool offset numbers **y**	DEC

Data Block GMC_DB_APP3

Address	Content	Format
DBDz	Tool length compensation x in [LU]	DEC
DBDz+4	Tool wear x in [LU]	DEC
DBDz+8	Wear x absolute = **0** / additive = **1**	DEC
DBDz+12	Tool length compensation x+1 in [LU]	DEC
DBDz+16	Tool wear x+1 in [LU]	DEC
DBDz+20	Wear x+1 absolute = **0** / additive = **1**	DEC
⋮	⋮	
DBDz+n	Tool length compensation x+y in [LU]	DEC
DBDz+n+4	Tool wear x+y in [LU]	DEC
DBDz+n+8	Wear x+y absolute = **0** / additive = **1**	DEC

Task Description

The "input/output tool offset" task can be used to input or output one, several or all tool offsets. You can choose between absolute or additive wear on input. The absolute wear is indicated on output.

8 GMC Technology Warning Tasks

Contents

In this chapter you will find a description of the tasks for the GMC technology warnings.

8.1 Acknowledge GMC Technology Warnings

Data Block GMC_DB_APP1

DBWx	Main task number = **6**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **0**	DEC
DBWx+10	**0**	DEC
DBWx+12	**0**	DEC

Task Description

The "acknowledge GMC technology warnings" task can be used to clear queued technology warnings. Warnings can also be cleared with the "acknowledge fault" [ACK_F] or "reset technology" [RST] control signal.

8.2 Output GMC Technology Warnings

Data Block GMC_DB_APP1

DBWx	Main task number = **-9**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Warning number 1 or 0	DEC
DBWy+2	Warning number 2 or 0	DEC
DBWy+4	Warning number 3 or 0	DEC
DBWy+6	Warning number 4 or 0	DEC
DBWy+8	Warning number 5 or 0	DEC
DBWy+10	Warning number 6 or 0	DEC
DBWy+12	Warning number 7 or 0	DEC
DBWy+14	Warning number 8 or 0	DEC

Task Description

The "output GMC technology warnings" task outputs up to eight GMC motion error messages. If fewer than eight error messages are available, the remaining digits are padded with 0. The fault [F] checkback signal indicates that a GMC motion error message is queued.

The "output GMC technology warnings" task outputs up to eight technology warnings. If fewer than eight warnings are available, the remaining digits are padded with 0. The [WARN] checkback signal indicates that a warning is queued.

9 NC Table Tasks

Contents

In this chapter you will find a description of the tasks for NC tables.

9.1 Delete NC Table and Output to OP

Data Block GMC_DB_APP1

DBWx	Main task number = **10**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **16**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **4**								DEC
DBWy+10	Z2 = NC table number								DEC
DBWy+12	Z3 = from interpolation point no. x								DEC
DBWy+14	Z4 = number of interpolation points y								DEC

Data Block GMC_DB_APP3

DBDz	NC table width [LU]	DEC
DBDz+4	Total number of interpolation points	DEC
DBDz+8	Interpolation point x	DEC
DBDz+12	Position of master axis [LU]	DEC
DBDz+16	Position of slave axis [LU]	DEC
DBDz+20	Interpolation point x+1	DEC
DBDz+24	Position of master axis [LU]	DEC
DBDz+28	Position of slave axis [LU]	DEC
	...	
DBDz+n	Interpolation point x+y	DEC
DBDz+n+4	Position of master axis [LU]	DEC
DBDz+n+8	Position of slave axis [LU]	DEC

Task Description

The "delete NC table and output to OP" task deletes the selected NC table (all interpolation points are initialized with 0) and outputs the specified number of interpolation points starting at the specified interpolation point. The NC table width and the total number of interpolation points are also output.

If "from interpolation point no. x" + "number of interpolation points y" is greater than the "total number of interpolation points", the interpolation point output starts at the "total number of interpolation points" – "number of interpolation points y".

If the "number of interpolation points y" is greater than the "total number of interpolation points", the interpolation points are output from point 1 up to the "total number of interpolation points".

If "0" is entered in "number of interpolation points y", the total number of interpolation points in the NC table is initialized again starting at "from interpolation point no. x". This allows you to modify your NC table quickly with respect to the number of interpolation points.

9.2 Accept NC Table

NOTE

MASTERDRIVES MC with F01:
From firmware version ≥ V1.4x, a maximum of 8 NC tables are available.

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 26**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1 to 10** for MCB **1 to 4** for MCT	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1 to 10** for MCB **1 to 4** for MCT	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2 Assignment for Technology on M7-FM

DBDz	NC table number	EP=1	DEC
DBDz+4	NC table function	EP=2	BIN
DBDz+8	DB number for NC table 1	EP=3	DEC
DBDz+12	DB number for NC table 2	EP=4	DEC
DBDz+16	DB number for NC table 3	EP=5	DEC
DBDz+20	DB number for NC table 4	EP=6	DEC
DBDz+24	DB number for NC table 5	EP=7	DEC
DBDz+28	DB number for NC table 6	EP=8	DEC
DBDz+32	DB number for NC table 7	EP=9	DEC
DBDz+36	DB number for NC table 8	EP=10	DEC

GMC_DB_APP2 Assignment for Technology on MASTERDRIVES MCT ≤ V1.4x

DBDz	NC table number	EP=1	DEC
DBDz+4	NC table function	EP=2	BIN
DBDz+8	A	EP=3	DEC
DBDz+12	B	EP=4	DEC

A = 1 and B = 0: One NC table with up to 200 interpolation points

A = 1 and B = 1: Two NC tables with up to 100 interpolation points

GMC_DB_APP2 Assignment for Technology on MASTERDRIVES MCT ≥ V1.4x

DBDz	NC table number	EP=1	DEC
DBDz+4	NC table function	EP=2	BIN
DBDz+8	NC table 1 present	EP=3	DEC
DBDz+12	NC table 2 present	EP=4	DEC
DBDz+16	NC table 3 present	EP=5	DEC
DBDz+20	NC table 4 present	EP=6	DEC
DBDz+24	NC table 5 present	EP=7	DEC
DBDz+28	NC table 6 present	EP=8	DEC
DBDz+32	NC table 7 present	EP=9	DEC
DBDz+36	NC table 8 present	EP=10	DEC

Table Function

The table below describes the bits of the NC table function.

<table>
<tr><th>Data Word</th><th>Meaning</th></tr>
<tr><td>DBDz+4</td><td>Representation:
<table>
<tr><td>DBBz+4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>BIN</td></tr>
<tr><td>DBBz+5</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>BIN</td></tr>
<tr><td>DBBz+6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>BIN</td></tr>
<tr><td>DBBz+7</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Z</td><td>Y</td><td>X</td><td>BIN</td></tr>
</table>
X=1: Reset NC table
Y=1: Accept NC table
Z=1: Accept all NC tables</td></tr>
</table>

Table 9-1 Meaning of the NC Table Function Bits

NC table present

When configuring the table, the following options are available:

Data word	Tables / supports				
	1 / 400	2 / 200	4 / 100	8 / 50	Variable max. 8 / max. 400
DBDz+8	≥ 1	≥ 1	≥ 1	≥ 1	0
DBDz+12	0	≥ 1	≥ 1	≥ 1	0
DBDz+16	0	0	≥ 1	≥ 1	0
DBDz+20	0	0	≥ 1	≥ 1	0
DBDz+24	0	0	0	≥ 1	0
DBDz+28	0	0	0	≥ 1	0
DBDz+32	0	0	0	≥ 1	0
DBDz+36	0	0	0	≥ 1	0

Task Description

The "accept NC table" task can be used to reset an NC table, to accept an NC table, or to accept all assigned NC tables.

Resetting an NC table means that an NC table is no longer assigned to the NC table number.

Accepting an NC table means verifying the contents of an NC table assigned to an M7 data block and, if no error is detected, assigning the specified table number to the NC table. If errors occur during verification, they are indicated as error number 1 and error number 2 in the task checkback message. The "output NC table status" task can be used to indicate the current status of the NC table. NC table numbers (1-8) which are assigned to data block number 0 are not accepted.

Only one bit of the NC table function can be enabled at a time.

9.3 Delete NC Table Interpolation Point and Output to OP

Data Block GMC_DB_APP1

DBWx	Main task number = **10**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**	DEC
DBBy+2	**0** **1** 0 0 0 0 0 0	BIN
DBBy+3	TI = **16**	DEC
DBWy+4	DB no.	DEC
DBWy+6	DBW no.	DEC
DBWy+8	Z1 = **3**	DEC
DBWy+10	Z2 = NC table number	DEC
DBWy+12	Z3 = from interpolation point no. x	DEC
DBWy+14	Z4 = number of interpolation points y	DEC

Data Block GMC_DB_APP3

DBDz	NC table width [LU]	DEC
DBDz+4	Total number of interpolation points	DEC
DBDz+8	Interpolation point x	DEC
DBDz+12	Position of master axis [LU]	DEC
DBDz+16	Position of slave axis [LU]	DEC
DBDz+20	Interpolation point x+1	DEC
DBDz+24	Position of master axis [LU]	DEC
DBDz+28	Position of slave axis [LU]	DEC
	⋮	
DBDz+n	Interpolation point x+y	DEC
DBDz+n+4	Position of master axis [LU]	DEC
DBDz+n+8	Position of slave axis [LU]	DEC

Task Description

The "delete NC table interpolation point and output to OP" task deletes the selected interpolation point and shortens the NC table by this point. The specified number of interpolation points is subsequently output starting at the specified interpolation point number. The NC table width and the total number of interpolation points are also output.

If "from interpolation point no. x" + "number of interpolation points y" is greater than the "total number of interpolation points", the interpolation point output starts at the "total number of interpolation points"– "number of interpolation points y".

If the "number of interpolation points y" is greater than the "total number of interpolation points", the interpolation points are output from point 1 up to the "total number of interpolation points".

9.4 Insert NC Table Interpolation Point and Output to OP

Data Block GMC_DB_APP1

DBWx	Main task number = **10**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **16**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **2**								DEC
DBWy+10	Z2 = NC table number								DEC
DBWy+12	Z3 = From interpolation point no. x								DEC
DBWy+14	Z4 = Number of interpolation points y								DEC

Data Block GMC_DB_APP3

DBDz	NC table width [µm]	DEC
DBDz+4	Total number of interpolation points	DEC
DBDz+8	Interpolation point x	DEC
DBDz+12	Position of master axis [LU]	DEC
DBDz+16	Position of slave axis [LU]	DEC
DBDz+20	Interpolation point x+1	DEC
DBDz+24	Position of master axis [LU]	DEC
DBDz+28	Position of slave axis [LU]	DEC
⋮	⋮	
DBDz+n	Interpolation point x+y	DEC
DBDz+n+4	Position of master axis [LU]	DEC
DBDz+n+8	Position of slave axis [LU]	DEC

Task Description

The "insert NC table interpolation point and output to OP" task inserts the selected interpolation point and extends the NC table by this point. The specified number of interpolation points is subsequently output starting at the specified interpolation point number. The NC table width and the total number of interpolation points are also output.

If "from interpolation point no. x" + "number of interpolation points y" is greater than the "total number of interpolation points", the interpolation point output starts at the "total number of interpolation points"– "number of interpolation points y".

If the "number of interpolation points y" is greater than the "total number of interpolation points", the interpolation points are output from point 1 up to the "total number of interpolation points".

9.5 Input NC Table from OP

Data Block GMC_DB_APP1

DBWx	Main task number = **10**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**	DEC
DBBy+2	**1** | **0** | 0 | 0 | 0 | 0 | 0 | 0	BIN
DBBy+3	TI = **17**	DEC
DBWy+4	DB no.	DEC
DBWy+6	DBW no.	DEC
DBWy+8	Z1 = **1**	DEC
DBWy+10	Z2 = NC table number	DEC
DBWy+12	Z3 = From interpolation point no. x	DEC
DBWy+14	Z4 = Number of interpolation points y	DEC

Data Block GMC_DB_APP3

DBDz	NC table width [LU]	DEC
DBDz+4	Reserved	
DBDz+8	Interpolation point x	DEC
DBDz+12	Position of master axis [LU]	DEC
DBDz+16	Position of slave axis [LU]	DEC
DBDz+20	Interpolation point x+1	DEC
DBDz+24	Position of master axis [LU]	DEC
DBDz+28	Position of slave axis [LU]	DEC
⋮	⋮	
DBDz+n	Interpolation point x+y	DEC
DBDz+n+4	Position of master axis [LU]	DEC
DBDz+n+8	Position of slave axis [LU]	DEC

Task Description

The "input NC table from OP" task inputs the "number of interpolation points" starting at the selected interpolation point number. The table width is also input.

NC tables can only be edited with this task. The number of interpolation points is determined by the amount of interpolation points edited.

9.6 Output NC Table to OP

Data Block GMC_DB_APP1

DBWx	Main task number = **10**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **9**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**	DEC
DBBy+2	**0** **1** 0 0 0 0 0 0	BIN
DBBy+3	TI = **17**	DEC
DBWy+4	DB no.	DEC
DBWy+6	DBW no.	DEC
DBWy+8	Z1 = **1**	DEC
DBWy+10	Z2 = NC table number	DEC
DBWy+12	Z3 = From interpolation point no. x	DEC
DBWy+14	Z4 = Number of interpolation points y	DEC
DBWy+16	Z5 = forwards = **1** / backwards = **2**	DEC

Data Block GMC_DB_APP3

DBDz	NC table width [LU]	DEC
DBDz+4	Total number of interpolation points	DEC
DBDz+8	Interpolation point x	DEC
DBDz+12	Position of master axis [LU]	DEC
DBDz+16	Position of slave axis [LU]	DEC
DBDz+20	Interpolation point x+1	DEC
DBDz+24	Position of master axis [LU]	DEC
DBDz+28	Position of slave axis [LU]	DEC
DBDz+n	Interpolation point x+y	DEC
DBDz+n+4	Position of master axis [LU]	DEC
DBDz+n+8	Position of slave axis [LU]	DEC

Task Description

The "output NC table to OP" task outputs the number of interpolation points starting at the selected interpolation point number. The NC table width and the total number of interpolation points are also output.

If, with "forwards", the result of "from interpolation point no. x" + "number of interpolation points y" is greater than the "total number of interpolation points", the interpolation point output starts at "total number of interpolation points" – "number of interpolation points y".

If, with "backwards", the result of "from interpolation point no. x" - "number of interpolation points y" is less than 1, the NC table is output starting at interpolation point 1.

If the "number of interpolation points y" is greater than the "total number of interpolation points", the interpolation points are output from point 1 to the "total number of interpolation points".

9.7 Input/Output NC Table

Data Block GMC_DB_APP1

DBWx	Main task number = **10**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**	DEC
DBBy+2	**E** | **A** | 0 | 0 | 0 | 0 | 0 | 0	BIN
DBBy+3	TI = **17**	DEC
DBWy+4	DB no.	DEC
DBWy+6	DBW no.	DEC
DBWy+8	Z1 = **2**	DEC
DBWy+10	Z2 = NC table number	DEC
DBWy+12	Z3 = From interpolation point no. x	DEC
DBWy+14	Z4 = Number of interpolation points y	DEC

Data Block GMC_DB_APP3

DBDz	NC table width [LU]	DEC
DBDz+4	Total number of interpolation points (on output only)	DEC
DBDz+8	Position of master axis Interpolation point x [LU]	DEC
DBDz+12	Position of slave axis Interpolation point x [LU]	DEC
DBDz+16	Position of master axis Interpolation point x+1 [LU]	DEC
DBDz+20	Position of slave axis Interpolation point x+1 [LU]	DEC
⋮	⋮	
DBDz+n	Position of master axis Interpolation point x+y [LU]	DEC
DBDz+n+4	Position of slave axis Interpolation point x+y [LU]	DEC

Task Description

The "input/output NC table" task inputs or outputs the number of interpolation points starting at the selected interpolation point number. The "NC table width" and the "total number of interpolation points" are also output.

If "from interpolation point no. x" + "number of interpolation points y" is greater than the total number of interpolation points, the interpolation points are output from "from interpolation point no. x" to the "total number of interpolation points".

The "input NC table" task is used to edit part or all of the NC table. The NC table width is also input.

If "from interpolation point no. x" + "number of interpolation points y" is less than or equal to the total number of interpolation points, the corresponding number of interpolation points is overwritten.

If "from interpolation point no. x" + "number of interpolation points y" is greater than the total number of interpolation points, existing interpolation points are overwritten, and the extra interpolation points are appended to the NC table. The total number of interpolation points is increased automatically by the technology.

If "from interpolation point no. x" is greater than the total number of interpolation points, the task is denied with an error message, since the NC table can only be extended contiguously.

9.8 Output NC Table Status

Data Block GMC_DB_APP1

DBWx	Main task number = **-29**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer (EP) = **1 to 8**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1 to 8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Status of table 1	EP=1
DBDz+4	Status of table 2	EP=2
DBDz+8	Status of table 3	EP=3
DBDz+12	Status of table 4	EP=4
DBDz+16	Status of table 5	EP=5
DBDz+20	Status of table 6	EP=6
DBDz+24	Status of table 7	EP=7
DBDz+28	Status of table 8	EP=8

Format of the Status Word

The table below describes the bits of the status word.

Data Word	Meaning
DBBz+n	Representation: DBBz+n | F | S | 0 | 0 | 0 | 0 | U | R | BIN Status bits: R=1: Reset NC table running U=1: Accept NC table running S= 1: Group error F= 1: Accept task finished and no errors
DBBz+n+1	Representation: DBBz+n+1 | Error code | DEC Error code: 0: No errors 1: No. of points 0 or greater than max. no. of interpolation points 2: Positional value of master axis greater than NC table width 3: Positional values of master axis do not increase 4: Data block does not exist 5: Data block too short
DBWz+n+2	Representation: DBWz+n+2 | Total number of interpolation points or last correct interpolation point number | DEC Number of interpolation points if the NC table was accepted without errors. Last correct interpolation point number if an error occurred during the accept task. The next interpolation point is the point with the error → see error code.

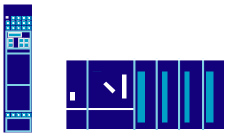
Table 9-2 Meaning of the NC Table Status Word

Task Description

The "output NC table status" task is used exclusively to indicate the current status of the NC table. The NC table status is generated by the "accept NC table" task.

9.9 Input/Output NC Table Parameters

NOTE



Function implemented from:		Elemet pointer EP:
MASTERDRIVES MC:	≥ V1.51	EP 7 assigned
SIMATIC Motion Control:	≥ V1.00.48	EP 7 assigned

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 12**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer (EP) = **1** to **9**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1** to **9**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	NC table mode	EP=1	BIN
DBDz+4	Master axis scale denominator	EP=2	DEC
DBDz+8	Master axis scale numerator	EP=3	DEC
DBDz+12	Slave axis scale denominator	EP=4	DEC
DBDz+16	Slave axis scale numerator	EP=5	DEC
DBDz+20	Set value master axis (0 to NC table width)	EP=6	DEC
DBDz+24	Window table actual value based master axis in LU	EP=7	DEC
DBDz+28	Reserved	EP=8	
DBDz+32	Reserved	EP=9	

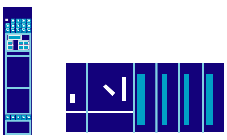
NC Table Mode

The table below describes the NC table mode.

<table>
<tr><th>Data Word</th><th>Meaning</th></tr>
<tr><td>DBDz</td><td>Representation:
<table>
<tr><td>DBBz</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>En F</td><td>En E</td><td>BIN</td></tr>
<tr><td>DBBz+1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>E</td><td>BIN</td></tr>
<tr><td>DBBz+2</td><td>0</td><td>0</td><td>0</td><td>D</td><td>0</td><td>0</td><td>0</td><td>C</td><td>BIN</td></tr>
<tr><td>DBBz+3</td><td>0</td><td>0</td><td>0</td><td>B</td><td>0</td><td>0</td><td>F</td><td>A</td><td>BIN</td></tr>
</table>
A=0: Interpolation values of slave axis are absolute
A=1: Interpolation values of slave axis are relative
B=0: Continuous output (automatic return jump to start of NC table)
B=1: Stop at end of NC table
C=0: Master axis scaling off
C=1: Master axis scaling on
D=0: Slave axis scaling off
D=1: Slave axis scaling on
E=0: Changes to scaling of slave axis have immediate effect
E=1: Changes to scaling of slave axis are only adopted on table restart (x=0)
F=0: Absolute table change
F=1: Relative table change
EnE=0: No takeover of mode E
EnE=1: Takeover of mode E
EnF=0: No takeover of mode F
EnF=1: Takeover of mode F</td></tr>
</table>

Table 9-3 Meaning of the NC Table Mode Bits

NOTE



Function implemented from:		NC table mode:
MASTERDRIVES MC:	≥ V1.4x	Extension E / EnE
	≥ V1.4x	Extension F / EnF
SIMATIC Motion Control:	≥ V1.00.48	Extension F / EnF

Task Description

The "input/output NC table parameters" task can be used to input or output one, several or all NC table parameters.

If a value greater than the NC table width is specified for "set value master axis", the set value is generated by modulus calculation within the NC table width.

NOTE

The scaling of the slave axis (y axis of the table) is absolute, which means that changes to the scale cause the setpoint of the slave axis to jump if the current y value of the table is not equal to zero at the point at which the change is made.

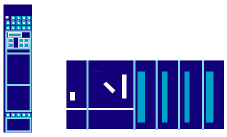
10 Synchronization Parameter Tasks

Contents

In this chapter you will find a description of the tasks for the synchronization parameters.

10.1 Input/Output Synchronization Parameters

NOTE



Function implemented from:	Element pointer EP:	
MASTERDRIVES MC:	≥ V1.2x	1 to 18
	≥ V1.4x	19
	≥ V1.5x	20 to 22
SIMATIC Motion Control:	≥ V1.0	1 to 18
	≥ V1.00.48	19 to 20

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 11**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1** to **22**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1** to **22**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Gear factor denominator	EP=1	DEC
DBDz+4	Gear factor numerator	EP=2	DEC
DBDz+8	Start/stop cycle: on/off length in [LU]	EP=3	DEC
DBDz+12	Start/stop cycle: on/off len. for ramp in [LU]	EP=4	DEC
DBDz+16	Start/stop cycle: on/off coupling pos. in [LU]	EP=5	DEC
DBDz+20	Start/stop cycle: on/off coupling position offset in [LU]	EP=6	DEC
DBDz+24	Position correction: position setpoint in [LU]	EP=7	DEC
DBDz+28	Position correction: maximum correction/controller cycle in [LU/cycle]	EP=8	DEC
DBDz+32	Position correction: Pressure mark not on separate axis (**0**) or pressure mark on separate axis (**1**)	EP=9	DEC
DBDz+36	Position correction: correction to fixed (**0**) or variable (**1**) position	EP=10	DEC
DBDz+40	Position correction: none (0) or at registration mark (1)	EP=11	DEC
DBDz+44	Master axis: linear/rotary axis in [LU] (only for MASTERDRIVES MC)	EP=12	DEC

Continued on next page.

Continued from previous page.

DBDz+48	Master axis: (axis number selection **1-n**) (only for technology on M7-FM)	EP=13	DEC
DBDz+52	Master axis: master val. source: **0**=act. val. control **1**= setpoint ctrl., **2**= virtual master	EP=14	DEC
DBDz+56	Virtual master: velocity setpoint in [10 * LU/min]	EP=15	DEC
DBDz+60	Virtual master: acceleration in [1000 * LU/s²]	EP=16	DEC
DBDz+64	Virtual master: set position in [LU]	EP=17	DEC
DBDz+68	Virtual master: linear/rotary axis in [LU]	EP=18	DEC
DBDz+72	Position correction: maximum correction speed in [10 * LU/min]	EP=19	DEC
DBDz+76	Position correction: acceleration in [10 * LU/s²]	EP=20	DEC
DBDz+80	Rated velocity master [10 * LU/min] only for MASTERDRIVES MC	EP=21	DEC
DBDz+84	Rated velocity slave [10 * LU/min] only for MASTERDRIVES MC	EP=22	DEC

Task Description

The "input/output synchronization parameters" task inputs or outputs the selected synchronization parameters.

Rated velocity master:
If the rated velocity of the master is not equal to zero the velocity input [%] is used and catch-up mode can be selected in the synchronization mode.

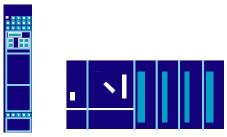
Rated velocity slave:
The “rated velocity slave” [10*LU/min] parameter is an alternative to the machine data MD23 [1000*LU/min]. Only use this parameter if you are exclusively using the synchronization mode. The use of the parameter lies within your responsibility!

Further Information

The meaning of the individual parameters is described in the "Function Description" in the chapter entitled "Synchronization Functions".

10.2 Current Synchronization Values and Diagnostics Output

NOTE



Function implemented from:		Element pointer EP:
MASTERDRIVES MC:	≥ V1.2x	1 to 14
	≥ V1.3x	15 to 18
	≥ V1.4x	19 to 22
SIMATIC Motion Control:	≥ V1.0	1 to 14
	≥ V1.00.48	15 to 18

Data Block GMC_DB_APP1

DBWx	Main task number = **-30**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1** to **22**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1** to **22**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Slave axis status	EP=1	BIN
DBDz+4	Slave axis actual position in [LU]	EP=2	DEC
DBDz+8	Slave axis velocity in [10 * LU]	EP=3	DEC
DBDz+12	Slave axis following error in [LU]	EP=4	DEC
DBDz+16	Internal table pos. for master axis in [LU]	EP=5	DEC
DBDz+20	Output table position for slave axis in [LU]	EP=6	DEC
DBDz+24	Number of active table	EP=7	DEC
DBDz+28	Virtual master status	EP=8	BIN
DBDz+32	Master value of axis in [LU]	EP=9	DEC
DBDz+36	Setpoint of axis in [LU]	EP=10	DEC
DBDz+40	Actual position of last registration mark in [LU]	EP=11	DEC
DBDz+44	0: Synchronization not active 1: Synchronization active	EP=12	DEC

Continued on next page.

Continued from previous page.

DBDz+48	Display selected cycle	EP=13	BIN
DBDz+52	Display selected function	EP=14	BIN
DBDz+56	Correction value in [LU] (position correction-difference value)	EP=15	DEC
DBDz+60	Current travel in [LU]	EP=16	DEC
DBDz+64	Remaining travel in [LU]	EP=17	DEC
DBDz+68	Current travel value in [LU] (= current position value - travel)	EP=18	DEC
DBDz+72	Master value correction status	EP=19	BIN
DBDz+76	Master value correction - Master value	EP=20	DEC
DBDz+80	Real master – position setpoint in [LU]	EP=21	DEC
DBDz+84	Real master- dead time compensation in [LU]	EP=22	DEC

Slave Axis Status

The table below describes the bits of the slave axis status.

Bit	MC	M7	Meaning
0	**V1.51**	**V1.00.48**	[B829] Print mark outside window 2
1	**V1.2**	**V1.0**	[B800] Correction of external synchronization running
2	**V1.4**	**V1.0**	[B803] Start / stop cycle active (MCT ≥ V1.4)
3	**V1.2**	**V1.0**	[B832] Start cycle active
4	**V1.2**	**V1.0**	[B833] Stop cycle active
5	**V1.4**	**V1.0**	[B831] Ramp inactive, start / stop cycle in constant motion
6	**V1.2**	**V1.0**	Internal NC table position < 0
7	**V1.2**	**V1.0**	Internal NC table position > NC table width
8	**V1.4**	**V1.0**	[B810] Adjustment active
9	**V1.4**	**V1.0**	[B811] Synchronous with master value
10	**V1.4**	**V1.0**	[B808] Axis referenced
11	**V1.4**	**V1.0**	[B809] Axis not referenced
12	**V1.4**	**V1.0**	[B820] Catch-up finished
13	**V1.4**	**V1.0**	[B821] Catch-up positioned
14	**V1.4**	**V1.0**	[B822] Catch-up velocity setpoint reached
15	**--**	**--**	Reserved
16	**V1.5**	**V1.00.48**	[B836] Synchronization in window 1

Table 10-1 Meaning of Slave Axis Status Bits

Bit	MC	M7	Meaning
17	**V1.5**	**V1.00.48**	[B837] Synchronization in window 2
18	**V1.5**	**V1.00.48**	[B812] Reversing lockout memory active
19	**V1.5**	**V1.00.48**	[B813] Reversing lockout currently not active
20 to 31	--	--	Reserved

Table 10-2 Meaning of Slave Axis Status Bits, continued

Virtual Master Status

The table below describes the bits of the virtual master status.

Bit	Meaning
0	Virtual master stationary
1	Acceleration active
2	Velocity setpoint reached
3	Deceleration active
4	Virtual master running
5 to 31	Reserved

Table 10-3 Meaning of the Virtual Master Status Bits

Display Selected Cycle

The table below describes the bits of "display selected cycle" (corresponds to the state of the [OPERATION] control signal).

Bit	Meaning
0	Continuous cycle
1	Start cycle
2	Stop cycle
3	Catch-up (MCT $\geq$ V1.4)
4 to 31	Reserved

Table 10-4 Meaning of the Bits of Display Selected Cycle

Display Selected Function

The table below describes the bits of the selected function (corresponds to the state of the [FUNCTION] control signal).

Bit	Meaning
0	1:1
1	Gearbox
2	Table
3 to 31	Reserved

Table 10-5 Meaning of the Bits of Display Selected Function

Master Value Correction Status

The following table describes the status bits for master value correction.

Bit	Meaning
0	Master value correction active
1	Trigger master value correction
2	Master value switch: 0 = Source master value 1 1 = Source master value 2
3	Enable direction change
4 to 31	Reserved

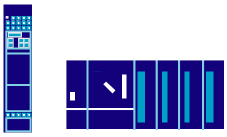
Table 10-6 Meaning of Status Bits for Master Value Correction

Task Description

The "Output actual values and diagnostic data" task can be used to output one, several or all actual values and/or diagnostic data.

10.3 Synchronization Parameters for Offset Angle Setting

NOTE



Function implemented from:		Element pointer EP:
MASTERDRIVES MC:	≥ V1.3x	1 to 12
	> V1.5	13 to 15
SIMATIC Motion Control:	≥ V1.00.48	1 to 19

Data Block
GMC_DB_APP1

DBWx	Main task number = **+/- 31**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer (EP) = **1** to **12**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1** to **12**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block
GMC_DB_APP2

DBDz	Synchronization mode	EP=1	DEC
DBDz+4	Offset angle correction type	EP=2	DEC
DBDz+8	Reserve	EP=3	DEC
DBDz+12	Absolute offset angle in [LU]	EP=4	DEC
DBDz+16	Relative offset angle in [LU]	EP=5	DEC
DBDz+20	Jog adjustment speed in [10 LU/min]	EP=6	DEC
DBDz+24	Reserve	EP=7	DEC
DBDz+28	Offset angle acceleration in jog mode in 10 LU/s²]	EP=8	BIN
DBDz+32	Offset angle delay in jog mode in [10 LU/s²]	EP=9	DEC
DBDz+36	Adjustment speed compensation movement in [10 LU/min]	EP=10	DEC
DBDz+40	Reserve	EP=11	DEC
DBDz+44	Acceleration of compensation movement in [10 LU/s²]	EP=12	DEC
DBDz+48	Relative offset angle setting mode	EP=13	DEC

Continued on next page.

Continued from previous page.

DBDz+52	Synchronization outer window in [LU]	EP=14	DEC
DBDz+56	Synchronization inner window in [LU]	EP=15	DEC
DBDz+60	Reserve	EP=16	DEC
DBDz+64	Offset angle setting value in [LU] (**only SIMATIC Motion Control**)	EP=17	DEC
DBDz+68	Proc. adjustment speed adaptation in jog mode in [%] (**only SIMATIC Motion Control**)	EP=18	DEC
DBDz+72	Proc. adjustment speed adaptation Compensation movement in [%] (**only SIMATIC Motion Control**)	EP=19	DEC

Synchronization Mode

The following table describes the input options for selecting the correction direction in the “Synchronization mode” data word.

Input	MC	M7	Meaning
0	**V1.3**	**V1.00.48**	shortest path
1	**V1.3**	**V1.00.48**	positive direction
2	**V1.3**	**V1.00.48**	negative direction
3	**V1.5**	**V1.00.48**	positive direction with window
4	**V1.5**	**V1.00.48**	Negative direction with window

Table 10-7 Meaning of Synchronization Mode

Offset Angle Correction Type

The following table describes the input option for selecting the correction direction in the “Offset angle correction” data word.

Input	Meaning
0	shortest path
1	As preset

Table 10-8 Meaning of Offset Angle Correction Type

Relative Offset Angle Setting Mode

The following table describes the input options for selecting the mode in the “Relative offset angle setting mode” data word.

Input	Meaning
0	Additive
1	Delete residual offset

Table 10-9 Meaning of Relative Offset Angle Setting Mode

Task Description

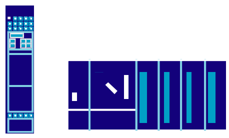
The task "Input / output synchronization parameters for offset angle adjustment" inputs or outputs the appropriate synchronization parameters.

Further Information

The meaning of the individual parameters is documented in the Function Description in the chapter on "Synchronization functions".

10.4 Catch-Up Synchronization Parameters

NOTE



Function implemented from:	Element pointer EP:	
MASTERDRIVES MC:	≥ V1.3x	1 to 6
	≥ V1.4x	7 to 9
SIMATIC Motion Control:	≥ V1.00.48	1 to 9

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 32**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer (EP) = **1** to **9**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1** to **9**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Set speed for catch-up in [10 LU/min]	EP=1	DEC
DBDz+4	Standstill position in [LU]	EP=2	DEC
DBDz+8	Deceleration delay in [1000 LU/s²]	EP=3	DEC
DBDz+12	Rounding time constant for deceleration in [ms]	EP=4	DEC
DBDz+16	Acceleration in [1000 LU/s²]	EP=5	DEC
DBDz+20	Rounding time constant for acceleration in [ms]	EP=6	DEC
DBDz+24	Position deceleration delay in [1000 LU/s²]	EP=7	DEC
DBDz+28	Position acceleration delay in [1000 LU/s²]	EP=8	DEC
DBDz+32	Rounding mode	EP=9	DEC

Rounding Mode

The following table describes the input options for the rounding mode.

Input	Meaning
0	Rounding only effective on sudden reduction in initial value during the acceleration process
1	Rounding always effective

Table 10-10 Meaning of Rounding Mode

Task Description

The “Input / output catch-up synchronization parameters" task inputs or outputs the appropriate synchronization parameters.

Further Information

The meaning of the individual parameters is documented in the Function Description in the chapter on “Synchronization functions”.

10.5 Input / Output Master Value Correction

NOTE

Function implemented from:	Element pointer EP:	
MASTERDRIVES MC:	≥ V1.4x	1 to 8
	≥ V1.5x	9 to 12

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 39**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer (EP) = **1** to **8**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1** to **8**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Offset acceleration correction in [1000 LU/s^2]	EP=1	DEC
DBDz+4	Offset max. correction speed in [10 LU/min]	EP=2	DEC
DBDz+8	Axis cycle length master value 1 in [LU]	EP=3	DEC
DBDz+12	Axis cycle length master value 2 in [LU]	EP=4	DEC
DBDz+16	Adjustment master value 2: numerator	EP=5	DEC
DBDz+20	Adjustment master value 2: denominator	EP=6	DEC
DBDz+24	Function selection	EP=7	DEC
DBDz+28	Axis cycle length master value output in [LU]	EP=8	DEC
DBDz+32	Rated velocity master value 1 [10 * LU/min]	EP=9	DEC
DBDz+36	Rated velocity master value 2 [10 * LU/min]	EP=10	DEC
DBDz+40	2nd adjustment master value 2: numerator	EP=11	DEC
DBDz+44	2nd adjustment master value 2: denominator	EP=12	DEC

Function Selection

The following table describes the input options for the function selection.

Input	Meaning
0	Master value correction
1	Master value adjustment

Table 10-11 Meaning of Function Selection

Task Description

The "Input / Output Master Value Correction" task inputs or outputs the appropriate synchronization parameters.

Further Information

The meaning of the individual parameters is documented in the Function Description in the chapter on “Synchronization functions”.

10.6 Input / Output Real Master

NOTE

Function implemented from:	Element pointer EP:	
MASTERDRIVES MC:	≥ V1.4x	1 to 6

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 38**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer (EP) = **1** to **6**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1** to **6**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Filter time input signal in [ms]	EP=1	DEC
DBDz+4	Dead time compensation in [ms]	EP=2	DEC
DBDz+8	Axis cycle length output in [LU]	EP=3	DEC
DBDz+12	Axis cycle length input in [LU]	EP=4	DEC
DBDz+16	Filter dead time compensation [ms]	EP=5	DEC
DBDz+20	Rated velocity in [1000 Lu/min]	EP=6	DEC

Task Description

The "Input / Output Real Master" task inputs or outputs the appropriate parameters.

Further Information

The meaning of the individual parameters is documented in the Function Description in the chapter on “Synchronization functions”.

11 SIMATIC Motion Control Tasks

Contents

In this chapter, you will find a description of the tasks for the SIMATIC Motion control solution and operation using the IM178.

11.1 Axis execution, enables

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 33**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **65**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBBz	Number of axis allocations calculated	DEC
DBBz+1	1st axis: local axis number on M7	DEC
DBBz+2	1st axis: axis enables	BIN
DBBz+3	1st axis: reduction factor	DEC
DBBz+4	1st axis: shift	DEC
DBBz+5	2nd axis: local axis number on M7	DEC
DBBz+6	2nd axis: axis enables	BIN
DBBz+7	2nd axis: reduction factor	DEC
DBBz+8	2nd axis: shift	DEC
	⋮	
DBBz+125	32nd axis: local axis number on M7	DEC
DBBz+126	32nd axis: axis enables	BIN
DBBz+127	32nd axis: reduction factor	DEC
DBBz+128	32nd axis: shift	DEC
DBBz+129	Reserved	DEC

Axis Enables

The following table describes the axis enable bits.

Datenwort	Bedeutung
DBBz+ (n-1)*4+2	Darstellung: 0 | 0 | 0 | 0 | EN_CU | EN_DAS | EN_CAM | EN_VM BIN EN_VM=1: Virtual master enable EN_CAM=1: Cam controller enable EN_DAS=1: Offset angle setting enable EN_CU=1: Catch-up enable

Table 11-1 Meaning of axis enable bits

NOTE

Function implemented as of:	Further enables:	
SIMATIC Motion Control:	≥ V1.00.48	EN_DAS and EN_CU

Task Description

The "Axis execution, enables" task parameterizes the actual axes calculated and their axis-specific enables. At the same time, the task sets the execution sequence for the axes. This makes it possible to calculate the local M7 axis 4 before the local M7 axis 2 (e.g. axis 4 master axis, axis 2 slave axis). The local M7 axes do not have to be parameterized in ascending order (M7 axis number) and contiguously.

Axes which are not defined through the axis execution are automatically inactive.

11.2 Cam Controller

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 34**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1 to 19**	DEC
DBWx+6	Data type = **5**	DEC
DBWx+8	Number of elements = **1 to 19**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBDz	Local M7 axis number for input variable	EP=1	DEC
DBDz+4	Input variable selection	EP=2	BIN
DBDz+8	Hysteresis	EP=3	DEC
DBDz+12	Cam 1 start in [LU]	EP=4	DEC
DBDz+16	Cam 1 end in [LU]	EP=5	DEC
DBDz+20	Cam 2 start in [LU]	EP=6	DEC
DBDz+24	Cam 2 end in [LU]	EP=7	DEC
DBDz+28	Cam 3 start in [LU]	EP=8	DEC
DBDz+32	Cam 3 end in [LU]	EP=9	DEC
DBDz+36	Cam 4 start in [LU]	EP=10	DEC
DBDz+40	Cam 4 end in [LU]	EP=11	DEC
DBDz+44	Reserved	EP=12	DEC
DBDz+48	Reserved	EP=13	DEC
	⋮		
DBDz+72	Reserved	EP=19	DEC

Local M7 Axis Number for Input Variable

The cams can be allocated to any axis using the “local M7 axis number for input variable” parameter (Element 1). This means that, for example, eight cams can be allocated to one axis. The axis itself is parameterized with the value zero or its own axis number.

Input Variable Selection

The following table describes the input options for selecting the input variable.

Input	Meaning
0	Set position value
1	Actual position value
2	Virtual master
3	X coordinate of cam disc
4	Y coordinate of cam disc

Table 11-2 Meaning of input variable selection

Task Description

The “cam controller” task sets all the parameters for the cam controller.

11.3 Input / Output Machine Data

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **7**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**E**	**A**	0	0	0	0	0	0	BIN
DBBy+3	TI = **6**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **1**								DEC
DBWy+10	Z2 = first MD number **51 ≤ x ≤ 70**								DEC
DBWy+12	Z3 = first MD number **1 ≤ y ≤ 20**								DEC

Data Block GMC_DB_APP3

DBDz	Machine data x	DEC
DBDz+4	Machine data x+1	DEC
DBDz+n	Last machine data x+y	DEC

Task Description

The “input / output machine data” order inputs or outputs the machine data (MD51 to MD70) required for operation with the IM178. By entering additional information 2 (Z2) and additional information 3 (Z3), you can specify whether to input or output a single machine data number, part of the machine data or all the machine data. Each machine data number takes up 4 bytes (1 double word) in the data block. The machine data has to be activated after it is input, and to do this you need to use the “activate machine data” task.

Further Information

A list and description of the machine data options can be found in the “machine data and technology parameters” chapter of the description of function.

11.4 Output Machine Data Limits

Data Block GMC_DB_APP1

DBWx	Main task number = **2**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **7**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWy	Number of DWs = **0**								DEC
DBBy+2	**0**	**1**	0	0	0	0	0	0	BIN
DBBy+3	TI = **6**								DEC
DBWy+4	DB no.								DEC
DBWy+6	DBW no.								DEC
DBWy+8	Z1 = **2**								DEC
DBWy+10	Z2 = MD number **(51 ≤ x ≤ 70)**								DEC
DBWy+12	Z3 = axis type in accordance with MD 1								DEC

Data Block GMC_DB_APP3

DBDz	Lower input limit	DEC
DBDz+4	Upper input limit	DEC

Task Description

The “output machine data limits” task outputs the upper and lower input limits for the machine data in question. Specifying the axis type means that the limits are output in accordance with the axis type selected. The axis type corresponds to the value in machine data 1.

11.5 Digital Inputs – Extended I/O Peripherals

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 35**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **97**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBBz	Number of allocations of digital inputs	DEC
DBBz+1	1st allocation: peripheral input number	DEC
DBBz+2	1st allocation: axis number	DEC
DBBz+3	1st allocation: axis input number	DEC
DBBz+4	2nd allocation: peripheral input number	DEC
DBBz+5	2nd allocation: axis number	DEC
DBBz+6	2nd allocation: axis input number	DEC
	⋮	
DBBz+190	64th allocation: peripheral input number	DEC
DBBz+191	64th allocation: axis number	DEC
DBBz+192	64th allocation: axis input number	DEC
DBBz+193	Reserved	DEC

Task Description

The "Digital Inputs – Extended I/O Peripherals" task provides all the data on the digital inputs for the M7 axes defined.

11.6 Digital Outputs – Extended I/O Peripherals

Data Block GMC_DB_APP1

DBWx	Main task number = **+/- 36**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **97**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBBz	Number of allocations of digital outputs	DEC
DBBz+1	1st allocation: peripheral output number	DEC
DBBz+2	1st allocation: axis number	DEC
DBBz+3	1st allocation: axis output number	DEC
DBBz+4	2nd allocation: peripheral output number	DEC
DBBz+5	2nd allocation: axis number	DEC
DBBz+6	2nd allocation: axis output number	DEC
	⋮	
DBBz+190	64th allocation: peripheral output number	DEC
DBBz+191	64th allocation: axis number	DEC
DBBz+192	64th allocation: axis output number	DEC
DBBz+193	Reserved	DEC

Task Description

The "Digital Outputs – Extended I/O Peripherals" task provides all the data on the digital outputs for the M7 axes defined.

12 MCT Parameter Tasks

Contents

In this chapter you will find a description of the tasks for the MCT parameters.

12.1 Input/Output MCT Parameters

Data Block GMC_DB_APP1

DBWx	Main task number =**+/-2000**	DEC
DBWx+2	Absolute axis number	DEC
DBWx+4	Element pointer = **1**	DEC
DBWx+6	Data type = **2**	DEC
DBWx+8	Number of elements = **4**	DEC
DBWx+10	DB no.	DEC
DBWx+12	DBW no.	DEC

Data Block GMC_DB_APP2

DBWz	Parameter identifier (PKE)	DEC
DBWz+2	Parameter index (IND)	DEC
DBWz+4	Parameter value (high word)	DEC or HEX
DBWz+6	Parameter value (low word)	DEC or HEX

Parameter values are represented in DEC format; binectors and connectors are represented in HEX format.

Further Information

For further information about parameter identifiers and the parameter index, please refer to the MASTERDRIVES MC Compendium – Chapter "Profibus Communication".

Task Description

The "input/output MCT parameters" task can be used to input or output parameters via Profibus.

13 Complete List of Tasks and Task Header Descriptions

List of Individual Tasks

In the table below you will find a complete list of the tasks with task header descriptions

Task	MT	EP	DT	No. E	No. DW	I	Q	TI	Extension					
									1.	2.	3.	4.	5.	6.
Input/output machine data	2	1	2	7	0	X	X	1	1	X	X	-	-	-
Input/output function parameters	2	1	2	7	0	X	X	5	1	X	X	-	-	-
Activate machine data	5	1	2	0	-	-	-	-	-	-	-	-	-	-
Machine data - output limit values	2	1	2	7	0	-	X	1	2	X	X	-	-	-
Function parameters – output limit values	2	1	2	7	0	-	X	1	2	X	1	-	-	-
Clear NC program memory	13	1	2	0	-	-	-	-	-	-	-	-	-	-
Delete NC program	14	1	2	1	-	-	-	-	-	-	-	-	-	-
Delete NC block	15	1	2	2	-	-	-	-	-	-	-	-	-	-
Output NC program number	2	1	2	7	0	-	X	2	4	X	X	-	-	-
Output NC block number	2	1	2	6	0	-	X	2	6	X	-	-	-	-
Output software version	-7	1	2	4	-	-	-	-	-	-	-	-	-	-
Output software version with generation date	-8	1	2	10	-	-	-	-	-	-	-	-	-	-
Output NC program to OP	2	1	2	10	0	-	X	3	1	X	X	X	X	X
Input NC block to OP	2	1	2	7	0	X	-	3	1	X	X	-	-	-
Input NC program	2	1	2	8	0	X	-	3	2	X	0	X	-	-
Output NC program	2	1	2	8	0	-	X	3	2	X	X	X	-	-
Input NC block	2	1	2	8	0	X	-	3	0	X	X	X	-	-
Output NC block	2	1	2	8	0	-	X	3	0	X	X	X	-	-
Input/output MDI block to OP	2	1	2	6	0	X	X	4	2	X	-	-	-	-
Input/output MDI block	2	1	2	6	0	X	X	4	1	X	-	-	-	-
Input/output roll feed loop count MDI	± 16	1	5	1	-	-	-	-	-	-	-	-	-	-
Input/output setup velocities	± 17	1 or 2	5	1 or 2	-	-	-	-	-	-	-	-	-	-
Input/output block search	± 18	1	2	8	-	-	-	-	-	-	-	-	-	-
Input automatic block search	19	1	2	0	-	-	-	-	-	-	-	-	-	-
Input teach-in	20	1	2	2	-	-	-	-	-	-	-	-	-	-
Input/output zero offset	± 21	1 to 6	5	1 to 6	-	-	-	-	-	-	-	-	-	-
Input/output roll feed velocity override	± 22	1	5	1	-	-	-	-	-	-	-	-	-	-

Task	MT	EP	DT	No. E	No. DW	I	Q	TI	Extension					
									1.	2.	3.	4.	5.	6.
Input/output simulation	± 23	1	2	1	-	-	-	-	-	-	-	-	-	-
Input / output write data to EEPROM or RAM	± 37	1	2	1	-	-	-	-	-	-	-	-	-	-
Input / output activate/deactivate encoder changeover	± 40	1	2	1	-	-	-	-	-	-	-	-	-	-
Output actual values	3	1	2	10	0	-	X	8	X	X	X	X	X	X
Output actual value data collection 1 to OP	3	1	2	5	0	-	X	7	1	-	-	-	-	-
Output actual value data collection 2 to OP	3	1	2	5	0	-	X	7	2 or 3	-	-	-	-	-
Output runtime data	-24	1	2	8	-	-	-	-	-	-	-	-	-	-
Input/output tool offset	2	1	2	6	0	X	X	14	X	X	-	-	-	-
Acknowledge GMC technology warnings	6	1	2	0	-	-	-	-	-	-	-	-	-	-
Output GMC technology warnings	-9	1	2	8	-	-	-	-	-	-	-	-	-	-
Delete NC table and output to OP	10	1	2	8	0	-	X	16	4	X	X	X	-	-
Accept NC table	± 26	1 to 10	5	1 to 10	-	-	-	-	-	-	-	-	-	-
Delete NC table interpolation point and output to OP	10	1	2	8	0	-	X	16	3	X	X	X	-	-
Insert NC table interpolation point and output to OP	10	1	2	8	0	-	X	16	2	X	X	X	-	-
Input NC table from OP	10	1	2	8	0	X	-	17	1	X	X	X	-	-
Output NC table to OP	10	1	2	9	0	-	X	17	1	X	X	X	X	-
Input/output NC table	10	1	2	8	0	X	X	17	2	X	X	X	-	-
Output NC table status	-29	1 to 8	5	1 to 8	-	-	-	-	-	-	-	-	-	-
Input/output NC table parameters	± 12	1 to 9	5	1 to 9	-	-	-	-	-	-	-	-	-	-
Input/output synchronization parameters	± 11	1 to 22	5	1 to 22	-	-	-	-	-	-	-	-	-	-
Output actual values and diagnostic data	-30	1 to 22	5	1 to 22	-	-	-	-	-	-	-	-	-	-
Synchronization parameters for offset angle setting	± 31	1 to 12	5	1 to 12	-	-	-	-	-	-	-	-	-	-
Catch-up synchronization parameters	± 32	1 to 9	5	1 to 9	-	-	-	-	-	-	-	-	-	-
Input / output master value correction	± 39	1 to 8	5	1 to 8	-	-	-	-	-	-	-	-	-	-
Input / output real master	± 38	1 to 6	5	1 to 6	-	-	-	-	-	-	-	-	-	-
Axis execution enable	± 33	1	2	65	-	-	-	-	-	-	-	-	-	-

Task	MT	EP	DT	No. E	No. DW	I	Q	TI	Extension					
									1.	2.	3.	4.	5.	6.
Cam controller	± 34	1 to 19	5	1 to 19	-	-	-	-	-	-	-	-	-	-
Input / output machine data	2	1	2	7	0	X	X	6	1	X	X	-	-	-
Output machine data limit values	2	1	2	7	0	-	X	6	2	X	X	-	-	-
Digital inputs– extended I/O peripherals	± 35	1	2	97	-	-	-	-	-	-	-	-	-	-
Digital outputs – extended I/O peripherals	± 36	1	2	97	-	-	-	-	-	-	-	-	-	-
Input/output MCT parameters	± 2000	1	2	4	-	-	-	-	-	-	-	-	-	-

Table 13-1 List of Tasks

Index

L

M

N

O

P

R

SIEMENS

System Solutions

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Part 3: User Interfaces 11.2002

GMC-OP-OAM Standard Software

SIEMENS

System Solutions

MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control

Part 3: User Interfaces

GMC-OP-OAM Standard Software

Edition 11.2002

Documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the "Remarks" column.

Status code in the "Remarks" column:

A .... New documentation.
B .... Unrevised reprint with new Order No.
C .... Revised edition with new status.

Edition	Order No.	Remarks
04.98	6AT1880-0AA00-1BA0	**A**
10.99	6AT1880-0AA00-1BC0	**C**
11.2002	6AT1880-0AA00-1BE0	**C**

Order No.: 6AT1880-0AA00-1BE0
Printed in the Federal Republic of Germany

Siemens-Aktiengesellschaft

Contents

1 Introduction

Contents of this Manual

This documentation contains:

- A description of all procedures necessary for installation and configuration
- Configuration and operation of the "GMC-OP-OAM" standard software
- Operator control using the standard screen forms of the OP25/27

Structure of this Manual

The documentation is organized into the following sections:

- Overview: An overview of General Motion Control
- Installation/Configuration: Installation of the S7 software and the ProTool project
- A description of the GMC-OP-OAM standard software
- Program and Configuration Examples: Examples of finished programs to help you get started
- Operating instructions for the standard screen forms on the OP25/OP27
- Expert Knowledge/Tips: How to integrate your own displays

2 Overview

Introduction

GMC (General Motion Control) describes the positioning and synchronization system solution for the application of the technology in SIMATIC Motion Control (M7-FM target systems) and in MASTERDRIVES Motion Control, Technology Option F01 (MASTERDRIVES MC target system).

SIMATIC Motion Control

In the solution with SIMATIC Motion Control, the technology is installed on one or more M7 computers. The M7-FM(s) is (are) connected to the individual MASTERDRIVES MC drive units across a high-speed fiber-optic link. The MASTERDRIVES MC units contain only the basic functionality with position control (MCB = MC with basic functionality).

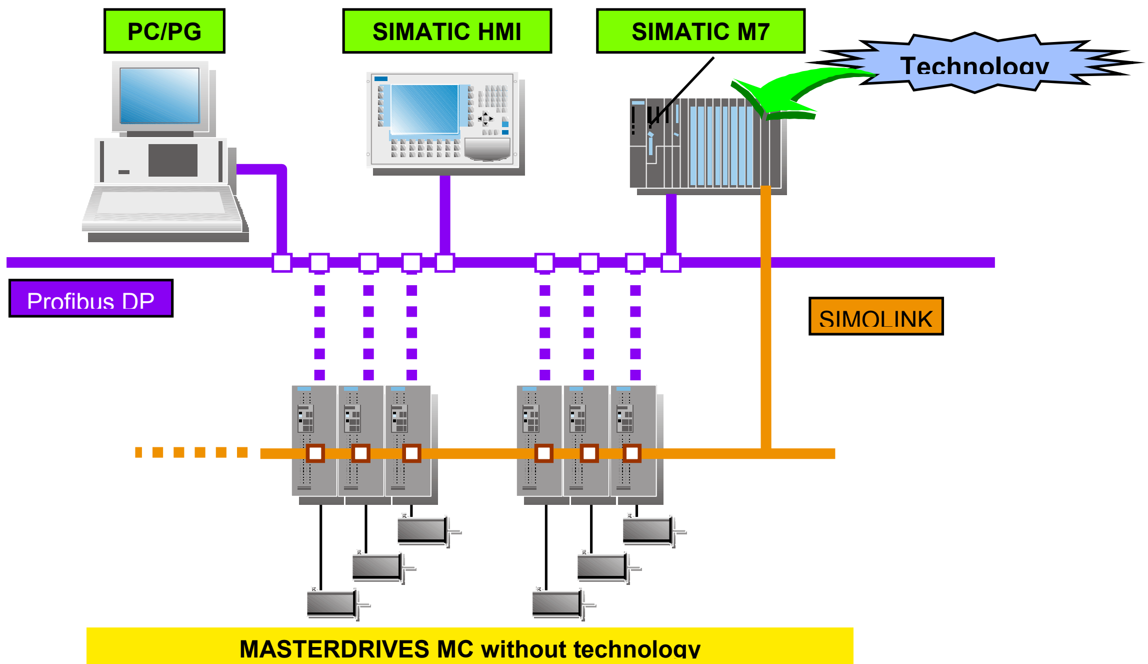


Fig. 2-1 Motion Control on M7-FM Target System with MASTERDRIVES MC

SIMATIC Motion Control with IM178

In the solution with SIMATIC Motion Control, the technology is installed on one or more M7 computers. M7-FMs are connected with the individual drive devices of the IM178 interface module via a synchronous Profibus module which only incorporate the basic functionality and the set speed output via an analog output.

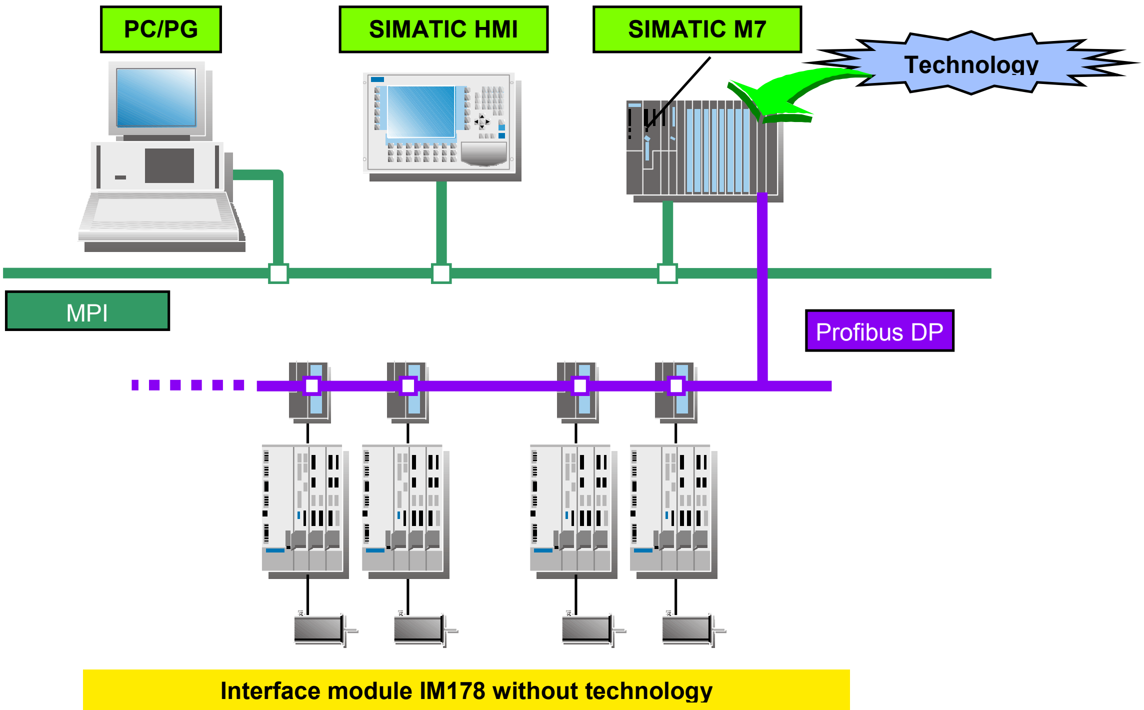


Fig. 2-2 Motion Control on M7-FM Target System with IM178

MASTERDRIVES Motion Control (Technology Option F01)

In the solution with MASTERDRIVES Motion Control (Technology Option F01), the technology is installed directly on the MASTERDRIVES MC (MCT = MC with technology). The S7-CPU is connected to the individual MASTERDRIVES MC drive units via PROFIBUS-DP. For the high-speed data communication required by the synchronization technology, the individual MASTERDRIVES MC units must be connected via the SIMOLINK fiber-optic link.

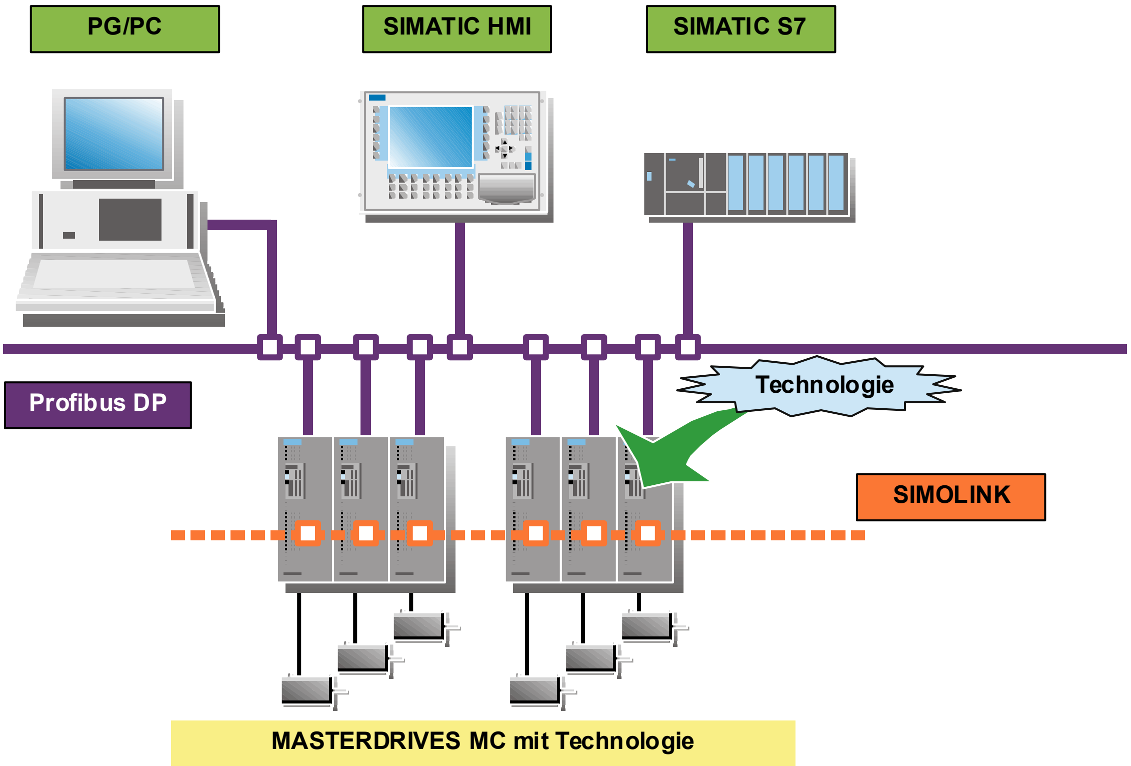


Fig. 2-3 Motion Control auf Zielsystem MASTERDRIVES MC

Extensions

MASTERDRIVES Motion Control and SIMATIC Motion Control can be used together. It is also possible for SIMATIC Motion Control (M7-FM) to communicate with the basic functionality of MASTERDRIVES MC via PROFIBUS-DP.

GMC-OP-OAM (**O**perating **a**nd **M**onitoring) provides a convenient user interface on the OP25/OP27 which allows you to communicate centrally (via the SIMATIC S7 system) with all axes. GMC-OP-OAM is designed to automatically track any modifications in system capacity with reference to the number of axes or a change in the target system configuration.

GMC-OP-OAM is based on the GMC-BASIC standard software. Runtime GMC-BASIC software is therefore required for the operation of GMC-OP-OAM.

The OP25/OP27 user interfaces can be connected to the SIMATIC S7 CPU via either MPI or PROFIBUS.

Hardware and Software

In addition to the hardware and software components detailed in the "GMC-BASIC" standard software description, you need:

- ♦ An OP25 or OP27 operator panel
- ♦ ProTool software V4.x or higher
- ♦ GMC-OP-OAM software for SIMATIC S7 and OP25/OP27

3 Installation / Configuration

Contents

In this chapter you will find the information required for the installation and configuration of the "GMC-OP-OAM" software.

Introduction

The SIMATIC S7 system is used as a centralized application interface to the technology and the OPs. The necessary S7 library and the example S7 and OP projects are generated automatically during installation of the "GMC-OP-OAM" software package.

The GMC-OP-OAM package allows you to visualize all configured axes independent of the target system (SIMATIC Motion Control or MASTERDRIVES MC with technology option F01).

Software Requirements

The "GMC-BASIC" software is a basic requirement for the operation of the GMC-OP-OAM package.

The installation diskettes for the GMC software packages can be read on a programming device/PC running the MS-Windows 95 / NT operating system.

The software packages cannot be installed until STEP 7 has been installed completely. If STEP 7 is deinstalled, the GMC software packages must be deinstalled first.

Hardware Requirements

- ♦ Programmable logic controller with an approved CPU
- ♦ M7-FM and/or MASTERDRIVES MC
- ♦ Approx. 8 MB free capacity on your hard disk
- ♦ Programming device/PC with STEP 7, V3.1 or higher

NOTE

The technology (positioning, synchronization) can be operated centrally on an M7-FM and/or decentrally on a MASTERDRIVES MC. If the technology is operated on a MASTERDRIVES MC, the term **MCT** (Motion Control with Technology) is used in this manual. If the technology is operated on an M7-FM and the MASTERDIVES MC is used without technology as a drive controller, the term **MCB** (Motion Control with Basic Functionality) is used.

3.1 Installation

Preparations

Before you start the setup program, please close all applications (e.g. STEP 7, MS-Word etc.), because you will need to reboot your programming device/PC after installing the example program, library and OP project, so that all the system variables are entered completely.

Installation

Insert the diskette in the drive and start the setup program. Important operating instructions are provided by the setup program during the installation routine. The drive on which you install the software is determined automatically from the installed version of STEP 7.

The following components are installed:

- A library named "P7MC9LIB" in directory STEP7\S7LIBS
- A project named "P7MC19EX" in directory STEP7\EXAMPLES
- Several OP projects in directory STEP7\S7LIBS\GMC_OAM\GMC_OPnn_Vnnn.pdb (nn stands for the OP number and revision level of the OP project).

The library contains all the blocks of the "GMC-OP-OAM" software package. The project contains the example program. The OP project contains all the standard displays for GMC and, as examples, several "supported user displays" (see the chapter entitled "Expert Knowledge/Tips").

NOTE

In order to edit the OP project and load it on the OP, you will need the "ProTool" configuring tool, version V4.x or higher.

WARNING

You should not use MS-Windows tools, such as the Explorer, to move STEP 7 files and folders or to modify the STEP 7 data in the MS-Windows registry. The correct operation of the program can no longer be assured after such modifications.

3.2 Configuration

Configuration

Configuration of the "GMC-OP-OAM" software package is limited to two entries in the GMC_DB_ORG data block.

+0.0	X_gmc_pict_pointer	STRUCT		
+0.0	i_db_no	INT	120	DB_No
+2.0	i_dbw_no	INT	0	DBW_No
+4.0	i_res1	INT	0	
=6.0		END_STR		
+6.0	X_gmc_order	STRUCT		
+0.0	i_db_no	INT	119	DB_No
+2.0	i_dbw_no	INT	0	DBW_No
+4.0	i_res1	INT	0	
=6.0		END_STR		

In data block GMC_DB_ORG, you must enter the DB number and the DBW number for data blocks GMC_DB_PICTPOINTER (X_gmc_pict_pointer.i_db_no and X_gmc_pict_pointer.i_dbw_no) and GMC_DB_ORDER (X_gmc_order.i_db_no and X_gmc_order.i_dbw_no) according to your project.

4 Function Blocks

Contents

In this chapter you will find a description of the function blocks.

Introduction

The function blocks administer and support all of the standard displays supplied for Motion Control in respect of display selection, user input and cyclic tasks such as "read actual values".

The function blocks detect a system restart and automatically initiate an initialization routine. The "GMC-OP-OAM" software can only run when all function blocks, including those contained in the "GMC-BASIC" package, have completed the initialization. Display processing commences when the "OP online" status is active.

All function block calls are already configured and parameterized. All you have to do is copy the example program into your project, complete the configuration of OB1 and GMC_DB_ORG and load the project into the S7 CPU. Provided that you have carried out the necessary hardware configuration, that you have inserted the technology software (on a flash card) if an M7 FM is in use, that you have made the necessary connections (MPI, Profibus, Simolink) and that you have loaded the OP project required onto your OP, you now have a functional system.

4.1 Function Block GMC_FB_PICTMAN (FB122)

Task

The GMC_FB_PICTMAN function block is the central block for task management of OP displays. It monitors the currently selected display and enters the tasks required for display selection, user input and cyclic tasks in the first application box in GMC_DB_COM.

For display on the OP, the status of the current task and any errors that occur are copied into the display area of the OP after termination.

The FB interacts with the following data blocks:

- GMC_DB_ORG (organization DB)
- GMC_DB_APP (application data block)
- GMC_DB_COM (DB for the application boxes for initiating a task)
- GMC_DB_ORDER (description of the tasks for the HMI)
- GMC_DB_PICT_POINTER (auxiliary data block for storing the DB/DBW pointers to the display-specific tasks in GMC_DB_ORDER)
- GMC_IDB_PICTMAN_DATA (instance DB of GMC_FB_PICTMAN **and** GMC_FB_PICTDATA)

Further Information

Data blocks GMC_DB_ORG, GMC_DB_APP and GMC_DB_COM are described in "GMC-BASIC Standard Software".

The data in the data blocks for GMC-OP-OAM which are relevant for your purposes are described in this manual in chapter 6.1 "Integrating Supported User Displays".

Integration

The GMC_FB_PICTDATA function block is called up in order to process the technology-specific displays. The function block cannot be called from within your user program. This function block merely has to be present in your project.

Any errors that occur are entered in the STATUS block parameter and in the GMC_DB_ORG data block.

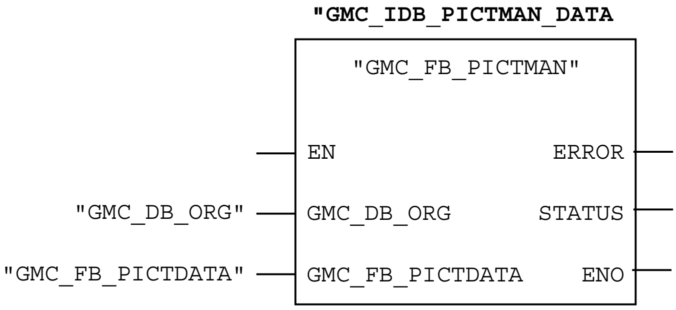
The GMC_FB_PICTMAN function block must be called up and supplied with parameters once during cyclic operation (e.g. OB1). A second call in another OB level, e.g. time-driven level OB35, is not permitted. Any FB number can be used. The function block is not multi-instance capable and requires its own instance data block.

NOTE

Before you start the system, you must configure GMC_DB_ORG correctly.

If you do not use the suggested instance DB number (DB122) as the instance DB, you must update all the variables of the OP project!

Block



Explanation of Input and Output Parameters

The table below explains the input and output parameters of the "GMC_FB_PICTMAN" function block

Name	Type	Data Type	Description	Default
GMC_DB_ORG	IN	BLOCK_DB	Data block number GMC_DB_ORG	DB100
GMC_FB _PICTDATA	IN	BLOCK_FB	CPU-specific, (zero not allowed); function block number GMC_FB_PICTDATA	FB123
ERROR	OUT	WORD	Error message from FB	
STATUS	OUT	WORD	Status message from FB	

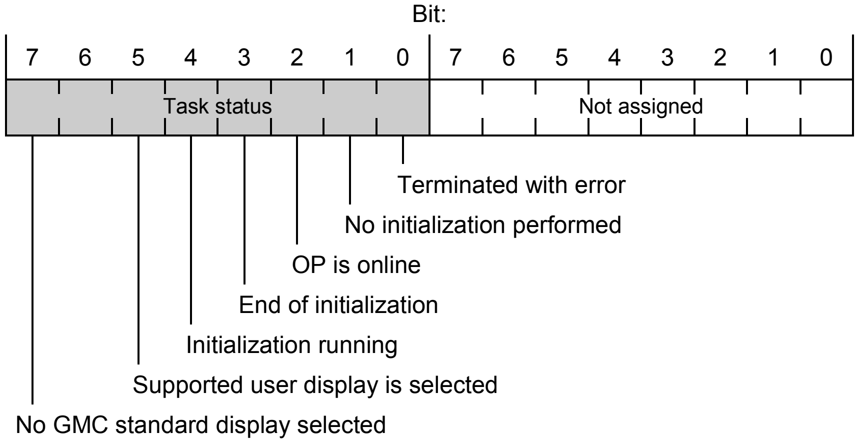
Description of Block Parameters

The hardware and software configuration is declared to the function block in the GMC_DB_ORG parameter. GMC_FB_PICTMAN calls up the GMC_FB_PICTDATA function block internally. This function block controls the technology-specific part. The associated function block number can be chosen freely and is declared to GMC_FB_PICTMAN in the GMC_FB_PICTDATA parameter.

The ERROR parameter outputs errors detected by the GMC_FB_PICTMAN function block. These messages are also stored in the GMC_DB_ORG data block.

STATUS Parameter

The assignment of the STATUS parameter is as follows:



- **Terminated with error**
  The FB has detected an error independently. It outputs the error in the ERROR parameter and enters it simultaneously in DB GMC_DB_ORG.
- **No initialization performed**
  No initialization has been performed by GMC_FB_START (e.g. the instance DB has been overwritten).
- **OP is online**
  The OP life bit is evaluated and the "OP is online" status bit is set or reset.
- **End of initialization**
  The startup initialization is complete.
- **Initialization running**
  The startup initialization is still active; HMI processing is not yet active.
- **Supported user display is selected**
  The display which is currently on the screen is within the number range from 1065 to 1096.
- **No GMC standard display selected**
  This status bit indicates that no GMC standard display or supported user display is currently on the OP screen (current display number < 1000 or >1096).

4.2 EN/ENO Feature

To enable the ENO (enable output) parameter if an error has occurred, the BR (binary result) bit is set to zero before quitting the function block. If the block is processed without an error, the BR bit is set to the "1" signal.

5 Program/Configuration Examples

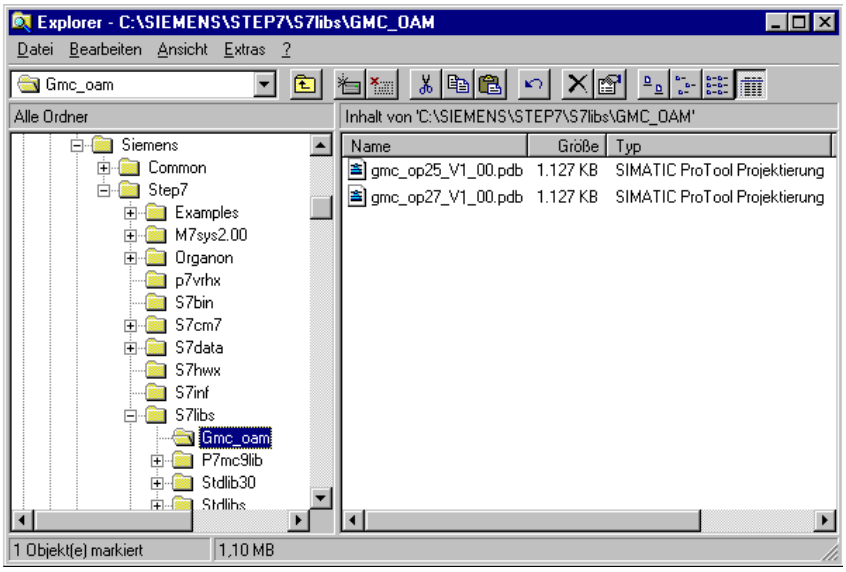
Contents

In this chapter you will find program and configuration examples.

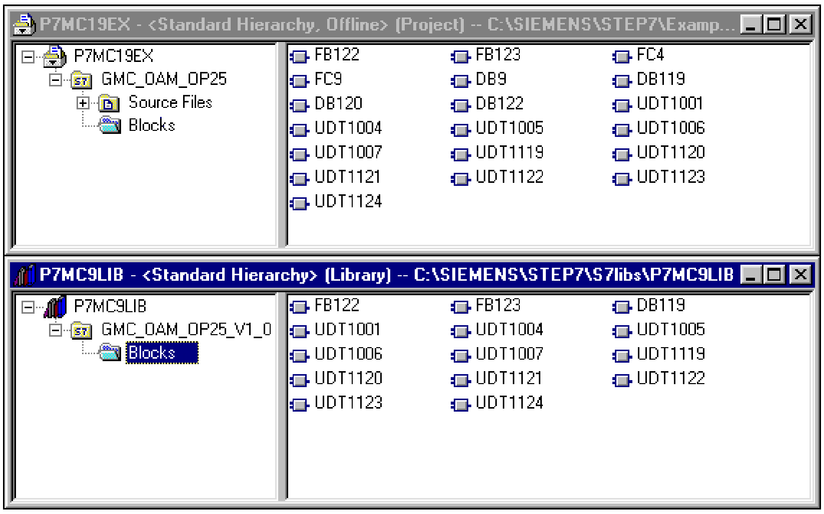
5.1 Introduction

OP Project

When the installation has been successfully completed, you will find the OP projects in directory SIEMENS\STEP7\S7LIBS\GMC_OAM using the Explorer.



When you have also installed the "GMC-OP-OAM" standard software with SETUP, call up the S7 File Manager. A library named "P7MC9LIB" and a project named "P7MC19EX" will have been installed.



The "P7MC9LIB" library contains all the blocks required for the operation of the "GMC-OP-OAM" standard software. The "P7MC19EX" project contains the example program and all the necessary standard blocks from the library as well as the necessary function block calls and function calls. All of the necessary data blocks are also generated and initialized.

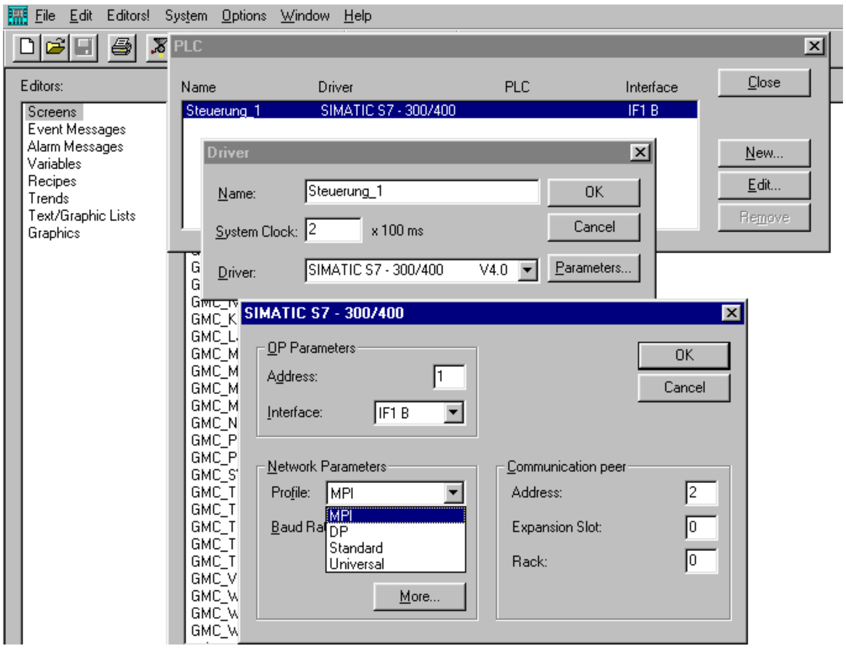
The UDTs (user-defined data types) in the library and project are used to structure the data blocks.

5.2 Opening and Loading the OP Project

After you open the standard project with ProTool, you should store it under a name of your choice for backup purposes.

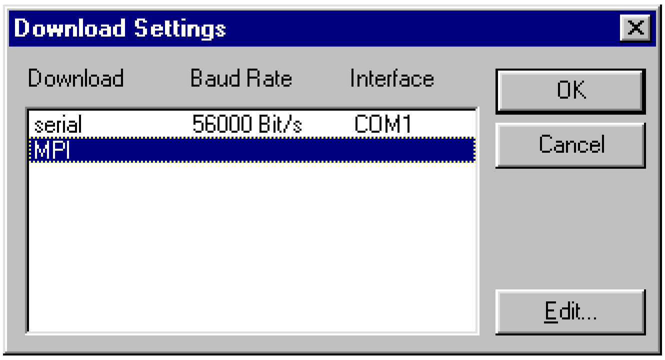
OP Connection

Define the OP connection to the S7-CPU (MPI or DP) via the "System/PLC" menu.



Communication

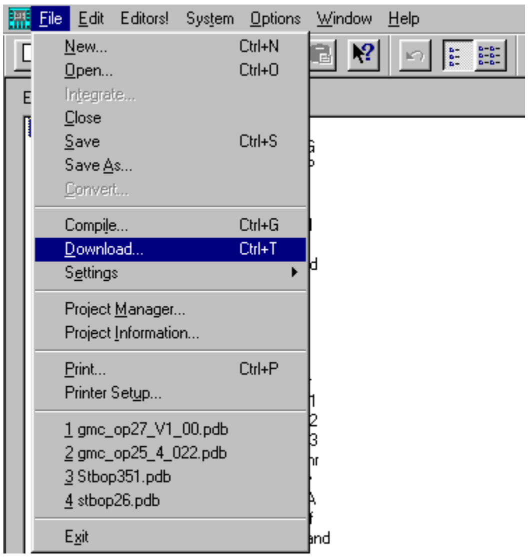
Set the communication parameters (MPI or COM1/2) for transferring your project from your programming device/PC to the OP via the "File\Settings\Download" menu.



Language Assignment

You can use the "System\Language" menu to select one or more of the supplied languages and load them on your OP. The standard OP project supports the German and English languages. German and English are selected by default, i.e. those two languages are transferred to the OP.

Now switch the OP to transfer mode (see the OP manual) and then load your project onto the OP via the "File\Download" menu.



NOTE

You need ProTool V4.0 or higher for your OP project. The first time you load an OP project, you must use the serial interface. You can subsequently use the MPI interface to transfer the project.

You must not renumber the displays of the standard OP project or change the display navigation. All of the variables displayed access a permanent data block (DB122 GMC_IDB_PICTMAN_DATA in the standard OP project). If you change the data block number in your S7 project, you must update all the variables of the OP project accordingly.

5.3 Creating the User Project

General Information

There are various ways of creating your individual user project. If you have already created your own user project from the example GMC-BASIC project, the easiest method is to copy all of the blocks from the example program supplied with the software (P7MC19EX) into your own project. You can, of course, use the library, however you will then have to program all the calls and parameters yourself.

Step 1: Create a New Project

The first step is to create a new project or open an existing one. Create your S7 hardware configuration according to the capacity of your system in terms of M7-FM or MASTERDRIVES MC units (see the chapter entitled "Installation").

Step 2: Copy the Data You Need into the New Project

The next step is to copy the blocks you need from the library or copy all the blocks from the example program into your S7 project. You will also need to copy the symbol table from the example project into your own project, in order to access the symbolic names.

NOTE

If you use the blocks from the library, you must program all the calls and the parameters for the function blocks yourself. You can follow the procedures used in the example programs or in the example calls described below.

If there is an overlap between the block numbers of the "GMC-OP-OAM" standard software and your own block numbers, you should consider renaming your own blocks. If this is not possible, please refer to the chapter entitled "Expert Knowledge/Tips". The information described in this chapter includes options for modifying the organization of the "GMC-OP-OAM" standard software.

Step 3: Include the GMC-OP-OAM Block Calls

You must add function call FC4 (FC_CALL_OAM_OP25) to OB1 in Network 4. Follow the procedure in the example call below.

Step 4: Configure GMC-OP-OAM in DB_ORG

Now open your GMC_DB_ORG data block. In the standard version, the data block numbers for the GMC-OAM package are initialized with the value –1 (i.e. "does not exist"). Now change to the data view of the data block in order to enter the data block numbers.

2.0	X_general.X_gmc_pict_pointer.i_db_no	INT	-1	120	DB_No
4.0	X_general.X_gmc_pict_pointer.i_dbw_no	INT	0	0	DBW_No
6.0	X_general.X_gmc_pict_pointer.i_res1	INT	0	0	
8.0	X_general.X_gmc_order.i_db_no	INT	-1	119	DB_No
10.0	X_general.X_gmc_order.i_dbw_no	INT	0	0	DBW_No
12.0	X_general.X_gmc_order.i_res1	INT	0	0	

When using the standard configuration, you enter number 120 for data block GMC_DB_PICTPOINTER and number 119 for GMC_DB_ORDER. Save the modified data block.

NOTE

Do not use the command "Edit -> Initialize Data Block", otherwise the initial values of the axis description (always 0) will overwrite the data. For this reason, you should always enter the values in the data view.

Step 5

When you have created your project correctly, you can load it into the S7-CPU with the CPU in stop mode. You should reset the CPU memory first.

Step 6

Switch the S7-CPU to RUN mode. When the S7-CPU starts up, the parameters in GMC_DB_ORG are checked and all required data blocks are set up automatically.

If all the parameters are correct, the S7-CPU will switch to RUN. If this is not the case, you should read out the S7 diagnostic buffer. If a block of the "GMC-BASIC" or "GMC-OP-OAM" standard software is responsible for the error, please refer to the chapter entitled "Expert Knowledge/Tips" (in the GMC-BASIC Configuring Guide). The information in this chapter includes a description of troubleshooting procedures in the event of problems with "GMC-BASIC" or "GMC-OP-OAM".

NOTE

If you have made changes to your project, you must switch the S7-CPU to stop in order to load the changes.

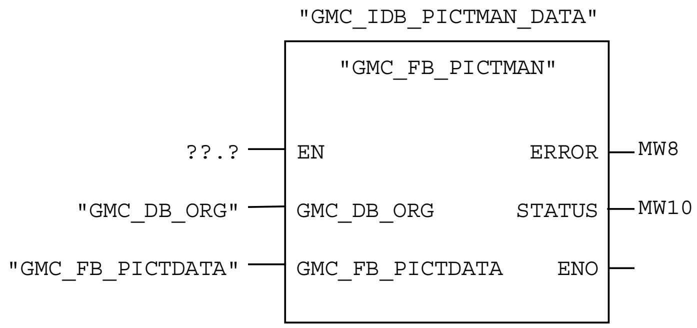
Example Calls

The example calls correspond to the example projects supplied with the software.

1. Startup
   The startup must not be changed when integrating the GMC-OP-OAM package.
2. Process interrupt for SIMATIC Motion Control
   The integration of the GMC-OP-OAM package does not require the modification of the process interrupt level in the program.
3. Cyclic operation
   Add the FC4 function call to OB1.
4. GMC_FB_PICTMAN call

FC4 :

Network 1 : Call GMC_FB_PICTMAN



Symbol information:

FB122	GMC_FB_PICTMAN	GMC FB for the higher-order functions of display management
DB122	GMC_IDB_PICTMAN_DATA	instance DB for GMC_FB_PICTMAN and _PICTDATA
DB100	GMC_DB_ORG	axis assignment
FB123	GMC_FB_PICTDATA	GMC FB for the display-specific function

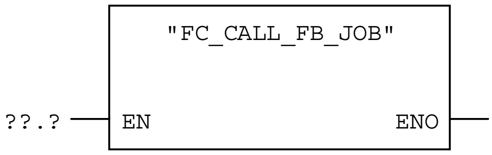
5. Block calls in OB1

   OB1 is used exclusively as a launching pad for the cyclic component of the "GMC-BASIC" and "GMC-OP-OAM" standard software.

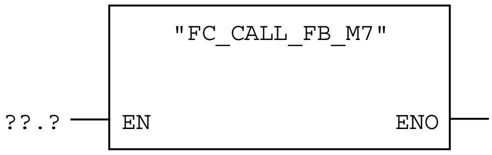
   The call in network 2 is only required for the M7-FM target system. The call in network 3 is only required for the MCT target system.

OB1 :

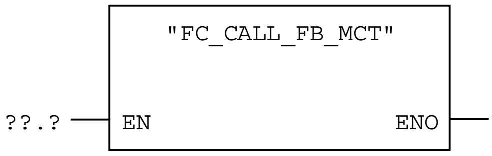
Network 1 : Call FC_CALL_FB_JOB



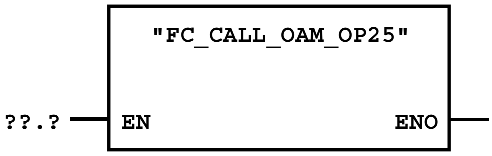
Network 2 : Call FC_CALL_FB_M7



Network 3 : Call FC_CALL_FB_MCT



Network 4 : Call FC_CALL_OAM_OP25



6 Expert Knowledge/Tips

Contents

In this chapter you will find various tips, for example on how to use the GMC-OP-OAM standard software support for your plant displays.

6.1 Integrating Supported User Displays

Introduction

What do we mean by "integrating supported user displays" and how do you benefit from this feature?

The following specifications apply within the GMC standard:

- OP standard displays are assigned to the number range from 1000 to 1010
- GMC standard displays are assigned to the number range from 1011 to 1064
- GMC-supported user displays are assigned to the number range from 1065 to 1096

All display numbers outside this number range are pure user displays and are not supported by the GMC-OP-OAM standard software.

However, the GMC-OP-OAM standard software does provide task support for certain displays. Simple features allow you to program one or more tasks for display selection, cyclic tasks and user input, and to use the existing data structure for these tasks.

Application Interface

The application interface comprises three data blocks which are required for the operation of the OP.

The GMC-OAM project distinguishes between standard displays (display numbers 1011 to 1064), supported user displays (display numbers 1065 to 1096) and user displays. All of the tasks for user displays are configured in the GMC_DB_ORDER data block. If you use the number range for "supported user displays", you must append the tasks you use contiguously to the GMC_DB_ORDER data block.

6.1.1 Instance Data Block GMC_IDB_PICTMAN_DATA (DB122)

Task

The GMC_FB_PICTMAN function block interacts with an instance data block. The DB number is specified with the call. All variables required for OAM operation are stored in the instance DB.

You are not allowed to access or modify these variables directly (e.g. with T DBW10).

The following variables present an exception to this rule:
You are allowed to access OP data, the interface area, the system keyboard and the display number area **symbolically**. Symbolic access to the parameters of "supported user displays" is also permitted.

DBW	Meaning	Symbolic Access Via ...
0	Parameter transfer for supported user displays, display numbers 1065 to 1096: • Display number • User task number	 i_SUPPORT_PICTURE_NO [int] i_JOB_NO [int]
n	Data areas required for the OP: • Interface area (32 bytes)	 z_OP0.a_OP_INTERFACE_RANGE[1] z_OP0.a_OP_INTERFACE_RANGE[2] z_OP0.a_OP_INTERFACE_RANGE[3] : z_OP0.a_OP_INTERFACE_RANGE[31] z_OP0.a_OP_INTERFACE_RANGE[32]
	• System keyboard (48 bool)	z_OP0.a_SYSTEM_KEYBOARD_IMAGE[1] z_OP0.a_SYSTEM_KEYBOARD_IMAGE[2] z_OP0.a_SYSTEM_KEYBOARD_IMAGE[3] : z_OP0.a_SYSTEM_KEYBOARD_IMAGE[47] z_OP0.a_SYSTEM_KEYBOARD_IMAGE[48]
	Display number area (5 words) • Current display type • Current display number • Reserved • Current input field number • Reserved	 z_OP0.x_PICTURE_NO_RANGE.i_AKT_PICT_TYP z_OP0.x_PICTURE_NO_RANGE.i_AKT_PICT_NO z_OP0.x_PICTURE_NO_RANGE.i_RES3 z_OP0.x_PICTURE_NO_RANGE.i_AKT_INPUT_NO z_OP0.x_PICTURE_NO_RANGE.i_RES5

Table 6-1 Description of Instance DB "GMC_FB_PICTMAN_DATA"

NOTE

DB122 is used as the instance data block. All variables displayed on the OP are also stored in this data block.
If you want to use a different instance DB number in your project, you have to modify all variables accordingly in the OP project!

6.1.2 Data Block GMC_DB_ORDER (DB119)

Task

Three blocks for task descriptions are defined for each display in the GMC_DB_ORDER data block. The first block describes the tasks for display selection; the second block describes the cyclic tasks, and the third block describes the user input tasks. Each block can consist of several contiguous tasks. The number of tasks in a block is defined in the "number of subtasks" parameter. These are then stored contiguously.

A user input is generally transferred with a task such as "transfer machine data" or "activate machine data". It is possible to formulate several individual tasks in order to transfer several user values. The parameter "number of user input tasks" is therefore also required in order to describe the user input tasks. The description of a task always has a constant length of 14 bytes.

The GMC_DB_ORDER data block is supplied for the standard displays and for the supported user display "control axis" (display 1065). If you use the "supported user display" number range, you must append your tasks to the standard displays using the same pattern.

NOTE

Version 2.0 and later also includes the real master and the master value correction as supported user displays.

DBW	Meaning	Description
n	Display number	There follows a task description for this display. The display number must match the display number on the OP. Display numbers from 1011 to 1064 are reserved for the standard displays. If no further display description follows, the data word contains W#16#FFFF.
m	Number of subtasks	This parameter defines the number of subtasks for the relevant "block". Multiple subtasks must be defined contiguously. If "number of subtasks" contains the value "zero", the next block is defined immediately instead of a subsequent task definition.
m+2	Main task number	Main task number
m+4	Absolute axis number	Absolute axis number
m+6	Element pointer	Points to the nth element of a data type, e.g. to the 5th DWORD of 10 DWORDs.
m+8	Data type	2 = WORD 3 = DINT 5 = DWORD 6 = STRUCT
m+10	Number of elements	Describes the number of elements immediately after the DB number and DBW number parameters.
m+12	DB no. GMC_DB_APP2	DB number and DBW number GMC_DB_APP2 describe the source area for "write data" or the destination area for "read data" on the CPU.
m+14	DBW no. GMC_DB_APP2	
m+16	User data	The data defined by the "number of elements" parameter are stored here.
k	Total number of user input tasks	This parameter only needs to be filled in for the user input tasks block. Appropriate range: 0 <= number of user input tasks <= 8. If this parameter has the value "zero", the next display definition follows immediately, instead of a task definition.

Table 6-2 Parameters of a Task

DBW	Meaning	Description
n	Display number	Task description for this display
n+2	Number of subtasks, e.g. 2	Tasks for display selection
n+4 n+6 n+8 n+10 n+12 n+14 n+16	Main task number Absolute axis number Element pointer Data type Number of elements DB no. GMC_DB_APP2 DBW no. GMC_DB_APP2	1st task
n+18 n+20 : n+30	Main task number Absolute axis number : DBW no. GMC_DB_APP2	2nd task
n+32	No. of subtasks, e.g. 1	Cyclic tasks
n+34 n+36 n+38 n+40 n+42 n+44 n+46	Main task number Absolute axis number Element pointer Data type Number of elements DB no. GMC_DB_APP2 DBW no. GMC_DB_APP2	1st task
n+48	Total number operator tasks, e.g. 2	User input tasks
n+50	Number of subtasks equals 1	
n+52 n+54 n+56 n+58 n+60 n+62 n+64	Main task number Absolute axis number Element pointer Data type Number of elements DB no. GMC_DB_APP2 DBW no. GMC_DB_APP2	1st user input task
n+66	Number of subtasks equals 1	
n+68 n+70 : n+80	Main task number Absolute axis number : DBW no. GMC_DB_APP2	2nd user input task
n+82	Next display number	Task description for next display
n+84	Number of subtasks	Tasks for display selection
:		
m	Next display number = W#16#FFFF	End of task description

Table 6-3 Structure of a Task

NOTE

If the value of parameter "number of subtasks" or "total number of user input tasks" is "zero", there is no task description for this block. In this case, the next block or the next display is then automatically interpreted.

6.1.3 Data Block GMC_DB_PICT_POINTER (DB120)

Task

The GMC_FB_START function block sets this data block up for the correct amount of standard displays when the CPU is powered up. It ensures a constant processing time when displays are selected, because a pointer is calculated and stored for each task block during an initialization phase. This pointer is used exclusively for subsequent accesses (at a later stage in the cycle).

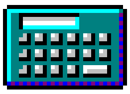
The standard displays are assigned to the number range from 1011 to 1064. Display numbers 1065 to 1096 are provided in case the GMC-OP-OAM standard software is also required to support user displays. Higher and lower display numbers are no longer supported.

If a display does not exist within the number range specified in the GMC_DB_ORDER data block, a pointer cannot be calculated for this display. In this case, the data area is reserved.

NOTE

If user display support is required, you must ensure that the GMC_DB_PICT_POINTER data block is the correct length.

In this case, you should create the GMC_DB_PICT_POINTER data block yourself and initialize it completely with zeroes. The DB which you have created is available in the project and must be loaded into the CPU.



EQUATION

The actual length is calculated according to the following equation:
Length in bytes = [(last supported user display number - 1010) * 6] + 2

Example

The number of the supplied user display (see Section 6.2) is 1065.
This produces a length of [(1065 – 1010) * 6] + 2 = 332 bytes

Adresse	Name	Typ	Anfangsw	Kommentar
0.0		STRUCT		
+0.0	highest_picture_no	INT	0	Highest picture nummer
+2.0	order_pointer	ARRAY[1..165]	165 (0)	Order pointer
*2.0		INT		
=332.0		END_STRUCT		

6.2 Integration Using Supplied User Display "Control Axis"

Introduction

The following procedure is recommended for the integration of your supported user display:

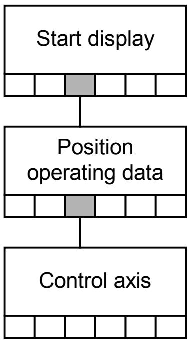
- Use ProTool to copy a similar display into your project and change this display to meet your needs.
- Then allocate a display number in the range from 1065 to 1096.
- Define the destination area for your variables in the CPU.
- Set up the destination area for the OP data in your STEP 7 project. You may require further static local variables which are also stored in this data block.
- If you have only configured tasks for display selection and/or cyclic tasks for one axis in your display, you only need to insert the task definitions in data block GMC_DB_ORDER before the end criterion W#16#FFFF.
- If you want to transfer user inputs from your display to the technology, or if you select a new axis, you need to provide the appropriate support in your user program in the form of an S7 function or an S7 function block.

6.2.1 Integration from the Perspective of the OP

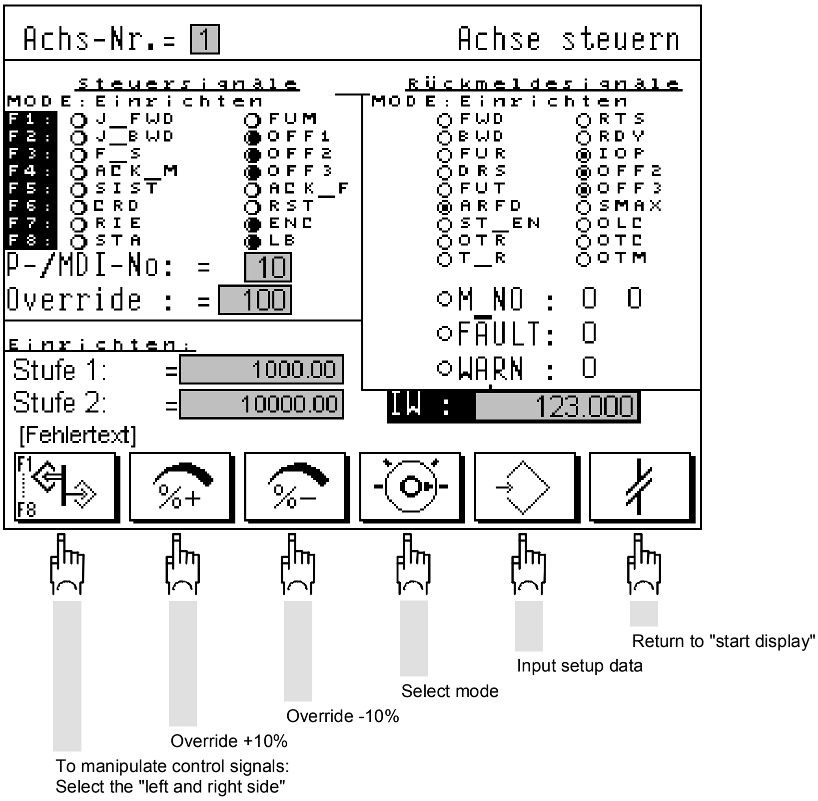
Display 1065 "control axis" is described here as an example for the OP25.

Function key F11 is used to select the "control axis" example display from the GMC start display (display number 1000). In order to allow the integration of several “control axes” without problems, the operating data mask is changed initially, with the display being selected subsequently.

The "control axis" example display is reached via the menu tree shown.



"Control axis" display



This display provides you with convenient functions for manipulating the control signals of an axis, monitoring the checkback signals, selecting the mode, selecting a program number and setting the override.

When you select the display, the velocities are read once for setup. The levels can be changed if necessary and transferred again to the technology by activating function key F13 "input setup data". The actual values of the axis are updated cyclically.

The OP data are stored in data block DB9 UAB_DB_CONTROL.

NOTE

You can only integrate your plant displays outside the menu tree of GMC-OP-OAM, e.g. from the start display.

6.2.2 Integration from the Perspective of the S7-CPU

Open your project and follow the instructions below.

Step 1: Define the User Display DB

The first step is to define the data block which is to contain the data you want to visualize on the OP screen. The data for the supported user display supplied with the software are stored in data block DB9 UAB_DB_CONTROL.

When you have created the user display DB, you need to define the user tasks in data block GMC_DB_ORDER by following the instructions in the steps below.

Step 2: Define the Screen Selection Tasks

All tasks that you configure in data block GMC_DB_ORDER are entered directly after the tasks of the GMC-OP-OAM standard software (before the data block end criterion W#16#FFFF).

Refer to your display to check whether you need to transmit a task to the technology for the data to be visualized on display selection:

- ♦ If not, proceed to step 3.
- ♦ If so, you now need to configure the associated task in data block GMC_DB_ORDER.

The "output setup velocities" display selection task (main task no. 17) is configured for our display 1065 "control axis".

972.0	uab1_pict_no	INT	1065	1065	first user picture number
974.0	a_uab1_jobs	INT	1	1	
976.0	a_uab1_ha	INT	-17	-17	
978.0	a_uab1_ua	INT	0	0	
980.0	a_uab1_element_pointer	INT	1	1	
982.0	a_uab1_data_type	INT	5	5	
984.0	a_uab1_no_elements	INT	2	2	
986.0	a_uab1_db_no	INT	9	9	
988.0	a_uab1_dbw_no	INT	60	60	

Step 3:
Define the Cyclic Tasks

Check the display again to see whether you need to transmit a cyclic task to the technology for the data to be visualized.

- If not, proceed to step 4.
- If so, you now need to configure the associated task in data block GMC_DB_ORDER.

The "read actual value of selected axis" cyclic task (main task no. 3) is configured for our display 1065 "control axis".

990.0	z_uab1_jobs	INT	1	1	
992.0	z_uab1_ha	INT	3	3	
994.0	z_uab1_ua	INT	0	0	
996.0	z_uab1_element_pointer	INT	1	1	
998.0	z_uab1_data_type	INT	2	2	
1000.0	z_uab1_no_elements	INT	5	5	
1002.0	z_uab1_db_no	INT	9	9	
1004.0	z_uab1_dbw_no	INT	88	88	

Step 4:
Define the User Input Tasks

Check the display again to see whether you need to transmit a user input task to the technology for the data to be visualized.

- If not, proceed to step 5.
- If so, you now need to configure the associated task in data block GMC_DB_ORDER.

Two user input tasks are configured for our display 1065 "control axis":

- "Input setup velocities" via F13 (main task no. 17), and
- The "output setup velocities" user input task (main task no. 17) is initiated after the input of a new axis.

In this case, the number of tasks (variable b_uab1_alljobs) is two.

NOTE

Please don't forget to terminate the block with the end criterion W#16#FFFF!

1006.0	b_uab1_alljobs	INT	2	2	
1008.0	b_uab1_jobs	INT	1	1	
1010.0	b_uab1_ha	INT	17	17	
1012.0	b_uab1_ua	INT	0	0	
1014.0	b_uab1_element_pointer	INT	1	1	
1016.0	b_uab1_data_type	INT	5	5	
1018.0	b_uab1_no_elements	INT	2	2	
1020.0	b_uab1_db_no	INT	9	9	
1022.0	b_uab1_dbw_no	INT	60	60	
1024.0	b1_uab1_jobs	INT	1	1	
1026.0	b1_uab1_ha	INT	-17	-17	
1028.0	b1_uab1_ua	INT	0	0	
1030.0	b1_uab1_element_pointer	INT	1	1	
1032.0	b1_uab1_data_type	INT	5	5	
1034.0	b1_uab1_no_elements	INT	2	2	
1036.0	b1_uab1_db_no	INT	9	9	
1038.0	b1_uab1_dbw_no	INT	60	60	
1040.0	w_ende	WORD	W#16#FFFF	W#16#FFFF	end definition

Step 5

You have now completed the first stage!

No task definitions are possible for the control signals and checkback signals, because these signals access the corresponding section of the selected axis directly in data block GMC_DB_CMD.

Step 6

You now need to check whether your display requires further support in the form of an S7 function or function block. This is the case if you use user input tasks directly or if you configure operator actions on the OP which lead indirectly to a user input task.

If you do not require further program support in the S7-CPU, proceed to step 9.

Program support is necessary in the form of S7 function FC9 UAB_FC_CONTROL. This function supports the control and checkback signals of all configured axes, input and output of the setup data and read-out of the actual value of the selected axis. When the user input task is initiated, the function formulates the associated task and enters it in the parameter interface of the GMC-OP-OAM standard software.

The parameter interface is stored in the instance data block of FB GMC_FB_PICTMAN, DB 122 GMC_IDB_PICTMAN_DATA.

NOTE

Please remember that you can only access the instance data block symbolically!

The relevant parameters are:

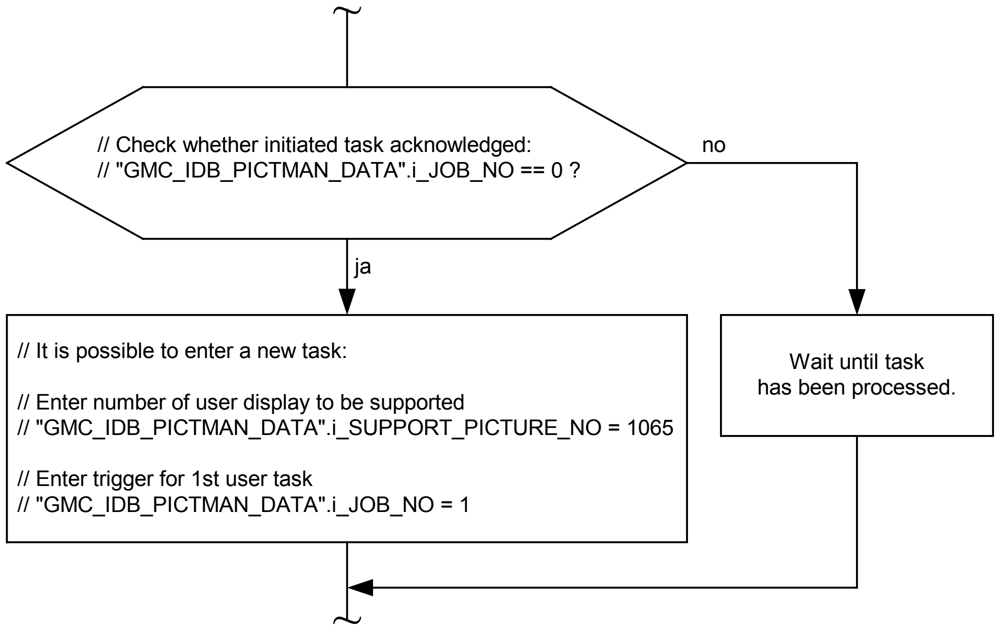
- The user display number "GMC_IDB_PICTMAN_DATA".i_SUPPORT_PICTURE_NO and
- The task number "GMC_IDB_PICTMAN_DATA".i_JOB_NO.

Adres	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	GMC_DB_ORG	BLOCK_DB	DB 1	DB 100	DB_ORG number
2.0	GMC_FB_PICTDATA	BLOCK_FB	FB 0	FB 122	GMC_FB_PICTDATA number
4.0	ERROR	WORD	W#16#0	W#16#0	Errornumber
6.0	STATUS	WORD	W#16#0	W#16#0	Statusbits
8.0	i_SUPPORT_PICTURE_NO	INT	0	1065	User picture number
10.0	i_JOB_NO	INT	0	1	Job number

Transmission of the task to the technology is initiated by entering the user display number and a task number in data block GMC_IDB_PICTMAN_DATA. Only one task number can be entered, however you can configure several user tasks under the same task number. These tasks are processed sequentially (see the structure of data block GMC_DB_ORDER).

Function block GMC_FB_PICTMAN detects the user task and attempts to process it. A task is processed when the task number has a value of zero again (acknowledgement for the user program).

You must program the coordination as follows in your user program:



Step 7

Complete the symbol table in your project. For the example supplied, this requires the entry of data block DB9 and function FC9.

UAB_DB_CONTROL	DB 9	DB 9	DB for supported user display
UAB_FC_CONTROL	FC 9	FC 9	sample user block for axis control

Step 8

Now include the function or function block call in the user program. The function which provides program support for the "supported user display" in the CPU is first described in detail in the section immediately below.

The UAB_FC_CONTROL function only supports display 1065, “control axis” (display name in ProTool: uab_steu_ein).
If the function detects the input of a new axis and/or the initiation of setup data transfer, it triggers a new task in data block GMC_IDB_PICTMAN_DATA.

The function interacts with the following data blocks:

- UAB_DB_CONTROL (local data of function, display area)
- GMC_DB_CMD (control and checkback signals of axes)
- GMC_DB_ORDER, (description of tasks for OAM project)
- GMC_IDB_PICTMAN_DATA, (instance DB of GMC_FB_PICTMAN **and** GMC_FB_PICTDATA, task interface)

The STATUS block parameter indicates whether or not the block is active. The UAB_FC_CONTROL function must be called up and supplied with parameters once during cyclic operation (e.g. in OB1). A second call in another OB level, e.g. time-driven level OB35 is not permitted. The FC number can be chosen freely.

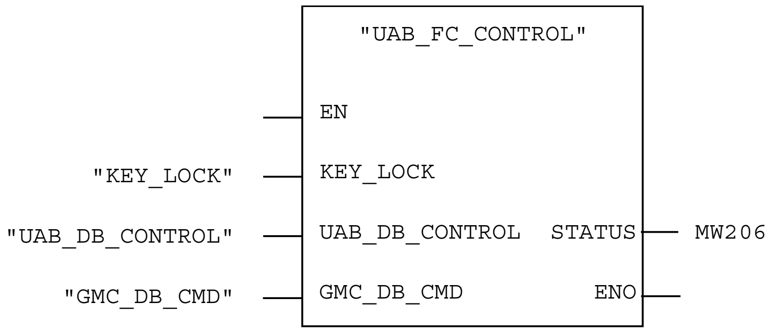
NOTE

The supplied “supported user display“ allows individual control of your axes. The associated function FC UAB_FC_CONTROL is equipped with a key switch function, parameter KEY_LOCK, in order to prevent misuse. The parameter KEY_LOCK is shown on the flag M0.7. Therefore, if you wish to control your axes, the KEY_LOCK flag must be set to signal state "1". In the simplest case, this could be an external key switch, connected via a digital port.

WARNING

If the display is deselected, your axis will continue to run in the state which had been selected last!

Block



Explanation of Input and Output Parameters

The table below describes the input and output parameters of the "UAB_FC_CONTROL" function.

Name	Type	Data Type	Description	Default
KEY_LOCK	IN	BOOL	Interlock for operation of selected axis 0 = Control not possible 1 = Control possible e.g. bit memory	M 0.7
UAB_DB _CONTROL	IN	BLOCK_DB	CPU-specific (zero is not allowed), CPU315-2DP: DB1-DB127 CPU413-2DP: DB1-DB511 CPU414-2DP: DB1-DB1023	DB9
GMC_DB_CMD	IN	BLOCK_DB	See above	DB117
STATUS	OUT	WORD	Status signal of function 0 => Function is not active 1 => Function is active MW, DB.DBW	MW206

Table 6-4

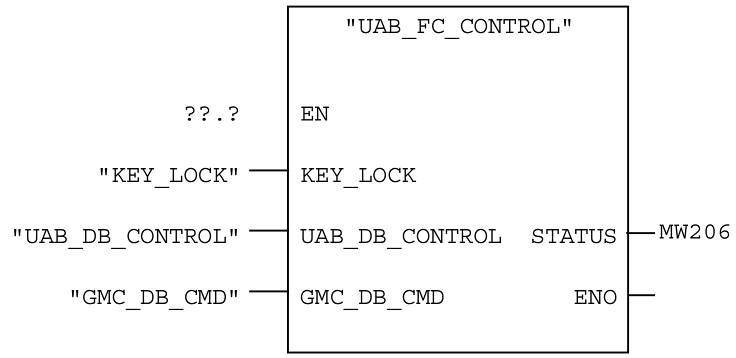
Description of Block Parameters

The function enables the control interface via the KEY_LOCK parameter. If the parameter signal is "0", it is not possible to control the axis. The UAB_DB_CONTROL parameter is used by the function to identify the destination area for its own static local data and the destination area for the variables on the OP. The GMC_DB_CMD parameter indicates where the control and checkback bits of the axes are stored. The STATUS output parameter indicates whether or not the function is currently active.

In the example program supplied with the software, the FC9 UAB_FC_CONTROL function call is in function FC4 FC_CALL_OAM_OP25, network 2. Function block GMC_FB_PICTMAN is called in network 1.

FC4 :

Network 2 : Call UAB_FC_CONTROL



Symbol information:

M 0.7	KEY_LOCK	parameter KEY_LOCK of function UAB_FC_CONTROL
DB9	UAB_DB_CONTROL	DB for supported user display
DB117	GMC_DB_CMD	extended control/checkback signals
MW206	STATUS	MW206

Step 9:
Load the Project in the CPU

When you have successfully completed this step, you are finished.

Now load your project in the CPU and test your function(s).

NOTE

If there is an overlap between the block numbers of the GMC-OP-OAM standard software and your own block numbers, you should consider renaming your own blocks. If this is not possible, simply rename blocks DB9 and FC9 in the example program, and change the block call manually. Then, open the OP project again and reconfigure all of the variables used in "supported user display uab_steu_ein" for the new destination area.

6.2.3 Other User Displays Supported by the Standard

General

With GMC-OP-OAM version V2.0 and later, supported user displays are provided for the "Real Master", the "Master Value Correction" and the "Master Value Correction Status" (see also menu tree V2.0, Chapter 8).

These displays are integrated analogously to the "Control Axis" display (see description in previous chapter).

Description of functions

For the sake of clarity, a separate function and a separate data block for operator control and monitoring have been implemented in the S7-CPU.

The following generally applies:
If the functions detect the input of a new axis and/or the initiation of the transfer of new parameters, then a new task is activated accordingly in data block GMC_IDB_PICTMAN_DATA.

The functions work with the following data blocks:

- UAB_DB_xxx (local function data, display area)
- GMC_DB_ORDER, (description of tasks for OAM project)
- GMC_IDB_PICTMAN_DATA, (instance DB of GMC_FB_PICTMAN **and** GMC_FB_PICTDATA, task interface),

Block parameter STATUS indicates whether or not the block is active. The UAB_FC_xxx functions must be called once and parameterized in cyclic mode (e.g. OB1). It is not permissible to call them a second time on another OB level, e.g. time-controlled level OB35.

For an indication of how the functions are represented and an explanation of their parameters, please refer to function "UAB_FC_CONTROL" described above.

In the sample program supplied, the functions are called in networks three, four and five of function FC4 FC_CALL_OAM_OP25.

NOTE

You can choose the number of the function(s) freely. The same basically applies to the data block number. To do so, just change the block call in your S7 program. Then open the OP project again and reconfigure all variables used in the "supported user display" to the new target range.

6.2.4 Hiding Supported User Displays

Procedure

Proceed as follows to hide supported user displays:

Delete from your S7 project the call(s) of the function(s) for the display(s) that you wish to hide. When you restart the CPU again, you will no longer be able to select the display(s) you have hidden on the OP.
If you want to go a step further, you can use the ProTool configuring software to delete the display call, any associated variables and finally the supported user display itself, from the appropriate function key(s).

All displays stored under the "deleted" menu tree branch will then be hidden.

6.2.5 Technical Data of Functions for Supported User Displays

Memory requirement

The block lengths are specified in number of bytes.

Display (Display No.)	**Control Axis (1065)**	**Real Master (1068)**	**Master Value Correction 1 (1069) and Master Value Correc-tion Status (1071)**	**Master Value Correction 2 (1070)**
Display name	*uab_steu_ein*	*uab_RM*	*uab_Lwkorr1* *uab_Lwk_Sta*	*uab_Lwkorr2*
FC name	UAB_FC-_CONTROL	UAB_FC-_REAL_MA	UAB_FC-_MA_VAL_CORR	UAB_FC-_MA_VAL_CORR2
Creation language	AWL	AWL	AWL	AWL
Block length (no. of bytes)	1282	320	360	314
Length of MC7 code (no. of bytes)	1100	208	242	202
Local data	26	6	6	6
Nesting depth	-	-	-	-
Called system functions	-	-	-	-
Data area allocation	UAB_DB-_CONTROL=136 GMC_DB_ORDER =68 (proportionate)	UAB_DB-_REAL_MA=74 GMC_DB_ORDER =68 (proportionate)	UAB_DB-_MA_VAL_CORR =82 GMC_DB_ORDER =92 (proportionate)	UAB_DB-_MA_VAL_CORR2 =58 GMC_DB_ORDER =54 (proportionate)

Table 6-5

6.3 Deactivating Standard Displays

Procedure

It is possible to deactivate standard displays which you do not require. To do this, use the ProTool configuring tool to delete the display call and any associated variables from the assigned function key.

This allows you to deactivate a complete branch of a menu tree so that it can no longer be called up or processed. All of the displays below the "deleted" branch of the menu tree are deactivated.

6.4 Technical Specifications of the GMC-OP-OAM V2.0 Standard Software

Memory Requirements and Runtime

The block lengths are specified in the number of bytes. The GMC-OP-OAM standard software runs in your user program at the specified execution time only. The execution time is independent of the number of axes and the number of M7-FM modules installed.

FB Name	**FB_PICTMAN**	**FB_PICTDATA**
Programming language	STL	STL
Block length (number of bytes)	(3.616) 3.6 kb	(7.096) 7 kb
MC7 code length (number of bytes)	(2.248) 2.2 kb	(5.260) 5.3 kb
Local data	72	44
Nesting depth	1	-
System functions called	-	-
Data area used	IDB_PICTMAN_DATA = 914 DB_PICT_POINTER >= 458 DB_ORDER >= 1.812	
Execution time in ms • CPU 315-2DP • CPU 413-2DP • CPU 414-2DP • CPU 416-2DP	 ≤ 1.,2 < 0.16 < 0.08 < 0.04	Runtime contained in FB_PICTMAN!

Table 6-6

Overview of Estimated Memory Requirements

The memory used by the GMC-OP-OAM package is independent of the number of axes and the number of M7-FM modules.

	GMC-OP-OAM
FB_PICTMAN	2248
FB_PICTDATA	5260
Total FB:	7508
DB_ORDER	1812
DB_PICT_POINTER	458
Instance DB	
GMC_IDB_PICTMAN_DATA	914
Total DB:	3184
Overall total:	**10692**

7 General Information for Operator Control

Contents

In this chapter you will find general information about the "GMC-OP-OAM" Standard Software.

7.1 SIMATIC OP25/OP27 Human-Machine Interface

Operator Control of GMC Functions

The "GMC-OP-OAM" standard software uses the SIMATIC OP25/27 HMI (human-machine interface). This is a convenient low-cost system with a standard user interface for controlling and monitoring the GMC functions.

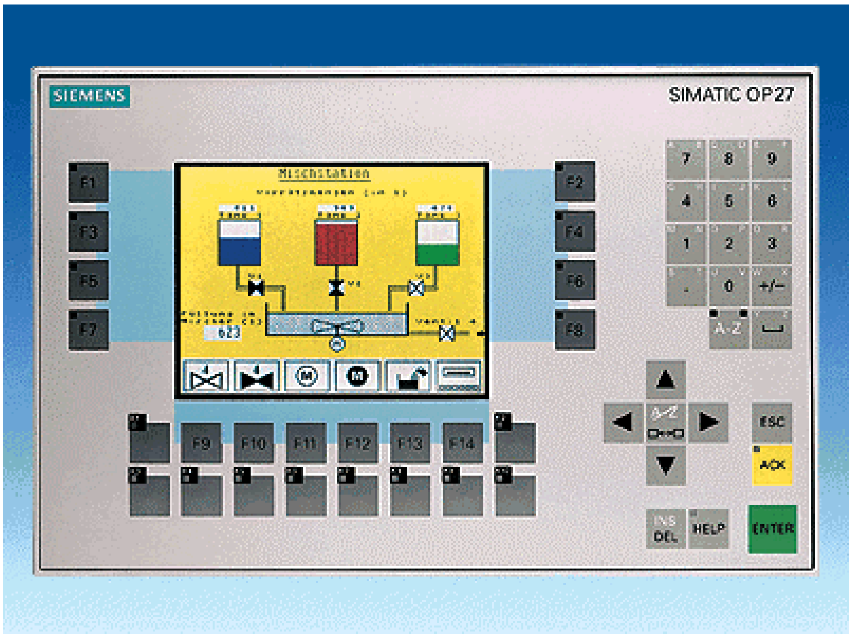
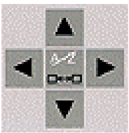
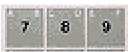
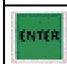


Fig. 7-1 SIMATIC OP 27 Graphics- Based Operator Panel

Input

Keys	Function
	Use the cursor control keys to position the cursor on an input field.
 ...	Enter a value with the numeric keypad.
	Confirm your entry with the Enter key.

Cancel Input

Keys	Function
	You can use the ESC key to cancel an entry you have already started.

Delete Input

Keys	Function
	You can delete an entry with the Delete key.

Info Key

Keys	Function
	In various displays, the Info key calls up additional information explaining the procedure for data input. When information is available, the LED in the Info key is illuminated.

7.2 Display Layout/Description

Display Layout

The following display layout is used in this documentation for an enhanced overview:

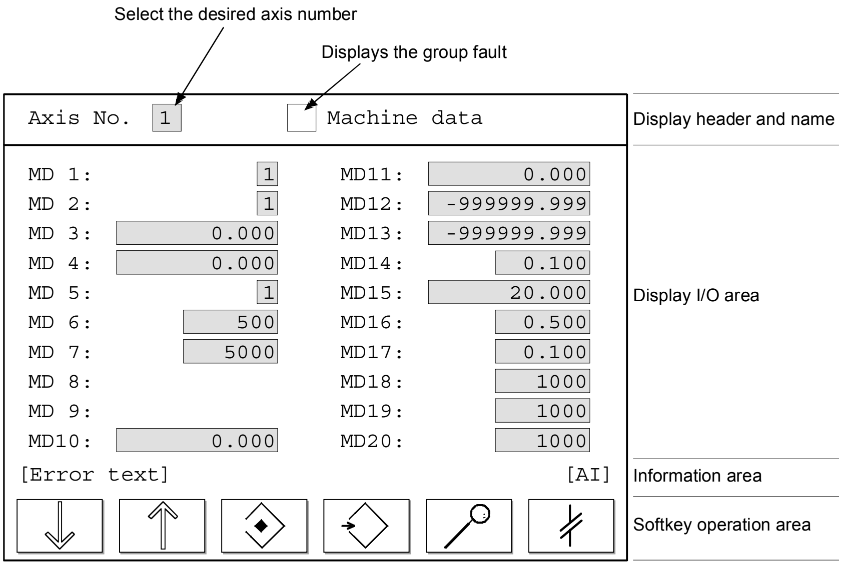


Fig. 7-2 Layout of Screen Forms

Input fields are highlighted in gray. [Error text] indicates an event message, a machining status or an error message. In various displays, additional information [AI] is output with an error message in the form of a number. The meaning depends on the error message.

When you select a display, the data are output from the selected axis. If you select a new axis number, the data are output direct from the selected axis in order to update the display.

If one of the axes in the total system outputs a fault [FAULT], a "lightning strike" indicates a group fault. You can call up the diagnostics display by activating the diagnostics softkey, in order to find out the axis in which the fault occurred.

All displays which allow input are protected by password. Different password levels (protection levels) are possible.

Individual passwords can be defined in the "passwords" system display. The master password is configured in ProTool.

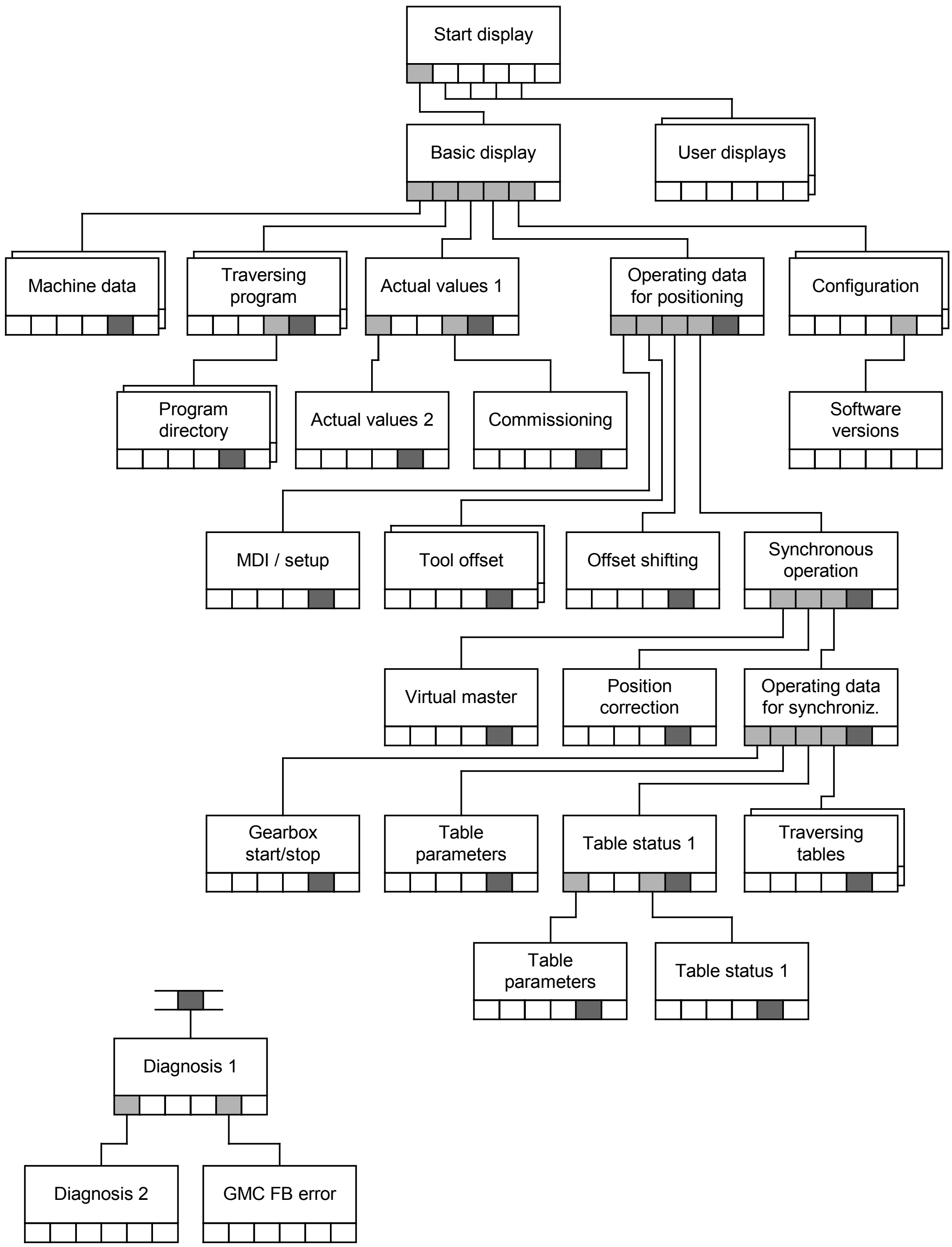
8 Screen Forms

Contents

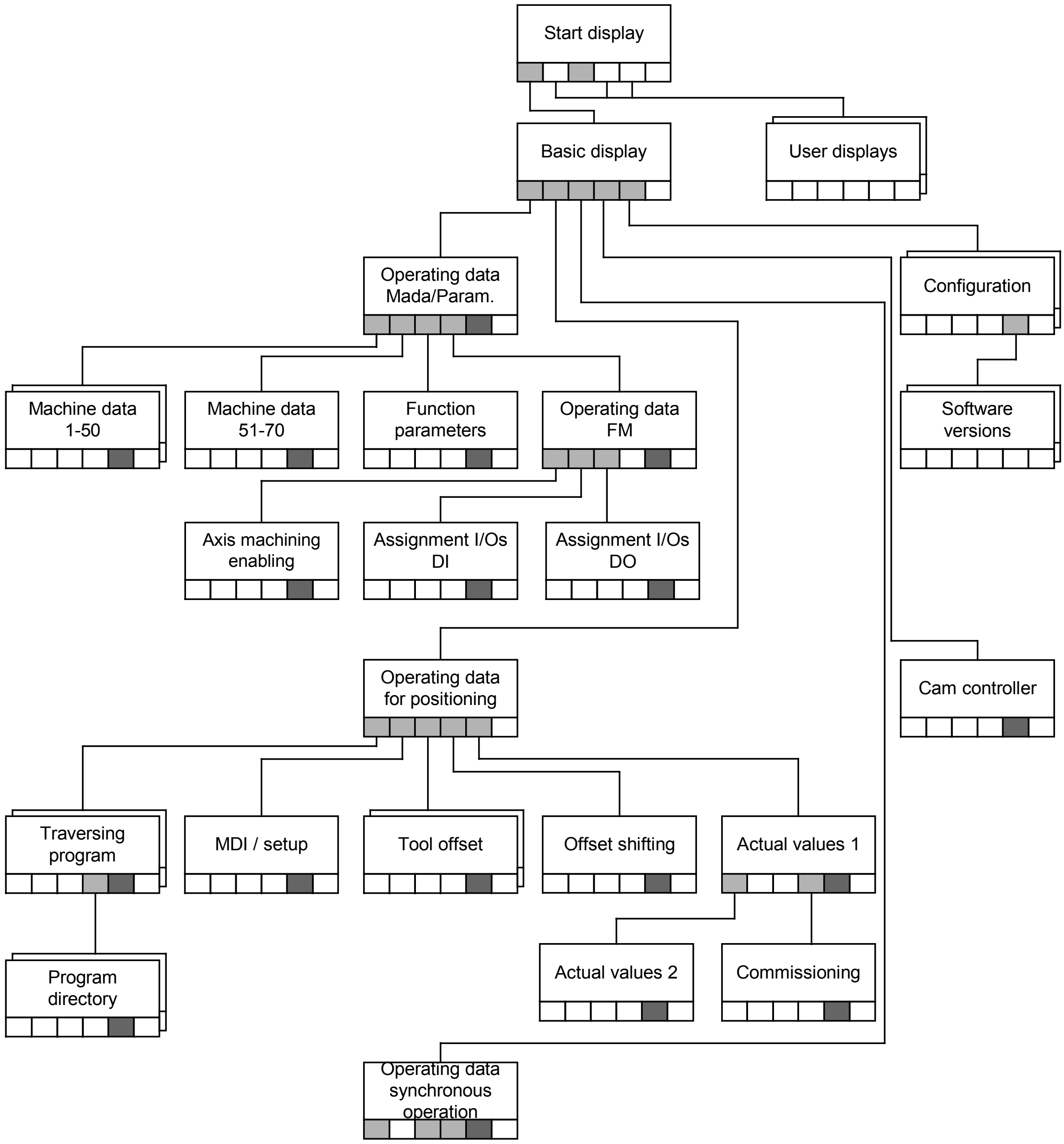
In this chapter you will find all the information about the screen forms of the "GMC-OP-OAM" standard software.

8.1 Menu Tree

Version V1.0

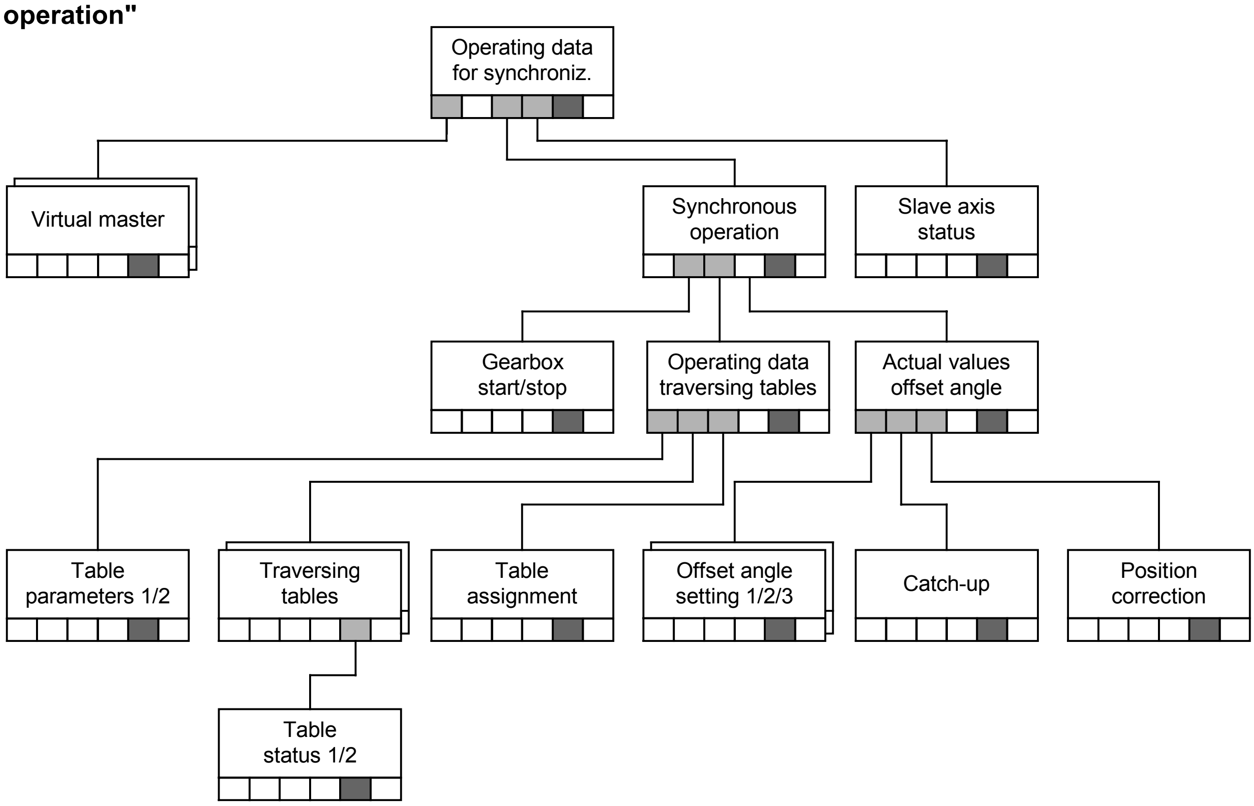


Version V2.0

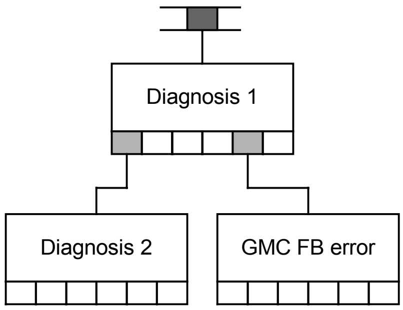


See next page

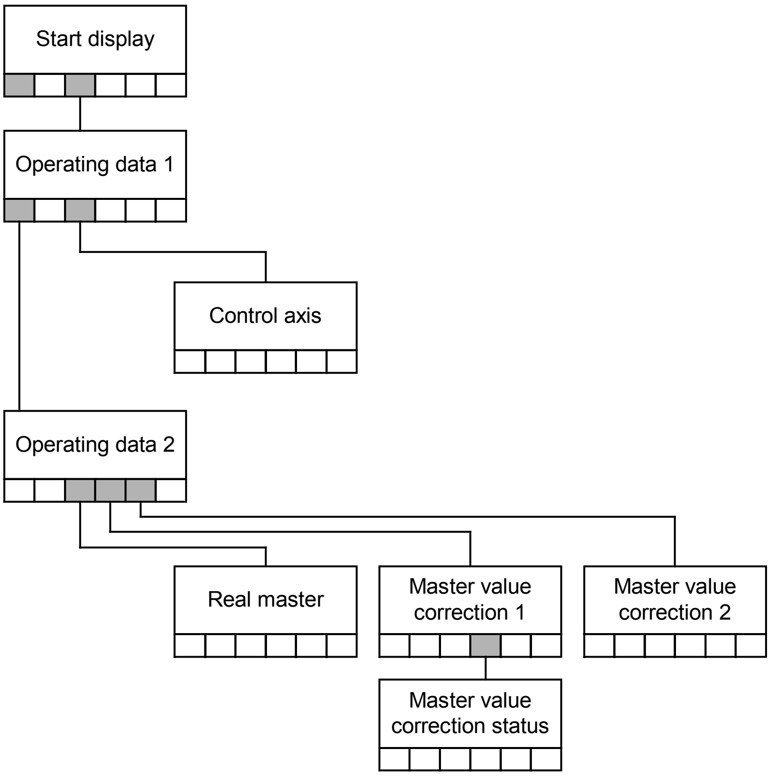
Continued "Operating data Synchronous operation"



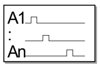
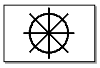
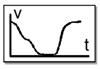
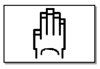
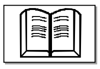
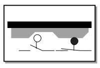
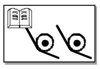
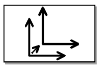
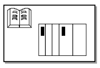
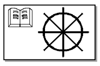
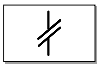
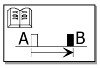
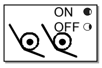
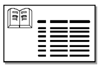
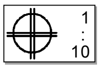
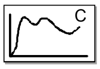
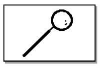
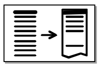
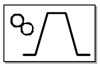
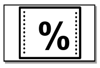
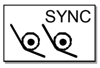
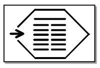
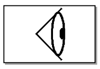
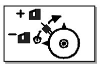
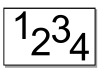
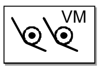
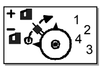
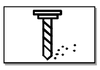
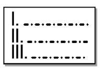
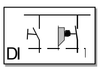
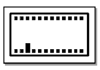
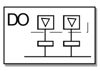
Diagnosis selection



Selection of supported user displays

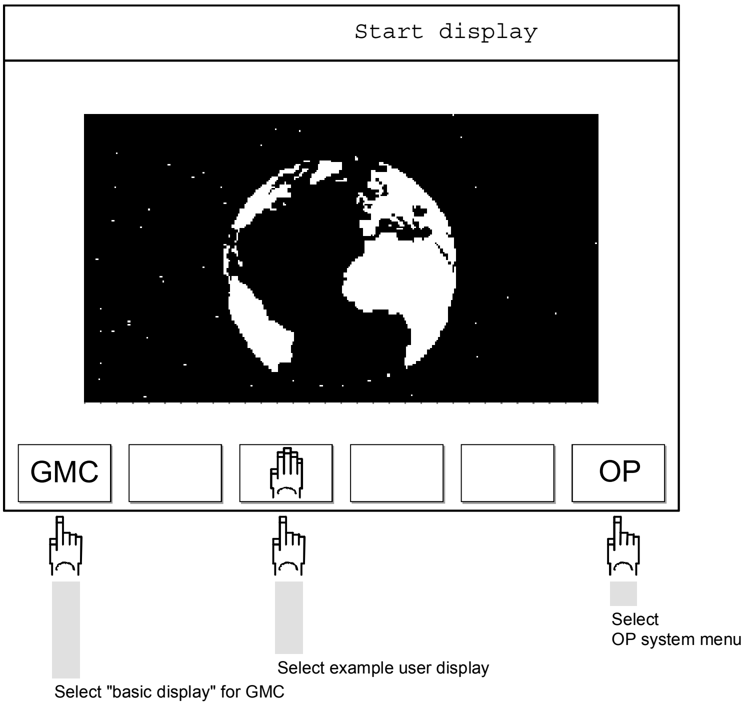


8.2 Icons Used for Display Selection

Icon	Select Display for...	Icon	Select Display for...
	Axis machining enabling		Machine data
	Catch-up		MDI/setup
	Operating data		Cam controller
	Operating data Synchronous operation		Offset shifting
	Operating data FM		Program directory
	Operating data Machine data / Parameters		Start display
	Operating data Positioning		Slave axis status
	Operating data Traversing tables		Table parameters
	Function parameters		Table status
	Diagnosis		Table assignment
	Gearbox start/stop		Traversing programs
	Synchronous operation		Traversing tables
	Commissioning		Offset angle setting
	Actual values		Virtual master
	Actual offset angle values		Tool offset
	Configuration or software versions		Assignment I/Os DI
	Position correction		Assignment I/Os DO

8.3 Start Display

Screen Form

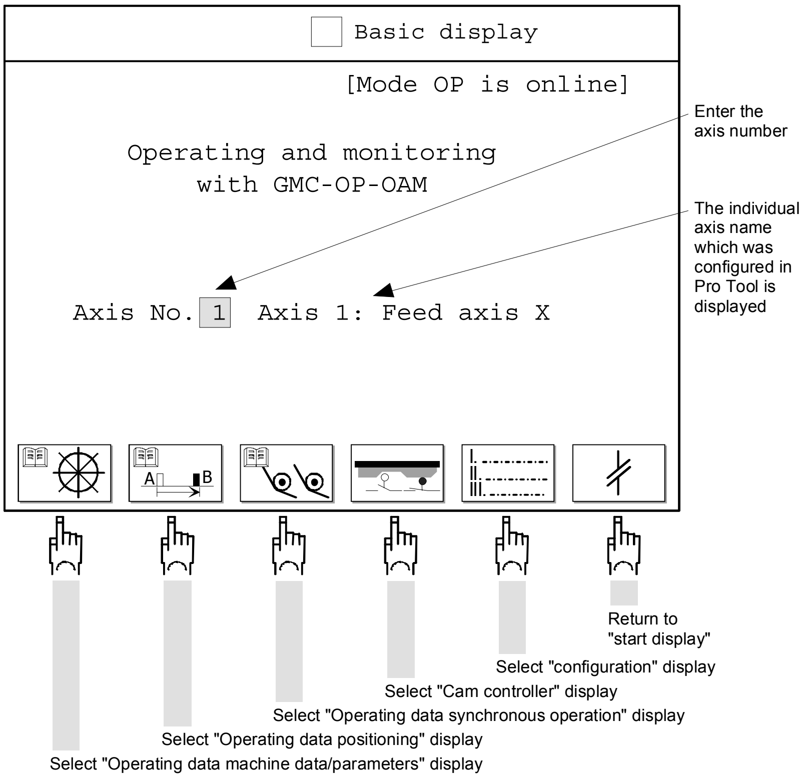


Description

The "start display" is an entry point to the various menu trees. The GMC softkey takes you to the standard menu tree of GMC.

8.4 Basic Display

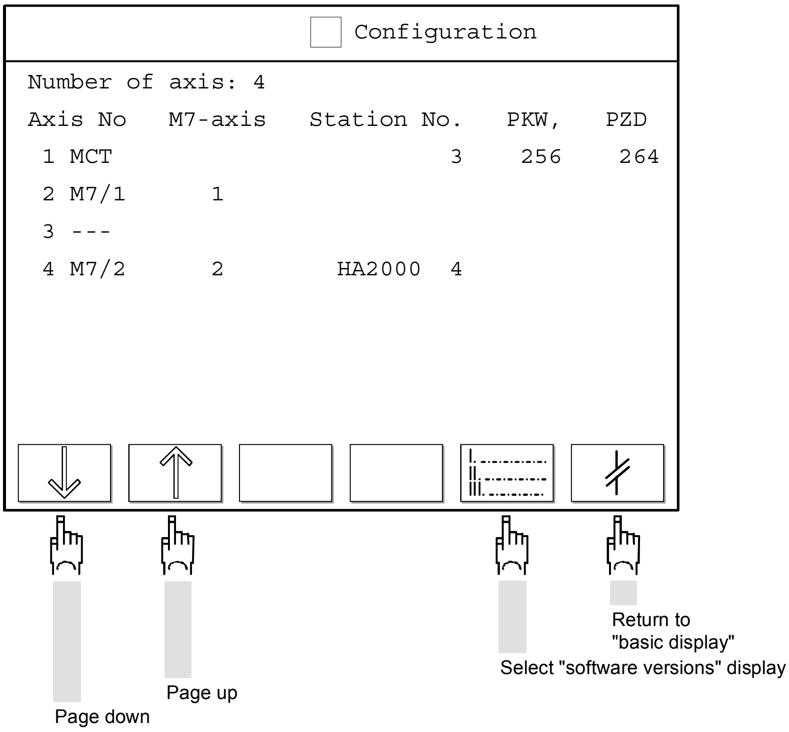
Screen Form



Description The "basic display" is at the top of the menu tree for GMC.

8.5 Configuration

Screen Form



Description

The "configuration" display shows a list of all axes in the complete system. The following parameters are shown for each axis:

- Axis No.:
  Global axis number and type of axis (MCT or M7).
  With M7, the M7 number is also shown (/1).
- M7 axis:
  Local axis number
- Station number:
  PROFIBUS address of the MCT
  HA2000 indicates that the axis is an M7 axis, and that a PROFIBUS connection to MASTERDRIVES MC is provided for communication with the basic functionality (HA2000).
- PKW, PZD:
  I/O addresses configured for the PKW and PZD components of the PROFIBUS protocol.

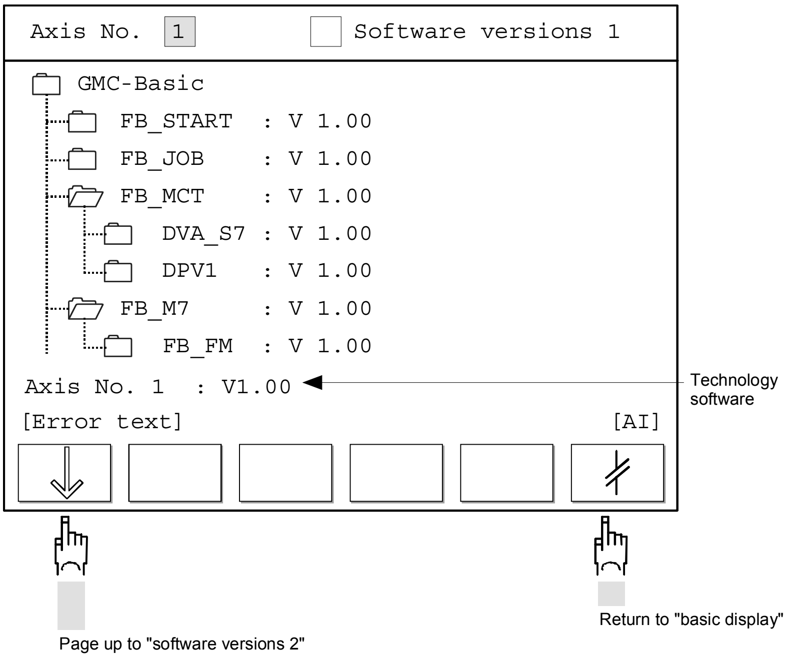
These parameters match the "axis descriptions" section in data block GMC_DB_ORG.

Password Protection

None

8.5.1 Software Versions 1

Screen Form



Description

The "software versions" displays show the software versions of the individual blocks and matches the "software versions" section in data block GMC_DB_ORG.

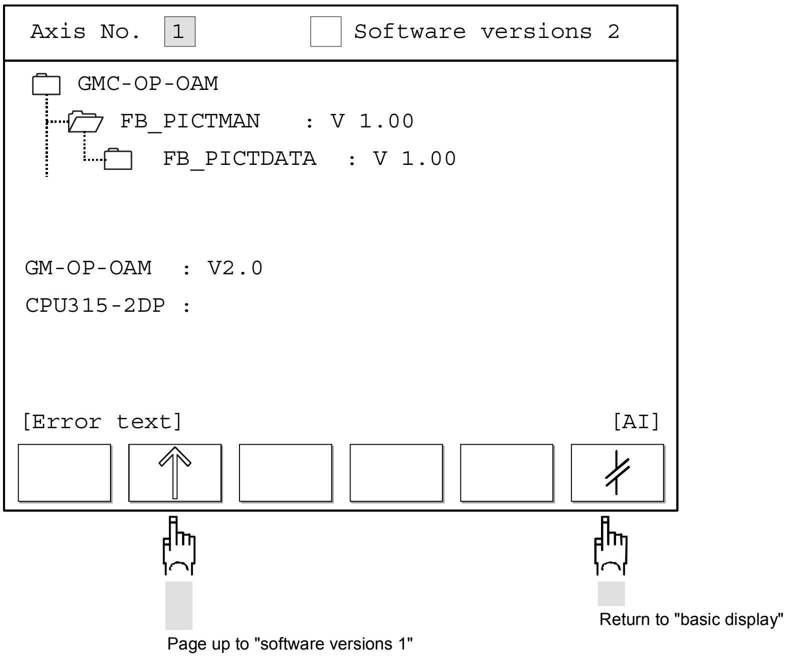
The "axis no." selection is only relevant with reference to the software version of the technology software.

Password Protection

None

8.5.2 Software versions 2

Screen Form



Description

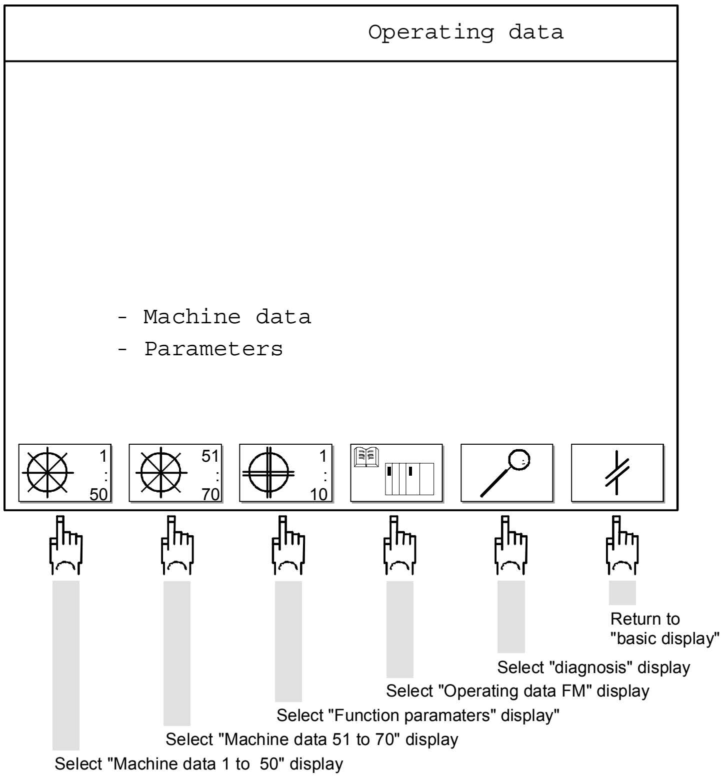
The "software versions" displays show the software versions of the individual blocks and matches the "software versions" section in data block GMC_DB_ORG.

Password Protection

None

8.6 Operating data for Machine data / Parameters

Screen Form



Description

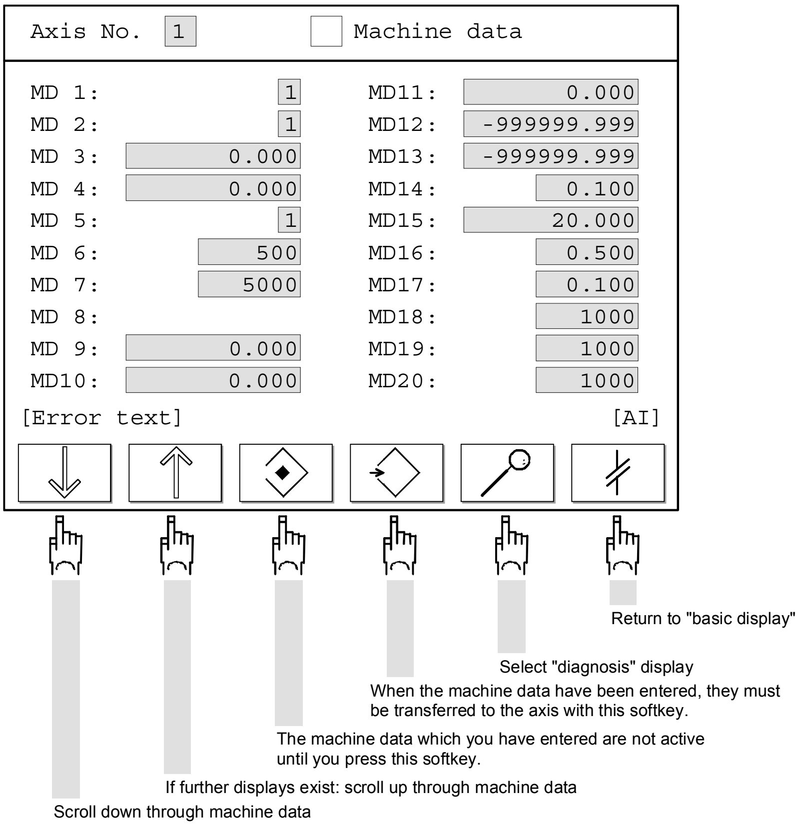
Intermediate display "Operating data for machine data / parameters" serves only as a distribution point to other displays.

Password Protection

None

8.6.1 Machine data

Screen Form



Description

The "machine data" display can be used to input or output the machine data of the technology. You can call up information on each machine data by pressing the "info" key.

NOTE

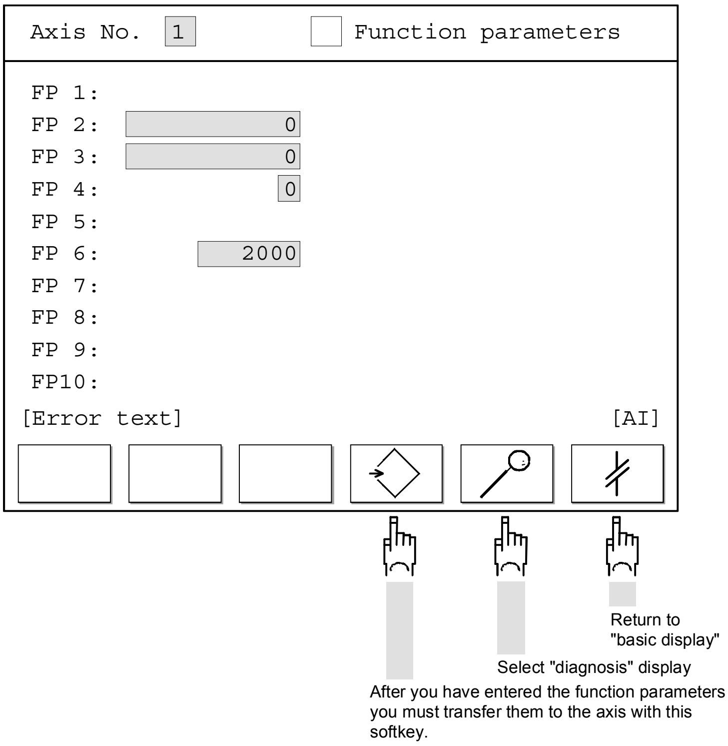
It is only possible to activate machine data when the axis is stationary.

Password Protection

Input is possible with level 3 or a higher-priority level.

8.6.2 Function parameters

Screen Form



Description

Function parameters can be input or output with the "Machine data" display. You can call information about each function parameter by selecting the Info key.

NOTE

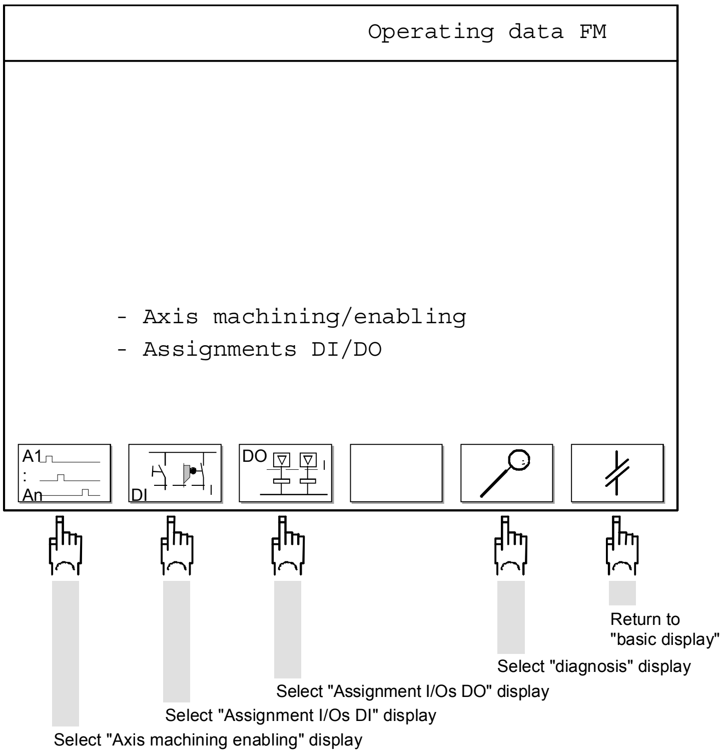
Function parameters are valid as soon as they have been transferred.

Password Protection

Input is possible with level 3 or a higher-priority level.

8.6.3 Operating data FM

Screen Form



Description

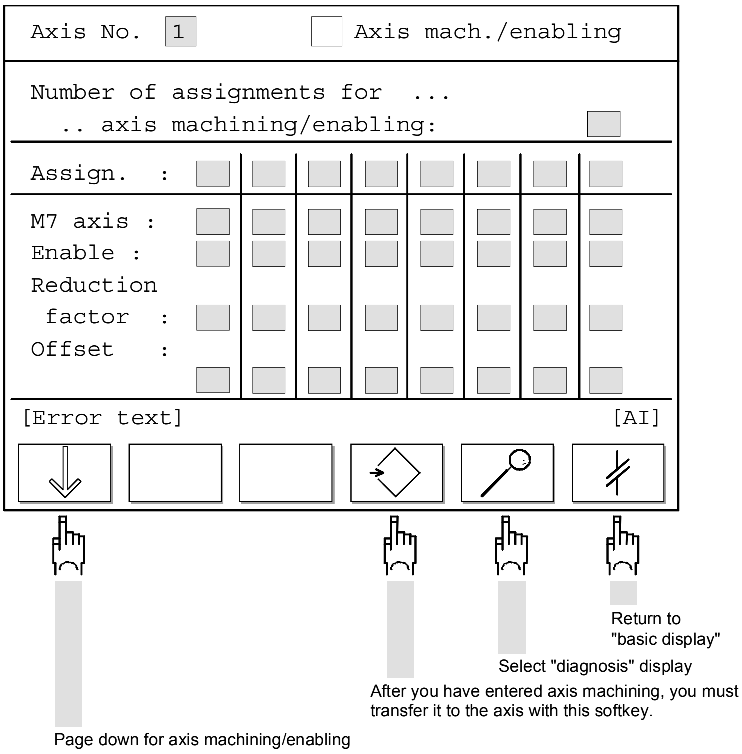
Intermediate display "Operating data FM" serves only as a distribution point to other displays.

Password Protection

None

8.6.4 Axis Machining Enabling

Screen Form



Description

Axes in a SIMATIC Motion Control system can be managed in display "Axis machining/enabling". The parameterization options provided enable you to adapt the application optimally to the requirements of the machine and to use the resources of the M7 computer in an optimum manner.

NOTE

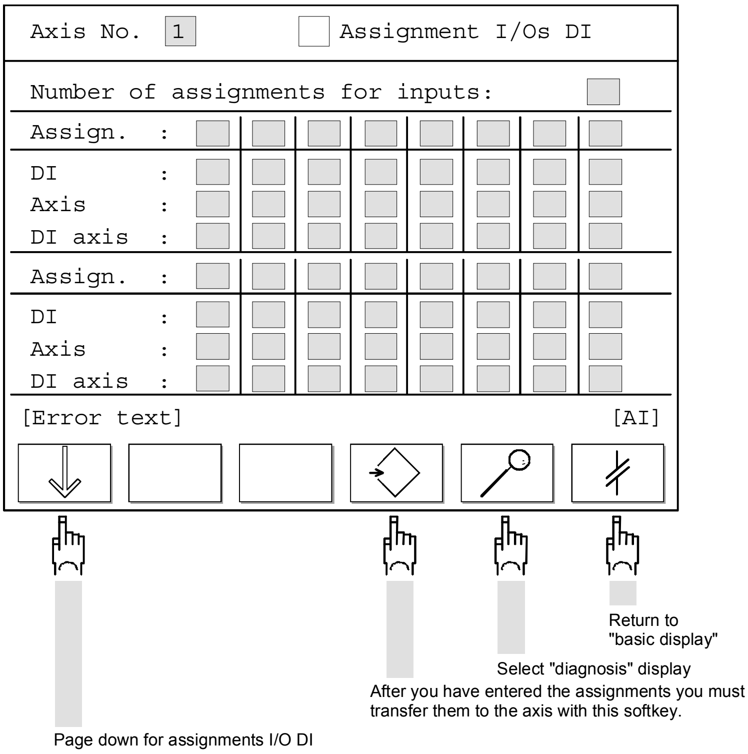
The axis management system is designed to be parameterized once by the S7 after system startup. None of the axes may be enabled for machining at this particular time.

Password Protection

Input is possible with level 3 or a higher-priority level.

8.6.5 Assignment I/Os DI

Screen Form



Description

You can define the general assignment between input data and an internal process image in display "Assignment I/Os DI".

NOTE

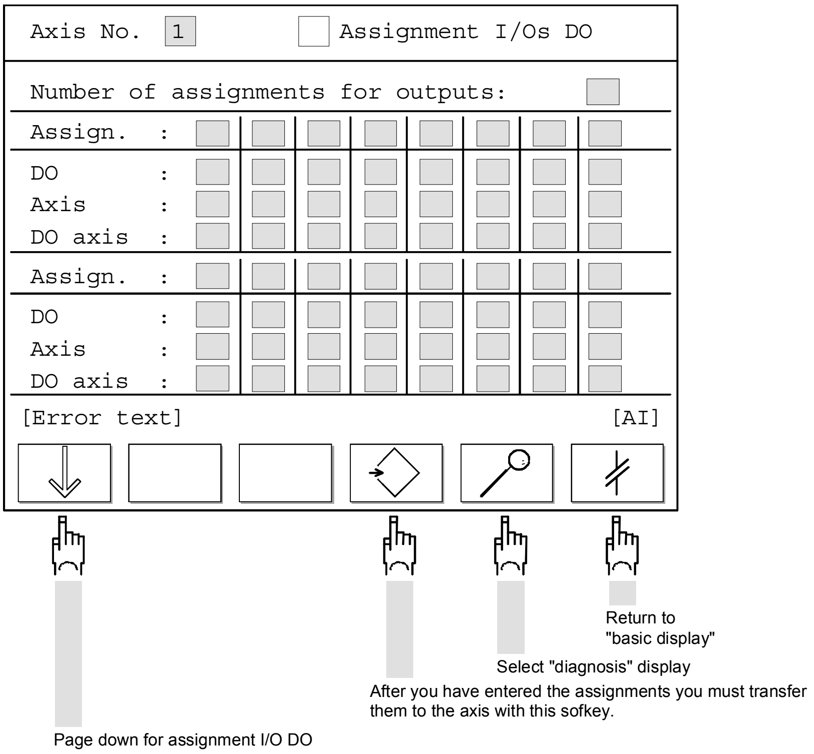
Please note that changing parameter settings when the system is running is equivalent to modifying the hardware configuration. Disconnected inputs and outputs can "freeze" at their last signal state. You can remedy this situation by restarting the system.

Password Protection

Input is possible with level 3 or a higher-priority level.

8.6.6 Assignment I/Os DO

Screen Form



Description

You can define the general assignment between output data and an internal process image in display "Assignment I/Os DO".

NOTE

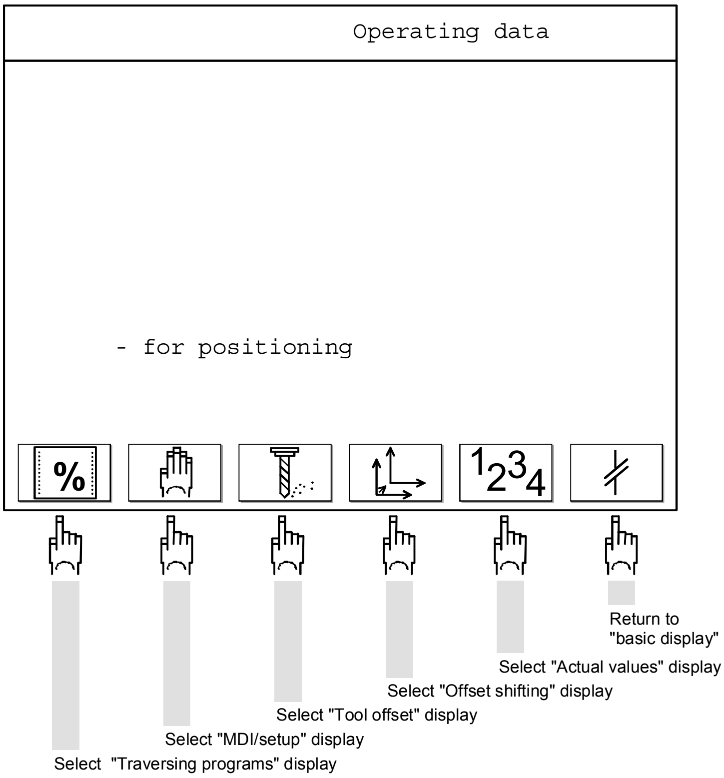
Please note that changing parameter settings when the system is running is equivalent to modifying the hardware configuration. Disconnected inputs and outputs can "freeze" at their last signal state. You can remedy this situation by restarting the system.

Password Protection

Input is possible with level 3 or a higher-priority level.

8.7 Operating data for Positioning

Screen Form



Description

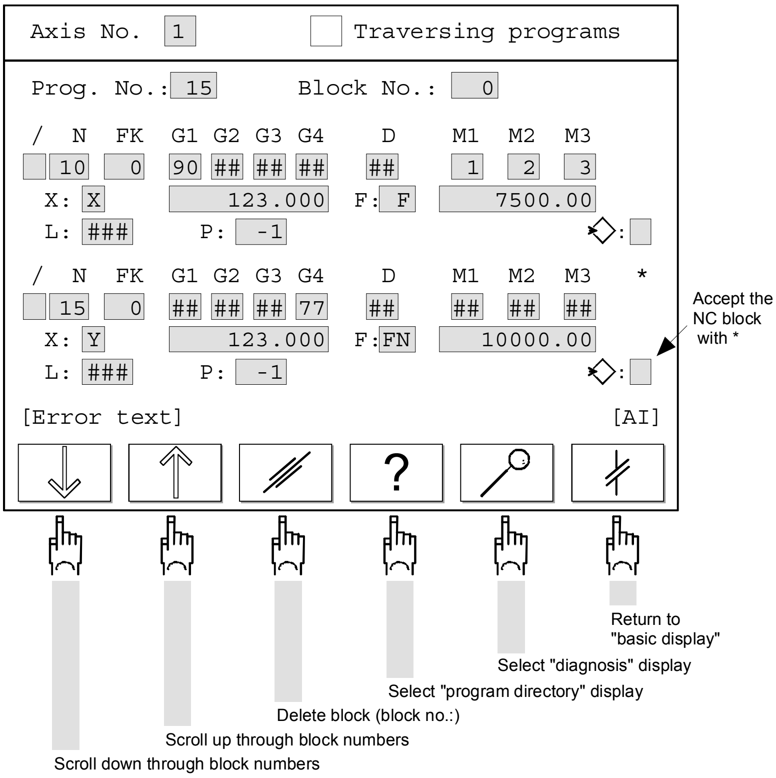
Intermediate display "Operating data for positioning" serves only as a distribution point to other displays.

Password Protection

None

8.7.1 Traversing Programs

Screen Form



Description

The "traversing program" display can be used to input and output NC programs. Individual blocks can also be deleted.

The "prog. no." and "block no." parameters specify the block number at which the NC program listing starts.

Blank input lines are required in order to enter new blocks in an existing NC program. You can create these blank input lines as follows:

- ♦ Scroll down through the NC blocks until you reach the end of the NC program, or
- ♦ In "block no.", enter a block number greater than the highest block number in the NC program.

You can then enter a new block.

The block elements are not displayed for input until you enter the block number under "N".

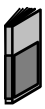
To delete an NC block, you must specify the block number in "block no." and press the "delete NC block" softkey twice.

Representation of the NC Block

```
 /  N   FK   G1  G2  G3  G4     D      M1   M2   M3    *
[ ][10] [ 0] [90][##][##][##]  [##]    [ 1] [ 2] [ 3]
  X: [X]      [   123.000]  F:[F]  [   7500.00]
  L: [###]      P: [ -1]                        ◇: [*]
```

/	Identifier for a skippable block
N	NC block number (1 and 200)
FK	Following block identifier (0 and 19)
G1	G1 function
G2	G2 function
G3	G3 function
G4	G4 function
D	Tool offset number
M1	M1 function number
M2	M2 function number
M3	M3 function number
X:	Axis name (X, Y, Z, A, B, C) and positional value (-999 999.999 to 999 999.999)
F:	Velocity type ((F: path velocity, FN: axis velocity) and velocity value (0 to 1000000.00)
L:	Subprogram number
P:	Loop count of subprogram calls

Non-assigned block elements are shown with a "#" (exception: "P: -1" for not assigned)



Further Information

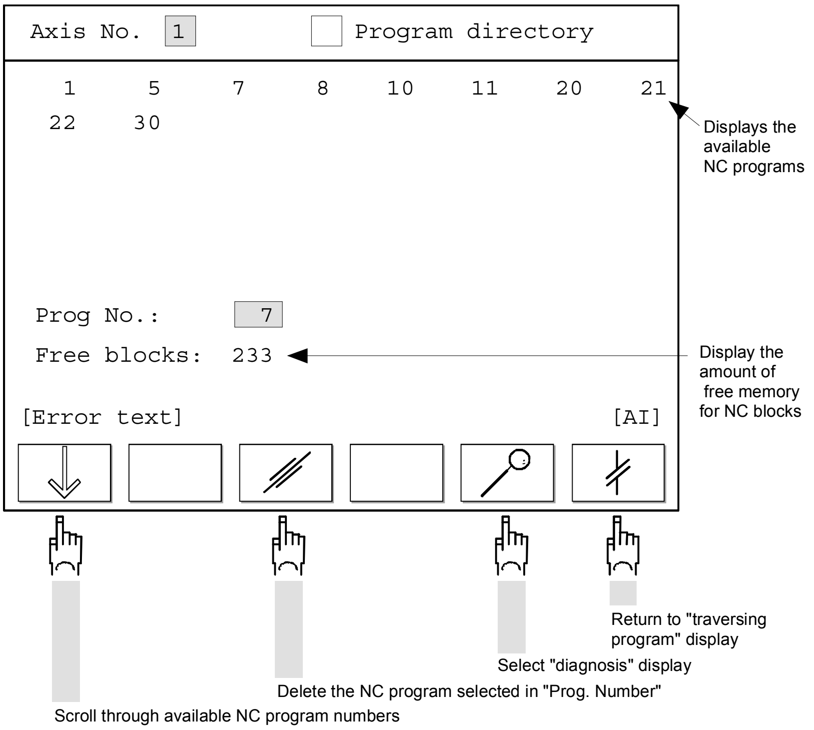
The procedure for writing NC programs is described in the "Programming Guide".

Password Protection

Input is possible with level 2 or a higher-priority level.

8.7.2 Program Directory

Screen Form



Description

The "program directory" display shows a listing of all NC program numbers available in the axis.

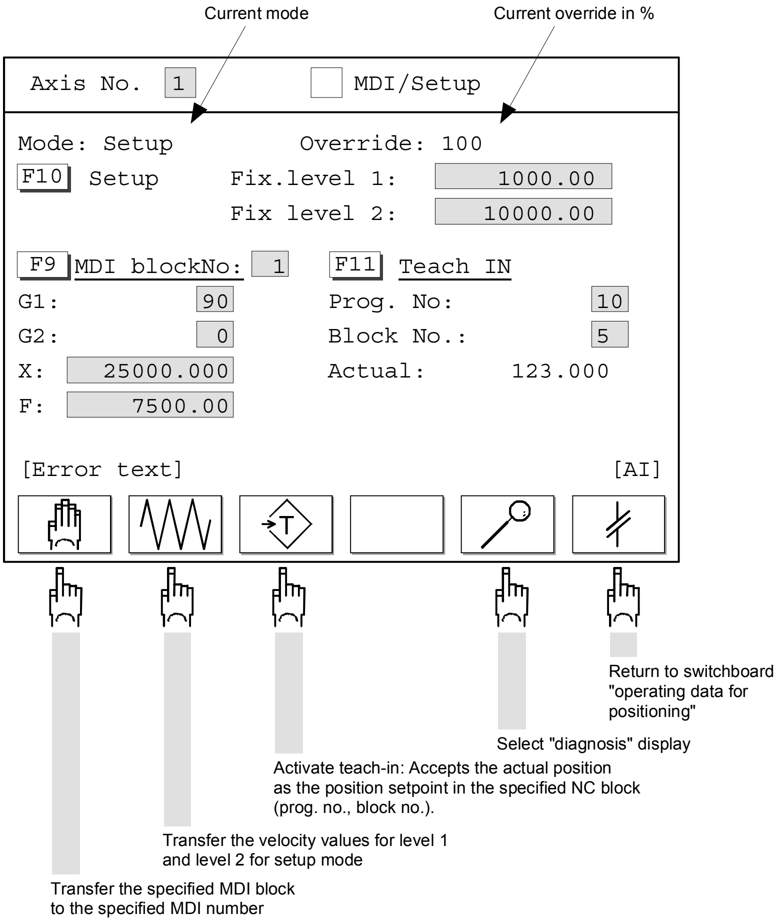
To delete an NC program, you must specify the program number in "prog. no." and press the softkey twice.

Password Protection

Deletion is possible with level 2 or a higher-priority level.

8.7.3 MDI/Setup

Screen Form



Description

The "MDI/setup" display is used exclusively for data input/output for the setup and MDI modes.

Level 1 and level 2 represent the two velocities for setup mode. If control signal [F_S] = 0, level 1 is selected. When [F_S] = 1, level 2 is selected.

G1 (G1 function), G2 (G2 function), X (position) and F (velocity) are the values for the MDI block. You can select from 10 MDI numbers.

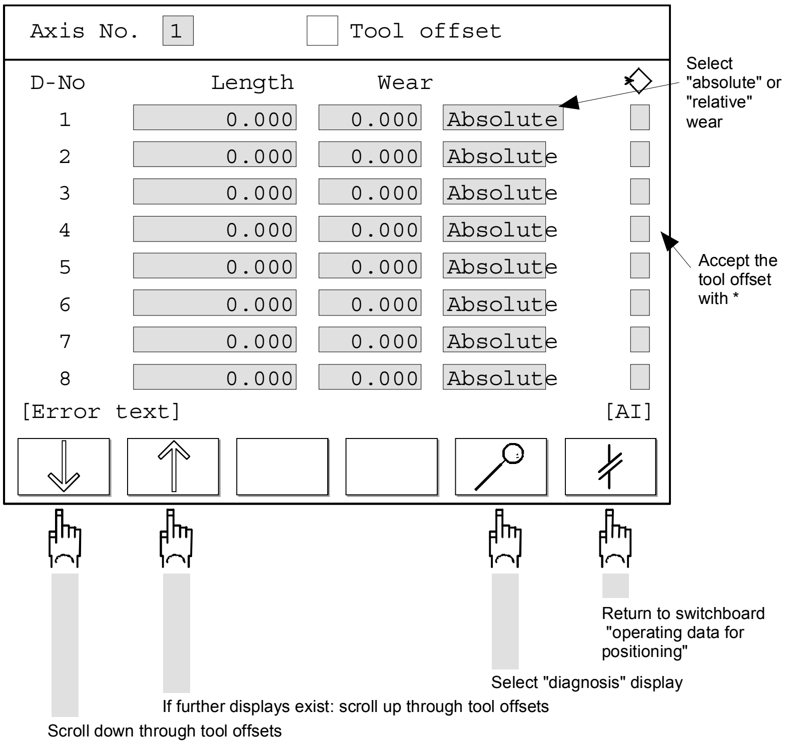
For teach-in, you must specify the NC program number and block number in "prog. no." and "block no.".

Password Protection

Input is possible with level 1 or a higher-priority level.

8.7.4 Tool Offset

Screen Form



Description

The 20 tool offset memories can be input and output in the "tool offset" display.

The tool wear can be specified in absolute or additive terms on input. The wear is always shown in absolute terms on output.

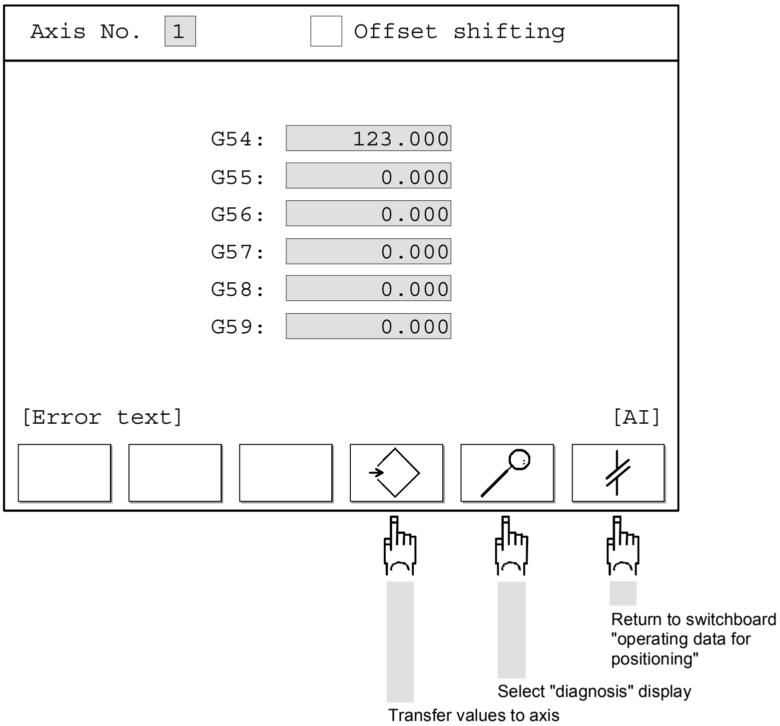
Each tool offset must be saved individually with *.

Password Protection

Input is possible with level 1 or a higher-priority level.

8.7.5 Offset Shifting

Screen Form



Description

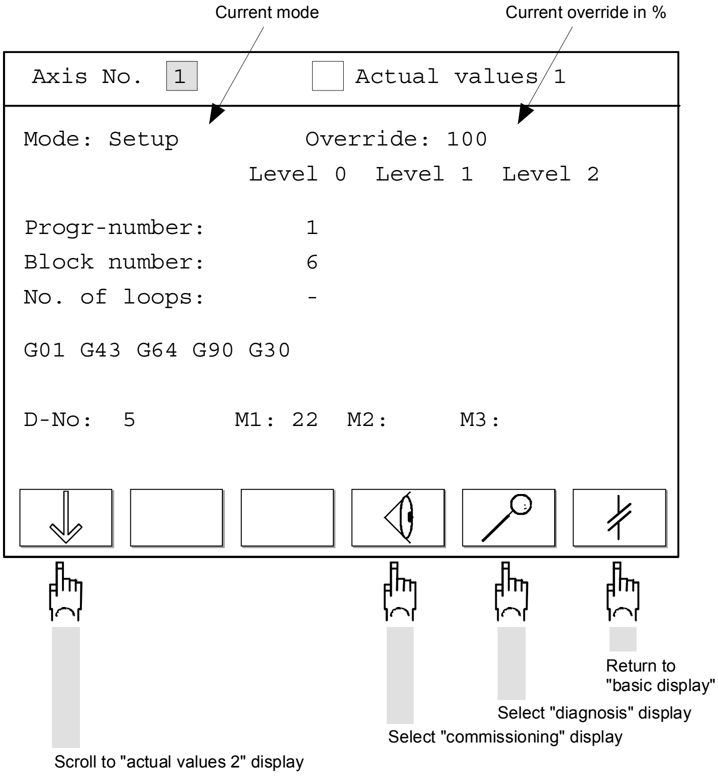
The zero offsets can be input or output in the "offset shifting" display. The zero offsets can be selected in NC programs using G functions G54 to G59.

Password Protection

Input is possible with level 1 or a higher-priority level.

8.7.6 Actual Values 1

Screen Form



Description

The current NC program runtime data are shown in the "actual values 1" display.

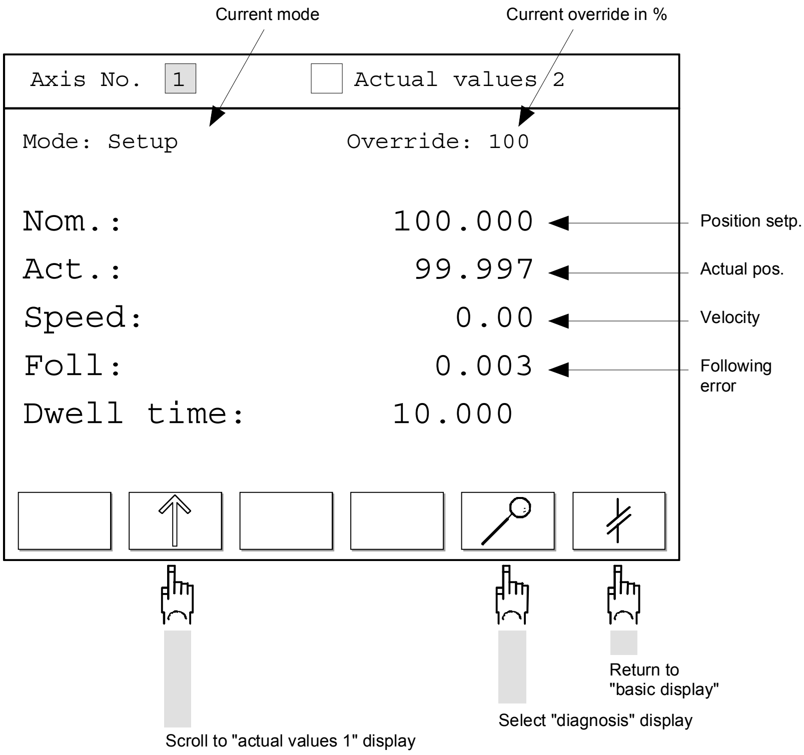
The current point of execution in the program is shown by "prog. no." and "block no.". Levels 1 and 2 represent the subprograms. "No. of loops" shows the remaining number of subprogram passes. Also displayed are the active G functions, the current tool offset number and the current M functions.

Password Protection

None

8.7.7 Actual Values 2

Screen Form



Description

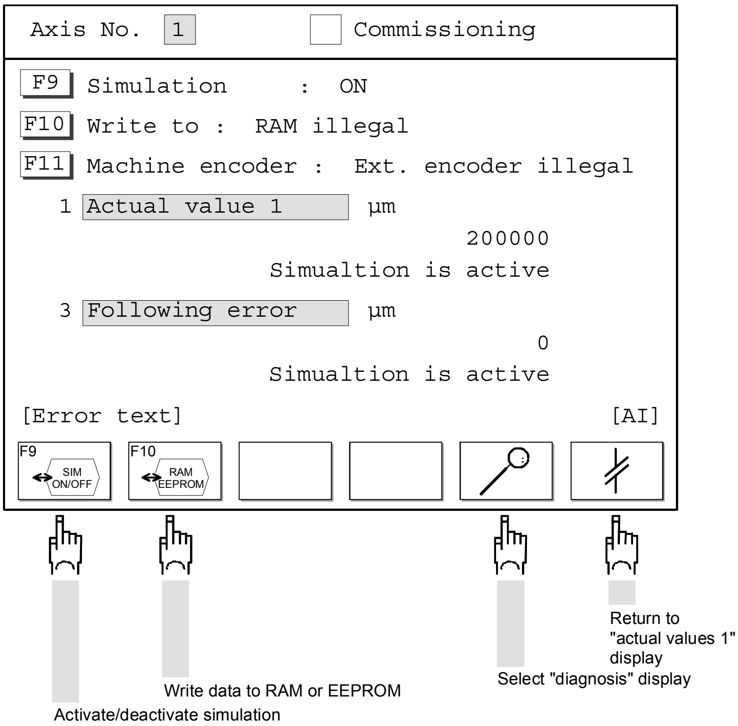
The "actual values 2" display shows the current actual values of an axis.

Password Protection

None

8.7.8 Commissioning

Screen Form



Description

The "Commissioning" screen displays a variety of actual values. You can select two of the following actual values from the list at one time.

Further Information

Please refer to document "Task Descriptions", Chapter 6.1, for a description of the meaning of individual actual values.

NOTE

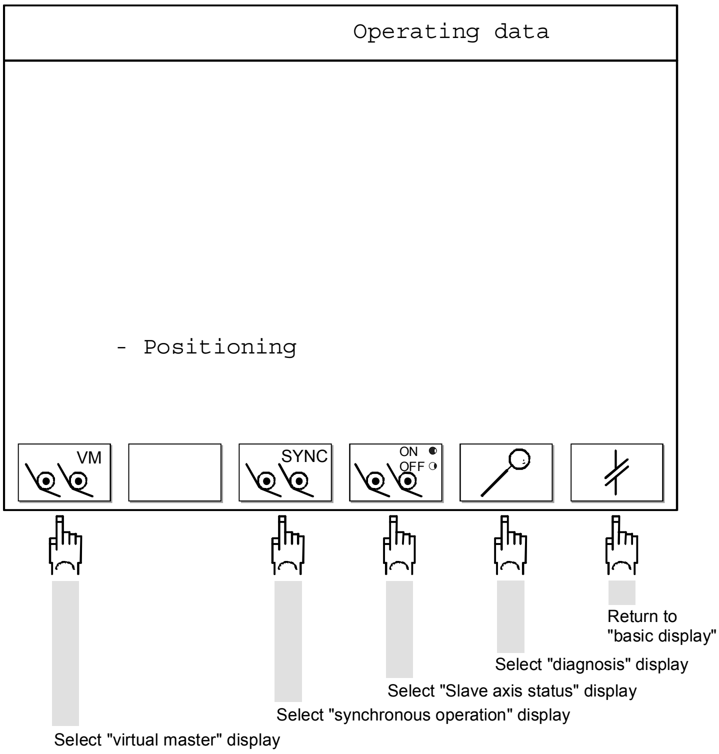
To ensure that a change in the Simulation state is accepted by the axis, the "Reset technology" control signal [RST] must be activated.

Password Protection

Simulation can be activated or deactivated from level 1 onwards.

8.8 Operating Data for Synchronous Operation

Screen Form



Description

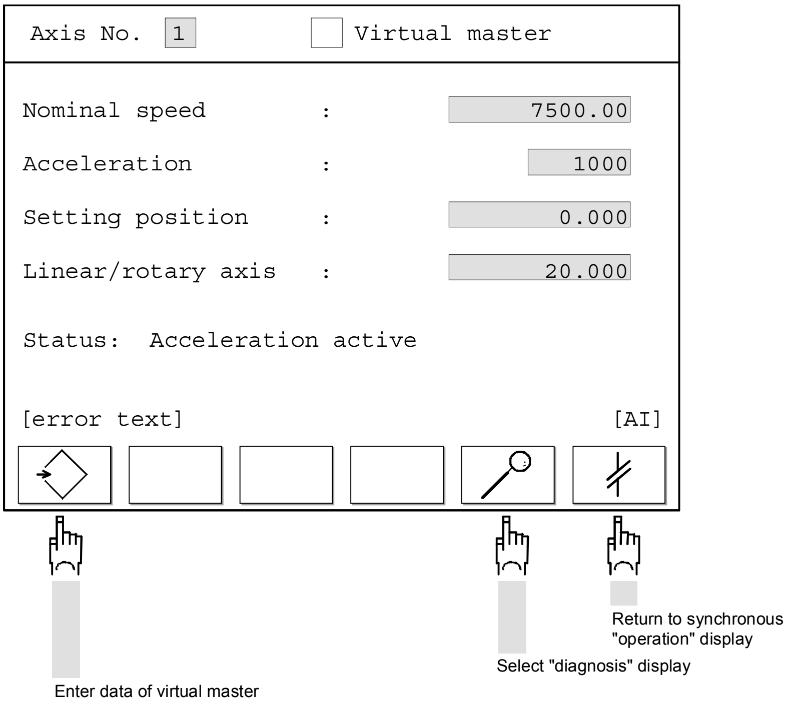
Intermediate display "Operating data for synchronous operation" serves only as a distribution point to other displays.

Password Protection

None

8.8.1 Virtual Master

Screen Form



Description

The "virtual master" display is used to input and output data for the virtual master.

The velocity setpoint ("nominal speed") and acceleration define the movement curve. "Setting position" can be used to specify a start position. Specifying a value of "0" for "linear/rotary axis" defines that the virtual master is a linear axis. A value "> 0" determines the length (circumference) of the rotary axis.

"Status" indicates the status of the virtual master. The following statuses are possible:

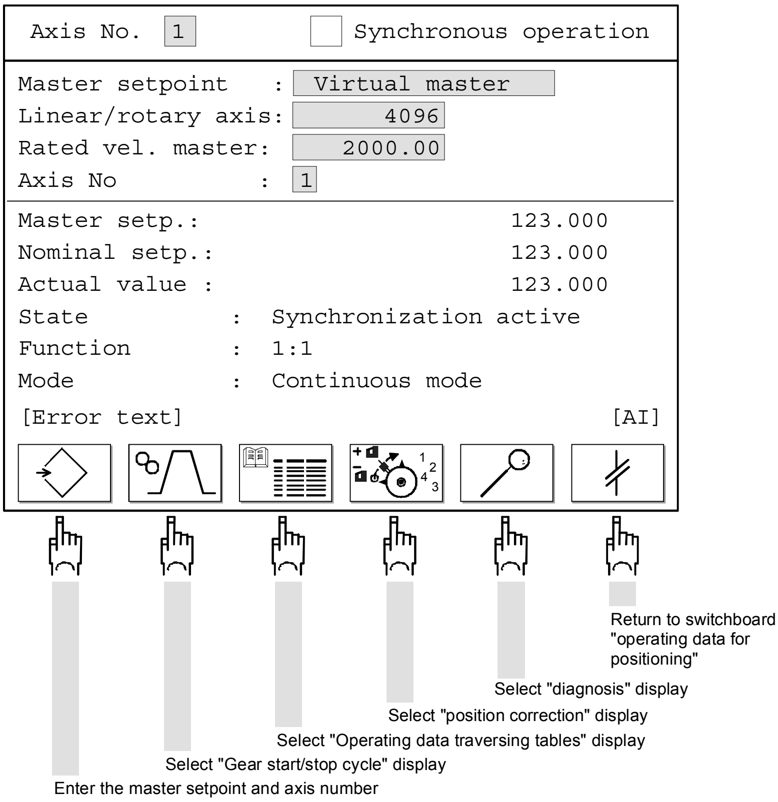
- Virtual master standstill
- Acceleration active
- Velocity setpoint reached
- Deceleration active
- Virtual master running

Password Protection

Input is possible with level 1 or a higher-priority level.

8.8.2 Synchronous Operation

Screen Form



Description

The "synchronous operation" display allows you to select the master value source, and shows the synchronization status data. Actual value control, setpoint control and virtual master can be selected as the master value source. If the technology is operated on an M7-FM, you must also specify the axis number (local axis number on the M7-FM) of the master axis used as the master value source.

The following synchronization status data are displayed:

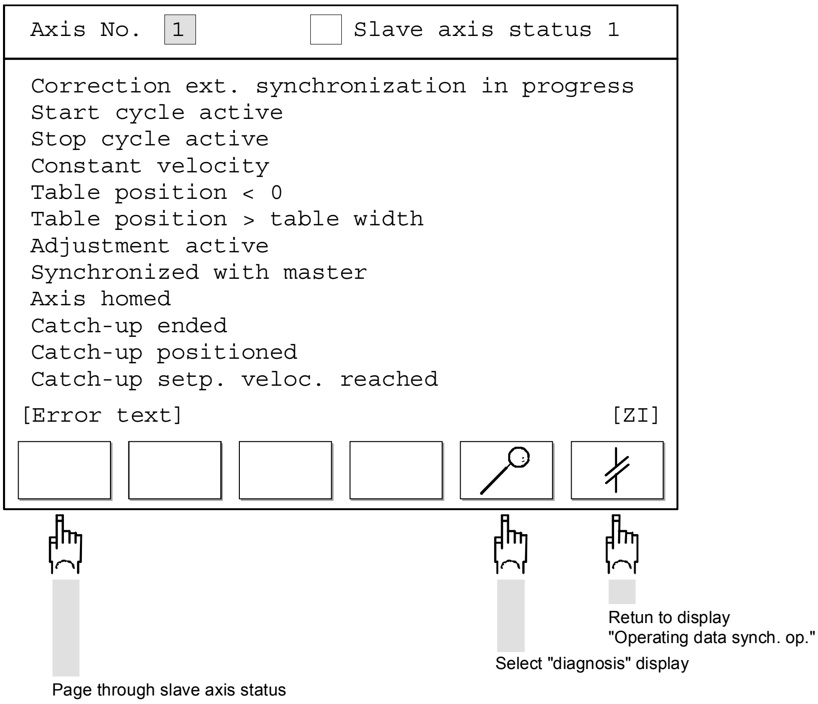
- Master setp.: Position of the master axis
- Nominal setp.: Position setpoint of the slave axis
- Actual value: Actual position of the slave axis
- State: Indicates whether synchronization is active or inactive
- Function: Displays the active function (1:1, gearbox or table)
- Mode: Indicates the selected cycle (continuous, start cycle or stop cycle)

Password Protection

Input is possible with level 1 or a higher-priority level.

8.8.3 Slave Axis Status

Screen Form



Description

The "Slave axis status" screen displays the status of different bits.

Further Information

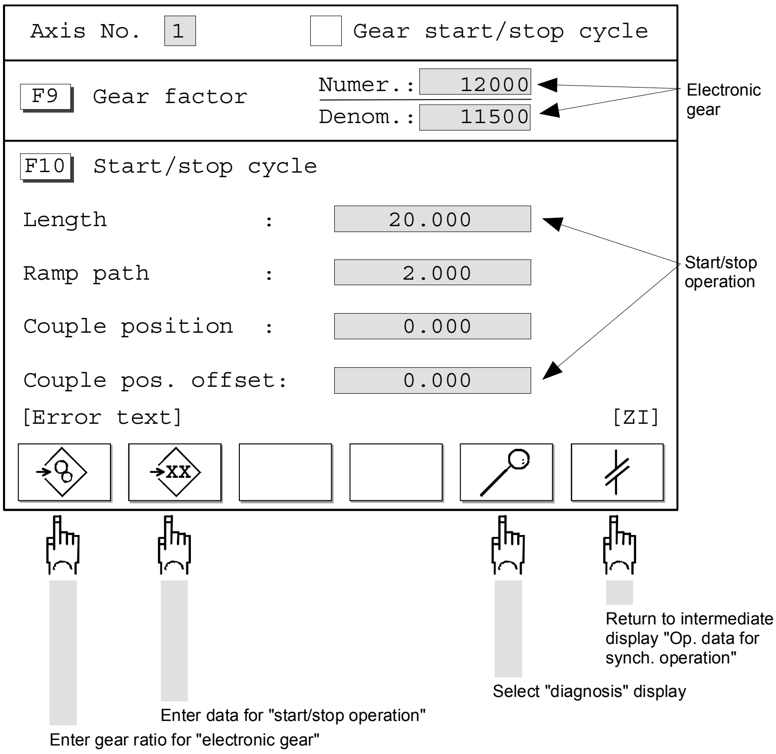
Please refer to document "Task Descriptions", Chapter 10.2, for a description of the meaning of individual status values.

Password Protection

None

8.8.4 Gear Start/Stop

Screen Form



Description

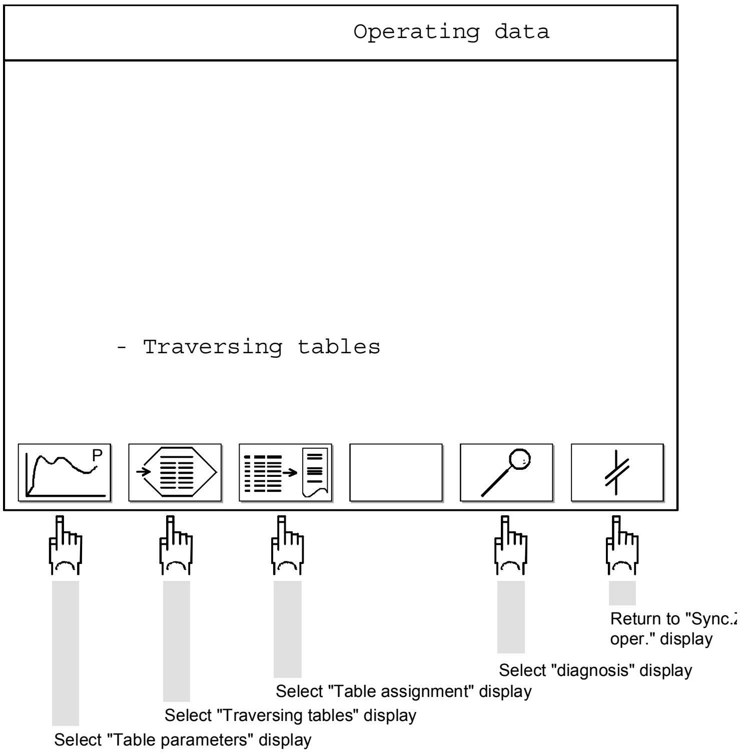
The data for the electronic gear and the start/stop cycle can be input and output in the "gear start/stop" display.

Password Protection

Input is possible with level 1 or a higher-priority level.

8.8.5 Operating Data for Traversing Tables

Screen Form



Description

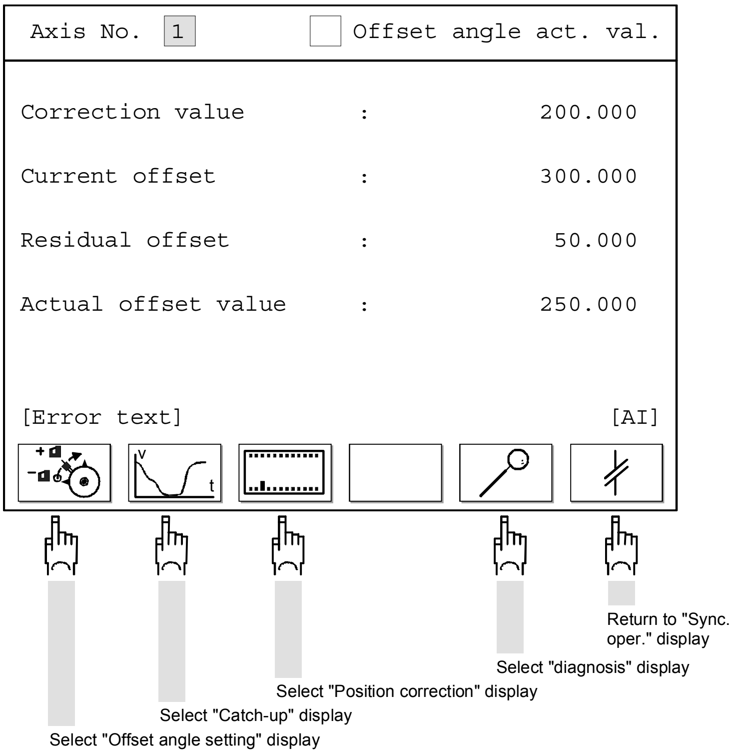
Intermediate display "Operating data for traversing tables" serves only as a distribution point to other displays.

Password Protection

None

8.8.6 Actual Offset Angle Values

Screen Form



Description

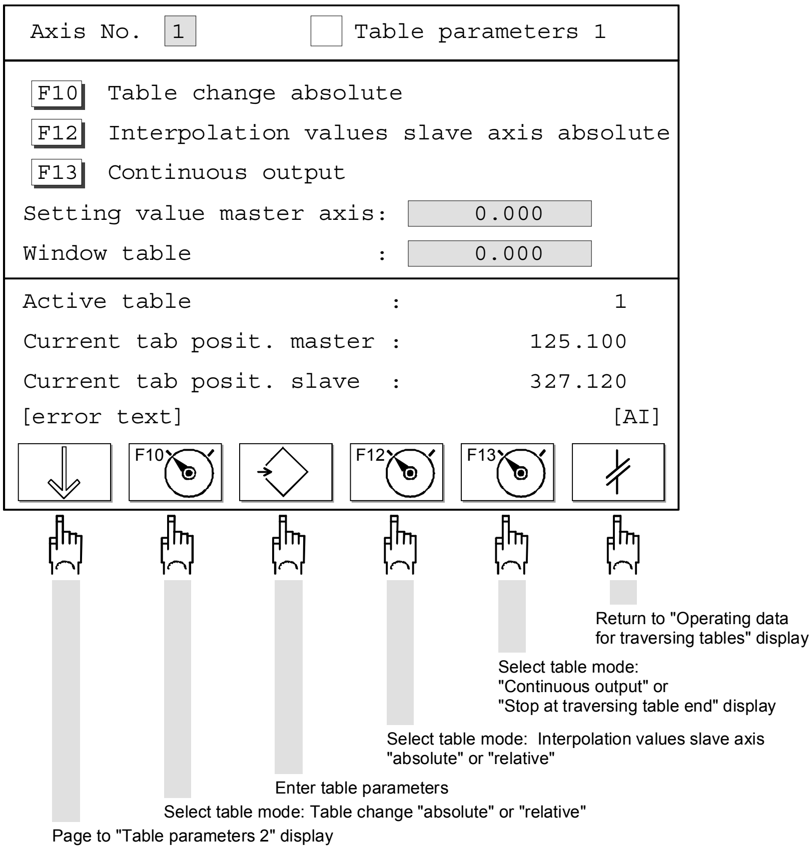
In form "Actual values offset angle" the actual offset angle settings are displayed.

Password Protection

None

8.8.7 Table Parameters 1

Screen Form



Description

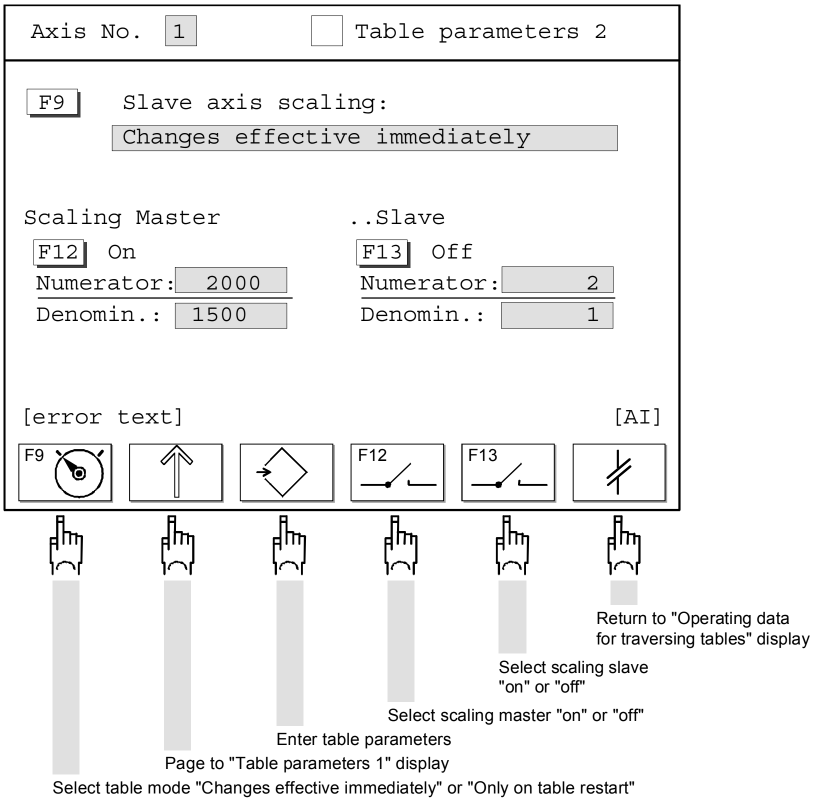
The "table parameters" display can be used to input and output the table parameters for scaling and mode. The display also shows the NC table currently being processed, together with the current table positions.

Password Protection

Input is possible with level 1 or a higher-priority level.

8.8.8 Table parameters 2

Screen Form



Description

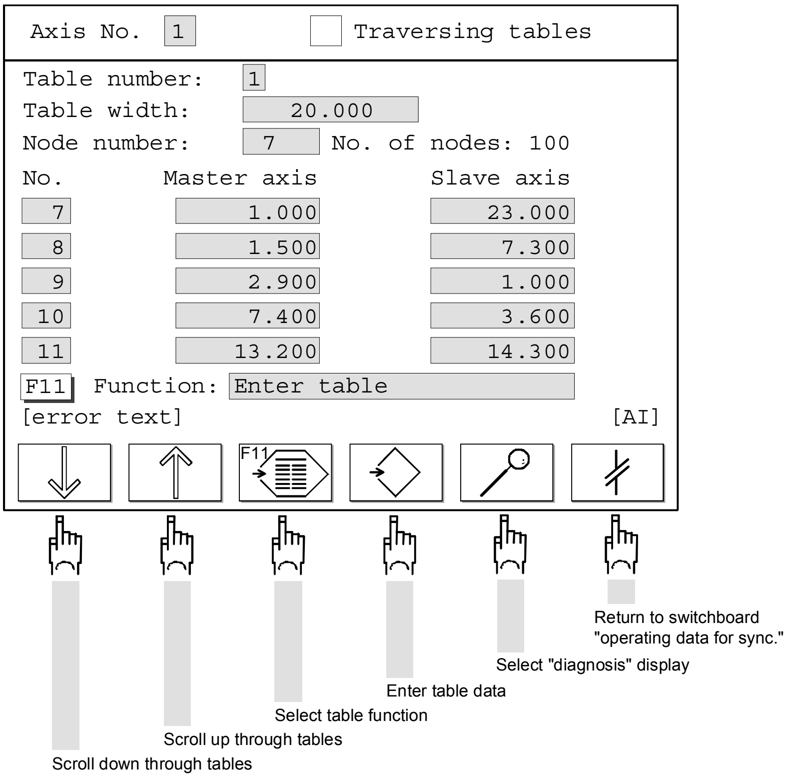
The "table parameters" display can be used to input and output the table parameters for scaling and mode. The display also shows the NC table currently being processed, together with the current table positions.

Password Protection

Input is possible with level 1 or a higher-priority level.

8.8.9 Traversing Tables

Screen Form



Description

Traversing tables can be input or output in the "Traversing tables" display. When you enter the "Table number", the traversing table is output from the specified "Interpolation point no.".

You can enter the following settings with the F11 table functions:

- Enter table:
  The contents of the display is entered in the NC table when you activate the "enter table data" softkey.
- Insert interpolation point:
  When you activate the "enter table data" softkey, an interpolation point is inserted at "node number". The positional values of the inserted point are initialized with "0.000".
- Delete interpolation point:
  When you activate the "enter table data" softkey, the interpolation point with the "node number" is deleted.
- Delete table:
  When you activate the "enter table data" softkey, the contents of the table are deleted.

NOTE

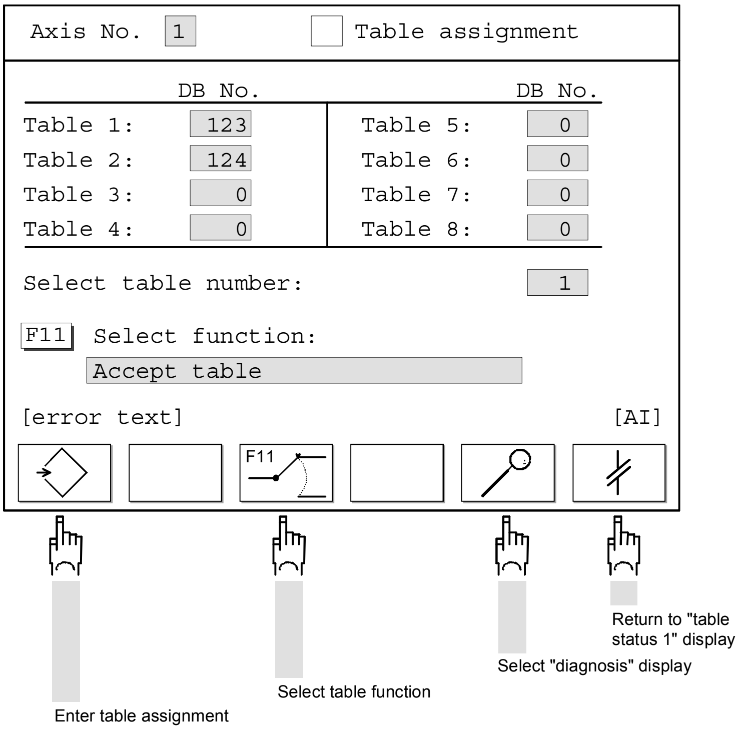
If you have changed any interpolation values, you must activate the "enter table data" softkey before you "scroll".

Password Protection

Input is possible with level 2 or a higher-priority level.

8.8.10 Table Assignment

Screen Form



Description

NC tables can be assigned, reset or accepted in the "table assignment" display.

If the technology is operated on an M7-FM, up to 8 tables can be assigned to each axis. Since the tables are stored in data blocks on the M7-FM, they must be assigned to the data block numbers. All NC tables are assigned to specific data blocks in the default system setting. You only need to change the default assignment if you want to use other data blocks (in this case you must set up the data blocks).

If the technology is operated on a MASTERDRIVES MC, you have the option of configuring one table with 200 interpolation points or two tables with 100 points each. You will need to enter the following data for this purpose:

- One table:
  Table 1 / DB no. = "1" and table 2 / DB no. = "0"
- Two tables:
  Table 1 / DB no. = "1" and table 2 / DB no. = "1"

When entering the NC table assignment, you can also activate a table function with F11:

- ♦ Accept assignment:
  Only the table assignments are accepted.
- ♦ Reset table:
  The "selected table no." is reset. It must be entered again for machining.
- ♦ Accept table:
  The "selected table no." is verified (calculated) and, if no error occurs, it is accepted.
- ♦ Accept all tables:
  All assigned NC tables are verified and, if no error occurs, are accepted.

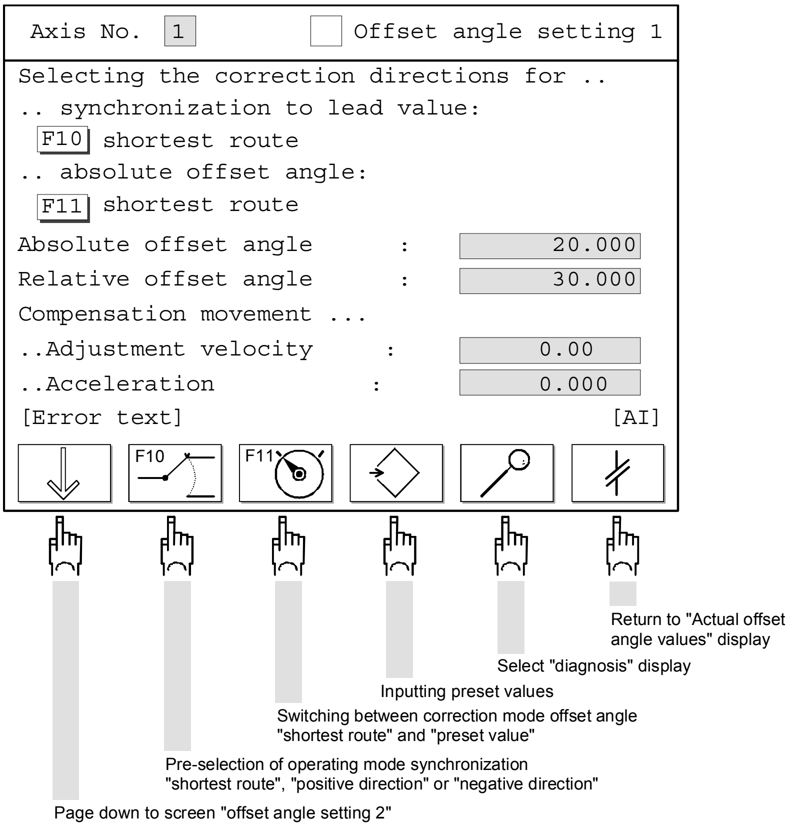
Errors are displayed in the message line and also in the "table status 1 and 2" displays.

Password Protection

Input is possible with level 2 or a higher-priority level.

8.8.11 Offset Angle Setting 1

Screen Form



Description

In form "Offset angle setting 1" the parameters for the offset angle setting can be set and output.

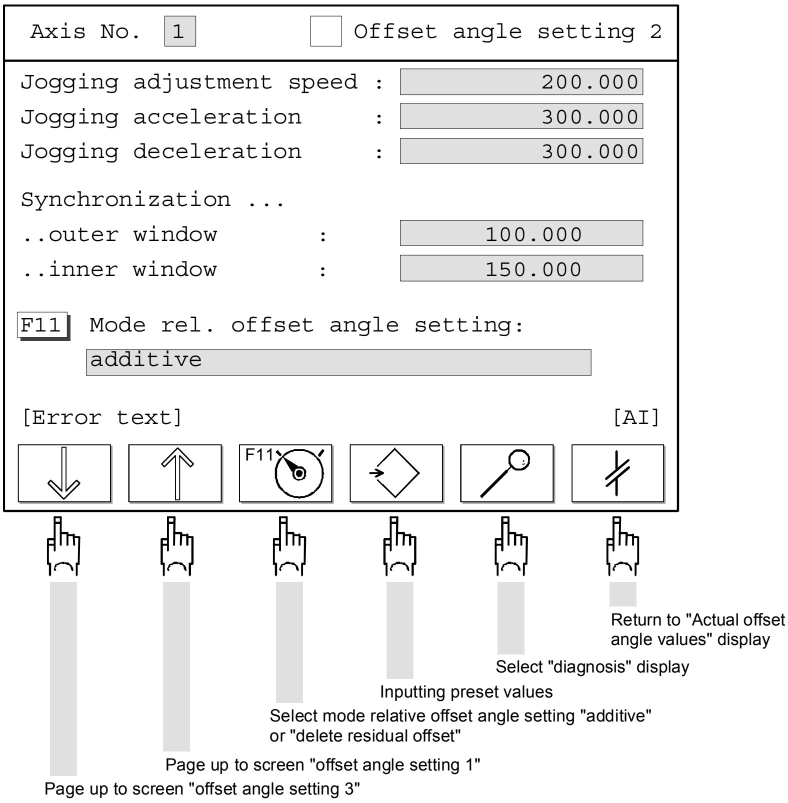
The offset angle setting consists of the offset memory functions, a positioning part for executing the offset movement, and various functions and interfaces for changing the offset angle.

Password Protection

Input is possible with level 1 or a higher-priority level.

8.8.12 Offset Angle Setting 2

Screen Form



Description

In form "Offset angle setting 2" the parameters for the offset angle setting can be set and output.

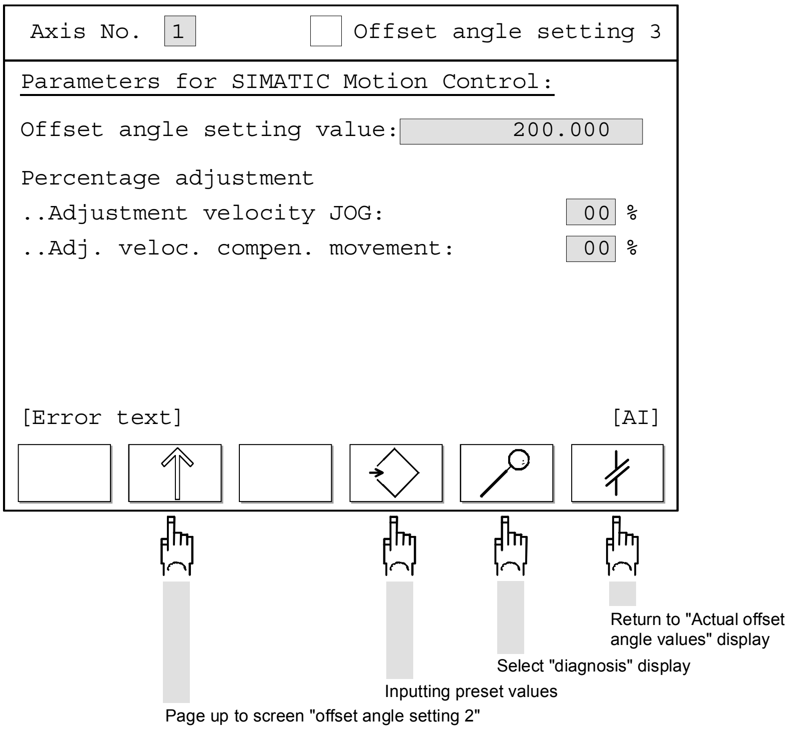
The offset angle setting consists of the offset memory functions, a positioning part for executing the offset movement, and various functions and interfaces for changing the offset angle.

Password Protection

Input is possible with level 1 or a higher-priority level.

8.8.13 Offset Angle Setting 3

Screen Form



Description

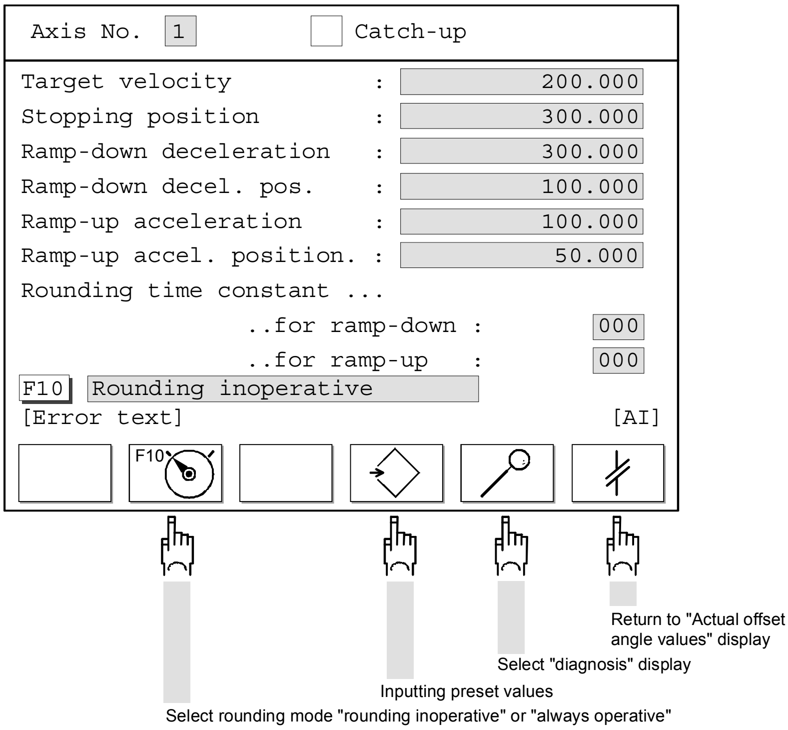
The offset angle setting parameters for SIMATIC Motion Control can be input or output in the "Offset angle setting 3" display.

Password Protection

Input is possible with level 1 or a higher-priority level.

8.8.14 Catch-up

Screen Form



Description

In Form "Catch-up" the parameters for the catch-up function can be set and output.

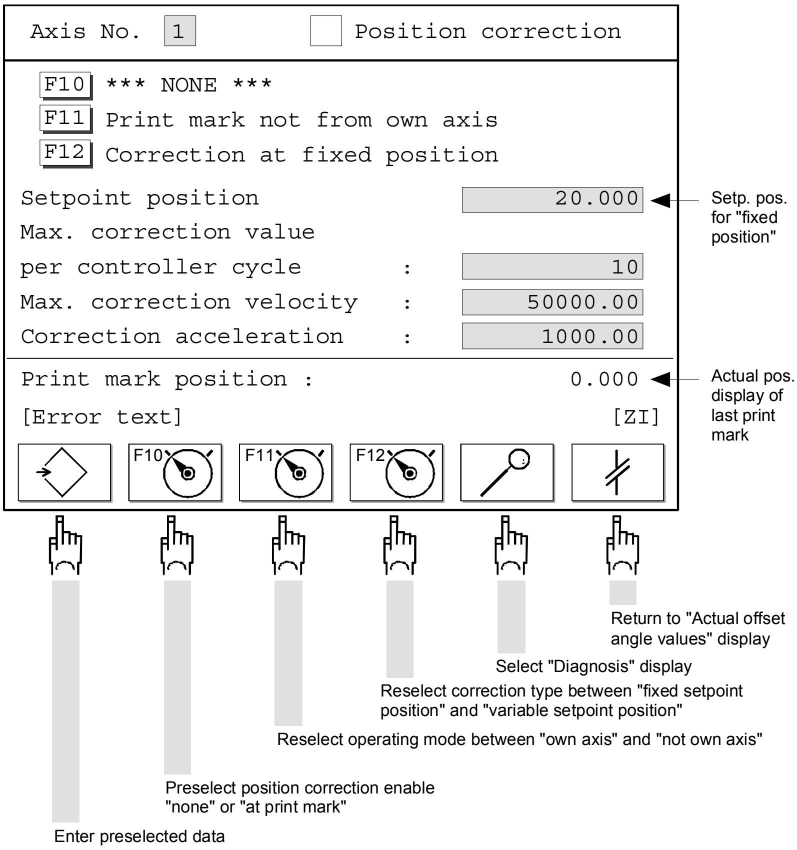
The catch-up function offers another way (in addition to the start/stop function) to decouple / couple a drive from / into interconnected, synchronously running motors (e.g. shaftless printing machine).

Password Protection

Input is possible with level 1 or a higher-priority level.

8.8.15 Position Correction

Screen Form



Description

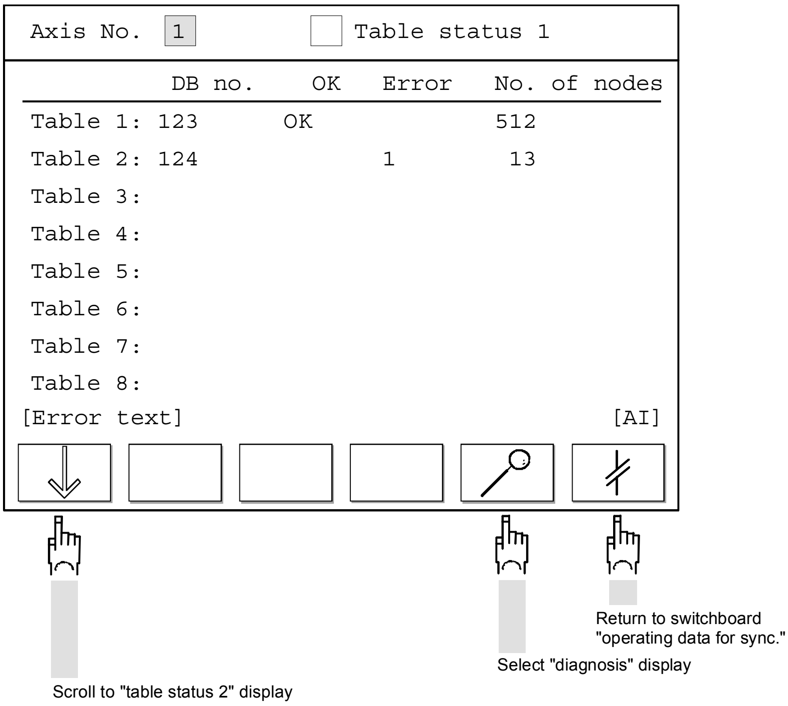
The parameters for print mark synchronization can be input and output in the "Position correction" display.

Password Protection

Input is possible with level 1 or a higher-priority level.

8.8.16 Table Status 1

Screen Form



Description

The "table status 1" display shows part of the current NC table status. The following information is displayed:

- **DB No.:** Number of the data block number which contains the NC table on the M7-FM. If the technology is installed on the MCT, a maximum of two NC tables only can be configured. When configuring "one table", DB no. "1" is displayed under table 1 / DB no. "1". When configuring "two tables", "1" is displayed under table 1 / DB no. and table 2 / DB no.
- **O.K.:** "O.K" indicates that the table has no errors.
- **Error:** If the table is not ok, an error code is displayed. You can look up the meaning of the error codes by scrolling through the "table status 2" display.
- **No. of nodes:** This is the number of interpolation points in the NC table. In the event of an error, the last error-free interpolation point number is displayed.

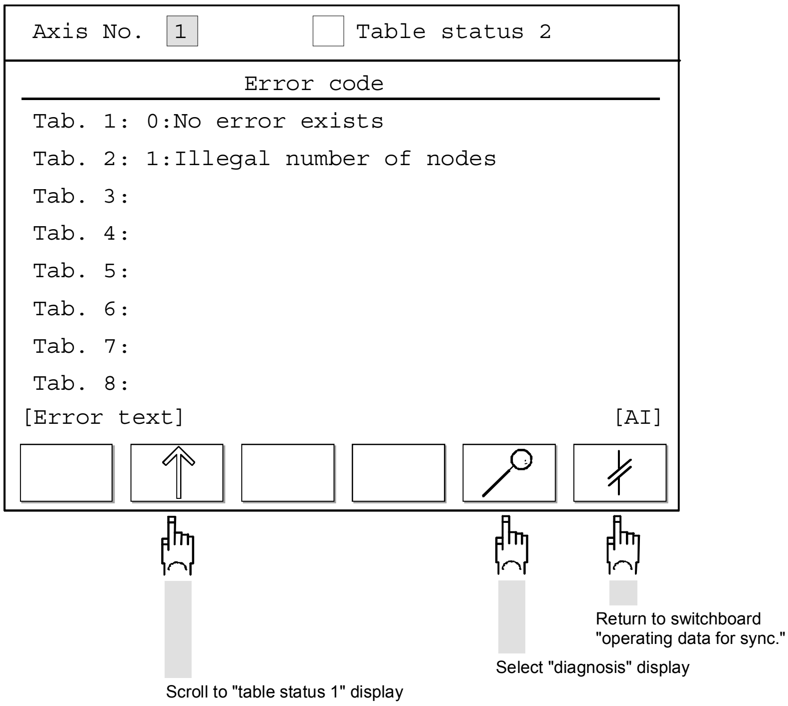
The NC table is defined and assigned in the "table assignment" display".

Password Protection

None

8.8.17 Table Status 2

Screen Form



Description

The "table status 2" display shows the error code of the table status in plain text.

The following error codes are displayed:

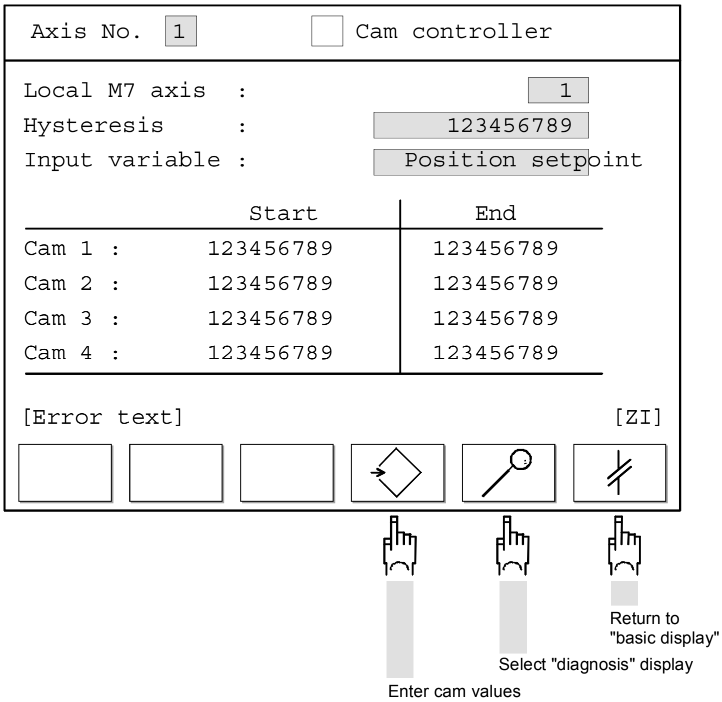
- 0: No error exists
- 1: Illegal number of nodes (0 or greater than the maximum possible number of interpolation points)
- 2: Master position greater than table width
- 3: Master position not increasing
- 4: Data block does not exist
- 5: Data block too short

Password Protection

None

8.9 Cam Controller

Screen Form



Description

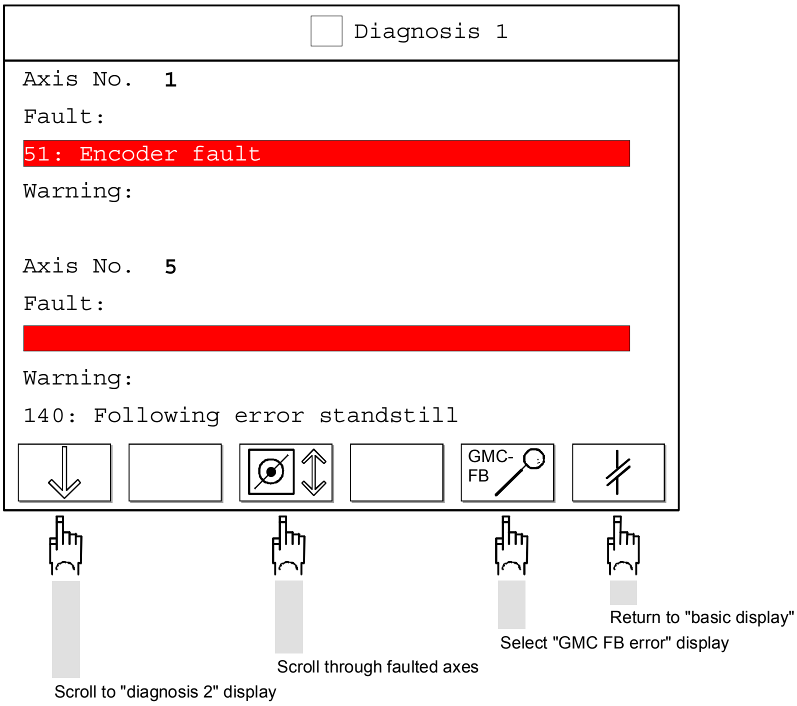
You can parameterize the four cams of a cam controller in this display. The cam controller is processed in the axis sampling time cycle.

Password Protection

Input is possible with level 2 or a higher-priority level.

8.10 Diagnosis 1

Screen Form



Description

The queued faults and warnings of the axes are shown in the "Diagnosis 1" display.

If more than warning is queued, you can scroll through the rest of the warnings in the "diagnosis 2" display.

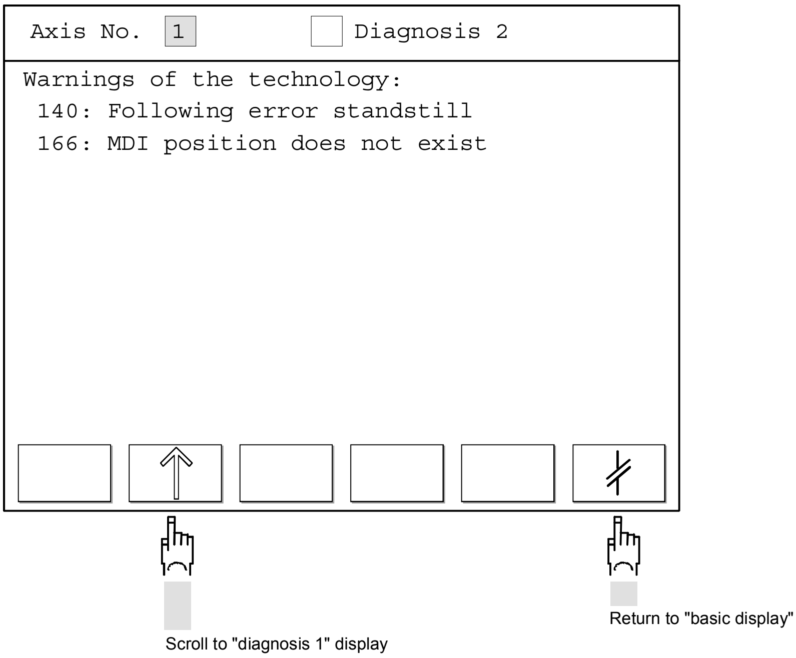
The "acknowledge fault" [ACK_F] control signal is used to clear faults and warnings (warnings starting at no. 129).

Password Protection

None

8.11 Diagnosis 2

Screen Form



Description

Up to 8 technology warnings of an axis can be visualized simultaneously in the "diagnosis 2" display.

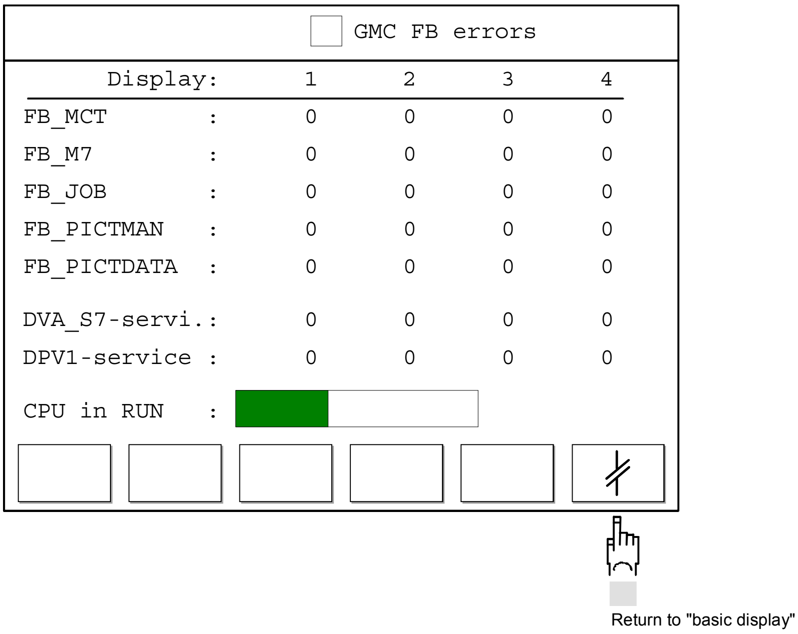
The "acknowledge fault" [ACK_F] control signal is used to clear faults and warnings (warnings starting at no. 129).

Password Protection

None

8.12 GMC FB Errors

Screen Form



Description

The "GMC_FB_errors" display indicates errors of S7 blocks. The displays match the "displays" section in data block GMC_DB_ORG.

When the bar is moving, the CPU is in RUN mode.

Password Protection

None

Index

M

N

O

P

R

S

T

U

V

A Appendix A: Error Messages

Contents

In this chapter you will find a description of all the error messages.

A.1 Warnings of the Technology

Output Location of the Warnings

The current warnings of the technology are entered in the checkback signals of the axis.

	7	6	5	4	3	2	1	0		**Axis_n.**
DBBy	RES	RES	RES	RES					BIN	OUT_1
DBBy+1	RES	WARN							BIN	OUT_2
DBBy+2									DEC	OUT_3
DBBy+3	WARN_NO								DEC	OUT_4

You can use the "output GMC technology warnings" task to output up to 8 queued warnings simultaneously.

The warnings are also displayed on the MASTERDRIVES MC.

Below is a complete list of technology warnings.

129	**Axis does not exist – machine data 1 = 0**
Cause:	Machine data 1 (encoder type/axis type) is 0 (axis does not exist).
Effect:	Operation of the axis is inhibited and the position controller is deactivated.
Remedy:	You must assign a valid value to machine data 1 in order to operate the axis.

130	**Operating conditions do not exist**
Cause:	The "in operation" [IOP] checkback signal was missing when a traversing command was initiated. The following conditions inhibit the "in operation" checkback signal: • MASTERDRIVES Motion Control (technology option F01) - Control signals [OFF1], [OFF2], [OFF3] and/or "enable controller" [ENC] are not activated. - Checkback signals [OFF2] and/or [OFF3] are not activated. - A fault [FAULT] is active. • SIMATIC Motion Control (IM178) - Control signal "enable controller" [ENC] is not activated. - Checkback signal "Converter working" not displayed on the IM178 [input I2].
Effect:	The traverse command is stopped or the drive is brought to a standstill via a ramp (machine data 43: deceleration time during errors).
Remedy:	• Activate control signals [OFF1], [OFF2], [OFF3] and "enable controller" [ENC]. • If checkback signals [OFF2] and/or [OFF3] are missing, check the supply of control word 1 (Masterdrive function diagram, sheet 180). • Analyze the queued fault number [FAULT_NO], remedy the fault, and then cancel the fault using the acknowledge fault [ACK_F] control signal. **NOTE:** To activate the "in operation" [IOP] status again, you must deactivate [OFF1] and then activate it again.

131	**OFF1 missing**
Cause:	Control signal [OFF1] was deactivated while a traversing command was being executed.
Effect:	The drive is brought to a standstill via a ramp (machine data 43: deceleration time during errors). There is a subsequent pulse disable.
Remedy:	Check the activation of control signal [OFF1] from the user program.

132	**OFF2 missing**
Cause:	• Control signal [OFF2] was deactivated while a traversing command was being executed. • Checkback signal [OFF2] was deactivated while a traversing command was being executed.
Effect:	The pulse disable is initiated immediately. If the motor is not braked, it coasts down.
Remedy:	• Check the activation of control signal [OFF2] from the user program. • If checkback signal [OFF2] is missing, check the supply of control word 1 (Masterdrive function diagram, sheet 180). **NOTE:** To activate the "in operation" [IOP] status again, you must deactivate [OFF1] and then activate it again.

133	**OFF3 missing**
Cause:	• Control signal [OFF3] was deactivated while a traversing command was being executed. • Checkback signal [OFF3] was deactivated while a traversing command was being executed.
Effect:	The motor decelerates at the current limit. There is a subsequent pulse disable.
Remedy:	• Check the activation of control signal [OFF3] from the user program. • If checkback signal [OFF3] is missing, check the supply of control word 1 (MASTERDRIVES function diagram, sheet 180). **NOTE:** To activate the "in operation" [IOP] status again, you must deactivate [OFF1] and then activate it again.

134	**Enable Controller ENC missing**
Cause:	The "enable controller" [ENC] control signal was deactivated while a traversing command was being executed.
Effect:	The pulse disable is initiated immediately. If the motor is not braked, it coasts down.
Remedy:	Check the activation of the "enable controller" [ENC] control signal from the user program.

135	**Error in measurement system – RESET necessary**
Cause:	The position logging system reported an actual position logging error. Plug unplugged, actual value cable broken, global actual value logging error.
Effect:	Position controller is switched off and stopped using the "deceleration time on error".
Remedy:	RESET, reference point search required for incremental position encoders.

136	**Machine data 1 changed - RESET necessary**
Cause:	Machine data 1 (encoder type/axis type) was changed.
Effect:	The activation of traversing commands is inhibited.
Remedy:	If machine data 1 has been changed, the "reset technology" [RST] control signal must be activated.

137	**Multiple axis assignment MD2 illegal – RESET necessary**
Cause:	The current axis has been assigned more than once. This warning is output after Power On, mode change and RESET (M7-FM only).
Effect:	The machine data last transferred cannot be activated. The previously valid machine data are still valid and will be activated as well.
Remedy:	A unique axis assignment must be parameterized for all the axes on an M7-FM. You may not, for example, define two axes as an X axis. RESET, correct machine data MD2 for affected axis and transfer again to the axis. Then activate the machine data.

138	**Axis assignment of roll feed incorrect**
Cause:	The NC block contains an axis number which is defined as a roll feed axis (M7 only) but the axis type is defined as an incremental or absolute encoder (machine data 1 = 1 or 2). The NC block for a roll feed axis type (machine data 1 = 3) contains: • No axis number (X, Y, Z...) • An incorrect axis number
Effect:	NC program execution is inhibited or aborted.
Remedy:	• Axis type 1 or 2: The block is not allowed to contain an axis number which is defined as a roll feed (M7 only). • Axis type 3: The axis number of the roll feed must be specified in every NC block.

139	**Deviation between encoder 1 and encoder 2 actual values too large - RESET necessary**
Cause:	The deviation between the actual values of encoder 1 and encoder 2 is higher than the value in function parameter FP6
Effect:	Machining is aborted and the axis stopped.
Remedy:	• Check and correct function parameter FP6 • Check connections in MASTERDRIVES MC (script file for encoder switchover) • Check your mechanical components as well Error message 139 can only be acknowledged by a RESET.

140	**Following error in standstill**
Cause:	The following error limit for standstill was exceeded at standstill: • Following error monitoring – at standstill (machine data 14) was entered incorrectly. • The value entered for "in position – exact stop window" (machine data 17) is greater than the value in "following error monitoring – at standstill" (machine data 14). • The axis was pushed out of position mechanically.
Effect:	The position control system is deactivated and the axis decelerates via "deceleration time during errors" (machine data 43).
Remedy:	• Check and correct the machine data. • Optimize the speed/current controller.

141	**Following error in motion**
Cause:	The following error limit for motion was exceeded during a traversing movement: • Following error monitoring – in motion (machine data 15) was entered incorrectly. • The mechanical system cannot follow the commands of the position controller. • Incorrect optimization of the position controller or speed controller. • The mechanical system is sluggish or blocked.
Effect:	The position control system is deactivated and the drive decelerates via "deceleration time during errors" (machine data 43).
Remedy:	• Check and correct the machine data. • Optimize the position controller or the speed controller. • Check the mechanical system.

142	**In position timer monitoring**
Cause:	The "in position – exact stop window" was not reached within the time specified in "in position – timer monitoring": • In position – exact stop window (machine data 17) too small • In position – timer monitoring (machine data 16) too short • Position controller or speed controller not optimized • Mechanical causes
Effect:	The position control system is deactivated.
Remedy:	Check and correct the machine data: • Optimize the position controller or speed controller • Check the mechanical system.

145	**Actual-value disable not allowed - axis standstill**
Cause:	The "digital input" with the "disable actual value" function was actuated while the roll feed was running.
Effect:	The axis movement is stopped via the deceleration ramp, the "disable actual value" function is not executed.
Remedy:	The "digital input" for "disable actual value" can only be actuated when the axis is stationary.

146	**Direction of movement not allowed**
Cause:	A positioning movement was aborted. When attempting to resume the movement at the point of interruption, the roll feed would have had to travel in the opposite direction to reach the programmed target position. This is inhibited by the setting of machine data 37 "response after abort". There are various possible reasons for the axis crossing the target position when a positioning movement is aborted: • Motor coastdown • The axis was moved intentionally, e.g. in setup mode.
Effect:	The axis movement is inhibited.
Remedy:	Move the axis in front of the target position in setup mode before continuing.

147	Machine data MD1 invalid
Cause:	Machine data MD1 can only be parameterized between 0 and 6
Effect:	The machine data is not adopted and cannot be activated.
Remedy:	The following parameter values are applicable if using the IM178: • 4 = IM178 with incremental position encoder • 5 = IM178 with absolute position encoder

148	Deceleration = 0
Cause:	The current deceleration value is 0, e.g. because of a RAM storage error or an error in the technology firmware.
Effect:	The position control system is deactivated and the drive is decelerated via the "deceleration time during errors" (machine data 43).
Remedy:	This error should not normally occur. It is used as an emergency stop feature for the technology software. Replace the hardware (M7; MCT).

149	Distance to go negative - RESET necessary
Cause:	Internal error in the technology software.
Effect:	The position control system is deactivated and the drive is decelerated via the "deceleration time during errors" (machine data 43).
Remedy:	RESET, this error should not actually occur. The technology software uses it as an emergency brake.

150	Slave axis already allocated to other master axis
Cause:	The selected NC program contains a slave axis which is already being used by another master axis (M7 only). Example: NC program 1, started in axis X, contains NC blocks for axes X and Y. NC program 2 is started in axis Z and contains NC blocks for axes Z and Y. This program is denied with warning 150, because axis Y is already being used by program 1.
Effect:	NC program execution is inhibited or aborted.
Remedy:	The same slave axis cannot be used simultaneously by several NC programs.

151	Slave axis operating mode not allowed
Cause:	The slave axis required by the master axis is not in "slave" mode (M7 only).
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	The slave axis must be switched to "slave" mode.

152	Slave axis operating mode changed during traversing
Cause:	The "slave" mode was deselected in the slave axis during the traversing movement (M7 only).
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	The slave axis must remain switched to "slave" mode.

153	Error in slave axis
Cause:	A warning is active in the slave axis required by the master axis (M7 only).
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	The NC program will only run if all of the axes it needs are error-free. To clear this warning, you must first clear all the warnings in the slave axis.

154	Follow-up mode in slave axis active
Cause:	The "follow-up mode" [FUM] control signal is active in the slave axis required by the master axis. A slave axis which is switched to follow-up mode cannot be operated by the master axis (M7 only).
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Deactivate follow-up mode in the slave axis.

155	Reset in slave axis active
Cause:	The "reset" [RST] control signal is active in the slave axis required by the master axis. A slave axis with an active reset cannot be used by the master axis (M7 only).
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Cancel the "reset" [RST] control signal in the slave axis.

156	Axis type (MD1) of slave axis not allowed
Cause:	An NC program was started in which a slave axis is defined as a roll feed axis type (M7 only). The warning is output in the master axis and indicates an illegal axis type in the slave axis.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Axes defined as roll feed axes can only be used in dedicated NC programs.

157	Slave axis, no axis enabling signal
Cause:	The axis enabling signal for the axis is not set in the control signals (M7 only).
Effect:	The axis cannot be moved.
Remedy:	Set the "Axis enable" control signal.

158	**Slave axis sampling time does not match master axis sampling time**
Cause:	Two different sampling times have been parameterized (M7 only).
Effect:	The axis cannot be moved.
Remedy:	Parameterize the sampling times to the same value and use main task "33" to transfer them back to the M7-FM.

159	**Master axis, no axis enabling signal**
Cause:	The axis enabling signal for the master axis was reset in the control signals as the axis was moving (M7 only). Either the master axis is traversing or one or more slave axes linked to it.
Effect:	All axes still moving are stopped.
Remedy:	Do not reset the axis enabling control signal until the axis or axes have reached a standstill.

160	**Setup speed = 0**
Cause:	The value entered in level 1 or level 2 for the [F_S] velocity level in setup mode is zero.
Effect:	The axis movement is inhibited.
Remedy:	Define a permissible velocity level for level 1 and/or level 2. The permissible value range is between 0.01 [1000*LU/min] and "traversing velocity – maximum" (machine data 23).

161	**Reference approach velocity = 0**
Cause:	The velocity value entered for "reference point - approach velocity" (machine data 7) is zero.
Effect:	The axis movement is inhibited.
Remedy:	Enter a permissible value for the approach velocity. The permissible value range is between 0.01 [1000*LU/min] and "traversing velocity - maximum" (machine data 23).

162	**Reference cutoff velocity = 0**
Cause:	The velocity value entered for "reference point - reducing velocity" (machine data 6) is zero.
Effect:	The axis movement is inhibited or stopped.
Remedy:	Enter a permissible value for the reducing velocity. The permissible value range is between 0.01 and 1000 [1000*LU/min].

165	**MDI block number not allowed**
Cause:	The MDI block number [MDI_NO] specified in the control signals is greater than 11.
Effect:	The axis movement is inhibited.
Remedy:	Define an MDI block number [MDI_NO] between 0 and 10.

166	**No position has been programmed in MDI mode**
Cause:	The "start" [STA] control signal was activated in MDI mode without initially transferring a positional value to the selected MDI block.
Effect:	The axis movement is inhibited.
Remedy:	Use the correct sequence: data transfer followed by axis start.

167	**No velocity has been programmed in MDI mode**
Cause:	The "start" [STA] control signal was activated in MDI mode without initially transferring a velocity value to the selected MDI block.
Effect:	The axis movement is inhibited.
Remedy:	Use the correct sequence: data transfer followed by axis start.

168	**G91 not allowed with MDI on the fly**
Cause:	G91 (incremental dimensions) was defined in the MDI block as the 1st G function for the MDI on-the-fly function.
Effect:	The axis movement is inhibited or stopped via the deceleration ramp.
Remedy:	The MDI on-the-fly function only allows G90 (absolute dimensions) as the 1st G function.

169	**Start conditions for flying MDI do not exist**
Cause:	• Control signal "reset technology" [RST] activated • Control signal "follow-up mode" [FUM] activated
Effect:	The "MDI on-the-fly" function is not executed.
Remedy:	Ensure that the control signals are activated correctly.

170	**Single block mode block does not exist**
Cause:	An NC block was started in single-block mode although a block has not yet been transferred.
Effect:	NC block execution is inhibited.
Remedy:	Transfer the block.

171	**Invalid axis parameterization (scanning time)**
Cause:	In the automatic program selected, axes are operated whose parameterization differs with respect to scanning time and shift.
Effect:	Automatic program cannot be started.
Remedy:	Check and correct axis parameterization.

172	**Program with this number does not exist**
Cause:	The program number specified in [PROG_NO] for automatic mode is not stored in the memory of the technology.
Effect:	NC program execution is inhibited.
Remedy:	• Transfer the program to the technology. • Select the correct program number.

173	**Program number not allowed**
Cause:	The program number specified in [PROG_NO] for automatic mode is not allowed.
Effect:	NC program execution is inhibited.
Remedy:	The permissible range for program numbers is between 1 and 200.

174	**Program number changed during traversing**
Cause:	The program number [PROG_NO] was changed while the program was running.
Effect:	NC program execution is aborted and the axis or axes are brought to a standstill via the deceleration ramp.
Remedy:	The program number must not be changed while the program is running.

175	**No block end programmed**
Cause:	The decoded NC block is not terminated with the following block identifier "0". You can use the "output actual values – decoder error location" task to read out the program number and block number where the block decoder detected an error.
Effect:	NC program execution is inhibited or aborted. Moving axes are stopped via the deceleration ramp.
Remedy:	Correct the block. The last block in the sequence must contain the following block identifier "0".

177	**Program number of block search forward does not exist**
Cause:	The program number for the main program (level 0), which was transferred with the block search function, does not exist
Effect:	NC program execution is inhibited.
Remedy:	Specify an existing main program number.

178	Program number of block search forward not allowed
Cause:	• The program number for the main program (level 0), which was transferred with block search, is different from the selected program number. • No breakpoint is known for the "automatic block search" function (a program abort has not yet occurred). • A different program number is stored as the breakpoint for the "automatic block search" function.
Effect:	NC program execution is inhibited.
Remedy:	For the block search function, the selected program number [PROG_NO] must be specified as the program number for the main program.

179	Program number of block search forward level 1/2 does not exist
Cause:	The subprogram number specified with block search for level 1 or level 2 does not exist.
Effect:	NC program execution is inhibited.
Remedy:	For the block search function, an existing program number must be specified as the subprogram number for level 1 or level 2.

180	Program number of block search forward level 1 <> program number of command
Cause:	The subprogram number transferred with block search for level 1 is not the same as the subprogram number in the NC block.
Effect:	NC program execution is inhibited.
Remedy:	For the block search function, the subprogram number specified in the NC block must be specified as the subprogram number for level 1.

181	Program number of block search forward level 2 <> program number of command
Cause:	The subprogram number transferred with block search for level 2 is not the same as the subprogram number in the NC block.
Effect:	NC program execution is inhibited.
Remedy:	For the block search function, the subprogram number specified in the NC block must be specified as the subprogram number for level 2.

183	Block number of block search forward level 0 does not exist
Cause:	The block number for the main program (level 0), which was transferred with block search, does not exist in the main program.
Effect:	NC program execution is inhibited.
Remedy:	For the block search function, an existing block number must be specified as the block number for the main program.

184	**Block number of block search forward level 0 is no UP call**
Cause:	The block number for the main program (level 0), which was transferred with block search, does not contain a subprogram call for subprogram level 1.
Effect:	NC program execution is inhibited.
Remedy:	For the block search function, a block number with a subprogram call must be specified as the block number for the main program (level 0) if a block search is to be performed in subprogram level 1.

185	**Block number of block search forward level 1 does not exist**
Cause:	The block number for subprogram level 1, which was transferred with block search, does not exist in the subprogram.
Effect:	NC program execution is inhibited.
Remedy:	For the block search function, a block number which exists in this subprogram must be specified as the block number for subprogram level 1.

186	**Block number of block search forward level 1 is no UP call**
Cause:	The block number for subprogram level 1, which was transferred with block search, does not contain a subprogram call for subprogram level 2.
Effect:	NC program execution is inhibited.
Remedy:	For the block search function, a block number with a subprogram call must be specified as the block number for subprogram level 1 if a block search is to be performed in subprogram level 2.

187	**Block number of block search forward level 2 does not exist**
Cause:	The block number for subprogram level 2, which was transferred with block search, does not exist in the subprogram.
Effect:	NC program execution is inhibited.
Remedy:	For the block search function, a block number which exists in this subprogram must be specified as the block number for subprogram level 2.

188	**Remaining number of loops block search forward level 1/2 not allowed**
Cause:	The remaining loop count transferred with block search for subprogram level 1 or 2 is greater than the programmed loop count.
Effect:	NC program execution is inhibited.
Remedy:	For the block search function, it is only allowed to specify a remaining loop count between 0 and the programmed loop count-1.

190	**Digital input not programmed**
Cause:	The NC block which was read in contains the "inprocess measurement" or "set actual value on-the-fly" function, although a digital input has not been programmed for this function (machine data 45).
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Program the digital input for the desired function.

191	**Digital input not actuated**
Cause:	Although the "external block change" function was programmed, the digital input was not actuated in order to trigger the external block change.
Effect:	The NC program is interrupted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	• Correct the program • Check the actuation of the digital input

195	**Negative overtravel reached**
Cause:	• Negative software limit switch position approached • "Software limit switches – negative" (machine data 12) entered incorrectly • The programmed position is less than the negative software limit switch • "Reference point – coordinate" (machine data 3) is less than the negative software limit switch • Incorrect encoder actual value
Effect:	The axis movement is stopped via the deceleration ramp.
Remedy:	• Check the machine data and the NC programs. • Check the encoder actual value.

196	**Positive overtravel reached**
Cause:	• Positive software limit switch position approached • "Software limit switches – positive" (machine data 13) entered incorrectly • The programmed position is greater than the positive software limit switch • "Reference point – coordinate" (machine data 3) is greater than the positive software limit switch • Incorrect encoder actual value
Effect:	The axis movement is stopped via the deceleration ramp.
Remedy:	• Check the machine data and the NC programs. • Check the encoder actual value.

200	**No position has been programmed in Automatic mode**
Cause:	No position has been programmed in the NC block for the roll feed version, although the axis number of the roll feed is specified.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	The axis number and the positional value must be specified in every NC block for the roll feed version.

201	**No velocity has been programmed in Automatic mode**
Cause:	The decoded NC block needs a path or axis velocity.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	When using linear interpolation with path velocity (G01), a path velocity must be defined with F. When using chaining with axis velocity (G77), the axis velocities must be defined with FX, FY, etc. When using roll feed with axis velocity (G01), the velocity must be defined with F.

202	**Axis unknown**
Cause:	An axis which does not exist was detected in the decoded NC block. A logical name (X, Y, Z, A, B, C) must be assigned to each axis with machine data 2 (axis assignment). Only these logical axis names can be used in the NC block. These errors cannot normally occur, since the logical axis names are verified when the NC blocks are entered. Exception: Machine data 2 (axis assignment) is changed afterwards. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

203	**1st G-function not allowed**
Cause:	The NC block which was read in contains an illegal 1st G function. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	The axis movement is inhibited or stopped via the deceleration ramp.
Remedy:	• MDI mode: Only G90 (absolute dimensions) or G91 (incremental dimensions) can be entered as the 1st G function. Only G91 is allowed for the roll feed version. • Automatic/single-block mode: Define a legal 1st G function according to the table (see the Programming Guide).

204	**2nd G- function not allowed**
Cause:	The NC block which was read in contains an illegal 2nd G function. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	The axis movement is inhibited or stopped via the deceleration ramp.
Remedy:	• MDI mode: Only G30 to G39 (acceleration override) can be entered as the 2nd G function. • Automatic/single-block mode: Define a legal 2nd G function according to the table (see the Programming Guide).

205	3rd G-function not allowed
Cause:	The NC block which was read in contains an illegal 3rd G function. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	The axis movement is inhibited or stopped via the deceleration ramp.
Remedy:	• MDI mode: No 3rd G function is allowed • Automatic/single-block mode: Define a legal 3rd G function according to the table (see the Programming Guide).

206	4th G-function not allowed
Cause:	The NC block which was read in contains an illegal 4th G function. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	The axis movement is inhibited or stopped via the deceleration ramp.
Remedy:	• MDI mode: No 4th G function is allowed • Automatic/single-block mode: Define a legal 4th G function according to the table (see the Programming Guide).

207	Zero offset not allowed without specified axis
Cause:	A zero offset without a specified axis has been found in the decoded NC block.
Effect:	The axis movement is inhibited or stopped via the deceleration ramp. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Remedy:	Correct the NC block.

208	D number is not allowed
Cause:	A D number greater than 20 was found in the decoded NC block.
Effect:	The axis movement is inhibited or stopped via the deceleration ramp. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Remedy:	Correct the NC block.

209	D number not allowed without specified axis
Cause:	A D number without a specified axis has been found in the decoded NC block.
Effect:	The axis is prevented from moving or stopped via the deceleration ramp. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Remedy:	Correct the NC block.

210	Interpolation of 3 axes not allowed
Cause:	The decoded NC block contains an interpolation of 3 or more axes. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block. Only 2D interpolation is allowed.

211	Shortest distance G68 and G91 not allowed simultaneously
Cause:	G function G68 (shortest path for rotary axis) was detected in the decoded NC block, although G91 (incremental dimensions) is active. Example: N10 G91 G68 X20.000 The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block. Function G68 can only be programmed in association with G90 (absolute dimensions).

212	Special function and axis combination not allowed simultaneously
Cause:	A different axis was programmed in the NC block following a special function (M7 only). Example: N10 G50 X100 F1000 N15 G90 Y200 wrong N15 G90 X200 right The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC program. The axis used in the NC block with the special function must also be programmed in the next NC block..

213	**Multiple D number not allowed**
Cause:	The decoded NC block contains several D numbers. Example: N1 G41 D3 D5. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

214	**Multiple acceleration behavior not allowed**
Cause:	The decoded NC block contains several mutually exclusive G functions from the acceleration override group (G30 to G39). Example: N1 G34 G35 The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

215	**Multiple special functions not allowed**
Cause:	The decoded NC block contains several mutually exclusive G functions from the special function group (G87, G88, G89, G50, G51). Example: N1 G88 G50 The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

216	**Multiple block transition not allowed**
Cause:	The decoded NC block contains several mutually exclusive G functions from the block transition group (G60, G64, G66, G67). Example: N1 G64 G66 X1.000 FX100.00 The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

217	Multiple axis programming not allowed
Cause:	The decoded NC block contains the same axis more than once. Example: N1 G90 G01 X100.000 X200.000 F100.00 The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

218	Multiple path condition not allowed
Cause:	The decoded NC block contains several mutually exclusive G functions from the preparatory function group (G00/G01/G76/G77). Example: N1 G01 (Linear interpolation) G77 (chaining) X10 F100. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

219	Multiple dimensions specification not allowed
Cause:	The decoded NC block contains several mutually exclusive G functions from the dimensional notation group (G90/G91). Example: N1 G90 G91. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

220	Multiple zero offset selection not allowed
Cause:	The decoded NC block contains several mutually exclusive G functions from the zero offset group (G53 to G59). Example: N1 G54 G58 The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

221	Multiple tool offset selection not allowed
Cause:	The decoded NC block contains several mutually exclusive G functions from the tool offset selection group (G43/G44). Example: N1 G43 G44 D2 The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

223	Subprogram number does not exist
Cause:	The decoded NC block contains a subprogram call, however the NC program which was called does not exist in the memory of the technology.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

224	Subprogram nesting depth not allowed
Cause:	The permissible nesting depth of subprograms was exceeded. Recursive calling of subprograms. The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC program. The permissible nesting depth for subprograms is 2 subprogram levels.

225	Simultaneous selection/deselection of collision monitoring not allowed
Cause:	The decoded NC block contains simultaneous selection and deselection of collision monitoring (G96/G97). Example: N1 G96 G97 X100 The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC block.

227	**Negative overtravel violated**
Cause:	The look-ahead function of the decoder has detected that the negative software limit switch will be crossed. See also error message "195: Negative overtravel reached". The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct traversing program. Check the machine data.

228	**Positive overtravel violated**
Cause:	The look-ahead function of the decoder has detected that the positive software limit switch will be crossed. See also error message "196: Positive overtravel reached". The NC program number and NC block number in which the NC block decoder detected the error can be read out with the "output actual values – decoder error location" task.
Effect:	NC program execution is inhibited or aborted, the axis is brought to a standstill via the deceleration ramp.
Remedy:	Correct the NC program. Check the machine data.

229	**IM 178 parameterization error**
Cause:	After starting up, the module reports a parameterization error (for a detailed description, see IM 178 manual).
Effect:	The module and the axes connected to it cannot be operated. The error cannot be acknowledged.
Remedy:	Proper parameterization of the module using Step7 HW Config. Once the amended parameterization has been loaded, the system has to be restarted.

230	**IM 178 sensor signal cable error (EXTF0) - RESET necessary**
Cause:	• Cable broken • Cable not properly connected • Short circuit (for a detailed description, see IM 178 manual)
Effect:	The axis is brought to a standstill via the deceleration ramp.
Remedy:	RESET, check sensor cable and repair or replace.

231	**IM 178 sensor error (EXTF1)**
Cause:	• Sensor cable problems • SSI sensor start / stop bit error (for a detailed description, see IM 178 manual)
Effect:	The axis is brought to a standstill via the deceleration ramp.
Remedy:	Check sensor / sensor cable, and repair or replace if necessary.

232	IM 178 zero mark error (EXTF2)
Cause:	Zero mark monitoring error (for a detailed description, see IM 178 manual).
Effect:	The axis is brought to a standstill via the deceleration ramp.
Remedy:	Check sensor / sensor cable, and repair or replace if necessary.

233	SSI sensor validity check operated
Cause:	The logged change in actual value from the SSI sensor signal has exceeded the set maximum over 4 consecutive cycles.
Effect:	The axis is brought to a standstill via the deceleration ramp.
Remedy:	Check parameterization and / or sensor.

241	Table assignment changed
Cause:	The table assignment has been changed from 1 table to 2 tables or vice-versa.
Effect:	NC tables cannot be processed.
Remedy:	Load the table again. **NOTE:** A table can only be loaded again if it is not selected. The warning is cleared automatically when the table has been successfully loaded.

242	Table 1 invalid
Cause:	Table 1 was not loaded correctly or has been reset.
Effect:	Table 1 cannot be processed.
Remedy:	Load table 1 again. **NOTE:** Table 1 can only be loaded again if it is not selected. The warning is cleared automatically when table 1 has been successfully loaded.

243	Table 2 invalid
Cause:	Table 2 was not loaded correctly or has been reset.
Effect:	Table 2 cannot be processed.
Remedy:	Load table 2 again. **NOTE:** Table 2 can only be loaded again if it is not selected. The warning is cleared automatically when table 2 has been successfully loaded.

244	Table 3 invalid
Cause:	Table 3 was not loaded correctly or has been reset.
Effect:	Table 3 cannot be processed.
Remedy:	Load table 3 again. **NOTE:** Table 3 can only be loaded again if it is not selected. The warning is cleared automatically when table 3 has been successfully loaded.

245	**Table 4 invalid**
Cause:	Table 4 was not loaded correctly or has been reset.
Effect:	Table 4 cannot be processed.
Remedy:	Load table 4 again. **NOTE:** Table 4 can only be loaded again if it is not selected. The warning is cleared automatically when table 4 has been successfully loaded.

246	**Table 5 invalid**
Cause:	Table 5 was not loaded correctly or has been reset.
Effect:	Table 5 cannot be processed.
Remedy:	Load table 5 again. **NOTE:** Table 5 can only be loaded again if it is not selected. The warning is cleared automatically when table 5 has been successfully loaded.

247	**Table 6 invalid**
Cause:	Table 6 was not loaded correctly or has been reset.
Effect:	Table 6 cannot be processed.
Remedy:	Load table 6 again. **NOTE:** Table 6 can only be loaded again if it is not selected. The warning is cleared automatically when table 6 has been successfully loaded.

248	**Table 7 invalid**
Cause:	Table 7 was not loaded correctly or has been reset.
Effect:	Table 7 cannot be processed.
Remedy:	Load table 7 again. **NOTE:** Table 7 can only be loaded again if it is not selected. The warning is cleared automatically when table 7 has been successfully loaded.

249	**Table 8 invalid**
Cause:	Table 8 was not loaded correctly or has been reset.
Effect:	Table 8 cannot be processed.
Remedy:	Load table 8 again. **NOTE:** Table 8 can only be loaded again if it is not selected. The warning is cleared automatically when table 8 has been successfully loaded.

A.2 Error Messages of the Technology for Task Management

Output Location of the Error Messages

The error messages of the technology are entered in the associated task box of data block GMC_DB_COM.

DBWx	Error number 1	HEZ	X_app{n}.I_fault1
DBWx+2	Error number 2	HEZ	X_app{n}.I_fault2

Error number 1 contains the actual error message. In the case of a large number of error messages, an extension is entered in error number 2 for more accurate diagnosis.

Below is a complete list of the error messages of the technology.

2001	**Jog feed rate 1 is too high**
Cause:	The velocity value in "level 1" for setup mode is greater than the maximum traversing velocity defined in machine data 23.
Effect:	The value entered for "level 1" is limited to the maximum traversing velocity (machine data 23).
Occurs with:	"Input setup velocities" task
Remedy:	Enter a legal velocity value for "level 1".

2002	**Jog feed rate 2 is too high**
Cause:	The velocity value in "level 2" for setup mode is greater than the maximum traversing velocity defined in machine data 23.
Effect:	The value entered for "level 2" is limited to the maximum traversing velocity (machine data 23).
Occurs with:	"Input setup velocities" task
Remedy:	Enter a legal velocity value for "level 2".

2003	**Feed rate in MDI is too high**
Cause:	The velocity value in the MDI block is greater than the maximum traversing velocity defined in machine data 23.
Effect:	The value entered for the MDI block is limited to the maximum traversing velocity (machine data 23).
Occurs with:	"Input MDI block" task
Remedy:	Enter a legal velocity value for the MDI block.

2004	**Negative feed rate values not allowed**
Cause:	A negative value was specified for the velocity.
Effect:	The original velocity value is retained.
Occurs with:	All tasks which contain a velocity value.
Remedy:	Enter a positive velocity value.

2006	Setup mode needed
Cause:	The teach-in function was not executed in setup mode.
Effect:	The task is not executed.
Occurs with:	"Input teach-in" task
Remedy:	Select setup mode.

2007	Negative dwell time not allowed or time value does not exist
Cause:	A negative dwell time or no dwell time (G04 with no time value) was specified in the NC program. "Error number 2" contains the NC block number in which the error occurred.
Effect:	The task is aborted or is not executed. The old value of the dwell time, if one exists, is not overwritten.
Occurs with:	"Input NC block/NC program" tasks
Remedy:	Enter a positive dwell time.

2008	Main command number not allowed
Cause:	The "general task header" contains an unknown main task number.
Effect:	The task is not executed.
Occurs with:	All tasks for which an unknown main task number is specified.
Remedy:	Specify a legal main task number.

2009	Command ID not allowed
Cause:	The "technological task header" contains an unknown task identifier.
Effect:	The task is not executed.
Occurs with:	All tasks which have a "technological task header" for which an unknown task identifier is specified.
Remedy:	Specify a legal task identifier.

2010	No transfer direction specified
Cause:	No transfer direction is specified in the "technological task header".
Effect:	The task is not executed.
Occurs with:	All tasks with a "technological task header".
Remedy:	Specify the transfer direction correctly.

2011	Both directions specified
Cause:	A unique transfer direction is not specified in the "technological task header", because both input (I) and output (Q) are enabled.
Effect:	The task is not executed.
Occurs with:	All tasks with a "technological task header".
Remedy:	Specify the transfer direction correctly.

2012	**Axis number not allowed**
Cause:	The absolute axis number entered in the general task header in GMC_DB_APP1 is less than 1 or greater than the total number of axes configured. The total number of axes is defined during commissioning in GMC_DB_ORG in the "axis descriptions" section. The absolute axis number entered is indicated in error number 2.
Effect:	The task is not executed.
Occurs with:	All tasks which require the specification of an axis number.
Remedy:	Specify a correct absolute axis number.

2013	**Command illegal for the axis type chosen (MD1)**
Cause:	A task which is not allowed for the selected axis type was initiated. The axis type is defined by machine data 1. Example: "Input/output roll feed velocity override" task with MD1=1 (axis type: axis with incremental encoder). This task is only allowed with MD1=5 (roll feed).
Effect:	The task is not executed.
Occurs with:	All tasks which depend on the axis type.
Remedy:	Use the correct tasks.

2014	**Read in not allowed**
Cause:	"Input (I)" is specified as the transfer direction in the "technological task header", although the selected task only allows "output (Q)".
Effect:	The task is not executed.
Occurs with:	All tasks with a technological task header which only allows "output (Q)".
Remedy:	Specify the correct transfer direction.

2015	**Read out not allowed**
Cause:	"Output (Q)" is specified as the transfer direction in the "technological task header", although the selected task only allows "input (I)".
Effect:	The task is not executed.
Occurs with:	All tasks with a technological task header which only allows "input (I)".
Remedy:	Specify the correct transfer direction.

2016	**Incorrect element pointer or number of elements**
Cause:	The element pointer in the "general task header" is negative or too large, or the value specified in "number of elements" is too large, or an incorrect data type has been specified.
Effect:	The task is not executed.
Occurs with:	All tasks.
Remedy:	Specify the element pointer, the number of elements and the data type correctly in accordance with the task description.

2017	Extension 1 is not allowed
Cause:	An illegal value is specified as extension 1 in the "technological task header".
Effect:	The task is not executed.
Occurs with:	All tasks with a "technological task header" which require extension 1.
Remedy:	Specify extension 1 correctly in accordance with the selected task.

2018	Extension 2 is not allowed
Cause:	An illegal value is specified as extension 2 in the "technological task header".
Effect:	The task is not executed.
Occurs with:	All tasks with a "technological task header" which require extension 2.
Remedy:	Specify extension 2 correctly in accordance with the selected task.

2019	Extension 3 is not allowed
Cause:	An illegal value is specified as extension 3 in the "technological task header".
Effect:	The task is not executed.
Occurs with:	All tasks with a "technological task header" which require extension 3.
Remedy:	Specify extension 3 correctly in accordance with the selected task.

2020	Extension 4 is not allowed
Cause:	An illegal value is specified as extension 4 in the "technological task header".
Effect:	The task is not executed.
Occurs with:	All tasks with a "technological task header" which require extension 4.
Remedy:	Specify extension 4 correctly in accordance with the selected task.

2021	Extension 5 is not allowed
Cause:	An illegal value is specified as extension 5 in the "technological task header".
Effect:	The task is not executed.
Occurs with:	All tasks with a "technological task header" which require extension 5.
Remedy:	Specify extension 5 correctly in accordance with the selected task.

2022	Extension 6 is not allowed
Cause:	An illegal value is specified as extension 6 in the "technological task header".
Effect:	The task is not executed.
Occurs with:	All tasks with a "technological task header" which require extension 6.
Remedy:	Specify extension 6 correctly in accordance with the selected task.

2023	Automatic operating mode necessary
Cause:	The block search was initiated but automatic mode was not selected.
Effect:	The task is not executed.
Occurs with:	"Input/output automatic/manual block search" task
Remedy:	Select automatic mode.

2024	Actual value analysis factor equals zero
Cause:	An actual value analysis factor with a value of 0 has been transferred.
Effect:	The task is not accepted.
Occurs with:	"Input machine data MD51 to MD55" task.
Remedy:	Enter an actual value analysis factor which is not equal to 0.

2026	NPV below lower limit
Cause:	The zero offset which was read in is below the lower limit defined for the input range. The number of the incorrect zero offset (54 to 59 for G54 to G59) is specified in "error number 2".
Effect:	The transfer is interrupted at the point where the incorrect zero offset occurs. The original contents of this zero offset and the following ones are retained.
Occurs with:	"Input zero offset" task.
Remedy:	Observe the input limits. The lower limit for the zero offset is -999 999 999.

2027	NPV above upper limit
Cause:	The zero offset which was read in is above the upper limit defined for the input range. The number of the incorrect zero offset (54 to 59 for G54 to G59) is specified in "error number 2".
Effect:	The transfer is interrupted at the point where the incorrect zero offset occurs. The original contents of this zero offset and the following ones are retained.
Occurs with:	"Input zero offset" task.
Remedy:	Observe the input limits. The upper limit for the zero offset is +999 999 999.

2030	First MD number is not allowed
Cause:	"Input / output MD1 to MD50" task: A value of 0 or greater than 50 has been specified for the "first MD number" (extension 2) in the "technological task header". "Input / output MD51 to MD750" task: A value less than 50 or greater than 70 has been specified for the "first MD number" (extension 2) in the "technological task header".
Effect:	The task is not executed.
Occurs with:	"Input/output machine data" task
Remedy:	Specify a start MD number between 1 and 50 or between 51 and 70.

2031	First MD number + number of MD numbers > Max. number of MD numbers
Cause:	A value is specified as the "Number of MD numbers" in the task header. This equals more than 50 or 70 when added to "First MD number". "First MD number" + "Number of MD numbers" <= 50 or 70.
Effect:	The task is not executed.
Occurs with:	"Machine data input / output" task.
Remedy:	Input a value for "Number of MD numbers" which does not exceed 50 or 70 when added to "First MD number".

2032	Multiple parameterization of axis assignment MD2 not allowed
Cause:	Axis is assigned more than once. As additional info, "Error number 2" specifies the number of the local M7 axis which has the same axis assignment.
Effect:	The task is not executed.
Occurs with:	"Activate machine data" task.
Remedy:	Correct machine data MD2 of the affected axis and transfer to the axis again. Then activate the machine data.

2033	First FP number not allowed
Cause:	A value of 0 or greater than 10 has been specified for the "first FP number" (extension 2) in the "technological task header".
Effect:	The task is not executed.
Occurs with:	"Input / output function parameters" task.
Remedy:	Specify a first FP number between 1 and 10.

2034	First FP number + number of FP number > max. number of FP numbers
Cause:	The value specified for "number of FP numbers" in the task header produces a value greater than 10 when added to the "first FP number". "First FP number" + "number of FP numbers" ≤ 10.
Effect:	The task is not executed.
Occurs with:	"Input / output function parameters" task.
Remedy:	Specify a "number of FP numbers" which does not produce a total greater than 10 when added to "first FP number".

2035	**Machine data value below lower limit**
Cause:	The machine data which was entered is below the lower limit defined for the input range. The number of the incorrect machine data is specified in "error number 2".
Effect:	The transfer is interrupted at the point where the incorrect machine data occurs. The original contents of this machine data and the following ones are retained.
Occurs with:	"Input machine data" task.
Remedy:	Observe the input limits.

2036	**Machine data value above upper limit**
Cause:	The machine data which was entered is above the upper limit defined for the input range. The number of the incorrect machine data is specified in "error number 2".
Effect:	The transfer is interrupted at the point where the incorrect machine data occurs. The original contents of this machine data and the following ones are retained.
Occurs with:	"Input machine data" task
Remedy:	Observe the input limits.

2038	**Machine data cannot be activated yet - axis standstill**
Cause:	Activation of the machine data while the axis was moving or while the NC program was running.
Effect:	The task is not executed.
Occurs with:	"Activate machine data" task
Remedy:	Wait until the axis is at a standstill, i.e. the "axis moves forwards" [FWD] and "axis moves backwards" [BWD] checkback signals must not be active. In automatic and single-block modes, the "function running" [FUR] checkback signal must not be active.

2039	**Limit switch negative > limit switch positive**
Cause:	The value of "software limit switches – negative" (machine data 12) is greater than the value of "software limit switches – positive" (machine data 13).
Effect:	The task is not executed.
Occurs with:	"Activate machine data" task
Remedy:	Specify the correct values for the software limit switches.

2040	**Ratio between acceleration and deceleration not allowed**
Cause:	The ratio between acceleration MD18 and deceleration MD19, or between acceleration breakpoint acceleration MD31 and deceleration breakpoint deceleration MD32 is illegal
Effect:	The task is executed. The ration is limited internally to 1:10.
Occurs with:	"Activate machine data" task
Remedy:	Parameterize the acceleration and deceleration settings correctly. The parameterized ratio must not exceed 1:10.

2042	Machine data 1 not allowed
Cause:	3 or 4 was entered in machine data 1 (axis type / encoder type).
Effect:	The task is not executed.
Occurs with:	"Input machine data" task
Remedy:	Enter MD1 correctly. Only values 1, 2 and 5 are allowed for MD1.

2045	Requested program does not exist
Cause:	The selected NC program number does not exist on the technology.
Effect:	The task is not executed.
Occurs with:	All output tasks which require the NC program number as an extension.
Remedy:	Specify an existing program number.

2047	Requested block does not exist
Cause:	The selected NC block number does not exist in the specified NC program.
Effect:	The task is not executed.
Occurs with:	All output tasks which require the NC block number as an extension.
Remedy:	Specify an existing block number.

2048	Program does not contain any blocks
Cause:	The selected NC program does not contain any blocks. A program will normally contain blocks, however it is possible for a program to have no blocks if all of the blocks were deleted by the "delete NC block" task.
Effect:	The task is not executed.
Occurs with:	"Output NC block number", "output NC program" and "output NC block" tasks.
Remedy:	Enter the blocks.

2049	No program exists
Cause:	The NC program memory is empty.
Effect:	The task is not executed.
Occurs with:	"Output NC program number" task.
Remedy:	Enter the programs.

2051	Program number not allowed
Cause:	A value of 0 or greater than 203 was entered as the program number.
Effect:	The task is not executed.
Occurs with:	All tasks which require the program number as an extension.
Remedy:	Specify a correct program number. The program numbers allowed are between 1 and 203.

2052	Block number not allowed
Cause:	A value of 0 or greater than 200 was entered as the block number.
Effect:	The task is not executed.
Occurs with:	All tasks which require the block number as an extension.
Remedy:	Specify a correct block number. The block numbers allowed are between 1 and 200.

2058	Block number in command is not the same as the block number in data
Cause:	The NC block number in the "technological task header" is not the same as the block number in the task data.
Effect:	The task is not executed.
Occurs with:	"Input NC block" task
Remedy:	The block numbers must match each other.

2059	Program number in command is not the same as the program number in data
Cause:	The NC program number in the task header is not the same as the program number(s) in the task data. The NC block number in which the program number is incorrect is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	"Input NC program" and "input NC block" tasks
Remedy:	The program numbers must match each other.

2060	Block number in data part not allowed
Cause:	NC block number 0 or a block number greater than 200 was detected in the task data on input.
Effect:	The task is aborted.
Occurs with:	"Input NC program" and "input NC block" tasks
Remedy:	Specify a correct block number. The block numbers allowed are between 1 and 200.

2061	Number of blocks/program not allowed
Cause:	Transfer of a program with more than 1000 (M7) or 50 (MCT) NC blocks
Effect:	The task is aborted.
Occurs with:	"Input NC program" and "input NC block" tasks
Remedy:	Enter programs with the correct length.

2062	Program memory full
Cause:	An attempt was made to transfer more than 1000 (M7) or 50 (MCT) NC blocks in S7 format to the technology.
Effect:	The task is aborted. In this state, 50 or 1000 NC blocks are stored in the memory of the technology.
Occurs with:	"Input NC program" and "input NC block" tasks
Remedy:	Delete NC programs which are no longer required.

2063	Deletion of program memory not allowed
Cause:	The NC program memory was cleared while an NC program was running.
Effect:	The task is not executed.
Occurs with:	"Clear NC program memory" task
Remedy:	Deselect automatic or single-block mode. An M7 computer contains several axes which share an NC program memory. You must therefore deselect automatic or single-block mode in all axes.

2069	Continuation ID not allowed
Cause:	A following block identifier greater than 19 was detected during transfer. The NC block number containing the illegal following block identifier is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	"Input NC program" and " input NC block" tasks
Remedy:	An NC block can only consist of a main block and up to 19 following blocks, i.e. the maximum following block identifier = 19.

2080	1st G function not allowed
Cause:	The 1st G function specified in the NC block is not allowed. For the roll feed version, a G function not equal to G91 was specified in MDI mode. The NC block number in which the 1st G function is not allowed is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	All tasks in which the 1st G function can be specified.
Remedy:	Specify a 1st G function which is allowed. Please refer to the Programming Guide for a description of permissible 1st G functions and their assignment in main and following blocks. Only G91 can be specified in MDI mode with the roll feed version.

2081	2nd G function not allowed
Cause:	The 2nd G function specified in the NC block is not allowed. The NC block number in which the 2nd G function is not allowed is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	All tasks in which the 2nd G function can be specified.
Remedy:	Specify a 2nd G function which is allowed. Please refer to the Programming Guide for a description of permissible 2nd G functions and their assignment in main and following blocks.

2082	3rd G function not allowed
Cause:	The 3rd G function specified in the NC block is not allowed. The NC block number in which the 3rd G function is not allowed is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	All tasks in which the 3rd G function can be specified.
Remedy:	Specify a 3rd G function which is allowed. Please refer to the Programming Guide for a description of permissible 3rd G functions and their assignment in main and following blocks.

2083	4th G function not allowed
Cause:	The 4th G function specified in the NC block is not allowed. The NC block number in which the 4th G function is not allowed is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	All tasks in which the 4th G function can be specified.
Remedy:	Specify a 4th G function which is allowed. Please refer to the Programming Guide for a description of permissible 4th G functions and their assignment in main and following blocks.

2084	Axis number not allowed
Cause:	• The axis number specified in the NC block is not allowed. • The axis number specified with teach-in cannot be found in the specified NC block.
Effect:	The task is aborted.
Occurs with:	"Input NC program", "input NC block" and "input teach-in" tasks
Remedy:	Specify a legal axis number: An axis name (X, Y, etc.) is generated for each axis in machine data 2 by means of an axis identifier. Only the defined axis identifiers can be used when defining an NC block: 1: X 2: Y 3: Z 4: A 5: B 6: C

2085	D number not allowed
Cause:	The D number specified in the NC block is not allowed. The NC block containing the D number is not allowed is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	"Input NC program" and "input NC block" tasks.
Remedy:	Only D numbers 0 (deselect tool offset) to 20 are allowed. D numbers can also only appear in the main block.

2086	Special function and axis combination not allowed simultaneously
Cause:	The special function and axis combination specified in the NC block are not allowed. Examples: N10 G50 X100 Y100 or N10 G88 X200 Z100 Only one axis name can be entered in an NC block containing G functions from the special function group. The NC block number in which the error occurred is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	"Input NC program" and "input NC block" tasks.
Remedy:	Correct the program.

2090	Subprogram number not allowed
Cause:	A value of 0 or greater than 200 was specified as the subprogram number. The NC block number containing the subprogram number which is not allowed is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	"Input NC program" and "input NC block" tasks.
Remedy:	Specify a correct subprogram number. The permissible subprogram numbers are between 1 and 200. The subprogram must be specified in the main block. Following blocks are not allowed.

2091	Number of loops not allowed
Cause:	A negative value or a value greater than 65535 was specified as the loop count. The NC block containing the illegal loop count is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	"Input NC program" and "input NC block" tasks
Remedy:	Specify a correct loop count. The permissible loop counts are between 0 and 65535. The loop count must be specified together with the subprogram number in the main block. Following blocks are not allowed.

2092	**Axis and path velocity not allowed simultaneously**
Cause:	The identifier for axis velocity (FN) and path velocity (F) were both programmed in the NC block. The NC block number in which the error occurred is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	"Input NC program" and "input NC block" tasks
Remedy:	Correct the program. The NC block can contain either an axis velocity or a path velocity. If the NC block is subdivided into a main block and following blocks, you can enter a path velocity and an axis velocity.

2096	**Block no. single block mode not allowed**
Cause:	A value not equal to 1 was specified as the block number.
Effect:	The task is not executed.
Occurs with:	"Input NC program" and "input NC block" tasks
Remedy:	Correct the program. Program numbers 201 (M7 and MCT) and 202-204 (M7) and block number 1 are defined for single-block mode. The NC block with block number 1 can contain following blocks.

2098	**Block search forward not allowed - program active**
Cause:	The block search function was initiated while the program was running.
Effect:	The task is not executed.
Occurs with:	"Input manual/automatic block search" tasks
Remedy:	A block search can only be performed when the program is inactive. A program is inactive for block search if: • It has already terminated; function running [FUR] is not active. • It has been interrupted; start [STA] is not active. • A new start is required in order to continue the program; [FUE] is active.

2099	**Position does not exist**
Cause:	No positional value was specified in an NC block with a special function (G50, G51, G88, G89 or G79). The NC block number in which the error occurred is specified in "error number 2".
Effect:	The task is aborted.
Occurs with:	"Input NC program" and "input NC block" tasks
Remedy:	Correct the program. A positional value must be specified in an NC block with a special function.

2110	Encoder changeover not allowed
Cause:	The supplementary conditions for encoder changeover were not fulfilled at the moment encoder changeover took place
Effect:	The task is not executed.
Occurs with:	"Activate encoder changeover input/output" tasks.
Remedy:	You can switch to another encoder only when the axis is stationary and checkback signal [FUR] "Machining in progress" = 0 signal.

2111	Command-value switchover not allowed
Cause:	Roll feed setpoint switchover • During an active movement • When a "warning" [WARN] is active • When another setpoint switchover is already active • When a function has already been initiated (function running [FUR])
Effect:	The task is not executed.
Occurs with:	"Input roll feed setpoint switchover" task
Remedy:	A setpoint switchover may only be performed when the axes are stationary.

2112	Invalid axis number
Cause:	When checking the axis number, it was ascertained that the authorized input limits had been exceeded. The invalid number is shown in error number 2 as additional information.
Effect:	The task is not executed.
Occurs with:	"Axis execution / enable, cam controller" task.
Remedy:	Observe the input limits.

2113	Enable value not allowed
Cause:	When enabling the virtual master and / or cam controller, an invalid value has been transferred. The invalid number is shown in error number 2 as additional information.
Effect:	The task is not executed.
Occurs with:	"Axis execution / enable" task.
Remedy:	Observe the input limit.

2114	Reduction factor / shift of an axis is not allowed
Cause:	When checking the reduction factor / shift, it was ascertained that the valid input limits have been exceeded. The invalid number is shown in error number 2 as additional information.
Effect:	The task is not executed.
Occurs with:	"Axis execution / enable" task.
Remedy:	Observe the input limits. The reduction factor has a value range of 1.. 32. Shift must be a value between 0 and 31. For a single axis, the shift cannot exceed the reduction value.

2115	Cam controller master value source not allowed
Cause:	A non-existent master value source has been selected. The source is shown in error number 2 as additional information.
Effect:	The task is not executed.
Occurs with:	"Cam controller" task.
Remedy:	Observe the input limits.

2116	Invalid cam parameterization
Cause:	A non-existent cam has been parameterized. The invalid cam is shown in error number 2 as additional information.
Effect:	The task is not executed.
Occurs with:	"Cam controller" task
Remedy:	Observe the input limits.

2117	Valid number of peripheral allocations exceeded
Cause:	During the peripheral allocations check, an infringement of the valid input limits was ascertained. The invalid number is shown in error number 2 as additional information.
Effect:	The task is not executed.
Occurs with:	"Allocation of peripheral DI / DO" task
Remedy:	Observe the input limits.

2118	Valid peripheral input / output number exceeded
Cause:	During the input / output number check, an infringement of the valid input limits was ascertained. The invalid number is shown in error number 2 as additional information.
Effect:	The task is not executed.
Occurs with:	"Allocation of peripheral DI / DO" task
Remedy:	Observe the input limits.

2119	Local axis number not allowed
Cause:	During the local axis number check, an infringement of the valid input limits was ascertained. The invalid number is shown in error number 2 as additional information.
Effect:	The task is not executed.
Occurs with:	"Allocation of peripheral DI / DO" task
Remedy:	Observe the input limits.

2120	Axis parameterization not allowed (operation status)
Cause:	The axis cannot be reparameterized as the checkback message "Axis is being executed" is active.
Effect:	The task is rejected.
Occurs with:	"Axis execution / enable" task
Remedy:	Stop axis and reset enable. Then trigger the task again.

2121	**Axis maximum exceeded or axis minimum not reached**
Cause:	When checking the axis number, it was ascertained that the number was in excess of the maximum or below the minimum valid input.
Effect:	The task is rejected. The original content is retained.
Occurs with:	"Axis execution / enable" task
Remedy:	Observe the input limits.

2122	**Axis input / output number not allowed**
Cause:	During the input / output number check, an infringement of the valid input limits was ascertained. The allocation in question is displayed as additional information in error number 2.
Effect:	The task is not executed.
Occurs with:	"Allocation of peripheral DI / DO" task
Remedy:	Observe the input limits.

2123	**Double configuration of inputs / outputs not allowed**
Cause:	During the input / output number check, a double configuration was discovered. The allocation in question is displayed as additional information in error number 2.
Effect:	The task is not executed.
Occurs with:	"Allocation of peripheral DI / DO" task
Remedy:	Avoid double configurations.

2124	**Transfer of a partial task is not allowed**
Cause:	When checking the task length, it was ascertained that only a partial task was transferred. The incorrect length in bytes is displayed as additional information in error number 2.
Effect:	The task is not executed.
Occurs with:	"Allocation of peripheral DI / DO" and "Axis execution / enable" tasks
Remedy:	Transfer of the complete task

2125	**Cam parameterization not allowed**
Cause:	Cams are limited to a range of 0 to rotary axis length for rotary axes. You have parameterized a cam which violates these limits. Error number 2 displays the affected cam as additional info.
Effect:	The task is not executed.
Occurs with:	"Cam controller" task
Remedy:	Parameterize within input limits.

2126	Hysteresis parameterization not allowed
Cause:	The hysteresis is limited to the 0 to rotary axis length range for rotary axes. Negative hysteresis values are rejected for linear axes.
Effect:	The task is not executed.
Occurs with:	"Cam controller" task
Remedy:	Parameterize within input limits.

2130	MCT parameter index not allowed
Cause:	A parameter index which does not exist was selected.
Effect:	The task is not executed.
Occurs with:	This error cannot normally occur.
Remedy:	Specify a correct MCT parameter index.

2131	MCT parameter number not allowed
Cause:	A parameter number which does not exist was selected.
Effect:	The task is not executed.
Occurs with:	This error cannot normally occur.
Remedy:	Specify a correct MCT parameter number.

2135	Master value synchronization: selection of synchronization mode not allowed
Cause:	The preset synchronization mode is invalid.
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters for offset angle setting" tasks
Remedy:	Transfer valid synchronization mode: 0: shortest path 1: in positive direction 2: in negative direction

2136	Offset angle setting: Correction type selection not allowed
Cause:	The correction type selected is invalid.
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters for offset angle setting" tasks
Remedy:	Transfer valid correction type: 0: shortest path 1: as set

2137	**Offset angle setting: Selection of mode relative to offset angle setting not allowed**
Cause:	The specified mode relative to the offset angle setting is not allowed
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters for offset angle setting" tasks.
Remedy:	Upload valid mode for relative offset angle setting: 0: Additive 1: Delete residual offset

2140	**Offset angle setting: jog offset angle – adjustment speed outside input limits**
Cause:	The adjustment speed setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters for offset angle setting" tasks
Remedy:	The adjustment speed setting must be between 0.00 and 20 000 000.00.

2141	**Offset angle setting: jog offset angle – acceleration outside input limits**
Cause:	The acceleration setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters for offset angle setting" tasks
Remedy:	The acceleration setting must be between 0.00 and 20 000 000.00.

2142	**Offset angle setting: jog offset angle – delay outside input limits**
Cause:	The delay setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters for offset angle setting" tasks
Remedy:	The delay setting must be between 0.00 and 20 000 000.00.

2143	**Offset angle setting: offset angle compensation movement – adjustment speed outside input limits**
Cause:	The adjustment speed setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters for offset angle" tasks
Remedy:	The adjustment speed setting must be between 0.00 and 20 000 000.00.

2144	**Offset angle setting: offset angle compensation movement - acceleration outside input limits**
Cause:	The acceleration setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters for offset angle" tasks
Remedy:	The acceleration setting must be between 0.00 and 20 000 000.00.

2145	**Real master: filter time for input signal outside input limits**
Cause:	The filter time setting is outside the valid limits
Effect:	The task is not executed.
Occurs with:	"Real master" tasks.
Remedy:	The filter time setting must be between 0 and 100.

2146	**Real master: dead time compensation outside input limits**
Cause:	The value for dead time compensation is outside the permissible input limits.
Effect:	The task is not executed.
Occurs with:	"Real master" tasks.
Remedy:	The dead time compensation value must be within the 0.00 to 100.00 range.

2147	**Real master: axis cycle length -output outside input limits**
Cause:	The axis cycle length output setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Real master" tasks.
Remedy:	The axis cycle length output setting must be between 0 and $2^{31} – 1$.

2148	**Real master: axis cycle length - input outside input limits**
Cause:	The axis cycle length input setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Real master" tasks.
Remedy:	The axis cycle length output setting must be between 0 and $2^{31} – 1$.

2149	**Real master: filter time constant for dead time compensation outside input limits**
Cause:	The filter time constant for dead time compensation setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Real master" tasks.
Remedy:	The filter time constant for dead time compensation setting must be between 0 and 100.

2150	**Real master: rated control speed outside input limits**
Cause:	The rated control speed setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Real master" tasks.
Remedy:	The rated control speed setting must be between 0.00 and 20 000 000.00.

2151	**Master value correction: acceleration outside input limits**
Cause:	The acceleration setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Master value correction" tasks
Remedy:	The acceleration setting must be between 0.00 and 20 000 000.00.

2152	**Master value correction: max. correction speed outside input limits**
Cause:	The max. correction speed setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Master value correction" tasks
Remedy:	The max. correction setting must be between 0.00 and 20 000 000.00.

2153	**Master value correction: master value 1 outside input limits**
Cause:	The master value 1 setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Master value correction" tasks
Remedy:	The master value 1 setting must be between 0 and $2^{31} - 1$.

2154	**Master value correction: master value 2 outside input limits**
Cause:	The master value 2 setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Master value correction" tasks
Remedy:	The master value 2 setting must be between 0 and $2^{31} - 1$.

2155	**Master value correction: master value 2 scaling outside input limits**
Cause:	The numerator / denominator settings for master value 2 are outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Master value correction" tasks
Remedy:	Set numerator -32768 to 32767, denominator 1 to 32767.

2156	**Master value correction: function selection not allowed**
Cause:	The function selection is invalid.
Effect:	The task is not executed.
Occurs with:	"Master value correction" tasks
Remedy:	Transfer valid function selection: 0: Master value correction 1: Master value adjustment

2157	**Master value correction: axis cycle length for master value output outside input limits**
Cause:	The axis cycle length for master value output setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Master value correction" tasks.
Remedy:	The axis cycle length for master value output setting must be between 0 and $2^{31} – 1$.

2158	**Master value correction: Rated velocity master value 1 not allowed**
Cause:	The value for rated velocity master value 1 is outside the permissible input limits.
Effect:	The task is not executed.
Occurs with:	"Master value correction" tasks.
Remedy:	The value must lie within range 0.00 to 20000000.00.

2159	**Master value correction: Rated velocity master value 2 not allowed**
Cause:	The value for rated velocity master value 2 is outside the permissible input limits.
Effect:	The task is not executed.
Occurs with:	"Master value correction" tasks.
Remedy:	The value must lie within range 0.00 to 20000000.00.

2160	**Catch-up: deceleration delay outside input limits**
Cause:	The deceleration delay setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Catch-up synchronization parameters" tasks
Remedy:	The deceleration delay setting must be between 0 and $2^{31} – 1$.

2161	**Catch-up: deceleration rounding time outside input limits**
Cause:	The deceleration rounding time setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Catch-up synchronization parameters" tasks
Remedy:	The deceleration rounding time setting must be between 0 and 6000.

2162	**Catch-up: acceleration outside input limits**
Cause:	The acceleration setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Catch-up synchronization parameters" tasks
Remedy:	The acceleration setting must be between 0 and $2^{31} – 1$.

2163	Catch-up: acceleration rounding time outside input limits
Cause:	The acceleration rounding time setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Catch-up synchronization parameters" tasks
Remedy:	The acceleration rounding time must be between 0 and 6000.

2164	Catch-up: deceleration delay positioning outside input limits
Cause:	The deceleration delay positioning setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Catch-up synchronization parameters" tasks.
Remedy:	The deceleration delay positioning setting must be between 0 and $2^{31} – 1$.

2165	Catch-up: acceleration positioning outside input limits
Cause:	The acceleration positioning setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Catch-up synchronization parameters" tasks
Remedy:	The acceleration positioning setting must be between 0 and $2^{31} – 1$.

2166	Catch-up: rounding mode not allowed
Cause:	The rounding mode selected is not allowed.
Effect:	The task is not executed.
Occurs with:	"Catch-up synchronization parameters" tasks
Remedy:	Transfer valid rounding mode: 0: Rounding not active on sudden reductions of the input value during the acceleration process 1: Rounding always active

2200	No data block for traversing table assigned
Cause:	An attempt was made to access an NC table to which an M7 data block number has not yet been assigned. The table number which has not been assigned is indicated in error number 2.
Effect:	The task is not executed.
Occurs with:	All tasks associated with the contents of the table.
Remedy:	Before the NC tables can be addressed, the M7 data blocks must be assigned to the table numbers with the "accept NC table" task.

2201	Table does not exist
Cause:	The selected table does not exist. The number of the table which does not exist is indicated in error number 2.
Effect:	The task is not executed.
Occurs with:	All tasks associated with the contents of the table.
Remedy:	Input an existing traversing table number.

2202	Node number > number of nodes not allowed
Cause:	An attempt was made to access an interpolation point number which is greater than the current number of points in the table. The illegal interpolation point number is indicated in error number 2.
Effect:	The task is not executed.
Occurs with:	All tasks associated with the contents of the table.
Remedy:	Specify an existing interpolation point number. The illegal interpolation point number specified in the task is stored in DBWy+12 (starting at interpolation point number x).

2203	Node number > max. number of nodes not allowed
Cause:	An attempt was made to access an interpolation point number which is greater than the maximum number of points in a table. The illegal interpolation point number is indicated in error number 2.
Effect:	The task is not executed.
Occurs with:	All tasks associated with the contents of the table.
Remedy:	Specify a permissible interpolation point number. The illegal interpolation point number specified in the task is stored in DBWy+12 (starting at interpolation point number x) If the technology is operated on an M7-FM, a table can contain a maximum of 1024 interpolation points. If the technology is operated on a MASTERDRIVES MC, a table (one or two tables), depending on the configuration, can have a maximum of either 100 or 200 interpolation points. From MASTERDRIVES MC Version V1.4, a maximum of 8 tables with a total of max. 400 interpolation points are available. The illegal interpolation point number specified in the task is stored in DBWy+12 (starting at interpolation point number x).

2204	**Too many nodes - insertion not possible**
Cause:	Inserting another interpolation point in the table would increase the number of interpolation points beyond the maximum limit. The permissible number of interpolation points is indicated in error number 2.
Effect:	The task is not executed.
Occurs with:	All tasks which increase the number of interpolation points in a table.
Remedy:	You cannot enter a greater number of interpolation points than can be stored in the table. If the technology is operated on an M7-FM, a table can contain a maximum of 1024 interpolation points. If the technology is operated on a MASTERDRIVES MC, a table (one or two tables), depending on the configuration, can have a maximum of either 100 or 200 interpolation points. From MASTERDRIVES MC Version V1.4, a maximum of 8 tables with a total of max. 400 interpolation points are available.

2205	**Node number < 1 not allowed**
Cause:	The specified interpolation point number is less than 1. The illegal interpolation point number is indicated in error number 2.
Effect:	The task is not executed.
Occurs with:	All tasks associated with the contents of the table.
Remedy:	Enter an interpolation point number greater than 0. The illegal interpolation point number specified in the task is stored in DBWy+12 (starting at interpolation point number x).

2206	**Node number gaps not allowed**
Cause:	The insertion of interpolation points would produce a gap in the table.
Effect:	The task is not executed.
Occurs with:	All tasks which increase the number of interpolation points in a table.
Remedy:	If interpolation points are inserted in a table, they must either be inserted within the interpolation points which already exist or starting at the highest interpolation point number. The illegal interpolation point number specified in the task is stored in DBWy+12 (starting at interpolation point number x).

2207	**Too many nodes - input not possible**
Cause:	Inserting another interpolation point in the table, starting at the selected interpolation point number, would increase the number of interpolation points beyond the maximum limit. The current number of interpolation points is indicated in error number 2.
Effect:	The task is not executed.
Occurs with:	All tasks which increase the number of interpolation points in a table.
Remedy:	You cannot enter a greater number of interpolation points than can be stored in the table. If the technology is operated on an M7-FM, a table can contain a maximum of 1024 interpolation points. If the technology is operated on a MASTERDRIVES MC, a table (one or two tables), depending on the configuration, can have a maximum of either 100 or 200 interpolation points. From MASTERDRIVES MC Version V1.4, a maximum of 8 tables with a total of max. 400 interpolation points are available.

2208	**Number of interpolation points not allowed**
Cause:	The number of interpolation points entered is less than 1 or exceeds the maximum value for a table. The current number of interpolation points is indicated in error number 2.
Effect:	The task is not executed.
Occurs with:	All tasks associated with the contents of the table.
Remedy:	Limit the number of interpolation points to within the valid range. If the technology is operated on an M7-FM, a table can contain a maximum of 1024 interpolation points. If the technology is operated on a MASTERDRIVES MC, a table (one or two tables), depending on the configuration, can have a maximum of either 100 or 200 interpolation points. From MASTERDRIVES MC Version V1.4, a maximum of 8 tables with a total of max. 400 interpolation points are available.

2210	**Table manipulation faulty**
Cause:	An error was discovered when the table was transferred. The exact cause of the error can be displayed using the "Output table status" order.
Effect:	The task is not executed.
Occurs with:	"Transfer table" task.
Remedy:	The interpolation points for the master axis absolutely must have increasing position values.

2211	**Table manipulation not allowed**
Cause:	While an NC table was active, an attempt was made to change the table assignment, to transfer the table again or to reset the table.
Effect:	The task is not executed.
Occurs with:	"Accept NC table" task with accept and reset table function.
Remedy:	Deselect the active table number or deselect the table function (FUNCTION).

2212	Table manipulation – table number selected not allowed
Cause:	A table has been selected whose number does not exist on the target system.
Effect:	The task is not executed.
Occurs with:	"Transfer table" task with "transfer and reset" table function.
Remedy:	Table number 1 to 8 on the M7-FM and 1 and 2 on MASTERDRIVES MC. From MASTERDRIVES MC Version V1.4, a maximum of 8 tables are available.

2213	Table manipulation – execution function not allowed
Cause:	The function selected does not exist.
Effect:	The task is not executed.
Occurs with:	"Transfer table" task with "transfer and reset" table function.
Remedy:	Transfer valid execution function. 1: Reset table 2: Adopt table 3: Adopt all tables

2214	Table manipulation – M7-FM: selected data block not allowed
Cause:	The number of the data block selected is invalid. The number of the invalid data block is displayed in error number 2.
Effect:	The task is not executed.
Occurs with:	"Transfer table" task with the "Adopt" table function.
Remedy:	Set the data block number to within the valid number range. Valid DB number ranges are between 1000 and 2000 and 4000 and 30000.

2215	Numerator and / or denominator settings outside input limits
Cause:	The numerator and / or denominator settings for the gear factor are outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks.
Remedy:	Numerator –32767 to +32767, denominator 1 to +32767.

2216	Inserter / ejector: length outside table input limits
Cause:	The parameterized insertion or injection length setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	The setting must not be less than 0 or greater than the maximum of $2^{31} - 1$.

2217	Inserter / ejector: ramp path outside input limits
Cause:	The parameterized ramp setting for insertion or rejection mode is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	The setting must not be less than 0 or greater than the maximum of $2^{31} - 1$.

2222	Position correction: max. correction setting / control cycle not allowed
Cause:	The maximum correction / control cycle setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	The maximum correction / control cycle setting must be within the valid range of 0 to 10000 [LU/Cyc].

2223	Position correction: printed mark definition not allowed
Cause:	The definition of the printed mark is invalid.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	Enter valid printed mark definition: 0: printed mark not on same axis 1: printed mark on same axis

2224	Position correction: set position definition not allowed
Cause:	The set position definition is invalid.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	MCT: only 0 or 1 are valid settings for the set position definition M7-FM: 0 is the only valid setting for the set position definition. 0: Correction to fixed set position 1: Correction to adjustable set position

2225	Position correction: enable definition
Cause:	The enable position correction definition is invalid.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	The enable value can only be set to 0 or 1. 0: No position correction 1: Position correction to printed mark

2226	M7 FM master axis: axis number not allowed
Cause:	The axis number given as the master axis is invalid.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	The axis number of the master on the M7–FM must be within the following ranges, which vary according to the version: M7-FM 8 axis version: 1 to 8 M7-FM 16+ axis version: 1 to 16+ A check is also run as to whether the actual axis number defined is still valid.

2227	Master axis: master value source not allowed
Cause:	The parameterized value for the master value source is not allowed.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	Valid values are: 0: actual value control 1: set value control 2: virtual master

2228	M7 FM master axis: actual value control on the same axis not allowed
Cause:	An attempt was made to set the actual value of one and the same axis as the set value.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	Avoid actual value control on the same axis.

2230	Virtual master acceleration / deceleration outside input limits
Cause:	The parameterized acceleration / deceleration setting for the virtual master is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	The parameterized acceleration setting must be between 0 and 2^{31} –1 [1000*LU/s^2].

2232	Virtual master: Linear / rotary axis outside input limits
Cause:	The linear / rotary (axis cycle) setting for the virtual master is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	The rotary axis setting must be between 1 and 2^{31} – 1.

2233	**Virtual master – speed too high**
Cause:	The speed of the virtual master is outside the valid speed range.
Effect:	The task is not executed.
Occurs with:	"Input synchronization parameters" tasks
Remedy:	The speed selected must not be greater than the traversing speed – at the maximum. This machine data (MD23) sets the maximum for all speed inputs. Furthermore, the velocity must not be higher than $\frac{\text{AxisCycleLength}}{2}$ per controller cycle

2234	**Master axis: Linear/rotary axis outside input limits**
Cause:	The value for linear/rotary axis (axis cycle) master is outside the permissible input limits.
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters input" tasks
Remedy:	The rotary axis value must be within the 1 to 2^{31} – 1 range.

2235	**Master axis: Rated velocity outside input limits**
Cause:	The value for rated velocity master is outside the permissible input limits.
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters input" tasks
Remedy:	The rated velocity value master must be within the 0.00 to 20 000 000.00 range.

2236	**Slave axis: Rated velocity outside input limits**
Cause:	The value for rated velocity slave is outside the permissible input limits.
Effect:	The task is not executed.
Occurs with:	"Synchronization parameters input" tasks
Remedy:	The rated velocity value slave must be within the 0.00 to 20 000 000.00 range.

2240	**Table manipulation: master axis scaling outside input limits**
Cause:	The numerator / denominator settings for the master axis (x axis) scaling are outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Input table parameters" tasks
Remedy:	Numerator –32768 to +32767, denominator 1 to +32767.

2241	**Table manipulation: slave axis scaling outside input limits**
Cause:	The numerator / denominator settings for the slave axis (y axis) scaling are outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Input table parameters" tasks
Remedy:	Numerator –32768 to +32767, denominator 1 to +32767.

2242	**Table manipulation: master axis set value outside input limits**
Cause:	The master axis set value setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Input table parameters" tasks
Remedy:	The set value must be between 0 and $2^{31} – 1$.

2250	**Parameter limits exceeded**
Cause:	The parameters setting is outside the valid input limits.
Effect:	The task is not executed.
Occurs with:	"Save data to RAM / EEPROM, traversing table parameters, offset angle setting and synchronization parameters" task. Error number 2 specifies the index (element pointer EZ) of the affected parameter as additional info.
Remedy:	Set parameters within input limits.

2251	**M7: data transfer not possible as data backup is still running**
Cause:	A backup process to save parameters, a table (after successful takeover) or an automatic block to the M7 memory card was still active when an attempt was made to transfer a new task to the M7.
Effect:	The task is not executed.
Occurs with:	All orders
Remedy:	Execute task again or stop the save to the memory card ("Write data to EEPROM / RAM" task) and save the data to the S7-CPU if necessary.

A.3 Error Messages of GMC_FB_MCT

Output Location of the Error Messages

The error messages of function block GMC_FB_MCT are entered in the associated task box of data block GMC_DB_COM.

DBWx	Error number 1	HEZ	X_app{n}.I_fault1
DBWx+2	Error number 2	HEZ	X_app{n}.I_fault2

Error number 1 contains the actual error message. In the case of a large number of error messages, an extension is entered in error number 2 for more accurate diagnosis.

Below is a complete list of error messages from function block GMC_FB_MCT.

4001	**Input check incorrect**
Cause:	The task transferred to GMC_FB_MCT was denied due to an error during the GMC_FB_MCT input check. The exact cause of the error is specified in error number 2.
Effect:	The task is not executed.

1	**Axis is not an MCT axis**
Cause:	The absolute axis number specified in the task header (DWx+2) is not an MCT axis according to the task description in GMC_DB_ORG (type of axis). This error cannot normally occur, because GMC_FB_JOB only passes a task to GMC_FB_MCT if an MCT axis is used. GMC_FB_JOB error or GMC_DB_AWP1 overwritten.

2	**Axis number < 1**
Cause:	The absolute axis number specified in the task header (DWx+2) is zero or negative. This error cannot normally occur, because the range has already been monitored by GMC_FB_JOB.

3	**Axis number > total number of axes not allowed**
Cause:	The absolute axis number specified in the task header (DWx+2) is greater than the total number of axes specified in GMC_DB_ORG, in the axis description section. This error cannot normally occur, because the range has already been monitored by GMC_FB_JOB.

4	**Data type not allowed**
Cause:	The data type specified in the task header (DWx+6) is not allowed. A definitive data type is specified for each task in the task descriptions. GMC_FB_MCT only checks whether the data type is <= zero and >= 6.
Remedy:	The data type must be specified correctly.

5	**Number of elements < 0 not allowed**
Cause:	The number of elements specified in the task header (DWx+8) is negative. The permissible entries for the number of elements are defined in the task descriptions.
Remedy:	The number of elements must be specified correctly.

6	**Command data too big**
Cause:	The length of the task data specified by the number of elements in the task header (DWx+8) is greater than 190 bytes. The permissible entries for the number of elements are defined for each task in the task descriptions.
Remedy:	The number of elements must be specified correctly. In special situations, where data are to be read into the technology more quickly, the task data can be transferred together with the technological task header. In this case, you should ensure that the sum of the general task header, technological task header and task data does not exceed 190 bytes.

4002	Command triggering incorrect
Cause:	After a successful input check, the task could not be issued to the MASTERDRIVES MC. The exact cause of the error is specified in error number 2.
Effect:	The task is aborted.

1	Time overflow
Cause:	The task issued by GMC_FB_MCT was not acknowledged within 200 OB1 cycles. Task acknowledgement error on MASTERDRIVES MC.
Remedy:	Restart the MASTERDRIVES MC.

2	PKW counter overflow
Cause:	The message frame transmitted via PROFIBUS was not answered. The task is repeated 3 times before an error is output. • The MASTERDRIVES MC addressed via PROFIBUS is not accessible. • The configured PROFIBUS address (S7 hardware configuration) or the PROFIBUS address set on the MASTERDRIVES MC (P918) is incorrect. • The PROFIBUS cable is defective.
Remedy:	Check the settings and connections.

4003	S7 function SFC5 signals error upon input
Cause:	GMC_FB_MCT uses S7 system function SFC5 (GADR_LGC) internally. SFC5 indicated an error during transfer of the data to the MASTERDRIVES MC (input). The exact cause of the error is specified in error number 2.
Effect:	The task is aborted.
Remedy:	Since SFC5 receives data from GMC_FB_MCT, the following errors should not occur (exception: error in hardware configuration). A description of SFC5 and the errors generated can be found in the S7 Reference Manual – Standard and System Functions.

8094	No subnet was configured with the specified SUBNETID
Cause:	The entry in the S7 hardware configuration and the entry in GMC_DB_ORG (X_mct.i_subnet_id) do not match.
Remedy:	Correct the entry in GMC_DB_ORG (X_mct.i_subnet_id).

8095	Illegal value for the RACK parameter
Cause:	The entry in the S7 hardware configuration (PROFIBUS address) and the entry in GMC_DB_ORG (X_axis.{axis_name}.i_station_no) do not match.
Remedy:	Correct the entry in GMC_DB_ORG (X_axis.{axis_name}.i_station_no).

8096	Illegal value for the SLOT parameter
Cause:	This error cannot occur, because the parameter is set permanently to 2.

8097	Illegal value for the SUBSLOT parameter
Cause:	This error cannot occur, because the parameter is set permanently to 0.

8098	Illegal value for the SUBADDR parameter
Cause:	This error cannot occur, because the parameter is set permanently to 0.

8099	The slot is not configured
Cause:	Check the S7 hardware configuration.

809A	The subaddress of the selected slot is not configured
Cause:	This error cannot occur, because the parameter is set permanently to 0.

4004	S7 function SFC58 signals error upon input
Cause:	GMC_FB_MCT uses S7 system function SFC58 (WR_REC) internally. SFC58 indicated an error during transfer of the data to the MASTERDRIVES MC (input). The exact cause of the error is specified in error number 2.
Effect:	The task is aborted.
Remedy:	Since SFC58 receives data from GMC_FB_MCT, the following errors should not occur (exception: error in hardware configuration). A description of SFC58 and the errors generated can be found in the S7 Reference Manual – Standard and System Functions.

8090	Specified logical base address invalid
Cause:	Check the PROFIBUS address and the I/O addresses in the S7 hardware configuration in GMC_DB_ORG in the axis descriptions section.

8092	The type specified in the ANY reference is not BYTE
Cause:	This error cannot occur.

8093	This SFC is not permitted for the module specified by LADDR and IOID
Cause:	This error cannot occur.

80A1	Negative acknowledgement when sending the data record to the module
Cause:	The MASTERDRIVES MC was unplugged during the write process or is defective.

80A2	DP protocol error at layer 2
Cause:	A hardware error may have occurred.
Remedy:	Try switching the unit off/on.

80A3	DP protocol error with direct data link mapper or user interface/user
Cause:	A hardware error may have occurred.
Remedy:	Try switching the unit off/on.

80A4	Communication problem on the K bus
Cause:	This error can only occur with an external DP interface.

80B0	Group error
Cause:	This error cannot occur.

80B1	**The length specified in the RECORD parameter is incorrect**
Cause:	This error can only occur with an external DP interface. The error cannot occur, because the parameter is set permanently to 206.

80B2	**The configured slot is not occupied**
Cause:	Check the PROFIBUS address and the I/O addresses in the S7 hardware configuration in GMC_DB_ORG in the axis descriptions section.

80C1	**The data of the previous write job for the same data record on the module have not yet been processed by the module**

80C2	**The module is currently processing the maximum possible number of jobs for a CPU**

80C3	**The required resources (memory etc.) are currently occupied**

80C4	**Communications error**

80C5	**Distributed I/Os not available**

80C6	**Data record transfer was stopped due to a priority class abort**

4005	**S7 function SFC59 signals error upon input**
Cause:	GMC_FB_MCT uses S7 system function SFC59 (RD_REC) internally. SFC59 indicated an error during transfer of the data to the MASTERDRIVES MC (input). The exact cause of the error is specified in error number 2 (see error number 1 = 4004).
Effect:	The task is aborted.
Remedy:	Since SFC59 receives data from GMC_FB_MCT, the following errors should not occur (exception: error in hardware configuration). A description of SFC59 and the errors generated can be found in the S7 Reference Manual – Standard and System Functions.

4006	Command cannot be executed upon input
Cause:	GMC_FB_MCT accesses the PROFIBUS communication system internally. The PROFIBUS communication system indicated reply identifier 7 during transfer of data to the MASTERDRIVES MC (input). The exact cause of the error is specified in error number 2.
Effect:	The task is aborted.
Remedy:	Since GMC_FB_MCT handles task supply, the following errors should not occur. Exception: The parameters on the MASTERDRIVES MC can be accessed directly using the special task "input/output MC parameters". A detailed description of PROFIBUS communication and the errors can be found in the "MASTERDRIVES MC – Compendium".

1	Inadmissible legal parameter number
Cause:	The parameter number does not exist.

2	Parameter value cannot be changed
Cause:	The parameter is a monitoring parameter.

3	Erroneous sub-index

4	No array

5	Incorrect type of data

6	Setting not permitted

7	Descriptive element cannot be changed

11	No operator control rights
Cause:	Check MASTERDRIVES MC parameter P53.

12	Key word missing
Cause:	The "access code" and/or "special access" hardware parameters are not set properly.

15	No text array available

17	Task cannot be executed due to operating status
Cause:	The converter status does not currently allow the issued task.

101	**Parameter number deactivated at the moment**

102	**Channel width too small**

103	**Number of PKWs incorrect**

104	**Parameter value not permissible**

105	**The parameter has been indexed**

106	**Task not implemented**

4007	**No operating authorization for PKW interface upon input**
Cause:	GMC_FB_MCT accesses the PROFIBUS communication system internally. The PROFIBUS communication system indicated reply identifier 8 (no operating authorization for PKW interface) during transfer of data to the MASTERDRIVES MC (input).
Effect:	The task is aborted.
Remedy:	Since GMC_FB_MCT handles task supply, the following errors should not occur. Exception: The parameters on the MASTERDRIVES MC can be accessed directly using the special task "input/output MC parameters". A detailed description of PROFIBUS communication and the errors can be found in the "MASTERDRIVES MC – Compendium".

4020	**Acknowledgement incorrect**
Cause:	After a task was successfully initiated, it could not be terminated because of a missing acknowledgement. The exact cause of the error is specified in error number 2 (see error number 1 = 4002).
Effect:	The task is aborted.

4021	**S7 function SFC5 signals error upon output**
Cause:	GMC_FB_MCT uses S7 system function SFC5 (GADR_LGC) internally. SFC5 indicated an error when fetching data from the MASTERDRIVES MC (output). The exact cause of the error is specified in error number 2 (see error number 1 = 4003).
Effect:	The task is aborted.
Remedy:	Since SFC5 receives data from GMC_FB_MCT, the following errors should not occur (exception: error in hardware configuration). A description of SFC5 and the errors generated can be found in the S7 Reference Manual – Standard and System Functions.

4022	**S7 function SFC58 signals error upon output**
Cause:	GMC_FB_MCT uses S7 system function SFC58 (WR_REC) internally. SFC58 indicated an error when fetching data from the MASTERDRIVES MC (output). The exact cause of the error is specified in error number 2 (see error number 1 = 4004).
Effect:	The task is aborted.
Remedy:	Since SFC58 receives data from GMC_FB_MCT, the following errors should not occur (exception: error in hardware configuration). A description of SFC58 and the errors generated can be found in the S7 Reference Manual – Standard and System Functions.

4023	**S7 function SFC59 signals error upon output**
Cause:	GMC_FB_MCT uses S7 system function SFC59 (RD_REC) internally. SFC59 indicated an error when fetching data from the MASTERDRIVES MC (output). The exact cause of the error is specified in error number 2 (see error number 1 = 4004).
Effect:	The task is aborted.
Remedy:	Since SFC59 receives data from GMC_FB_MCT, the following errors should not occur (exception: error in hardware configuration). A description of SFC59 and the errors generated can be found in the S7 Reference Manual – Standard and System Functions.

4024	**Command cannot be executed upon output**
Cause:	GMC_FB_MCT accesses the PROFIBUS communication system internally. The PROFIBUS communication system indicated reply identifier 7 (command cannot be executed) when fetching data from the MASTERDRIVES MC (output). The exact cause of the error is specified in error number 2 (see error number 1 = 4006).
Effect:	The task is aborted.
Remedy:	Since GMC_FB_MCT handles task supply, the following errors should not occur. Exception: The parameters on the MASTERDRIVES MC can be accessed directly using the special task "input/output MC parameters". A detailed description of PROFIBUS communication and the errors can be found in the "MASTERDRIVES MC – Compendium".

4025	**No operating authorization for PKW interface upon output**
Cause:	GMC_FB_MCT accesses the PROFIBUS communication system internally. The PROFIBUS communication system indicated reply identifier 8 (no operating authorization for PKW interface) when fetching data from the MASTERDRIVES MC (output).
Effect:	The task is aborted.
Remedy:	Since GMC_FB_MCT handles task supply, the following errors should not occur. Exception: The parameters on the MASTERDRIVES MC can be accessed directly using the special task "input/output MC parameters". A detailed description of PROFIBUS communication and the errors can be found in the "MASTERDRIVES MC – Compendium".

A.4 Error Messages of GMC_FB_M7

Output Location of the Error Messages

The error messages of function block GMC_FB_M7 are entered in the associated task box of data block GMC_DB_COM.

DBWx	Error number 1	HEZ	X_app{n}.I_fault1
DBWx+2	Error number 2	HEZ	X_app{n}.I_fault2

Error number 1 contains the actual error message. In the case of a large number of error messages, an extension is entered in error number 2 for more accurate diagnosis.

Below is a complete list of error messages from function block GMC_FB_M7.

0305	**Invalid target range on the M7-FM**
Cause:	Index and length give a target range which exceeds the data range for this task (main task / subtask).
Effect:	The task is not executed.
Remedy:	The definition of the order on the M7-FM in the data block GMC_DB_COMENGINE_ORDER must be corrected and reloaded into the M7-FM from your project. Restart required.

HA number	**Display of the main task number (HA no.)**
Cause:	There is a contradiction between the task parameterized and the task description on the M7-FM (DB_POINTER on the M7-FM).

0307	**Main task is not defined on the M7-FM**
Cause:	The main task is not defined on the M7-FM.
Effect:	The task is not executed.
Remedy:	The definition of the order on the M7-FM in the data block GMC_DB_COMENGINE_ORDER must be corrected and reloaded into the M7-FM from your project. Restart required.

HA number	**Display of the main task number (HA no.)**
Cause:	The task parameterized is not included in the task descriptions on the M7-FM (DB_POINTER on the M7-FM).

1F00	Task code incorrect
Cause:	Task code is incorrect. Internal error in ComEngine (M7-FM).
Effect:	The task is not executed.
Remedy:	The definition of the order on the M7-FM in the data block GMC_DB_COMENGINE_ORDER must be corrected and reloaded into the M7-FM from your project. Restart required.

1F02	Execution of task refused
Cause:	The execution of the task is refused because the internal interface is still busy. Internal error in ComEngine (M7-FM).
Effect:	The task is not executed.
Remedy:	Internal error in ComEngine, restart required.

1F03	Start or processing of a task failed
Cause:	The main task is not properly defined on the M7-FM (main task / subtask information) or the technology module allocated to the main task / subtask is not loaded.
Effect:	The task is not executed.
Remedy:	The definition of the order on the M7-FM in the data block GMC_DB_COMENGINE_ORDER must be corrected and reloaded into the M7-FM from your project. Restart required.

1F04	Combined operation / data supply task failed
Cause:	There is a contradiction between the information in the values to be read and the actual data ranges on the M7-FM or between the number of values to read in requested and the maximum number of values to be read in (113 data words).
Effect:	The task is not executed.
Remedy:	Initiate the task again. Restart required.

1F05	Operation monitoring time expired
Cause:	The technology block addressed has not executed the operation task within the configured operation monitoring time. Either the technology block is not available or it needs more time to be executed than specified in the configured operation monitoring time.
Effect:	The task is not executed.
Remedy:	Load technology block. Restart required.

1F06	Operation task rejected
Cause:	The operation task was rejected by the function block FB-FM because the M7-FM is not in RUN mode.
Effect:	The task is not executed.
Remedy:	Set M7-FM to RUN mode and repeat task.

1F07	M7-FM in stop condition or M7 application does not run
Cause:	M7-FM, key switch to STOP, parameterization error (yellow USR LED flashes). HW Config incorrect (IF module incorrectly parameterized / connected).
Effect:	No communication is possible, the application hangs.
Remedy:	M7-FM, key switch to RUN, correct parameterization error.

4444	Monitoring of operation task expired
Cause:	The operation task started has not been completed within a particular number of S7-CPU OB1 cycles. The FM was still busy saving data to the EEPROM and therefore unable to process the new task straight away.
Effect:	The task might not have been executed correctly.
Remedy:	Increase the timeout counter (default = 16000) Access to the variable is **symbolic** only: Example: Double monitoring time: Open data block GMC_IDB_M7, Set Z_fb_m7.i_time_out_max = 32000. Alternatively, you can configure a minimum cycle time, e.g. 40 ms, via HW Config for an S7-400 CPU. We would however recommend that you save your data only to the RAM (user memory), especially data which are frequently changed. You can speed up task processing significantly by doing this. Your data will however be lost when you switch the power supply off.

A.5 Error Messages of GMC_FB_JOB

Output Location of the Error Messages

The error messages of function block GMC_FB_JOB are entered in the associated task box of data block GMC_DB_COM.

DBWx	Error number 1	HEZ	X_app{n}.I_fault1
DBWx+2	Error number 2	HEZ	X_app{n}.I_fault2

Error number 1 contains the actual error message. In the case of a large number of error messages, an extension is entered in error number 2 for more accurate diagnosis.

Below is a complete list of error messages from function block GMC_FB_JOB.

4101	**GMC_DB_APP1 is incorrect**
Cause:	The GMC_DB_APP1 data area (DB number, DBW number DB_APP1) specified in the task box in GMC_DB_COM is incorrect. The exact cause of the error is specified in error number 2.
Effect:	The task is not executed.

1	**Data block number not allowed**
Cause:	The specified data block number is zero or greater than the number allowed for this S7-CPU.
Remedy:	Specify a permissible data block number.

2	**Data block does not exist**
Cause:	The specified data block number does not exist or the specified DB number is incorrect.
Remedy:	Set up the data block or correct the DB number.

3	**Data block is too short**
Cause:	The data block is shorter than the specified DBW number or the DBW number is incorrect.
Remedy:	Increase the data block or correct the DBW number.

4	**Data block is write-protected**
Cause:	The data block is write-protected or the specified DB number is incorrect.
Remedy:	Cancel the write protection or correct the DB number.

5	**Command dataset = 0**
Cause:	A copy task of 0 bytes was passed to the internal copy function.
Remedy:	This error cannot occur, because GMC_FB_JOB initializes at least 10 bytes.

6	**Odd DBW number not allowed**
Cause:	The specified DBW number is an odd number.
Remedy:	DBW numbers must always be even.

4102	**GMC_DB_APP2 is incorrect**
Cause:	The data area specified for GMC_DB_APP2 (DB number, DBW number DB_APP2) in GMC_DB_APP1 in the general task header is incorrect. The exact cause of the error is specified in error number 2 (see error number 1 = 4101).
Effect:	The task is aborted.

4103	**GMC_DB_APP3 is incorrect**
Cause:	The data area specified for GMC_DB_APP3 (DB number, DBW number DB_APP3) in GMC_DB_APP2 in the general task header is incorrect. The exact cause of the error is specified in error number 2 (see error number 1 = 4101).
Effect:	The task is aborted.

4108	**Acknowledgement in status read acknowledgement incorrect**
Cause:	The anticipated acknowledgement with 0 has not been received from the technology. The incorrect acknowledgement is entered in error number 2. This error cannot normally occur.
Effect:	The task is aborted.

4109	**Acknowledgement in status acknowledgement general incorrect**
Cause:	The anticipated acknowledgement with 0 or –4 has not been received from the technology. The incorrect acknowledgement is entered in error number 2. This error cannot normally occur.
Effect:	The task is aborted.

4110	**Command ID in status acknowledgement general incorrect**
Cause:	An incorrect task identifier was received from the technology in the acknowledgement. The incorrect task identifier is entered in error number 2. This error cannot normally occur.
Effect:	The task is aborted.

4111	**Number of part commands < 1 not allowed**
Cause:	The number of subtasks in GMC_DB_APP1 (DBWx-2) is less than 1. The number of subtasks which was entered is indicated in error number 2.
Effect:	The task is aborted.
Remedy:	Specify a correct number of subtasks.

4112	**Absolute axis number not allowed**
Cause:	The absolute axis number entered in the general task header in GMC_DB_APP1 is less than 1 or greater than the total number of axes configured. The total number of axes is defined during commissioning in GMC_DB_ORG in the "axis descriptions" section. The absolute axis number entered is indicated in error number 2.
Effect:	The task is aborted.
Remedy:	Specify a correct absolute axis number.

4113	**Command aborted through new startup**
Cause:	The initialization bit was enabled while the task was being executed. The initialization bit is normally enabled by GMC_FB_START on an S7-CPU system startup. The bit-coded state of GMC_FB_JOB is entered in error number 2.
Effect:	The task is aborted.
Remedy:	Initiate the task again.

4114	**Status in GMC_FB-JOB incorrect**
Cause:	The status control system of GMC_FB_JOB indicates an incorrect internal status. The bit-coded status of GMC_FB_JOB is entered in error number 2. This error cannot normally occur.
Effect:	The task is aborted.
Remedy:	Restart the system.

4116	**Axis does not exist or type of axis > 3 not allowed**
Cause:	The absolute axis number entered in the general task header in GMC_DB_APP1 does not exist or the type of axis is > 3. The type of axis is entered in the axis descriptions for each axis in GMC_DB_ORG during commissioning. If the type of axis entered is 0 or > 3, the axis cannot be processed by GMC_FB_JOB. Either the absolute axis number (type of axis = 0) or the invalid axis type is indicated in error number 2.
Effect:	The task is aborted.
Remedy:	Specify a correct absolute axis number or type of axis.

A.6 Displays of GMC_FB_START

Output Location of the Error Messages

GMC_FB_START is executed during the startup phase (OB100). It checks the contents of data block GMC_DB_ORG and sets up the necessary data structure. If any errors occur, they are entered in the displays section in GMC_DB_ORG as follows:

DBWx	Display 1	HEX	X_fb_start.i_res1
DBWx+2	Display 2	DEC	X_fb_start.i_res2
DBWx+4	Display 3	DEC	X_fb_start.i_res3
DBWx+6	Display 4	HEX	X_fb_start.i_res4

If GMC_FB_START does not detect any errors, the value "0" is entered in displays 1 to 4.

If errors occur, the following list (display 1) describes the causes.

4200	**GMC_DB_ORG error**
Cause:	Data block GMC_DB_ORG contains an error. The exact cause is specified in **displays 2** to **4**.
Effect:	The S7-CPU does not start up.
Remedy:	GMC_DB_ORG must be loaded from your project into the CPU with the correct configuration.

1	**GMC_DB_ORG does not exist**
Cause:	Data block GMC_DB_ORG does not exist and cannot be set up. The data block number of GMC_DB_ORG is indicated in display 3. The error number of SFC22 (CREAT_DB) is indicated in display 4: 8091 The nesting depth was exceeded. 8092 The compress function is currently active. 80A1 Error in DB number:·The number is 0.·The number is greater than the maximum number of DBs allowed for this CPU. 80A2 Error in length: The length is 0.·The length was specified with an odd number. The length is greater than the length allowed by the CPU. 80B1 No DB number is available. DB number already assigned. 80B2 Not enough memory available. 80B3 Not enough contiguous memory is available (remedy: compress the memory!).

2	**GMC_DB_ORG not initialized**
Cause:	GMC_DB_ORG was set up but does not yet contain data. If GMC_DB_ORG does not exist, it is set up, in order to make the "displays" section available; no structure data are entered, however. The data block number of GMC_DB_ORG is indicated in **display 3**.

3	**GMC_DB_ORG too short**
Cause:	GMC_DB_ORG exists but does not have the required length. The data block number of GMC_DB_ORG is indicated in display 3. The required length is indicated in bytes in display 4.

4201	**0 or negative total number of axes not allowed**
Cause:	0 or a negative number is specified as the total number of axes in the axis descriptions section in GMC_DB_ORG. **Displays 2** to **4** are not relevant.
Effect:	The S7-CPU does not start up.
Remedy:	Specify a value greater than 0 for the total number of axes.

4202	**Data block does not start with DBW0**
Cause:	All necessary data blocks are entered in GMC_DB_ORG; the data block number and the DBW number define the start of the data area. For some data blocks, DBW0 must be specified as the start of the data area. This error is generated if a DBW number <> 0 is entered for this data block. The data block number of the incorrectly configured data block is indicated in **display 2**. The incorrectly configured DBW number is indicated in **display 3**. The DBW address of the incorrectly configured data block is indicated in **display 4**.
Effect:	The S7-CPU does not start up.
Remedy:	Check the DBW entry. If you use the GMC_DB_ORG supplied with the software, all the structure parameters are entered correctly. The following data blocks must start with DBW0: GMC_DB_CMD, GMC_DB_PARA, GMC_DB_IN, GMC_DB_OUT, GMC_DB_AWP1_M7, GMC_DB_AWP2_M7, GMC_IDB_FM, GMC_DB_ORDER.

4203	Data block area overlap
Cause:	All necessary data blocks are entered in GMC_DB_ORG. During the startup phase, the system detected that the same data block numbers are defined for different areas. The incorrectly configured DBW number is indicated in **display 2**. The DBW address of the 1st incorrectly configured data block is indicated in **display 3**. The DBW address of the 2nd incorrectly configured data block is indicated in **display 4**.
Effect:	The S7-CPU does not start up.
Remedy:	Check the entries in GMC_DB_ORG. If you use the GMC_DB_ORG supplied with the software, all the structure parameters are entered correctly.

4204	Error setting up data block
Cause:	During the startup phase, GMC_FB_START attempts to set up all necessary data blocks. This error is indicated if an attempt to set up a data block has failed. The number of the data block which could not be set up is indicated in **display 2**. The DBW address of the data block which could not be set up is indicated in **display 3**. The error number of SFC22 (CREAT_DB) is entered in **display 4**: 8091 The nesting depth was exceeded. 8092 The compress function is currently active. 80A1 Error in DB number:·The number is 0.·The number is greater than the maximum number of DBs allowed for this CPU. 80A2 Error in length: The length is 0.·The length was specified with an odd number. The length is greater than the length allowed by the CPU. 80B1 No DB number is available. DB number already assigned. 80B2 Not enough memory available 80B3 Not enough contiguous memory is available (remedy: compress the memory!).
Effect:	The S7-CPU does not start up.
Remedy:	Check the entries in GMC_DB_ORG. Compress the memory.

4205	M7 number 0, negative or too high
Cause:	The M7 number entered in the axis descriptions section in GMC_DB_ORG is 0, negative or too high. The number of the incorrectly configured axis, for which the invalid M7 number is entered in the axis description, is indicated in **display 2**. The maximum permissible M7 number is indicated in **display 3**. The DBW address of the incorrectly configured M7 number is indicated in **display 4**.
Effect:	The S7-CPU does not start up.
Remedy:	Specify a valid M7 number.

4206	**Local axis number for M7 < 1 not allowed**
Cause:	The "local axis number for M7" entered in the axis descriptions section in GMC_DB_ORG is less than 1. The number of the incorrectly configured axis, for which the invalid local axis number is entered in the axis description, is indicated in **display 2**. The number of the incorrectly configured local axis is indicated in **display 3**. The DBW address of the incorrectly configured local axis is indicated in **display 4**.
Effect:	The S7-CPU does not start up.
Remedy:	Specify a valid value for "local axis number for M7".

4207	**Local axis number for M7 > 32 not allowed**
Cause:	The "local axis number for M7" entered in the axis descriptions section in GMC_DB_ORG is greater than 32. The number of the incorrectly configured axis, for which the invalid local axis number is entered in the axis description, is indicated in **display 2**. The number of the incorrectly configured local axis is indicated in **display 3**. The DBW address of the incorrectly configured local axis is indicated in **display 4**.
Effect:	The S7-CPU does not start up.
Remedy:	Specify a valid value for "local axis number for M7".

4208	**Identical local axis numbers for M7 not allowed**
Cause:	The same "local axis number for M7" has been entered for multiple axes in the axis descriptions section in GMC_DB_ORG. The number of the first incorrectly configured axis, for which the first invalid local axis number is entered, is indicated in **display 2**. The number of the second incorrectly configured axis, for which the second invalid local axis number is entered, is indicated in **display 3**. The DBW address of the first incorrectly configured local axis is indicated **in display 4**.
Effect:	The S7-CPU does not start up.
Remedy:	The local axis number of an M7 number can only be assigned once.

4209	**Identical PROFIBUS addresses for MCT not allowed**
Cause:	The M7 number entered in the axis descriptions section in GMC_DB_ORG is 0, negative or too high. The number of the incorrectly configured axis, for which the invalid M7 number is entered in the axis description, is indicated in **display 2**. The maximum permissible M7 number is indicated in **display 3**. The DBW address of the incorrectly configured M7 number is indicated in **display 4**.
Effect:	The S7-CPU does not start up.
Remedy:	Find setting for configured Profibus address in HW Config and correct in the incorrect axis description.

4210	I/O address overlap for MCT not allowed
Cause:	There is an overlap between the "parameter interface (PKW) I/O address for MCT" and/or "process data interface (PZD) I/O address for MCT" entered in the axis descriptions section in GMC_DB_ORG and/or the M7 address from the M7 description. The number of the first incorrectly configured axis, for which the invalid I/O address is entered in the axis description, is indicated in **display 2**. The number of the second incorrectly configured axis, for which the invalid I/O address is entered in the axis description, is indicated in **display 3**.
Effect:	The S7-CPU does not start up.
Remedy:	The "parameter interface (PKW) I/O address for MCT" and/or "process data interface (PZD) I/O address for MCT" must not overlap.

4211	Incorrect data block number
Cause:	An incorrect data block number has been specified in the "general parameters", "MCT description" or "M7 description" section. The number of the incorrectly configured data block is indicated in **display 2**. The DBW address of the incorrectly configured data block is indicated in **display 3**.
Effect:	The S7-CPU does not start up.
Remedy:	The data block numbers must be used in accordance with the selected S7-CPU.

A.7 Displays of GMC_FB_PICTMAN

Output Location of the Error Messages

GMC_FB_PICTMAN is used by the "GMC-OP-OAM" standard software (user interface with COROS OP25/27). If errors occur, they are entered in the displays section of GMC_DB_ORG as follows:

DBWx	Display 1	HEX	X_fb_pictman.i_res1
DBWx+2		DEC	X_fb_pictman.i_res2
DBWx+4		DEC	X_fb_pictman.i_res3
DBWx+6		HEX	X_fb_ pictman.i_res4

If GMC_FB_PICTMAN does not detect any errors, the value "0" is entered in display 1.

If errors occur, the following list describes the causes.

4501	**Invalid display number in GMC_DB_ORDER during initialization**
Cause:	Configuration error in GMC_DB_ORDER.
Effect:	The associated display cannot be operated on the OP.
Remedy:	Correct the configuration in GMC_DB_ORDER.

4502	**Display not defined**
Cause:	No display pointer for the selected display (display numbers 1011 to 1060) in data block GMC_DB_PICT_POINTER.
Effect:	The associated display cannot be operated on the OP.
Remedy:	Correct the configuration in GMC_DB_PICT_POINTER.

4503	**Supported user display not defined**
Cause:	No display pointer for the selected user display (display numbers 1061-1090) in data block GMC_DB_PICT_POINTER.
Effect:	The associated display cannot be operated on the OP.
Remedy:	Correct the configuration in GMC_DB_PICT_POINTER.

B Appendix B: Glossary

Absolute coordinates

Entering a set position as an absolute value relative to the path zero point ("Move up", See also → G functions, → incremental coordinates).

Absolute measurement data relates to a shared → reference point, e.g. the workpiece zero point W or the machine zero point M.

Absolute dimensions

Identical to → absolute coordinates

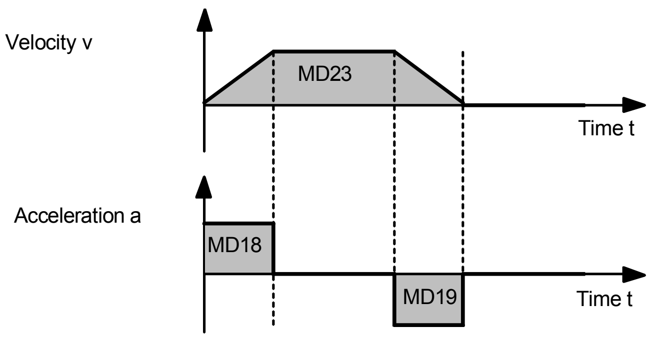
Absolute value device

Position or angle device which, once its supply voltage has been switched on, gives the position of the axis immediately as an absolute actual value. At any point, it provides absolute information about the current angle or linear position. If an absolute value device is used as a position detector, no → reference search is required after startup.

Opposite: → incremental position device

Acceleration

The transfer behavior of the reference input variable is adjusted to the transfer behavior of the controlled system using the machine data for acceleration and deceleration.



The gradient of the acceleration and deceleration ramp should be selected such that a high level of current is required for acceleration and deceleration, but the current never reaches the maximum current level.

Acceleration override

Percentage evaluation of the acceleration value

Acceleration ramp

Sets the acceleration time between standstill and rated speed.

Acceleration time

"Acceleration time" determines the → acceleration ramp in control and reference point approach mode.

Accuracy

See → resolution

Actual value evaluation factor
The actual value evaluation factor indicates which traversing path (e.g. in [µm]) represents an increment of the position encoder.

Actual value commutation
If the master axis is derived from the actual position value of another axis, or, to put it more simply, from an external actual position detector, this is known as actual value commutation.

Address code
The address code consists of a series of alphanumeric characters. Together with a sequence of digits, it forms a word, which is an element of an NC block.

Angle synchronization
During angle synchronization, the actual angle value of a → slave axis is permanently compared to the angle signal from the → master axis. The resulting difference in position is passed on to a position controller. The position controller corrects the speed controller such that the angular position of the slave is always exactly the same as the master. In the special case of "gear synchronization", the angle information is multiplied by a gear factor. This multiplication often needs to be highly precise and to an accuracy of many decimal places.

Angle synchronization must not be confused with → speed synchronization, where the relationship between the speeds of the master and the slave is highly precise, but not in perfect, constant synchronization.

Approach direction
Direction of motion of the →reference point

Approach speed
Speed at which the → reference point cam is moved.

Automatic mode
In automatic mode, complete programs are executed fully automatically. In order to be able to start a → traversing program, the traversing program data must be transferred to the MASTERDRIVES MC in the form of parameter blocks using, for example, PROFIBUS or USS. The different programs are distinguished by their program numbers. If programs are to be executed step by step, this can be achieved by setting the control signals [SIST (=Single Step)].

In the → roll feed variant, the positioning for an active start is triggered by an internal [RIE] or external read-in enable (rapid input). This allows extremely rapid signal processing times. Each block has to be started separately by the [RIE]. The actual value is also initially set to zero for each positioning.

Axis allocation
See → axis code

Axis code
The axis code generates an axis name (X, Y, etc.) for each axis. When an NC block is defined, only the defined axis codes can be used.

Axis cycle length
The axis cycle length serves to mark the distinction between → linear axes and → rotary axes. If the axis cycle length is "0", the axis is defined as a linear axis. If the axis cycle length is "> 0", the axis is defined as a rotary axis. The value defines the covered by the rotary axis during one rotation (360°).

Axis length
→ axis cycle length

Axis speed

A distinction is made between entering a → tool path speed and an axis speed.

For → chaining with axis speed, the speed is given together with the address letters FX, FY, etc. The speeds given relate to the axes X and Y. Therefore, each axis moves at the programmed axis speed. The resulting path depends on the paths to be traveled and the axis speeds.

Axis type

A distinction is made between 2 axis types:

- The "axis" type can be used in conjunction with either incremental or absolute position devices. This type of axis is used if an absolute reference to the position system is required throughout the time the system is in operation.
- The "roll feed" type can only be used in conjunction with incremental position devices, as there is no absolute reference to the position system. All movements are only executed on a relative basis, and the actual position value of the axis automatically resets before every movement.

Baud rate

Measurement unit for the transfer speed through data connections.

1 baud is equal to 1 bit per second.

BERO

The BERO is the reference point cam required for synchronization of the measurement system.

Binector

Freely connectable binary signal in MASTERDRIVES MC, e.g. digital output, output of a timing element, control bit from Profibus, etc.

Block

See → NC block

Block search

The "block search" function allows the execution of the traversing program to begin at any point. This is required if, for example, the execution was cancelled and the program does not have to start again from the beginning.

Block transition

Traversing programs are executed in accordance with the sequence or NC blocks programmed. There are various block transitions, which depend on the content of the NC blocks, for example: maximum loop, G60 accuracy, loop window 1 or 2, path-dependent chaining, etc.

Break point

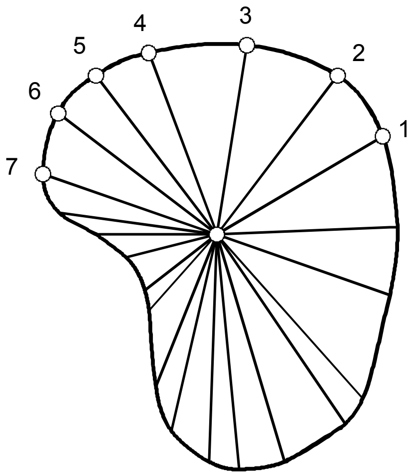
Every mass counters the movement change in its natural mass inertia, and this is also the case for the material to be transported on a roll feed. If the system accelerates, the rollers cannot spin and the material cannot tear. The positioning must be rapid and high-precision during deceleration, but without compressing the material. In order to meet these requirements, there are break points in the acceleration and braking phases where the traversing curve can be optimized to the mechanical conditions.

Built-on sensor

See → external sensor

Cam controller A cam controller switches binary outputs on or off when a parameterizable actual position value is reached. This means that external switching elements – e.g. pneumatic valves – can be started at defined points of the movement sequence (position cams).

Cam disc A mechanical cam disc can be reproduced by means of → table synchronization



Position Master	Position Slave
0°	100 mm
30°	95 mm
60°	90 mm
90°	95 mm
:	:
:	:

Cam disc and associated table

Cancel A program is cancelled either if a certain error message occurs while the program is being executed or if the mode is changed.

Chaining The chaining function allows the sequence of movement for several axes to be set for traversing programs.

Checkback signal Checkback signal is the name given to the signals made available by the technology in the process data interface. They allow the technology to report on its current execution status.

Collision The "collision" function allows the movement of the axis (or axes) to be brought to a rapid halt by removing the digital input during traversing programs where collision monitoring is enabled.

Connector Freely connectable "analog signal" in MASTERDRIVES MC. There are word connectors (16 bit signals) and double word connectors (32 bit signals). Connectors are, for example, the measured value of an analog input, a PROFIBUS setpoint or an adder output.

Continuous mode In continuous mode, the slave axis is permanently in operation and follows any changes to the lead value. This is the standard procedure for any axes which are rigidly connected and for which synchronization acts as a replacement for mechanical drives or line shafts.

Control Control mode is ideal for setting up the speed controller, moving to a fixed stop or for motors which have to run at a certain speed.

Control axis See → master axis

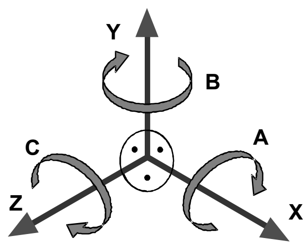
Control signal Control signals are used to control the technology.

See also → checkback signals

Control value The control value is the set value for the → slave axes, sent from the → master axis. The control value represents a set position value. It is then evaluated (electronic gear, table, etc.).

(See also → actual value commutation, → setpoint commutation)

Coordinate



The movement directions of a tool machine can be attributed to a system of coordinates adapted to the movement directions of the machine. The system of coordinates used is right-handed and right-angled and uses the axes X, Y and Z. The coordinate system is adapted to the main machine axes.

Cross cutter See → flying cutter

Data block While the → function blocks implement the sequence programs required, the data blocks represent the → user interface.

Data transfer Data is transferred between the user program and the technology using the GMC_DB_APP data blocks. The execution of a task is triggered using the GMC_DB_COM data block.

Deceleration See → acceleration

Deceleration ramp Determines the deceleration time between the rated speed and standstill.

Deceleration time The "deceleration time" is used to determine the → deceleration ramp in control and reference point search mode.

Default entry Default settings are standard values preset by the manufacturer.

Destination reached and stop (DRS)

This status bit indicates that a positioning task has been successfully executed and concluded. It is transferred to the controller during a handshake procedure in order to ensure that the execution of traversing orders can be recognized even when the paths are extremely short.

When the target position is reached, [DRS] is activated and remains for as long as the start signal is active. In order for [DRS] to be output, a parameterizable position tolerance (DRS tolerance window) must be achieved within a parameterizable monitoring time. If this is not the case, an error bit is set and the positioning process is cancelled.

D functions

The D functions (D1 to D20) allow one of 20 parameterisable second tool functions to be selected in a → traversing program – these can be used to compensate for the length of the tool. This enables automatic → tool compensation. "D0" enables a tool compensation function to be deselected again.

The → G functions G43 and G44 allow you to select whether the tool compensation is positive or negative (i.e. whether it is added to or subtracted from the set position value).

Direct position logging

Direct position logging is when the position device used for logging the actual position value is not integrated into a motor, but built onto the machine (→ external sensor, → built-on sensor) (e.g. → SSI sensor, linear scale or → incremental sensor).

This process is used if, for example, elasticity and internal clearance cannot be avoided, and a genuine actual position signal is nonetheless required.

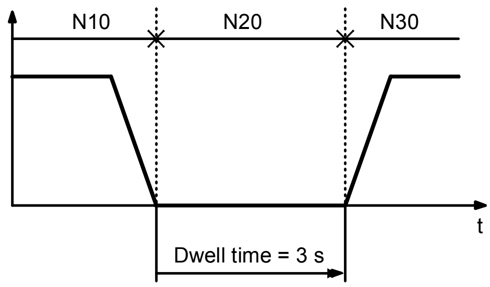
See also → indirect position logging

Disengage

See → start cycle → intermittent mode

Dwell time

Dwell times are required to hold the machine controller to a specific time.



Dwell times are only effective on a per-block basis and must be reselected if required.

Editor

An editor supports inputs and changes.

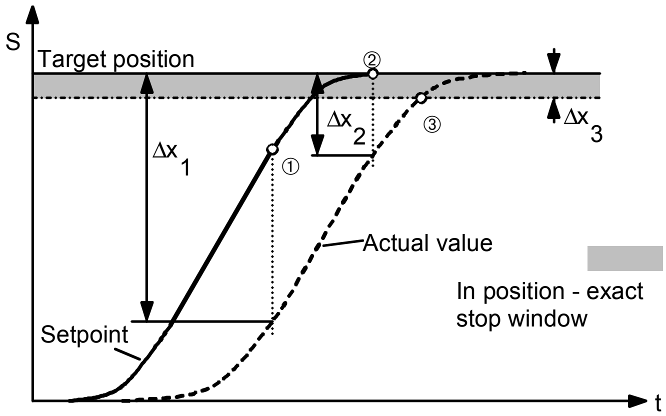
Enable virtual master

Irrespective of the current mode, the virtual master is always executed. The execution is controlled using the "enable virtual master" input.

Encoder	An encoder is a measurement system (sensor) which logs the actual values for the speed and / or angle or positions and makes them available for electronic processing. Depending on the mechanical design, sensors can be built into the drive motor (→ installed sensor) or built onto the motor or the axis (→ built-on sensor). Depending on the sort of movement, a distinction is made between rotary sensors (rotary transducers) and linear sensors (→ linear scales). A distinction is also made between →absolute value sensors (code sensors) and → incremental sensors. See also → ERN sensor
EnDat interface	Bi-directional synchronous serial interface from Heidenhain for transferring actual position values from an → absolute value sensor to the drive controller using the → EnDat protocol.
EnDat protocol	Proprietary serial transfer protocol from Heidenhain for transferring actual position / angle values from an → absolute value sensor to the drive controller. The EnDat protocol also allows parameterization and diagnosis of the sensor.
Endless loop	If the M function M18 is programmed as the first M function in the last traversing set of a traversing program, the system automatically jumps back to the start of the traversing program. An endless loop can also be achieved by a delay time in the last NC block of a traversing program.
End stop	See → hardware limit switch See → software limit switch
Entity data block	An entity data block is allocated to a function block. The allocation is fixed, and it contains the static local data of the function block.
Engage	See → stop-start
EQN sensor	EQN sensors are → built-in multiturn absolute value sensors with the → EnDat protocol. In addition to the absolute measured values for the angle / position, incremental signals are provided in the form of a sine and cosine voltage signal of V_{SS}. The signals can be used for speed and position control or for speed and rotor position control.
ERN sensor	ERN sensors are → built-in incremental sensors (sine-cosine encoders). Depending on the type, the output is either sinus-wave voltage signals of 1 V_{SS} or TTL signals. Of these, two signals can be used for speed or position control and one zero signal can be used as the sensor reference point signal.
Errors	Errors are essentially only generated by the basic functionality of the MASTERDRIVES MC and result in the power immediately being switched off from the motor (pulse inhibitor).

Exact positioning

If programmed positions need to be exactly accurate, then exact positioning, or G60, must be programmed. G60 is effective on a modal basis. If G60 is programmed, the program execution is retained until all the axes involved have reached their positions and stop windows (machine data 17).



① Maximum corner rounding occurs on a block change at the deceleration start point.
② Low corner rounding is achieved with traversing to "digital zero".
③ With exact stop G60, the axis approaches the exact position, because execution of the NC program is suspended until the actual value has reached the exact stop window.

External sensor

Position sensor which is not built into the sensor, but built onto the machine externally or via a mechanical interim part. External sensors (also known as "built-on sensors" or "machine sensors") are used for → direct position logging.

Flying cutter

A cutter which moves with the material (longitudinal movement) during the cutting process (crossways movement) and then automatically moves back to its home position afterwards. The flying cutter has at least two moving axes: one for the crossways movement and one for the longitudinal movement.

The longitudinal movement must be synchronous to the flow of material during cutting in angle / position synchronization mode. After cutting, the cutter must be moved back to its starting position before the next cutting process can begin.

In order to cut, the cutter is moved from one side of the material to the other (crossways movement).

Follower axis

See → slave axis

Following error

The following error is the difference between the set position value and the actual position value for a position or angle synchronization controller. The following error is a measurement of the quality of the position control system and should be as low as possible.

The following error can be minimized using a sophisticated → pilot control system for all known influencing factors.

Follow-up mode Follow-up mode means that the set position value is matched to the actual position value.

Function block The function blocks incorporate the actual administration of the communication between the user interface and the technology. They process the task requests and ensure that the control and checkback signals are exchanged on a cyclical basis. On startup, they set the system to a defined initial status and monitor all the operator functions with respect to their correctness.

Gear synchronization See → angle synchronization

G function The G functions are used both for defining the type of a movement and for calling up corrections, shifts and special functions. The G functions are divided up into G groups on the basis of their function. In each G group, there is a switch status or a basic setting, i.e. this G function is active without being selected. Example: G00 - interpolation with rapid mode for the first G group. The effectiveness of G functions is described as either "modal" (for all the NC blocks in a program) or "local" (G function valid for a single NC block).

Various different G functions can be defined within one NC block. Each traversing program is a unit in itself which is distinct from other traversing programs. Therefore all G groups are in their initial setting when the program begins and must be changed to the setting desired if required.

G group See → G function

Impulse injection The impulse injection corrects a discrepancy between the set position and the actual position. It also triggers an offset between the master axis and the slave axis.

In-built sensor Position sensor built into the motor, e.g..: → resolver, → encoder or → incremental sensor.

Opposite: → external sensor, → built on-sensor

Incremental dimension Relative position command (e.g. "move 10 mm / 10 degrees on a positive direction", see → G functions).

Incremental dimension commands represent incremental (relative) measurement information which relates to the previous position.

Incremental mode If an MDI traversing set with incremental coordinates is transferred, the programmed path is traversed each time the system starts.

Incremental sensor Incremental sensors (incremental position encoders) are generally → pulse generators, → encoders or → resolvers.

Indirect position logging

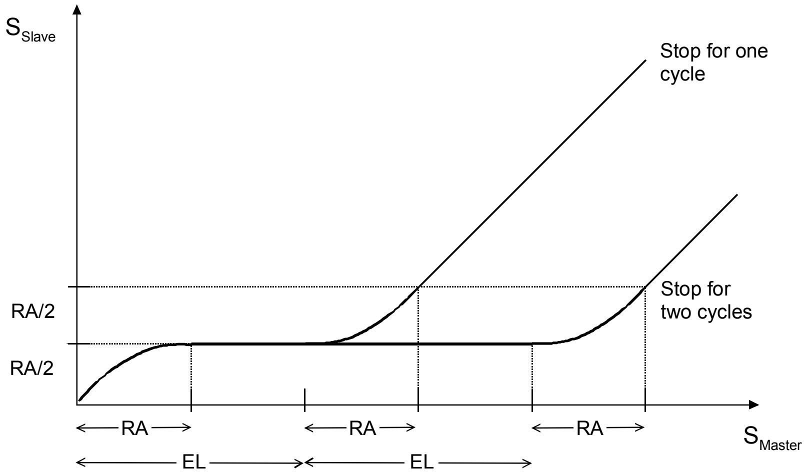
Unlike with → direct position logging, the current position value (angle / path) is not measured directly at the mechanically moving axes of the machine, but via mechanical interim parts instead. Interim parts are gears, measurement spindles, gear wheels, etc. Rotary built-on and installed sensors are used as the position encoders. Sensors built into the drive motor or sensors attached externally measure indirectly.

See also → direct position logging

Intermittent mode

In intermittent mode, the slave axis moves synchronously to the master axis in the initial state. The start cycle is activated by a trigger signal (SST or SSC). The start cycle is started once the master axis has crossed the couple position.

The movement sequence of the master axis and slave axis is shown in the following graph.



RA = On/off ramp
EL = On/off length

Interpolation

Between two axes, there is "interpolation at rapid speed" and "interpolation at tool path speed".

The path programmed for G00 "interpolation at rapid speed" is traversed at the greatest possible speed in a straight line.

The path programmed for G01 "interpolation at tool path speed" is traversed in a straight line at the programmed tool path speed.

Jog mode

In setup mode, the position of the axis is controlled using the direction buttons "jog forwards" and "jog backwards". See → inching mode

Jump

A jump is a programming command used to influence the block execution sequence in a → traversing program.
A distinction is made between:

- conditional jump:
  The program only branches off to the jump destination (block number, → sub-program) if the jump condition is fulfilled.
- unconditional jump:
  The program always jumps
- absolute jump:
  The jump destination is given as an absolute address
- relative jump:
  The jump destination is given as a relative address, i.e. a signed number indicates how many addresses (→ blocks) the program should jump backwards or forwards.

KV factor

Proportional amplification of the position controller. See → position controller

Limit switch

See → software limit switch, → hardware limit switch, → linear axis

Linear axis

A linear axis has a limited traversing range with its limits formed by → hardware limit switches (e.g. BEROs, initiators) and → software limit switches. The set position value is kept within an allowable range using software limit switches.

Linear scale

A linear scale is used for → direct position logging for linear movements and often takes the form of an SSI sensor (→ absolute value sensor).

Loop

If → NC blocks need to be executed more than once, they are programmed into a single traversing program and called up as a sub-program. This sub-program can also be run more than once in a row as a loop. Once called, a sub-program is executed block by block, starting with the first NC block number, until the program jumps back to the level above. Once the programmed loop executions are complete, the main program continues.

Loop number

Entering an MDI loop number allows to restrict the number of positioning moves. The loop number determines the number of repetitions.

The "input / output loop number for MDI roll feed" task inputs or outputs the loop numbers for MDI mode.

Machine data

Machine data is a concept which is used in digital controllers, and they basically represent setup variables through which the position control can be adapted to the mechanical conditions of the system, e.g. max. acceleration, limit switch positions, etc.

Machine zero point

The absolute zero point of an axis, which is physically defined by the manufacturer of the machine. This is used to define the machine coordinate system. The measurement system and electronics are adjusted to this point on setup and → reference point search.

Main program

See → traversing program

Manual data input (MDI)

In "Manual Data Input" mode, the axes are operated using control signals (which are entered manually).

The following options are available:

- MDI for "axes with incremental or absolute measurement systems":
  In MDI mode, individual MDI NC blocks consisting of a position and a speed can be executed.
- MDI for "roll feed" type axes:
  For "roll feed" type axes, positioning for an active start is only triggered by the read-in enable control signal or an external read-in enable. This enables very rapid signal processing times.
- MDI traversing set 0:
  In order to accelerate the transfer of the MDI data still further, there is an option to save the MDI data in the optional range of the control signals.
- Floating MDI:
  Changing a traversing order (position / speed) while the system is in motion.

Master axis

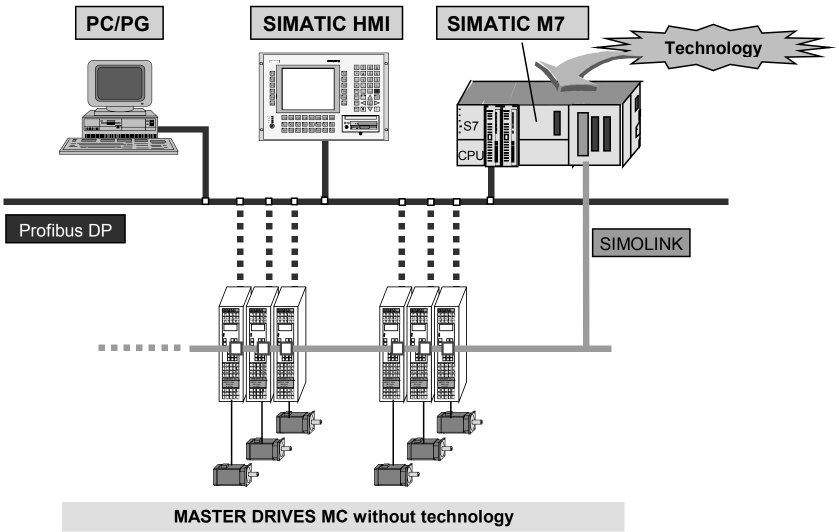
The master axis is the axis which sets the movement course of the → slave axes moving in perfect synchronization during angle synchronization or cam disc functions. The master axis does not require the angle synchronization function. It often only has very little intelligence and needs to know nothing from the slave axes.

The master axis can be a drive ("internal master axis"), a built-on pulse generator ("external master axis") from an upstream machine part or a fictitious movement. The latter is also known as a → "virtual master axis".

MCB

Motion control with basic functionality (centralized solution)

In the centralized solution, the technology is installed on one or more M7 computers. The M7-FM(s) is (are) connected via rapid fiber optic cables to the individual MASTERDRIVES MC drive devices which only incorporate the basic functionality from position control.



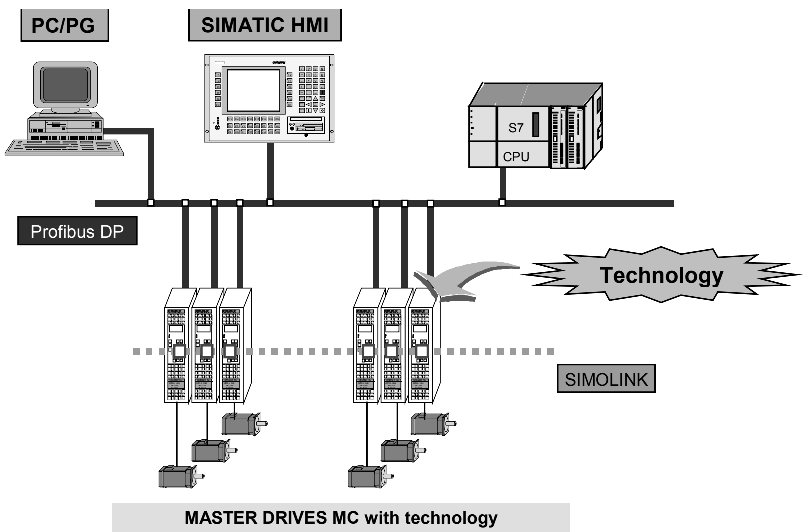
MC parameter

Specialist task which enables direct access to the parameters of the MASTERDRIVES MC.

MCT

Motion control with technology (decentralized solution)

In the decentralized solution, the technology is directly on the MASTERDRIVES MC. The S7-CPU is connected to the individual MASTERDRIVES MC drive devices via Profibus DP. In order to ensure rapid data transfer for synchronization technology, the individual MASTERDRIVES MC must be connected using SIMOLINK fiber optic cable.



MDI
See → Manual Data Input

MDI traversing block
An MDI traversing block contains all the information required to execute a traversing procedure from a start position to a target position and then bring the drive to a halt. (Point to point positioning; see → NC block).

The MDI traversing block is maintained in RAM until such a point when it is overwritten by a new MDI traversing block or the operating mode changes. Positioning can begin once it has been transferred. A new MDI traversing block can be transferred before positioning is complete.

With MASTERDRIVES MC and SIMATIC M7, an MDI traversing block can easily be defined via a serial interface (e.g. PROFIBUS or USS). Alternatively, you can use a control command to select one of a maximum of 10 MDI traversing sets which can be stored in MASTERDRIVES MC. An MDI traversing set consists of the following parameters:

- first G function: type of positioning G90=absolute dimensions or G90=incremental dimensions (relative)
- second G function: acceleration override 10% to 100% in 10% increments
- position
- speed

M function
The M functions (machine functions) are optional elements of a → NC block. They can be used to set or reset binary signals at defined points in the traversing process when this is required by the external machine controller.

Modal
G functions which are effective on a modal basis remain active until they are replaced by another G function in the same group.

Movement block
See → NC block

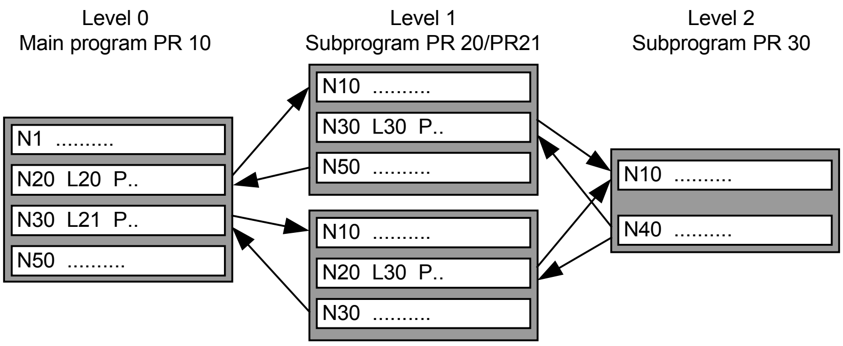
Multi-turn sensor
An → absolute value sensor which provides an absolute reflection of the angular position over a series of rotations (typically 4096 rotations). The number of rotations is transferred in the higher value bits of the position information, while the lower-value bits contain the angular position within one rotation. Multi-turn sensors often have several slotted discs which are connected via gear wheels. The angular position value is often transferred to the control electronics via a serial protocol, e.g. → EnDat, → SSI; see also → EQN sensor.

NC block
A NC block contains all the data required to execute a stage, e.g. positioning, delay or output of switching functions.

NC program
See → traversing program

NC language
Programming language which enables numeric movement control.
(Numeric Control)

Nesting depth A nesting depth of two is allowed, and the number of sub-program calls is only limited by the memory space available.



Offset compensation

Operator panel input / output unit

Optional value optional extension

Overlap When movement is transferred from one axis to another, the movement overlaps (one axis is accelerated before the other stops completely). This results in very rapid movement sequences.

Override The speed override function allows the speed set for a traversing process to be altered on a floating, percentage basis.

The acceleration override (control signal) allows the acceleration and deceleration ramps of traversing processes to be altered by a percentage during motion (e.g. to adjust to workpieces / products with different inertia momentum).

Password Passwords can be entered to control access permissions.

Pilot control Pilot control is the name given to a position control system when the position controller does not set the whole stroke for the speed, and the → curve generator (= position acceleration generator) provides not only the position course, but also the appropriate speed at any point and the acceleration momentum (e.g. by differentiating the position course) and is not controlled by the position controller, but directly by the speed and momentum controller as a set value.

Position acceleration generator See → travel path generator

Position controller

The position controller is normally a P controller (and sometimes a PI controller) which compares the internal digital set position value with the actual digital value from the position encoder on a cyclical basis. The result of this comparison of the set value and the actual value is a signed difference between the setpoint and the actual value. This is known as the → following error. The proportional amplification of the position controller is known as the KV factor. The output signal from the position controller is effective on the speed controller, and tends towards correcting the position error. This correction is small, if → pilot control is used.

Position control system

All the software and hardware used for positioning. The position control system is a tight, sometimes time-critical amalgamation of a controller and feedback control.

The only way to achieve results which are optimized to the design of the machine is to generate control algorithms which take all the physical requirements into consideration (which also involves a highly developed pilot control system).

Position correction

Position correction evaluates the discrepancy between a prominent product position and a defined setpoint. The product position is logged using an appropriate actual position logging system, e.g. a printed mark reader. The setpoint in question is defined by means of a parameter. Any discrepancy between the set position and the actual position is corrected by means of impulse injection. The impulse injection triggers an offset between the master and the slave axis.

Position encoder

See → encoder

Position generator

The position generator incorporates the current speed and displays it across the length of the linear / rotary axis.

Positioning

Positioning is the movement of a drive from its current position to a preset target position. There are two different types:

- Simple → trip positioning and → rapid / creep positioning, where positioning accuracy is not of primary importance. With these systems, the drive is purely speed-controlled and switches either off or to creep speed once it has reached a preset position. A position controller is not required.
- Controlled positioning, where the target position is reached via a path defined by a → travel curve generator. Deviations from the path curve (→ following error) are corrected by the position controller at every stage.
  There are two types of controlled positioning: simple point to point positioning (→ MDI) and the execution of an automatic program (→ automatic mode).

Positioning time

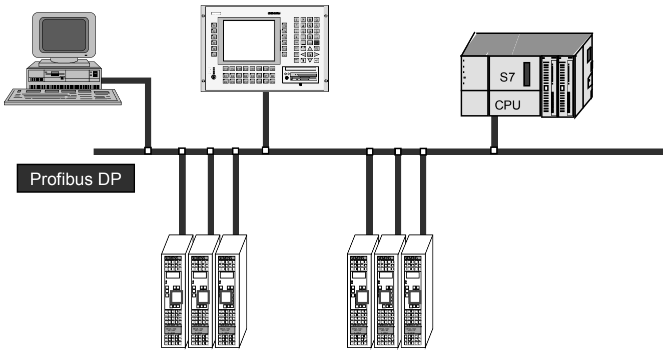
The positioning time is dependent on the → override, the → time override, the → traversing speed and the change of NC blocks.

PPO type In the Profibus profile of adjustable speed drives, the usable data for the cyclical channel is described as the Parameter Process data Object (normal channel).

Printed mark Mark on the edge of the material to be processed (film, cardboard, paper) which can be detected electronically by a detector head. The printed mark is in a fixed geometrical relationship to, for example, an image or a fold on the material to be processed. The detector head makes it possible to point the material in the right direction for the next processing stage.

Process data interface The process data interface is the address range between the master and the slave. Here, the data is exchanged on a cyclical basis. The size is determined by the → PPO type.

Profibus A Profibus subnet is a network of several RS485-based devices.



Pulse generator Incremental path and speed sensor which generally gives two pulse trains which are 90° offset from one another and one zero impulse per rotation.

Pulse generator reproduction See → resolver

Rapid / creep positioning Simple form of → positioning with low positioning accuracy requirements. The drive moves in rapid mode until a set position value (preliminary switch point) is reached. At this point the speed is reduced to creep speed. The drive moves at this speed until the selected final position value is reached and then switches off. No compensation is made if the target position is overshot.

Rated guide speed Adjustable speed to which a connector relates, which gives a speed in %.

Rate of change limiting Prevention of jumps in the acceleration and deceleration procedures using a rounded speed curve to protect the mechanics.

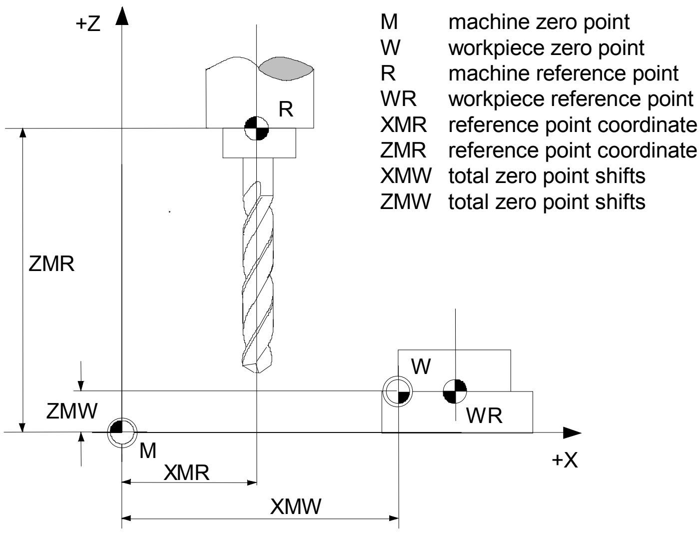
Read-in enable When the read-in enable is triggered, the system begins to decode the traversing program.

Reducing cam See → BERO

Refer See → reference search

Reference point Reference points are set within the traversing range of a positioning drive to make programming, operation and handling easier, especially when incremental position devices are used. (Examples: → machine zero point,

workpiece zero point, → reference point, workpiece reference point, etc..)



Reference point shift If, after the reference point has been found / set, the axis is not to remain at the synchronization point, but needs to be shifted by a certain amount, the amount required can be entered in machine data number 4. After the reference point shift, the current position value is set to the reference point coordinate.

Reference point switch The reference point switch is allocated to → reference search mode. Together with the → zero mark signal from the connected sensor, it determines the physical position of the reference point.

Reference search When using → incremental position encoders, once the drive is switched on, there is no connection between the measurement system and the mechanical position of the axis. Therefore, every time the drive is switched on, it must be moved to a defined reference position.

Once the reference point switch (e.g. BERO or BERO signal connected to the zero pulse) is reached, the position controller sets the actual position value to the parameterized zero point value (reference point value).

Register controller
In printing presses, the register controller ensures that the different colors are printed exactly on top of one another. In order to do this, registers, which are also known as passer marks or → printed marks, are printed on the paper. In textile printing, the register can also take the form of a metal thread woven into the material. These marks are evaluated by optical scanners arranged between the printing lines. A complicated measurement and control electronics system detects and registers errors and adjusts the course of the material using adjustable register rollers such that the registers overlap again.

In a → shaftless printing press, register rollers are not used, and the output signal from the register controller has a direct effect by adjusting the offset angle.

One reason why register errors can occur is that paper is a ‘living product’, which changes in length constantly as a result of the different wetting and drying processes in the printing press (extends and shrinks).

As the "degree of extension and shrinkage" only changes slowly, the register controller can be relatively slow (scanning times in the region of 20 ms).

Relative positioning
Movement with → incremental dimensions ("move by"); see also → G functions

Remaining loop number
The remaining loop number indicates how many sub-programs still need to be executed.

Remaining path
The remaining path is defined as the difference between the programmed target position and the current position.

Reset
reset, delete

Resolution
For a position logging system, the resolution is the smallest measurable path increment.

In principle, a distinction should be made between the theoretical resolution of a position measurement system and the positioning accuracy which can be achieved. The latter is also dependent on the setpoint resolution and the rigidity of the system as a whole (mechanical transfer elements, sensor installation, elasticity figures, internal clearances, temperature of the logging system, etc.). In practice, positioning accuracy is about 1 to 10 times lower than the resolution.

Resolver
Mechanically robust and cost-effective sensor which is built onto the motor and which has no in-built electronics as it works entirely on magnetic principles: a cosine and a sine signal are induced in two coils which are 90° offset from one another. The resolver provides all the signals required for the converter to operator in speed-controlled mode or for position control. The number of sine and cosine periods supplied per rotation is equal to the number of pole pairs in the resolver.

Retrigger
Each retrigger extends the length of the stop / start cycle by a stop / start cycle length.

Roll feed

The roll feed function is required, for example, for presses and cutters where an exact length of the sheet belt has to be conveyed into the press / cutter.

This is an endlessly rotating → rotary axis without → stops and with relative position settings (cutting length).

Rotary axis

A rotary axis is an axis whose position is given as an angular position. After one complete rotation, the starting point (e.g. 0 degrees) and the end point (e.g. 360 degrees) coincide.

Unlike for a → linear axis, a rotary axis has no end stops. The drive can move in the same direction endlessly without any restrictions to its path.

Rotor position encoder

See → resolver

Rotor sensor zero mark

Zero mark on the sensor which returns once per rotation.

Scaling

In order to be able to program the machines as flexibly as possible, it is possible to scale the tables either in the x-direction (master) or in the y-direction (slave). This means that various variables for one product using one and the same table can be produced. However, care must be taken to ensure that the figures do not go out of the allowable position and speed ranges.

Screens

Control and monitoring, e.g. using SIMATIC HMI OP25/OP27. Standard screens allow parameters to be changed and actual values to be displayed.

S curve

S-shaped rounded speed ramp generated by a → travel curve generator. The s-shaped rounding allows → rate of change limiting and thus protects the mechanics.

Sensor

See → encoder

Set floating actual value

The "set floating actual value" function (G87, G88, G89 see

→ G function) is triggered by a rapid input, where the block change is floating and at the same time the actual value is set to a new measurement. There is also an option to read out the actual value recovered at the time the rapid input was received ("read out floating measured value" task). The rapid input must be programmed with the "set floating actual value" function.

Setup mode

In setup mode, the traversing movements of the axis or the roll feed are determined using the non-latching direction buttons (e.g. jog forwards [J_FWD] and jog backwards [J_BWD]). You can also select a stage to select one of two independent speeds. There is also the option of influencing the speeds using an → override.

The final positions set using the direction buttons can be adopted as setpoints (→ teach in - function).

Set value control

If the lead value is derived from the set position value for another axis, this is known as set value control.

It is not the actual value, but the more stable set value of the → master axis is transferred as the lead value to the slave axes during synchronization. This means that fluctuation in the actual value are not reflected in the lead value.

Set value formation

The "set start value" command sets the position output to a parameterizable start value.

Shaftless printing press

Rotary printing press where the longitudinal shaft, and often the stationary shafts are eliminated and replaced by individual drives connected together by means of → angle synchronization. Normally, the angle synchronization is subordinate to a →register controller which (requires little of the dynamics and) corrects the angle synchronization to compensate for the material stretching or shrinking or for slip.

The technical requirements of the drive control system are very high for a shaftless printing press and can be estimated as follows:

- replacing the longitudinal, stationary and cross shafts in packaging and textile printing presses: **demanding**
- replacing the longitudinal shaft between the printing works and the folding system in rotary printing machines: **demanding**
- shaftless packaging and textile printing presses: **demanding**
- shaftless offset rotary printing press (for newspapers, etc.): **very demanding**
- shaftless rotation intaglio printing press (for magazines, etc.): **extremely demanding**
- replacing the gear wheel connection in sheet-fed printing press (e.g. for printing books, artwork, etc.): **exceptionally demanding**

SIMOLINK

SIMOLINK is a high-performance fiber optic ring for connecting drives to one another or for connecting drives to a control system (e.g. SIMADYN D or SIMATIC M7).

SIMOLINK allows set values / actual values and control bits / status bits to be transferred at 11 MBaud. For example, the transfer time for 16 double words (at 32 bit) is 100 µs. The scanning times of all relevant control circuits are synchronized with one another with quartz accuracy and without jitters using special "sync telegrams".

Single step mode

Under normal circumstances, the programs are processed automatically, i.e. the axis executes the NC blocks automatically one after another.

If the control signal [SIST] (=Single Step) is set in automatic mode, then only one traversing set is processed for each start signal [STA]. Removing the control signal [SIST] has the effect that [STA] sets off automatic traversing set programming again.

Single step mode is useful when setting up the machine and testing out → traversing programs.

As traversing programs are essentially executed in single steps anyway for the → roll feed function, the control signal [SIST] is not effective.

Single turn sensor

A single turn sensor is a rotary → absolute value sensor which only logs the actual path value for one rotation and can store it in non-volatile memory in the event of power outage, unlike a → multi-turn sensor which can log the actual path value over several rotations.

Slave axis

In the → angle synchronization and → cam disc function, a slave axis is an axis which is in perfect synchronization with the → master axis.

Slip

Shifts between the product and the axis which moves the product.

Software limit switch

A software limit switch defines the working range for an axis; see also → linear axis.

Speed synchronization

The drives are given their set speed value via a set speed value cascade. The set speed can be sensitively adjusted by multiplying it by a gear factor and maintained with a constancy of about 0.006%. Unlike with → angle synchronization, the angle between the master axis is not maintained in perfect synchronization during speed synchronization; this means that if a one-off angle error occurs because of a load impulse, this is not caught up again with speed synchronization.

See also → angle synchronization

Speed override

Percentage evaluation of the speed

SSI protocol

The SSI protocol is very often used in conjunction with built-on sensors, linear scales, laser measurement systems, etc. In an SSI sensor, the data and the shift rate are both transferred along the sensor cable as 5V differential signals (RS422 standard).

SSI sensor

An → absolute value sensor for logging the absolute position and transfer the position information to the control electronics using the SSI protocol (Synchronous Serial Interface).

Start command

When the start conditions (→ start enable) are fulfilled, a movement can be triggered using the start command [STA].

Start cycle

The start cycle serves to leave the → angle synchronization or the → cam disc function at a specific angle position. At the disengage point, the → slave axis leaves the synchronous mode, i.e. it disengages from the → master axis and is brought to a halt using a deceleration ramp at a parameterized position. The beginning and length of the deceleration ramp can be adjusted relative to the movement of the master axis.

See also → stop cycle

Start enable

The start enable [ST_EN] is a status or checkback bit used by the position controller to signal that all the conditions required before a traversing program can start have been fulfilled and therefore that a traversing program can be initiated using a → start command.

Start position

The initial position is set to a parameterizable start value using the "set start value" command.

Start signal

In order to be able to start a saved traversing program, the basic conditions must be fulfilled, and the traversing program number, the read-in enable (RIE) and the start signal (STA) must be defined.

While the read-in enable activates the NC block decoder, the start signal enables the execution of the traversing sets.

Start value

See → start position

Stop at end of table

If "stop at end of table" is set, the table remains at the last interpolation point when the last end of the table is reached.

Stop cycle

The stop cycle is used to start the → angle synchronization or → cam disc function at a parameterizable (start) position. The stop cycle synchronizes the → slave axis using a ramp to the position of the → master axis at a parameterized position. The beginning and length of this acceleration ramp can be adjusted and coupled to the angle / path of the master axis. Once the stop cycle function is finished, the drive is brought to a halt from a parameterized position to the start position.

See also → start cycle

Stop cycle mode

In stop cycle mode, the slave axis is initially in a "standstill" status. The stop cycle is activated using a trigger signal (SST or SSC). The start cycle is started once the master axis passes the couple position.

Sub-program

There is no essential difference between sub-programs and main programs, they only become sub-programs when they are called from a traversing program. They are programmed for movement sequences which are often required and can be called from a number of different → main programs.

Sub-programs consist of a code containing the sub-program number, a sequence of incremental dimension blocks and the command "end of sub-program – return to main program".

A nesting depth of two is possible.

Generally, sub-programs are not independent programs, as specific data, such as feed, is stored in the main program.
They are programmed using incremental dimensions. This means that the sub-programs are adjustable, i.e. they can start from any axis position.

Synchronicity

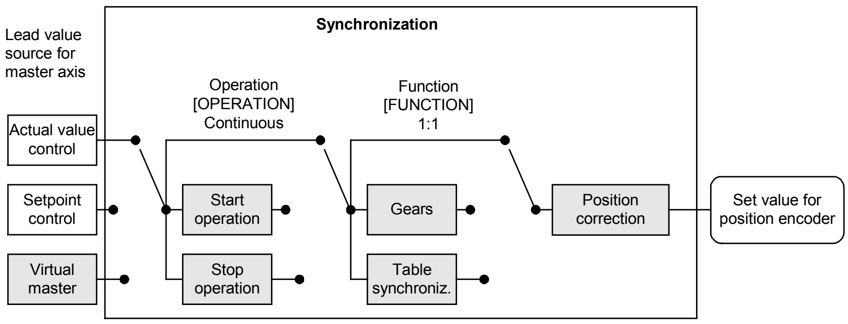
See → synchronization mode

Synchronization

While the axis is moving itself, i.e. logging the set position value entirely on its own, in one of the positioning modes (set-up, MDI, automatic, etc.), the axis becomes an externally controlled axis (slave axis) in synchronization mode. The set position value is derived from a master axis.

For synchronization functions, a distinction must be made between precision → speed synchronization and fixed angle → angle synchronization.

The synchronization technology option supports angle synchronization. Speed synchronization can be achieved using the functions incorporated in the basic device.



Synchronization mark

See → printed mark

Synchronization pulse

The synchronization pulse sets the measurement system to a specific coordinate and thus creates an absolute position reference to the mechanics.

Table interpolation Table interpolation is based on a table made up of n interpolation points. Each interpolation point allocates an angle / position of the → master axis to an angle / position of the → slave axis. Until the transition process, the slave axis moves between two interpolation points in → angle / position synchronization with the master axis.

Enables a relative movement in perfect synchronization between a → master axis and a → slave axis. The angle allocation between the master and slave axis is stored in a table.

One example application for table interpolation is to replace an eccentric disc with a cam disc function.

Table synchronization See → table interpolation

Task header A task header describes what task data (e.g. machine data or NC programs) is to be input or output.

Every task has a task header, the definition of which is fixed and which has a main task number and various other specifications.

Task list A task list (consisting of several tasks) can be executed in series using only a single task command.

Task section A task section consists of an input range and an output range. The input range is used to specify and start data transfer. The output range represents the current status of the task and reports the end of the data transfer or an error, if one has occurred, by outputting an error number.

Teach in In → setup mode, a position is moved to which will later be used as a set position for an operational traversing process. This can be transferred to a pre-selected → traversing set using the program and block number by means of the "teach in" task. Additional movement commands must be available in this traversing set or a previous one (G , F commands, etc.).

"Teach in" is only allowed in setup mode.

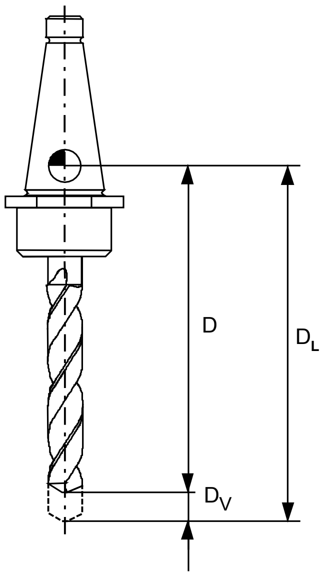
Test mode See → control

Time monitoring The time monitoring function generates a warning and cancels positioning if a specific tolerance is not achieved within a specific time.

Time override The time override is switched on or off using the "time override" function. A time override can be between 0% and 100%. This means that both the speed, and the acceleration and deceleration are affected such that the time is directly related to the override value.

Tool compensation

Tool compensation allows a compensation for the tool length (e.g. drill length) which is automatically incorporated into the calculation of the set position value. Tool compensation is activated and deactivated using the → D functions.



Tool compensation breaks down into tool length correction and tool length wear.

D	tool length calculated
DL	effective tool length (tool length correction)
DV	wear (tool length wear)

Example:

The tool used is 100 mm long (DL = 100 mm). There is also wear of 1 mm (DV = –1 mm).

Therefore the active tool compensation is calculated as
D = DL + DV = (100) + (–1) = **99 mm**.

Tool path speed

A distinction is made between entering a tool path speed and an axis speed.

If there is interpolation with tool path speed, the speed is given in conjunction with the address letter F. The speed value F given relates to the path programmed. The individual axis speeds are calculated on the basis of the paths to be traversed. The resulting path is a straight line.

Travel curve generator

The travel curve generator (also known as a "position acceleration sensor" or "setpoint generator") generates the speed sequence over time for a positioning movement. The acceleration and deceleration ramps and the maximum speed are adjustable. To ensure a gentle start and protect the mechanics, user-friendly travel curve generators have a → rate of change limiting, e.g. a speed sequence representing an S curve (acceleration sensor with rounding). A modern travel curve generator has a → pilot control for the speed and the acceleration momentum.

Traversing program A traversing program (or main program) contains the information required for a movement sequence. They are programmed by the user. An instruction in a traversing program is called a → NC block.

Traversing programs are divided into main programs and → sub-programs.

Traversing range The traversing range is determined by two machine data numbers which set the positive and negative end positions (→ software limit switch).

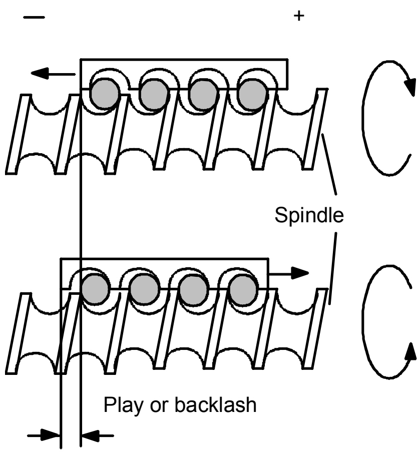
Traversing speed The maximum traversing speed sets the upper limit for all speed inputs. The traversing speed can also be influenced on a percentage basis using the override.

Traversing table The slave axis is connected to the master axis via a traversing table. The traversing table incorporates the movement pattern for the slave axis relative to the master axis.

Trigger signal Triggering of a process by means of a signal.

Trip positioning Simple form of → positioning without a defined movement curve. If a preset set position is reached, the drive is brought to a halt. It is not impossible for the target position to be overshot in this instance. See also → rapid / creep positioning.

Turning clearance



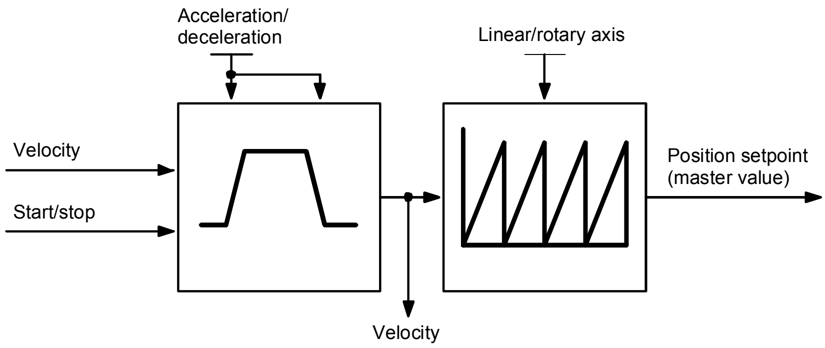
The path / angle that a spindle or motor must leave behind when it changes direction until the axis (carriage, machine table, tool holder) moves is known as the turning clearance (also clearance, play, air).

User-configured display User-configured displays are supported by the standard GMC-OP-OAM software when it is appropriately configured. Display numbers 1065 to 1096 are set aside for this.

User interface The user interface consists of a series of → data blocks which are used to execute all the tasks required to operate the technology.

Virtual master axis

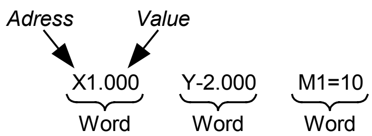
A virtual → master axis is a master axis not allocated to a real existing drive. It simulates the set position / angle value of a rotating axis (which only exists at the software level). The virtual master axis is often generated by a speed acceleration generator. The virtual master axis can be used, for example, as a set value generator for several axes operated in → angle synchronization.



The negative effects, which can occur when the actual position value of a real existing master axis (internal or external master axis) is used and passed on to the slave axis, do not occur with a virtual master. The control instability of a master drive cannot be propagated to the slave drives. Therefore, a virtual master axis is recommended for synchronization.

Word

A word is an element of a NC block and consists of an address code and a sequence of digits. The address code is a sequence of alphanumerical characters. The sequence of digits can have a sign or a decimal point. The sign is between the address characters and the series of digits, although it is assumed to be positive if no sign is entered. There can be no blank spaces between the address and the value.

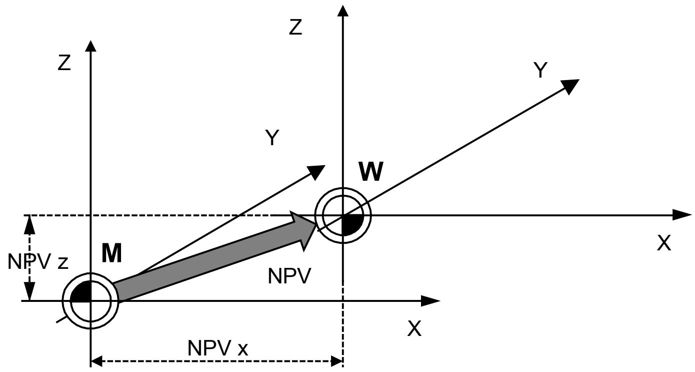


Workpiece zero point

The workpiece zero point is generally set by the programmer of the workpiece for all axes and entered into the diagram. All the dimensions programmed relate to the workpiece zero point.

Zero displacement

The zero displacement is the signed distance between the workpiece zero point W and the machine zero point M.



How many zero displacements can be saved and the range of allowable figures both depend on the control system. The zero displacement is determined by the user and must be in memory before the program starts.

Zero mark

The zero mark can be found on the track of an → incremental sensor. With rotary incremental sensors, one zero mark signal is generally output per rotation, while with linear incremental sensors (→ linear scale), one zero mark signal is output for each axis length, or several are output at specific intervals.

C Appendix C: General Index

E

F

G

H

I

J

K

L

M

P

R

T

Siemens AG A&D MC V4 P.O. Box 3180 D-91050 Erlangen Federal Republic of Germany	**Suggestions** **Corrections** **for Publication:** System Solutions MASTERDRIVES Motion Control (Technology Option F01) and SIMATIC Motion Control Order No.: 6AT1880-0AA00-1BE0 Edition: 11.2002
From Name: ____________________ Address of your company/department Street: ____________________ Postcode: __________ Town/city: __________ Telephone: ________ / ______________ Telefax: ________ / ______________	Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.

Suggestions and/or corrections

Siemens AG
Automation and Drives Group
Motion Control Systems
P.O. Box 3180, D - 91050 Erlangen
Federal Republic of Germany

Siemens Aktiengesellschaft

Order No.: 6AT1880-0AA00-1BE0
Printed in the Federal Republic of Germany
135/892017 BS 04980.2

Progress
in Automation.
Siemens