SIEMENS

SINUMERIK 840D/840Di/810D

Description of Functions 07.2000 Edition

Tool Management

Manufacturer/Service Documentation

Overview of SINUMERIK 840D/840Di/810D/FM-NC Documentation (07.00)

General Documentation

SINUMERIK
840D/810D/ FM-NC
Brochure

SINUMERIK
840D/840Di/ 810D/ FM-NC/611
Catalog Ordering Info NC 60.1 ***)**
Technical Info. NC 60.2

SINUMERIK SIROTEC SIMODRIVE
Accessories
Catalog Accessories NC-Z

User Documentation

SINUMERIK
840D/810D/ FM-NC
AutoTurn
– Short Guide
– Programming (1)
– Setup (2)

SINUMERIK
840D/810D
Operator's Guide
– **Unit Operator Panel**
– **HPU**
– **HT 6**

SINUMERIK
840D/840Di/ 810D/ FM-NC
Diagnostics Guide *)

SINUMERIK
840D/810D/ FM-NC
Operator's Guide
– Short Guide
– Operator's Guide ***)**

User Documentation

SINUMERIK
840D/840Di/ 810D/ FM-NC
Program. Guide
– Short Guide
– Fundamentals ***)**
– Advanced ***)**
– Cycles
– Measuring Cycles

SINUMERIK
840D/810D
Operator's Guide
– **ManualTurn**
– Short Guide ManualTurn
– **ShopMill**
– Short Guide ShopMill

SINUMERIK
840Di
System Overview

Manufacturer / Service Documentation

SINUMERIK
840D/810D
Description of Functions
– **ManualTurn**
– **ShopMill**

SINUMERIK
840D/840Di/ 810D
Description of Functions
Synchronized Actions
Wood, Glass, Ceramics

SINUMERIK
840D/810D
Descr. of Functions
– **Computer Link**
– **Tool Data Information System**

SINUMERIK
Configuring (HW) ***)**
– FM-NC
– 810D
– 840D

Manufacturer / Service Documentation

SINUMERIK
840D/840Di/ 810D/ FM-NC
Operator Components (HW) ***)**

SINUMERIK SIMODRIVE
611D 840D/810D
Description of Functions
Drive Functions *)

SINUMERIK
840D/840Di/ 810D/ FM-NC
Description of Functions
– Basic Machine ***)**
– Extended Functions
– Special Functions

SINUMERIK
840D/810D/ FM-NC
Configuring Kit
MMC 100/101
– Configuring Syntax
– Development Kit

SINUMERIK
840D/810D/ FM-NC
Screen Kit
MMC 100/101
SW Update and Configuration

SINUMERIK
840D/810D
Description of Functions
Tool Management

SINUMERIK
840D/810D/ FM-NC
Description of Functions
Operator Interface **OP 030**

Manufacturer / Service Documentation

SINUMERIK SIMODRIVE
Description of Functions
SINUMERIK **Safety Integrated**

SINUMERIK
840D
Description of Functions
Digitizing

SINUMERIK SIMODRIVE
Installation & Start-up Guide *)
– FM-NC
– 810D
– 840D/611D
– MMC

SINUMERIK SIMODRIVE
840D/840Di/ 810D FM-NC 611D
Lists *)

SINUMERIK SIMODRIVE
840D 611D
Description of Functions
Linear Motor

SINUMERIK SIMODRIVE
840D 611D
Description of Functions
– **Hydraulics Module**
– **Analog Module**

SINUMERIK SIMODRIVE SIROTEC
EMC Guidelines

Electronic Documentation

SINUMERIK SIMODRIVE
840D/810D/ FM-NC 611, Motors
DOC ON CD *)
The SINUMERIK System

Manufacturer / Service Documentation

SINUMERIK
840D/810D
Descr. of Functions
ISO Dialects for SINUMERIK

SINUMERIK
Descr. of Functions
CAM Integration
DNC NT-2000

SINUMERIK
840Di
Manual
(HW + Installation and Start-up)

***)** These documents are a minimum requirement for the control

SIEMENS

SINUMERIK 840D/840Di/810D

Tool Management

Description of Functions

Valid for

Control	*Software version*
SINUMERIK 840D	5
SINUMERIK 840DE (Export variant)	5
SINUMERIK 840Di	5
SINUMERIK 810D	3
SINUMERIK 810DE (Export variant)	3
SINUMERIK MMC	5 and 6

07.00 Edition

SINUMERIK® 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 edition with new Order No.
C . . . . . Revised edition with new status.
If factual changes have been made on the page since the last edition, this is indicated by a new edition coding in the header on that page.

Edition	Order No.	Remarks
04.95	6FC5 297-2AC60-0BP0	**A**
09.95	6FC5 297-3AC60-0BP0	**C**
03.96	6FC5 297-3AC60-0BP1	**C**
08.97	6FC5 297-4AC60-0BP0	**C**
12.97	6FC5 297-4AC60-0BP1	**C**
12.98	6FC5 297-5AC60-0BP0	**C**
08.99	6FC5 297-5AC60-0BP1	**C**
07.00	6FC5 297-5AC60-0BP2	**C**

This manual is included in the documentation available on CD-ROM **(DOCONCD)**

Edition	Order No.	Remarks
10.00	6FC5 298-6CA00-0BG0	**C**

Trademarks
SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIROTEC®, SINUMERIK® and SIMODRIVE® are trademarks of Siemens. Other product names used in this documentation may be trademarks which, if used by third parties, could infringe the rights of their owners.

Further information is available on the Internet under:
http://www.ad.siemens.de/sinumerik

This publication was produced with Interleaf V7

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

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. 6FC5 297-5AC60-0BP2
Printed in the Federal Republic of Germany

Siemens Aktiengesellschaft

Preface

Notes for the reader

The SINUMERIK documentation is organized on 3 different levels:

- General documentation
- User documentation
- Manufacturer/Service documentation

This manual is intended for the machine-tool manufacturer. It gives a detailed description of the functions available in the SINUMERIK 840D/810D and SINUMERIK FM-NC controls.

The Description of Functions only applies to the software versions specified. When a new software version is published, the Description of Functions for that software version should be ordered. Old Description of Functions are not necessarily applicable for new software versions.

Please consult your local Siemens office for more detailed information about other SINUMERIK 840D/810D and SINUMERIK FM-NC publications as well as the publications that apply to all SINUMERIK controls (e. g. Universal Interface, Measuring Cycles, ...).

Note

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

Aim

The Descriptions of Functions provide the information required for configuration and installation/start-up.

Target group

The Descriptions of Functions therefore contain information for:

- The configuring engineer
- The PLC programmer who writes the PLC user program by providing signal lists
- The system start-up engineer after configuration and installation of the machinery and equipment
- The service technician for checking and interpreting the status displays and alarms

Structure of the manual

The Description of Functions is structured as follows:

- General list of contents of the manual
- Descriptions of functions, installation and start-up, programming, data backup, data and alarms, PLC sample programs
- Appendix with list of abbreviations, lexicon, references
- Index

Note

Pages indicated provide the following information:
Part of the Description of Functions / Book / Chapter – Page

If you require information on a certain function, you will find the function as well as the code under which the function is organized on the inside title page of the manual.

If you only require information on a certain term please refer to the index in the Appendix. There you will find the code of the Description of Functions, the Section number as well as the page number on which the information about this term is to be found.

Within each of the Description of Functions in Sections 4 and 5 you will find definitions on effect, data format, input limits etc. for the various signals and data definitions.
These definitions are explained in the “Technical comments” section below.

Specification of the software version

The software versions specified in this documentation refer to the SINUMERIK 840D control; the parallel software version for the SINUMERIK 810D control (if the function is released, see /BU/, Catalog NC 60.1) is not specified explicitly. The following applies:

Table 1-1 Equivalent software versions

SINUMERIK 840D software version		SINUMERIK 810D software version
4.3 (12.97)	corresponds to	2.3 (12.97)
3.7 (03.97)	corresponds to	1.7 (03.97)

Symbols

!

Important

This symbol always appears in the documentation when important information is being conveyed.

Ordering Data Option

In this documentation, you will find this symbol with a reference to an ordering option. The function described is executable only if the control contains the designated option.

⚙

Machine Manufacturer

This symbol appears in this documentation whenever the machine manufacturer can influence or modify the described functional behavior. Please observe the information provided by the machine manufacturer.

⚠

Danger

This symbol appears whenever death, severe bodily injury or substantial property damage **will** occur if the appropriate precautions are not taken.

⚠

Caution

This symbol appears whenever minor bodily injury or property damage **could** occur if the appropriate precautions are not taken.

⚠

Warning

This symbol appears whenever minor bodily injury or property damage **can** occur if the appropriate precautions are not taken.

Technical Comments

Notation

The following notation and abbreviations are used in this documentation:

- PLC interface signals – IS “signal name” (signal data)
  e.g.: – IS “MMC-CPU1 ready” (DB10, DBX108.2), i.e. the signal is stored in data block 10, data byte 108, bit 2.
  – IS “Feed/spindle offset” (DB31–48, DBB0), i.e. the signals are stored for each axis and spindle separately in data blocks 31 to 48, data block byte 0.
- Machine data –> MD: MD_NAME
- Setting data –> SD: SD_NAME
- The character “≙” means “corresponds to”

Explanation of the abbreviations used in Chapters 8 and 9

In Chapters 8 and 9 of the Description of Functions you will find a description of the data and signals that bear relevance to the function concerned. Terms and abbreviations used in these tabular descriptions are explained here.

Values in the table

The machine data indicated in the Descriptions of Functions are always values for an NCU572.2.

For the values of the other NCUs (e.g. NCU570, NCU571, NCU573), please refer to the “Lists” documentation.

Reference: /LIS/ “Lists”

Default value

The machine/setting data is preset with this value during installation. If these default values are different for each channel, this is marked by an “/”.

Value range

States the input limits. If no value range is specified, the input limits are dependent on the data type and “∗∗∗” is displayed next to the field.

Activation

When machine data, setting data etc. are altered they are not immediately active. Information about activation of alterations is therefore always stated. The following is a list of the possible activation conditions in order of priority:

- POWER ON (po) "RESET" key on the front panel of the NC module
  or by switching the power off/on
- NEW_CONF (cf) – "Re-configuring" of PLC interface function
  – "RESET" key on operator panel
- RESET (re) "RESET" key on operator panel
- Immediately (im) After input of value [immediately]

Protection levels

Protection levels 0–7 are available, the protection of levels 0 to 3 (4 to 7) can be removed by setting a password (keyswitch setting). The user only has access to information for a certain protection level or for the lower protection levels. The machine data are assigned different protection levels as standard.

The table lists write protection levels only because read protection levels are derived from the write protection levels:

Write protection level	Read protection level
0	0
1	1
2	4

Reference: /BA/, Operator's Guide
/FB/, A2, "Various Interface Signals"

Unit

The unit refers to the machine data default setting
SCALING_FACTOR_USER_DEF_MASK and
SCALING_FACTOR_USER_DEF.
If the MD is not based on a physical unit a "–" is entered.

Data type

The following data types are used in the control:

- DOUBLE
  Real or integer values
  input limits from +/–4.19*10^{-307} to +/–1.67*10^{308}
- DWORD
  Integer values
  Input limits from –2.147*10^{9} to +2.147*10^{9}
- BOOLEAN
  Possible input values: true or false or 0 or 1
- BYTE
  Integer values from –128 to +127
- STRING
  consisting of max. 16 ASCII characters (capital letters, digits and underscore)

Data management

The PLC interface descriptions in the individual Description of Functions assume a theoretical maximum number of components:

- 4 mode groups (associated signals stored in DB11)
- 8 channels (associated signals stored in DB21–30)
- 31 axes (associated signals stored in DB31 to 61)

Please consult the Description of Functions for the number of components actually implemented for each of the software versions
References: /FB/, K1, “Mode Groups, Channels, Program Operation Mode”

■

Notes

Contents

Introduction 1

The purpose of the Tool Management function (TM) is to ensure that the correct tool is in the correct location on the machine at any given time and that the data assigned to the tool are up to date. The function is used on machine tools with circular, chain or box magazines. It also allows tools to be loaded quickly, prevents wastage by monitoring periods of tool usage and machine idle times by taking into account replacement tools.

TM functions

The main tool management functions are as follows:

- Tool selection from all magazines and turrets for active tools and replacement tools (WZMG)
- Identification of suitable empty locations as a function of tool size and location type (WZMG)
- Tool-related location coding (fixed and variable) (WZMG)
- Initiation of tool changes via T or M06 command (WZBF)
- Axis motions during tool change with automatic synchronization on next D number (WZBF)
- Monitoring of workpiece count, tool life and wear with prewarning limit alarm (WZMO)

The function is capable of managing up to 30 real magazines with a total of 600 magazine locations and 600 tools, and up to 12 edges per tool (max. 1500 tool edges). The maximum number of edges per tool depends on the software version (12 edges in SW version 5.1 and later) and machine data settings.

With the MMC 103 installed to provide the most user-friendly configuration, the full range of tool management functions is available. But even with an OP 030 or MMC 100.2, the main functions can be utilized on a task-related basis.

New structure

The range of functions to be executed by the tool management system has been extended ever further over time. A new structure based on the following categories of function will be selected in future:

WZBF Basic functions of tool management (available even when tool management is not active)
WZMO Tool monitoring
WZMG Tool magazine management (only available when tool management is active)
WZFD Tool management with “flat” D numbers (only when tool management is not active)

Main tool management functions (standard)	MMC 103	MMC 100.2	OP 030
System displays in standard software	X	X	X
Options for configuring screenforms and softkeys	X		
User-friendly start-up via system displays	X		
Editing of tool data	X	X	X
Magazine and tool lists	X	X	X
Empty location search and location positioning	X	X	X
Loading and unloading of tools	X	X	X
Easy search for empty locations via softkeys	X		
Several real magazines possible	30	1 magazine from SW 5.3: 30 magazines	1 magazine from SW 5.3: 30 magazines
Several load/unload positions per magazine	X		
Tool cabinet and tool catalog	X		
“Relative” D number with free numbering	X		
Adapter data	X		
Location-dependent offsets	X		
Loading and unloading via code carrier system	X		
Data backup via V.24 interface	X	X	
Data backup to hard disk	X		

1.1 General overview

This document describes and illustrates the scope of tool management functions. TM functions are implemented on the MMC, NCK and PLC. The appropriate functions are shown in the function structure (see Section 2.1). The tool management is divided into several subareas, which were outlined in the introduction.

Basic functions

The WZBF area of the TM structure contains the basic functions. The latter are generally available, even in systems without an active TM system. Basic functions include, for example, creating and deleting tools, entering offsets and tool changes. On the basic function level, a specific number (max. 12) of tool edges (D numbers) is assigned to each T number.
In systems without an active TM system, the “WZFD or flat D number” function (optional D number selection independent of T number) can be activated alternatively. As regards the number of tool edges, there is no fixed maximum limit of 12 edges per tool, but a flexible quantity. With the “flat D number” option, the user is responsible for managing and allocating T numbers to D numbers.

Special functions

The miscellaneous tool management functions include magazine management, searches for tools and empty locations and monitoring of tool service life, workpiece count or wear values. These miscellaneous functions are available only when the TM system (option) is active.
Magazine management functions must be implemented by the machine manufacturer in systems without active TM. These will generally be executed via the PLC.

Magazine management

“Magazine management” refers to the administration of magazine locations. These locations might be empty, loaded with tools or assigned to oversized tools in adjacent locations.
Empty locations can be “loaded” with other tools. The TM function enables the machine manufacturer to manage tools and magazine locations optimally.
The magazine management systems provides extended functions such as load, unload or position tools and includes searches for tools, magazine locations and search strategies for replacement tools. As regards monitoring functions, tools are disabled and excluded from further use after a monitor timeout. The machining operation is continued automatically using an enabled, identical tool (duplo tool) if one is available.

It is most easy to define the configurations of magazines, load magazines, spindles, grippers, etc. when the tool management function is active. Furthermore, the interfaces (DB 71 to DB 73) must be processed in the PLC (see Section 2.3). Task-related tool motions (e.g. position chain, swivel gripper) must be extracted from the interface processing sequence. On completion of a tool motion, the positions and task status must be acknowledged via basic program blocks (FC 7 and FC 8). A cycle (or ASUB) may also be generated for the NC program in which the tool change is programmed with the requisite travel motions. An identifier is programmed for the tool change or tool preselection when the TM system is active. A duplo number is also available to support unique identification of replacement tools. Tool identifier and duplo number are imaged onto an internally allocated T number. The latter is utilized to address the variables described below.

OPI variables

Additional functions are made available via OPI variables (see Section 2.2, PI commands) from the PLC or MMC (see Subsection 5.12.5). Suitable language commands can be included in the NC program (e.g. cycle, ASUB) to optimally adapt the tool management system to machinery features. The user can obtain a clear overview from the data structures (NC data blocks) forming the basis of the TM system.

1.2 Key data

Operator panels

The following operator panels can be used for tool management as from SW 3.2:

- MMC 100.2 (supersedes MMC 100)
  Two interfaces are available:
  – Standard
  – Workshop (Shopmill) as of SW 5.3
- MMC 103 (supersedes MMC 102)
- OP 030 e.g. for load magazines in conjunction with MMC 100.2/103

The following functions are implemented differently on the MMC 100.2 and MMC 103 user interfaces:

- Installation and start-up
- Data backup (hard disk)
- Operation via configured displays.

The following are not implemented in the MMC 100.2:

- Configuring of user softkeys for empty location search
- Start-up via system displays.
- Several load/unload positions per magazine
- Tool cabinet, tool catalog
- “Relative” D numbers with free numbering
- Adapter data
- Location-dependent offsets
- Loading and unloading via code carrier system

Data

Data handling and management are performed in the NC and MMC 103. All data can be read and written by manual entry, NC program or data communication.

Operation

The system is operated on the basis of screen displays. These include screens for start-up (MMC 103 only) and screens for tool management operation (magazine lists, tool lists, loading/unloading).

Programming in the NC part program:

The tool management function makes it possible to call a tool using a name (identifier), e.g. T = “end mill 120 mm”, in the part program.

It is still possible to call a tool with a T number. The T number is then the name of the tool (e.g. T=“12345678”).
A tool is uniquely defined by its name and duplo number. However, every tool can also be identified uniquely via its “internal” T number. The internal T number is generally assigned by the NCK, but not used in the programming of a tool change in the part program.
With magazine type “Circular”, the T call is the change command.
With magazine types “Chain” and “Box-type”, the T call prepares for a change. The M06 function loads the prepared tool into the spindle.

Note

M06 is the generally used term for tool change.

PLC

The tool management function utilizes data blocks (DB71–73) to receive TM commands and function blocks (FC7, 8) to acknowledge them. Another block, FC22, is used as a direction selection for magazines.
Tool management data can also be read and written via FB2 and 3. Complex TM services can be initiated via FB4.

Magazine types

Circular, chain magazines and box magazines can be managed. Other magazine types such as pick-up magazines are imaged on the latter three types.
Real magazines can be defined as a circular, chain or box-type. Load points or load stations must be used as the magazine type for loading and unloading.
Type designation “magazine buffer” covers all other locations in which tools can be placed (spindle, gripper,...).

Location coding

Tools are supported both by fixed location coding and variable location coding.

Location type

The location type defines the type and shape of the location. By assigning location types to magazine locations it is possible to subdivide a magazine into areas. This means that different types of special tools, e.g. “especially_large”, “especially_heavy” can be assigned to specific locations.

The location types can be placed in ascending order or hierarchy. This order determines that a tool that is supposed to be inserted in a “small” location type can also be placed in a “larger” location type if no “smaller” location type is vacant.

Monitoring

In tool management, it is possible to select either workpiece counts or tool life monitoring (with reference to the cutting edges). Tool wear monitoring is also available with SW 5 and higher. Replacement tools (duplo tools) are identified by a duplo number.

Search strategy

Customizable search strategies are available for tool change. Various strategies are possible for tool search and to search for empty location of the “old tool”.
You can still set a search strategy for loading tools.

Excerpt from TM basic data

Term	Data / range
Magazine configurations per channel	**1**
Total number of magazines	max. 32
Total number of magazine locations	max. 600
Total number of tools	max. 600
Programming the tools in the NC program using an identifier (name) with 32 alphanumeric characters	e.g. T = “Angle head cutter_32”
Duplo no.	1–32000
Total number of cutting edges	max. 1500
Location type definition	yes
Adjacent location consideration in half locations	Two-dimensional
Location coding	fixed or variable

Term	Data / range
Strategy for tool search	can be set (programmed) via system variables
Strategy for empty location search	can be set (programmed) via system variables
M06 command for tool change	M code, settable via MD, channel-specific
Tool change with M06 or T command	Settable via MD, channel-specific
Wear monitoring	for every cutting edge
Wear monitoring according to tool life	resolution msec
Wear monitoring according to number of workpieces	counter
Access to TM data via NC program	system variables
Automatic decoding stop until tool is selected.	yes
T = Location no.	can be set via MD

Option

Tool Management is an OPTION.

Overview 2

2.1 Tool management function structure

MMC

- Tool data display, input/output
- Magazine data display, input/output
- Offset data display, input/output
- Tools and material management
  - Master data
  - Particular tool data
  - Code carrier
- Load/unload dialog

NCK

- Tool data management
  - Status
  - Monitoring
  - Offsets
- Magazine data management
  - Magazines
  - Magazine locations
- Tool management
  - Find tool
  - Find empty location
  - Change tool
  - Load, unload

PLC

- Magazine control
- Gripper control
- Spindle control
- Safety interlocks
- Perform tool change
- Calculation of position, if necessary
- Special change strategy, if required

2.2 Data structure MMC/PLC – NCK (OPI)

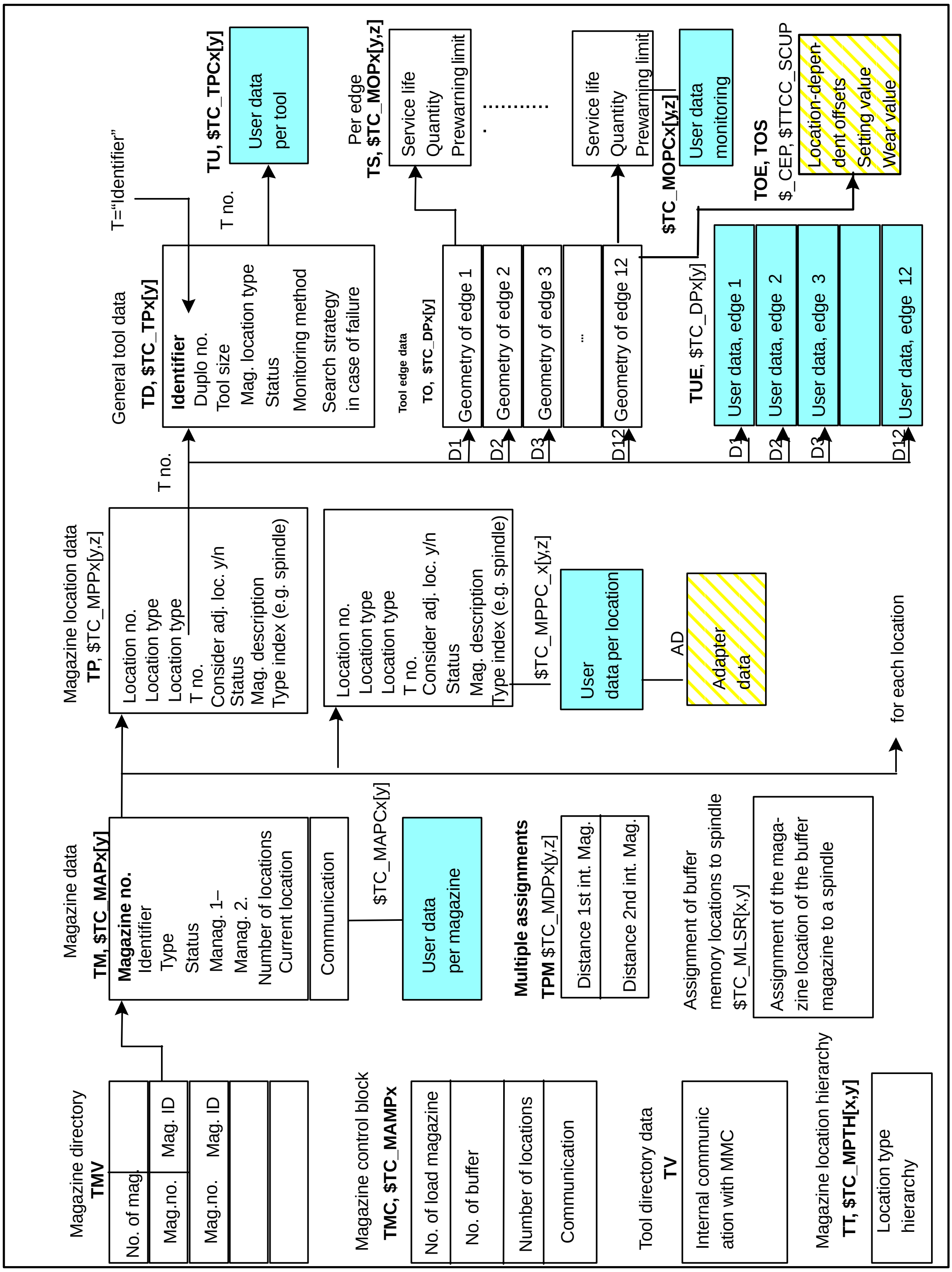


Fig. 2-1 Structure of magazine and tool data

Unchecked boxes mark the previous data of the tool management. Checked boxes show the user data.
New data blocks are displayed as checked and grayed boxes.

TOA area

A TOA area represents an independent area of the tool management system. There is no connection to other TOA areas.
Up to 10 independent TOA areas may be created depending on the number of channels available. Several channels can be assigned to one TOA area, but one channel cannot be assigned to more than one TOA area. A partial quantity of magazines, buffer memories and load magazines can be assigned to one TOA area.

2.3 PLC–NCK interfaces

Overview

The heart of the SINUMERIK 840D tool management system is located on the NCK. The PLC merely contains the interfaces for the machine-specific part (see Fig. 2-2).

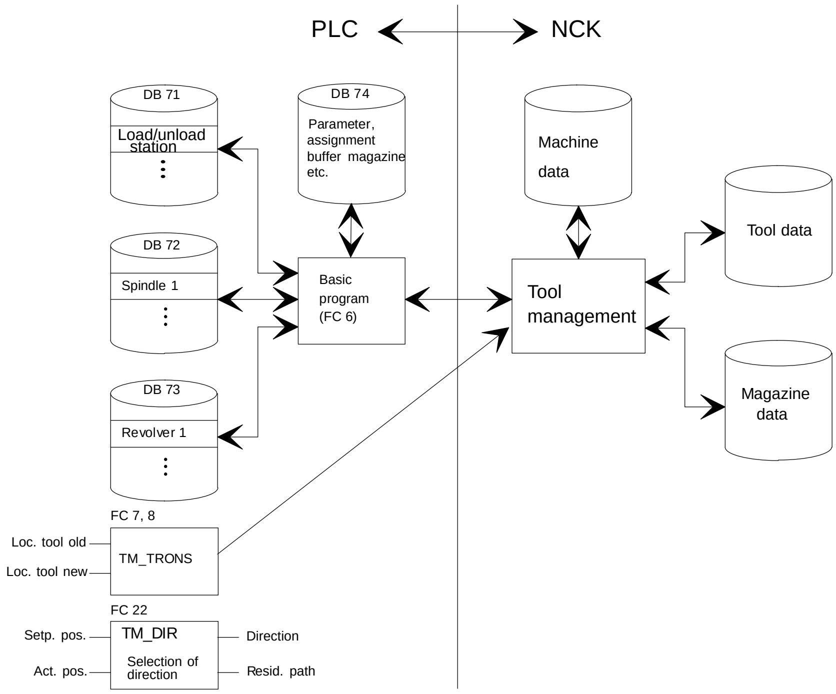


Fig. 2-2 Data structure and PLC–NCK interface

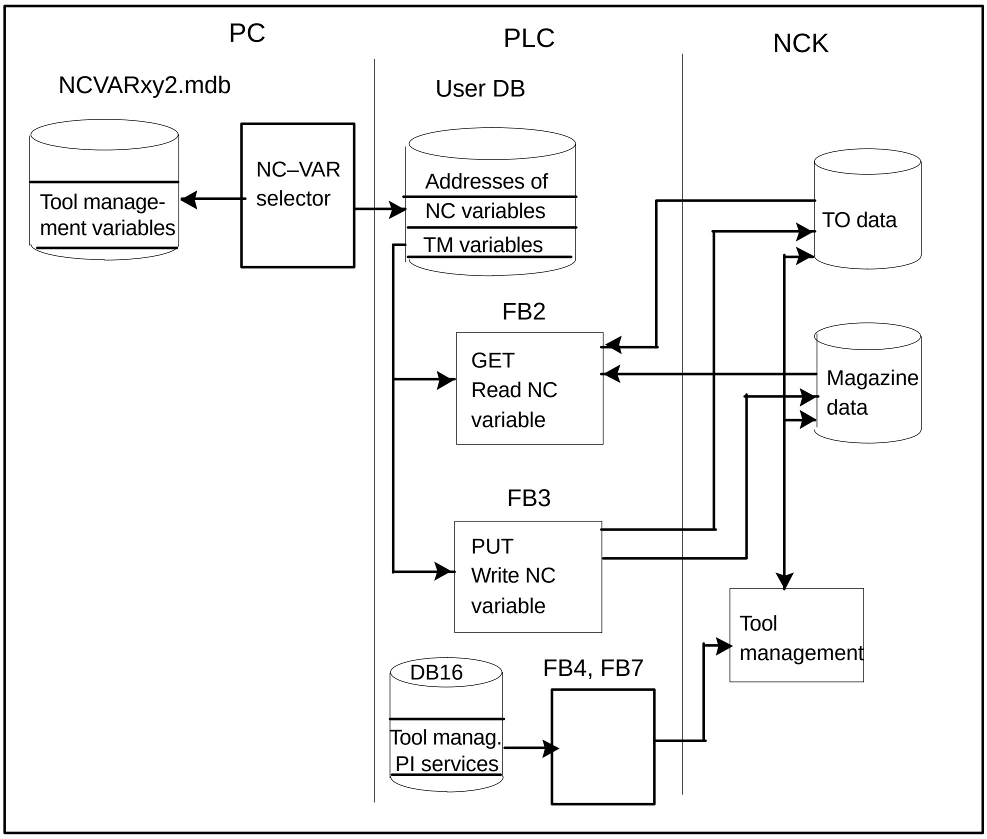


Fig. 2-3 Extended interface for tool management between PLC and NCK

2.4 Magazine configuration

Magazine configuration

In one configuring process, it is possible to create a magazine configuration which includes one or several real (actual magazine for storing tools, NCK is capable of managing several real magazines) magazines. All the magazines of one configuration can be operated simultaneously in one channel. Several magazine configurations can be defined but only one configuration can be active in one channel at one time.

The magazine and tool data are stored in the TO area of the NC. The TO area can also be subdivided into individual TO units in the machine data. The machine data must also specify which channel or channels operate on which TO units. Only one magazine configuration can ever be active for each TO unit. If several channels are assigned to a TO unit the magazine configuration is applicable to all assigned channels.

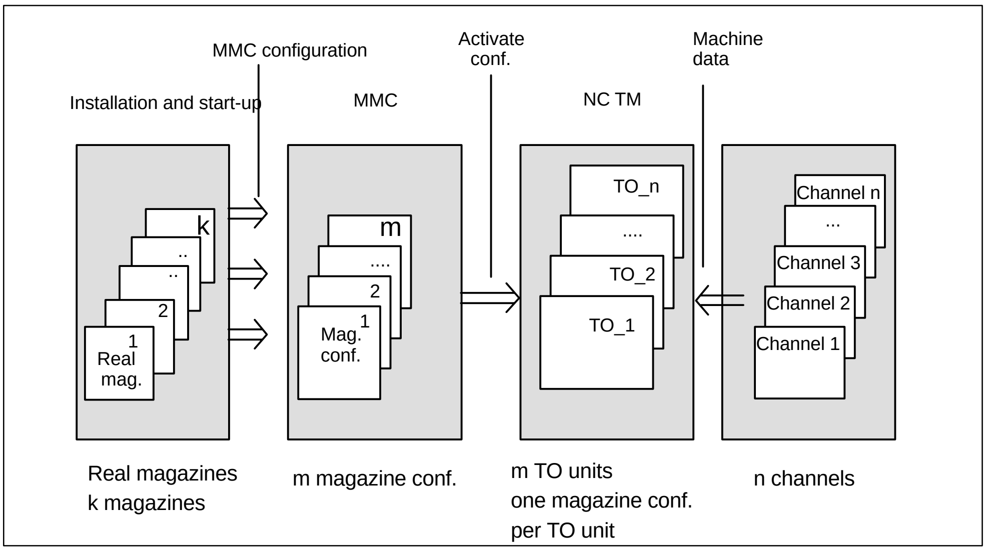


Fig. 2-4 Assignment of magazines to channels

Please refer to Section 3.1 and Subsection 4.2.1 for more information.

2.5 Magazine list

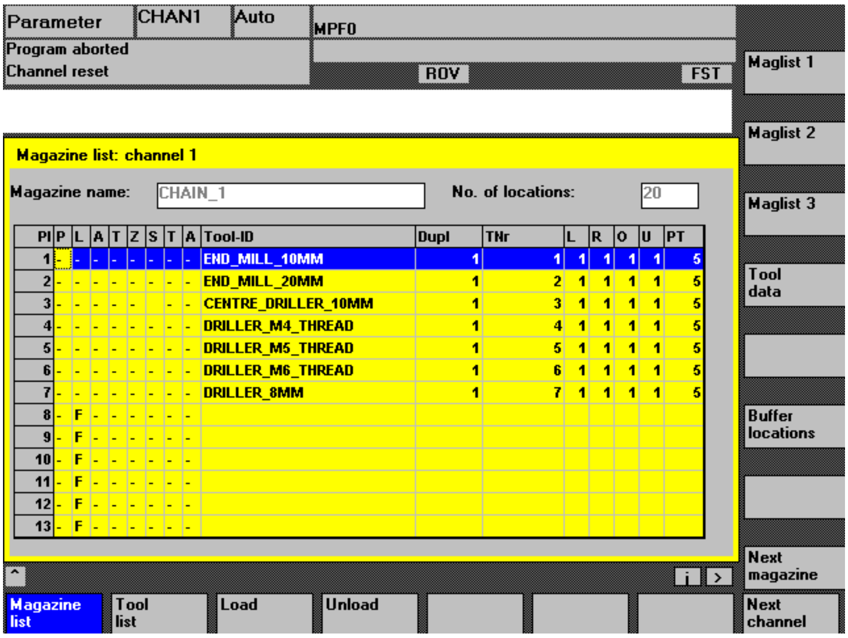
The magazine list is a location-oriented map of the tool magazine, gripper and spindle. Tool management only works with the tools from the magazine list.

Additional tools without a magazine assignment can also be selected for tool changes. The tool must be inserted in the machine manually and removed again manually after machining (manual tool). The same applies to the tool list. For manual tools, see Subsection 3.2.11.

MMC 103

The structure of the magazine list (i.e. which data are to be displayed) is defined by the machine manufacturer via the PARAMTM.INI file. Up to three user-definable displays (screen forms) are available for dividing up the various types of data, e.g. offsets, wear, general data. These displays can be called up via their own softkeys. In the example below: *Maglist 1, Maglist 2, Maglist 3*. The tools that are located in the selected magazine are listed in the magazine list.

The magazine list configured under *Maglist 1* is the tool management basic display. All operations can be selected from this display. One magazine list is available for each channel.



Parameter CHAN1 Auto MPF0
Program aborted
Channel reset
ROV FST

Magazine list: channel 1

Magazine name: CHAIN_1 No. of locations: 20

Pl	P	L	A	T	Z	S	T	A	Tool-ID	Dupl	TNr	L	R	O	U	PT
1	-	-	-	-	-	-	-	-	END_MILL_10MM	1	1	1	1	1	1	5
2	-	-	-	-	-	-	-	-	END_MILL_20MM	1	2	1	1	1	1	5
3	-	-	-	-	-		-	-	CENTRE_DRILLER_10MM	1	3	1	1	1	1	5
4	-	-	-	-	-	-	-	-	DRILLER_M4_THREAD	1	4	1	1	1	1	5
5	-	-	-	-	-	-	-	-	DRILLER_M5_THREAD	1	5	1	1	1	1	5
6	-	-	-	-	-	-	-	-	DRILLER_M6_THREAD	1	6	1	1	1	1	5
7	-	-	-	-	-	-	-	-	DRILLER_8MM	1	7	1	1	1	1	5
8	-	F	-	-	-	-	-	-								
9	-	F	-	-	-	-	-	-								
10	-	F	-	-	-	-	-	-								
11	-	F	-	-	-	-	-	-								
12	-	F	-	-	-	-	-	-								
13	-	F	-	-	-	-	-	-								

Fig. 2-5 Example of a magazine list

2.6 Tool list

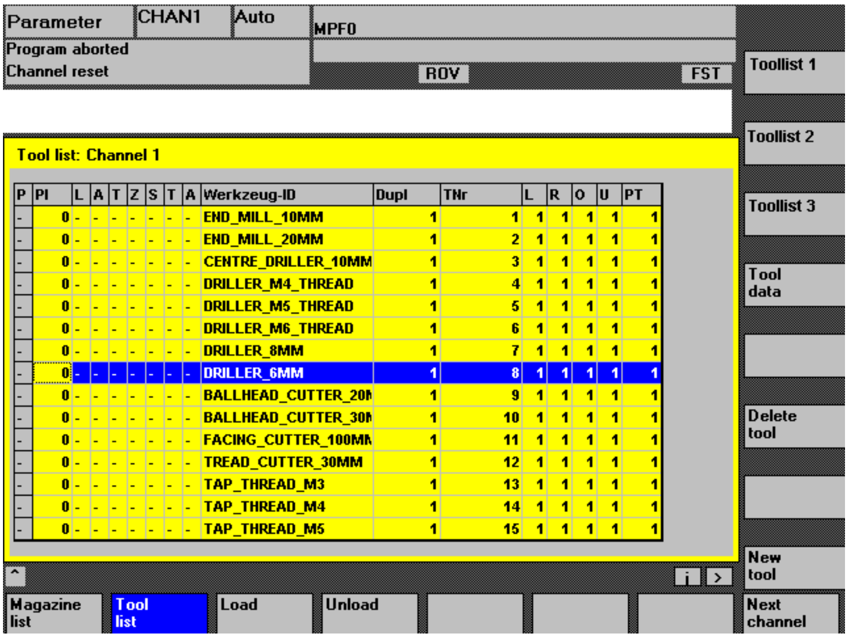
The tool list contains all the tools known to the NC. These are the tools in the magazine and tools which have been unloaded but whose data are to be retained.

The tool management function works with loaded tools from the magazine list.

MMC 103

The structure of the tool list is defined by the user. The data can be displayed in up to three user-definable displays (screen forms). In the example below: *Toollist 1, Toollist 2, Toollist 3*.

In the tool list, all tools of the TO area are sorted according to their internal T number, including tools which are not assigned to any magazine location. The display under softkey "*Toollist 1"* is the main display.



Tool list: Channel 1

P	Pl	L	A	T	Z	S	T	A	Werkzeug-ID	Dupl	TNr	L	R	O	U	PT
-	0	-	-	-	-	-	-	-	END_MILL_10MM	1	1	1	1	1	1	1
-	0	-	-	-	-	-	-	-	END_MILL_20MM	1	2	1	1	1	1	1
-	0	-	-	-	-	-	-	-	CENTRE_DRILLER_10MM	1	3	1	1	1	1	1
-	0	-	-	-	-	-	-	-	DRILLER_M4_THREAD	1	4	1	1	1	1	1
-	0	-	-	-	-	-	-	-	DRILLER_M5_THREAD	1	5	1	1	1	1	1
-	0	-	-	-	-	-	-	-	DRILLER_M6_THREAD	1	6	1	1	1	1	1
-	0	-	-	-	-	-	-	-	DRILLER_8MM	1	7	1	1	1	1	1
-	0	-	-	-	-	-	-	-	DRILLER_6MM	1	8	1	1	1	1	1
-	0	-	-	-	-	-	-	-	BALLHEAD_CUTTER_20M	1	9	1	1	1	1	1
-	0	-	-	-	-	-	-	-	BALLHEAD_CUTTER_30M	1	10	1	1	1	1	1
-	0	-	-	-	-	-	-	-	FACING_CUTTER_100MM	1	11	1	1	1	1	1
-	0	-	-	-	-	-	-	-	TREAD_CUTTER_30MM	1	12	1	1	1	1	1
-	0	-	-	-	-	-	-	-	TAP_THREAD_M3	1	13	1	1	1	1	1
-	0	-	-	-	-	-	-	-	TAP_THREAD_M4	1	14	1	1	1	1	1
-	0	-	-	-	-	-	-	-	TAP_THREAD_M5	1	15	1	1	1	1	1

Fig. 2-6 Example of a tool list

Modification of tool identifiers and duplo numbers in the lists

Renaming tools

In SW 5.2 and higher, the operator can change the tool name and the duplo number directly in the magazine, tool and working offset lists and in the tool details display. It is not possible to change the tool type directly in the working offset list.

Modifying tool identifier and duplo number

MD 9240: USER_CLASS_WRITE_TOA_NAME can be set to determine whether or not the tool identifier and duplo number can be altered in lists by the user. The default value is always "0". With this value, no changes can be made by the operator in the lists.

Modifying tool type

MD 9241: USER_CLASS_WRITE_TOA_NAME can be set to determine whether or not the user can change the tool type directly in the tool list, magazine list and the tool details display (see Subsection 4.5.1). The default value for the MD is "0".

With this setting, the operator cannot make direct changes in the lists.

Note

It is not possible to change the tool type directly in the working offset list. Changing the tool type of a cutting edge automatically changes the tool type for all cutting edges of the same tool. The cutting edges of a tool do not appear in direct succession in the working offset list, because they are sorted according to user-assigned D numbers.

Tool type function

If the operator changes the tool type of a cutting edge, the tool type of the other cutting edges of the same tool is also changed.

The following data are set to 0:

- Tool user data
- Tool offset parameters of all cutting edges
  (the cutting edge adapter data are not changed if the tool is located at a magazine location and the “magazine location adapter data” function is active on the NC.)
- Cutting edge user data of all cutting edges
- Edge monitoring data of all cutting edges
- Location-dependent offset parameters of all cutting edges (wear values and setup values)

Configuration of the modification procedure

Before the changes are made to the tool type, the operator is asked to confirm the changes. This prompt can be suppressed, depending on the current access rights, by setting the following data in paramtm.ini:

```
[ACCESSLEVEL]
ChangeToolTypeWithoutConfirmation=–1        ;Value range –1 to 7
```

Confirmation is always requested with the default “–1”. Entering an access level (values 1–7) specifies the lowest access level at which the prompt is to appear.

Confirmation is requested with keyswitch “0” (access level 7). The prompt is suppressed with keyswitch “1” and higher (access level 6 and lower).

Example:

```
[ACCESSLEVEL]
ChangeToolTypeWithoutConfirmation=6         ;Value range –1 to 7
```

New magazine list with several lines (SW 5.2 and later)

Cutting edges in magazine list

Several lines are available for each tool in the magazine list in SW 5.2 and higher. The edges for each tool are included in every magazine display.

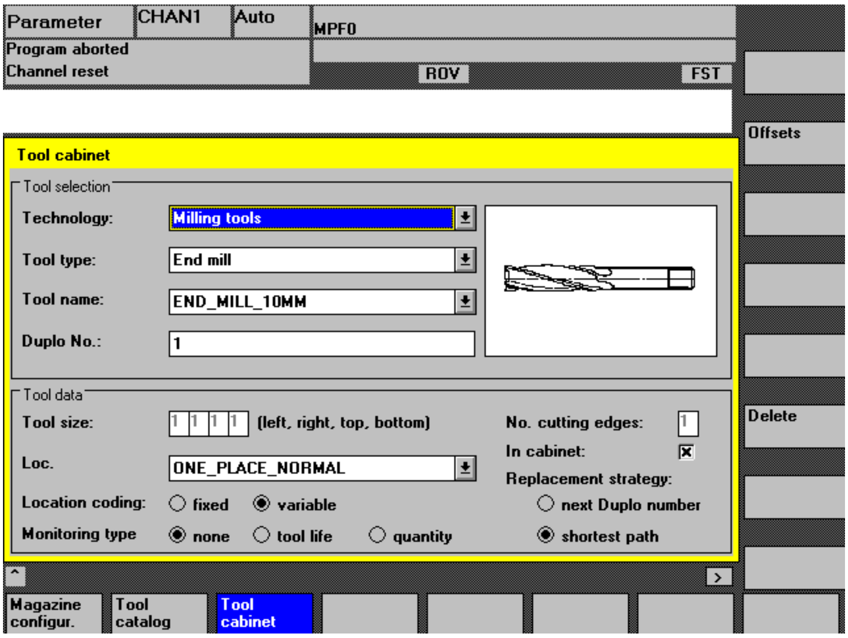
2.7 Tool cabinet (MMC 103 only)

Tool cabinet (MMC)

The data of the tools employed can be stored in the tool cabinet. These data are called particular tool data. When the tools are unloaded the data corrected while the tool was in use can be stored in the tool cabinet. The user can retrieve these data again when he loads the tool. To do this, however, he must know the duplo no. of the tool.

A complete set of tool data is kept in the cabinet for each individual tool in the control, called a selected tool. A replacement tool is identified in the catalog by its technology, tool type, its unique name (for tool type) and its unique (in relation to tool type and name) duplo no. (>0). Every replacement tool thus has a different duplo no., even those with identical technology, type and tool name.

You can store or enter tool data for selected tools in the tool cabinet.



Parameter | CHAN1 | Auto | MPF0

Program aborted

Channel reset

ROV | FST

Tool cabinet

Tool selection

Technology: Milling tools

Tool type: End mill

Tool name: END_MILL_10MM

Duplo No.: 1

Tool data

Tool size: 1 1 1 1 (left, right, top, bottom)

No. cutting edges: 1

In cabinet: ☒

Loc. ONE_PLACE_NORMAL

Replacement strategy:

Location coding: ○ fixed ● variable

○ next Duplo number

Monitoring type ● none ○ tool life ○ quantity

● shortest path

Fig. 2-7 Example of tool cabinet

2.8 Tool catalog (MMC 103 only)

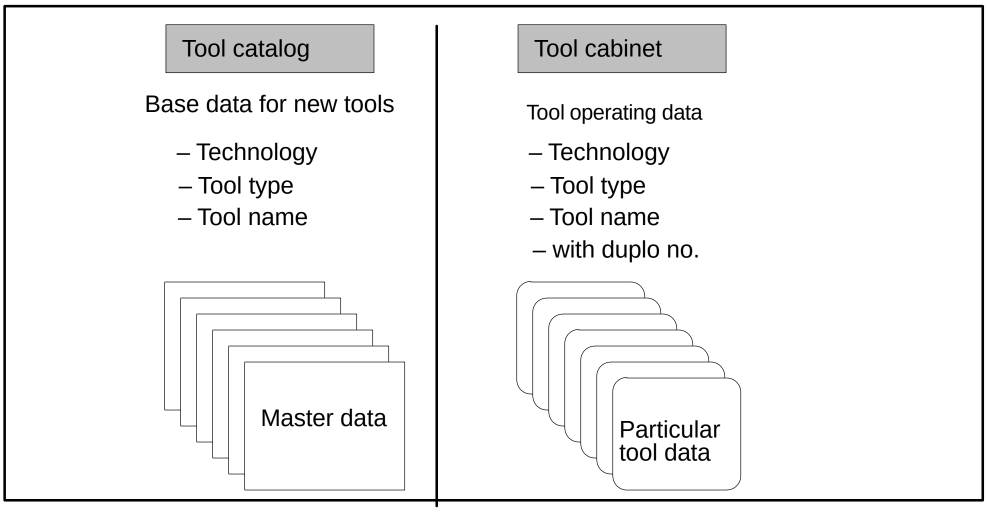


Fig. 2-8 Differences between tool catalog and cabinet

Tool catalog

The tool catalog is empty when supplied. Tool data must be entered before a new tool can be loaded via the catalog. Technology and tool type are selected for this purpose and a toolname specified. The tool and cutting edge data are then entered.

As a result of this process, so-called “master data” have been set up for the tools.

When loading a new tool the user can call these master data. It is not possible to store the data of tools already used. Exactly one master data record therefore exists for each tool, defining a certain technology, a certain type and a certain tool name.
The tool master data are sorted in the tool catalog. The generally applicable tool data as well as setpoint and technology data for tools are handled in the tool master data.

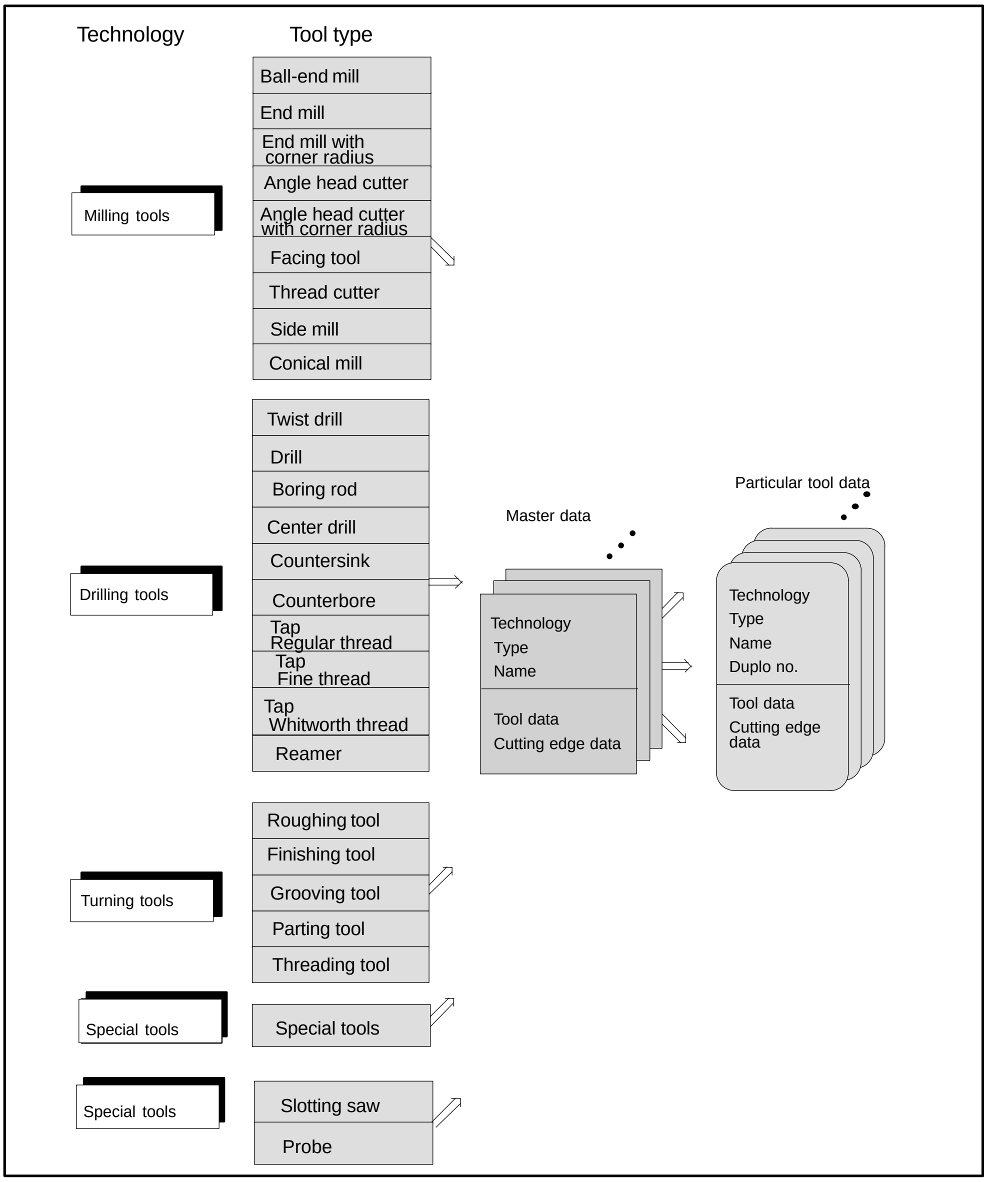
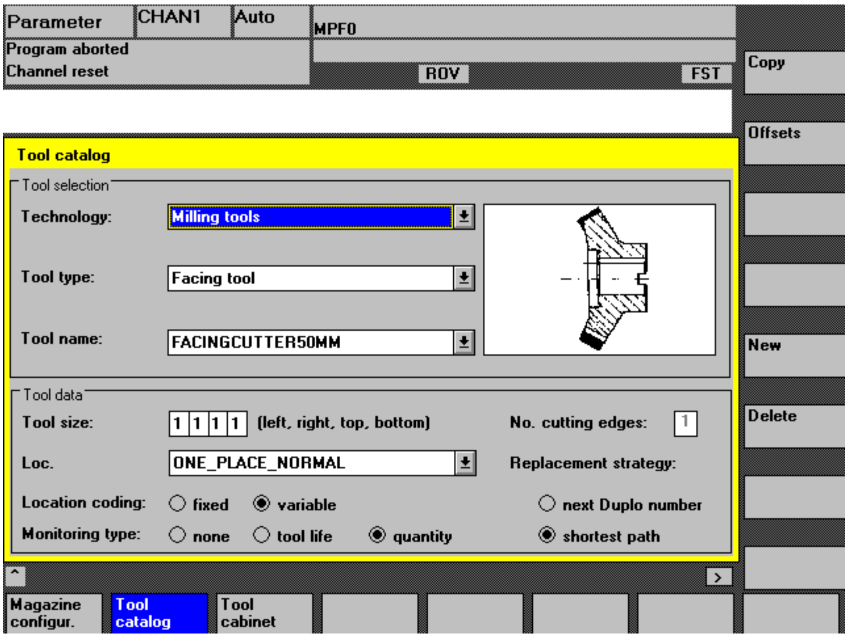


Fig. 2-9 Structure of tool catalog with master and operating data

The full list of tool types is contained in the Programming Guides.



Program aborted
Channel reset

Tool catalog

Tool selection

Technology: Milling tools

Tool type: Facing tool

Tool name: FACINGCUTTER50MM

Tool data

Tool size: 1 1 1 1 (left, right, top, bottom)   No. cutting edges: 1

Loc. ONE_PLACE_NORMAL   Replacement strategy:

Location coding: ○ fixed ● variable   ○ next Duplo number

Monitoring type: ○ none ○ tool life ● quantity   ● shortest path

Fig. 2-10 Example of tool catalog

Note

Master data can only be read out of the control and transferred to another in their entirety.

Tool data:

The following data can be entered for every tool:

- Tool size e.g. 2222 (left, right, bottom, top)
- location type
- Location coding (fixed, variable)
- Tool monitoring (none, tool life, workpiece count, wear)
- Number of cutting edges (display of defined edges only)
- Search strategy for replacement tool (next duplo no., shortest path, etc.)

Cutting edge data:

- Cutting edge position
- Number of cutting edges (display only)

- Cutting edge number (display only)
- Offset parameters (geometry, wear, base)
- Monitoring data (setpoint, prewarning limit)

Note

The database of the particular tool data can only be read out of the control and transferred to another control in its entirety.

2.9 Access protection, access levels

The access to programs, data and functions is protected via 8 hierarchical levels according to customer requirements. These are subdivided into

- 4 password levels for Siemens, machine manufacturer and end user
- 4 keyswitch settings for end users

Protection level	Locked by:	User
0	Password	Siemens
1	Password	Machine manufacturer: Development
2	Password	Machine manufacturer: Start-up engineer
3	Password	End user: Servicing
4	Keyswitch position 3	End user: Programmer, machine setter
5	Keyswitch position 2	End user: Skilled operator without programming knowledge
6	Keyswitch position 1	End user: Trained operator without programming knowledge
7	Keyswitch position 0	End user: Semi-skilled operator

For further information, see Subsection 4.4.1.

MMC 103: The access protection is defined in file c:\user\paramtm.ini (see Chapter 4). It must be entered after vocabulary word

[ACCESSLEVEL].

MMC 100.2: Protection is set via display machine data.

Examples of functions that can be disabled:

- Loading
- Unloading

- Magazine list, tool list display
- Tool cabinet, tool catalog
- Loading the magazine configuration

2.10 Openess in MMC

OA / OEM package

The OEM / OA Package for MMC103, HMI-Advanced can also be used to extend the operator screen forms and functions of the tool management.

OEM package MMC / Open Architecture Package MMC, Order number FC5 253-0AX20-0AB0. Please refer to the order catalog NC 60.1 for the current status.

The OPI variables and PI services are available for extending the functionality. The additional function are accessed with OEM softkeys.

The OPI variables are described in the OPI_GR.HLP or OPI_UK.HLP files in the MM2\HLP directory.
The help files OPI_GR.HLP or OPI_UK.HLP are provided in the OEM package.

For more information, please refer to Subsection 5.12.5 in this description.

Description of Functions 3

In this chapter, reference is made to variables, alarms and machine data. A detail description of these features can be found in the following chapters:

Chapter 5:	Programming
Chapter 8:	Machine Data
Chapter 9:	Signal Description, PLC Interfaces
Chapter 10:	Alarms

3.1 Magazines

The position of a tool is identified by a magazine identifier and a location identifier. In a real magazine (chain, turret, etc.), the position of the tool is identified by the magazine number and the location within the magazine assigned during start-up.

3.1.1 Buffers

Buffers are located on the second internal magazine. The buffer includes the spindle, toolholderm gripper, loader and transfer location. The buffers are located at magazine number 9998. Each buffer element is assigned a unique location. Any location numbers may be assigned. It is recommended that all spindles and toolholder be numbered in ascending order starting at 1. The assignment to real magazines or of spindles/toolholders to other buffers is made during start-up ($TC_MDP2, $TC_MLSR).

Example: Assigning the locations in the buffer magazine

No.	Name	Type	Index	Assignment to spindles	Distances to magazine
1	Spindle_1	Spindle	1		1
2	Gripper_1	Gripper	1		
3	Gripper_2	Gripper	2		
4	Loader_1	Loader	1		

No.	Name	Type	Index	Assign-ment to spindles	Distances to magazine
5	Loader_2	Loader	2		
6	Transfer_1	Transfer location	1		

3.1.2 Load magazine

The load magazine is the 1st internal magazine and is assigned magazine number 9999. The load magazine contains the loading points.
A distinction is made between

- load points and
- load stations

Load points are provided for loading and unloading tools. The allocation of locations is fixed, all other locations can be assigned freely In the case of fixed assignment, location 1 in the load magazine is used.

Location 1 is reserved for loading/unloading to all spindles/toolholders. All positioning tasks for relocation actions to any locations (not load points) are still handled via the 1st location. The jobs stated, which refer to a particular magazine location, are output at the interface of the load point. The load points are assigned to magazines during start-up ($TC_MDP1). A load point is an open access to the magazine where a tool can be directly added to or removed from the magazine **manually**.

A load point is viewed as an “external magazine location” which a gripper, for example, can access to transfer a tool to the magazine during loading.

3.1.3 Box, chain magazines

The setting in MD 22550 (TOOL_CHANGE_MODE) must always be 1 for these types of magazine.

Chain and box magazines are normally provided with additional buffers for transporting tools between the magazine and spindle. These additional buffers can contain tools temporarily.

Commands are distributed in the PLC by FC 6. In this case, DB 72 acts as the user interface. There is a separate interface area for each spindle in the interface. A new command from the NCK is not entered in the interface until the previous command has been acknowledged with status values less than 100 (today 1 ..7) via FC 8.

1. The programming function T = identifier or T = location is implemented in the PLC in data block DB 72. Bit “Prepare tool” is activated in the associated interface.
2. Programming function M06 is also implemented in DB 72. In this instance, bit “Change tool” is set in the activated interface. The “Prepare tool” bit from a previous T command is not reset in this case. If the “Prepare tool” bit must not be set for the M06 function, then measures must be taken in the user program to reset the bit on the final acknowledgement of the T command.
3. Programming functions T and M06 in the same block set the “Prepare tool” and “Change tool” bits simultaneously in the activated DB 72 interface.

Exceptional cases which are imaged in the PLC identically to “3.” above are as follows:

- Initiation of a tool change after block search (last accumulated tool change for the active tool)
- Initiation of a tool change with Init blocks.

Caution: In these exceptional cases, the subroutine (macro, cycle) in which M06 is normally programmed is not executed.

Examples for machine tools with chains and flat magazines

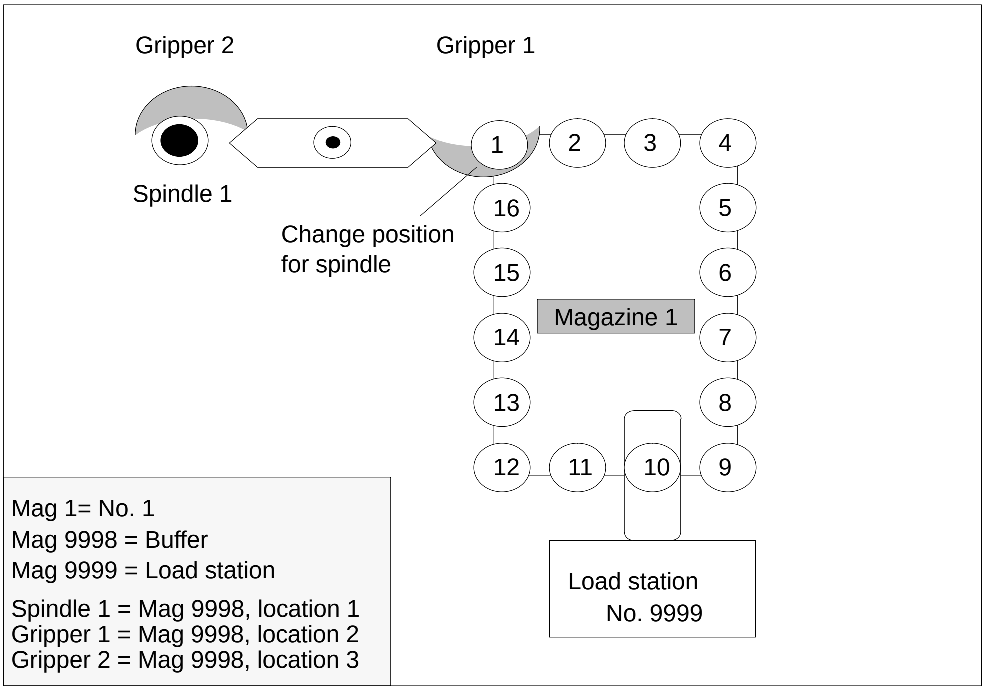


Fig. 3-1 Machine tool with chain magazine

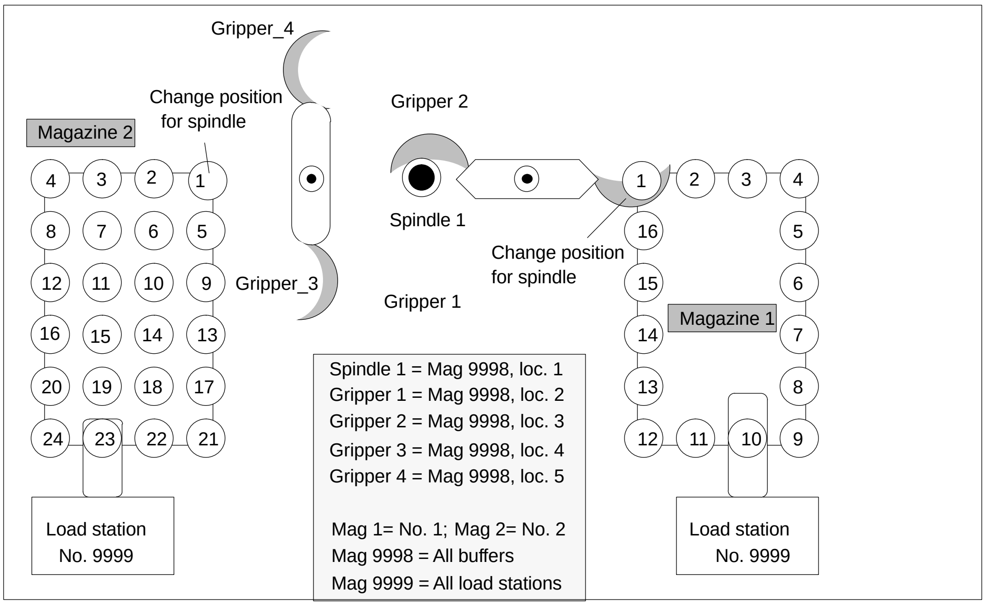

Gripper_4
Change position for spindle
Magazine 2
Gripper 2
4 3 2 1
8 7 6 5
12 11 10 9
16 15 14 13
20 19 18 17
24 23 22 21
Gripper_3
Spindle 1
Change position for spindle
Gripper 1
1 2 3 4
16 5
15 6
14 7
13 8
12 11 10 9
Magazine 1
Spindle 1 = Mag 9998, loc. 1
Gripper 1 = Mag 9998, loc. 2
Gripper 2 = Mag 9998, loc. 3
Gripper 3 = Mag 9998, loc. 4
Gripper 4 = Mag 9998, loc. 5
Mag 1= No. 1; Mag 2= No. 2
Mag 9998 = All buffers
Mag 9999 = All load stations
Load station No. 9999
Load station No. 9999

Fig. 3-2 Machine tool with chain and box magazines

The magazine zero point is defined by $TC_MDP2 (with value assignment of 0). The change position (spindle) is normally assigned on this basis.

3.1.4 Circular magazine

The setting in MD 22550 (TOOL_CHANGE_MODE) is normally 0.

Circular magazines do not have any additional buffer with which tools can be transported from the magazine to the spindle. The tools on circular magazines are not physically transported to the spindle, but are moved into a defined position through rotation of the turret so that machining can take place with one particular tool. The tool is transported to the spindle or holder only in the software.

If TOOL_CHANGE_MODE is set to 1 for a turret, then the description above for chain and box magazines applies analogously.

The description below applies when TOOL_CHANGE_MODE = 0.

Programming command T = identifier initiates the tool change. T = location can be programmed as an alternative. When T = location, no tool need actually be stored in the location.

The commands is distributed in the PLC by FC 6. In this case, DB 73 is the user interface. There is a separate interface area for each turret. The turret numbers are assigned successively in ascending sequence according to magazine numbers during start-up. The permissible magazine range is 1 ... max. number of real magazines. A new command from the NCK is not entered in the interface until the previous command has been acknowledged via FC 7 (alternatively via FC 8).

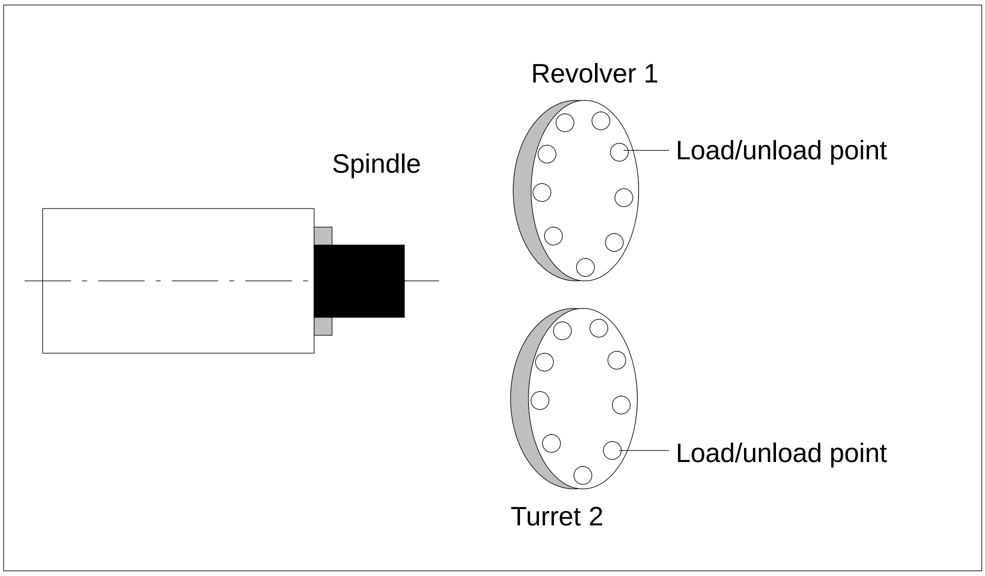


Fig. 3-3 Double-slide turning machine with direct loading/unloading point in turret

3.1.5 Other magazine types

In practice, there are other types of magazines in addition to the ones listed above. For example, disk-type, wheel-type, pickup, storage, basket magazines (to name but a few). Such types must be mapped to the three types of magazines supported by the tool management .

3.1.6 Wear group (SW 5.1 and later)

Locations in a magazine are linked to form an area referred to as the “wear group”. In this way, location groups of a magazine can be activated for particular machining operations.

A wear group number is assigned to each of these locations and the magazine thereby divided into several different areas. Only tools from one of the areas are then used for a specific machining operation.

The wear group number for each magazine location is defined via system parameter **$TC_MPP5[m,p]** (m: Magazine number, p: Location number).

Values in the range – 32000 ... + 32000 can be assigned.

Values > 0: The specified number is assigned to the location (e.g. **$TC_MPP5[1,3] = 2** assigns the third location of the magazine to wear group number 2).

Value = 0: The location is not assigned to a wear group, as a result the magazine locations are not generally included in tool searches.
If the parameter is set to 0, the data will be fully compatible with magazine data generated in older NCK software versions.

Values < 0: The wear group whose number corresponds to the absolute value of this number is disabled (e.g. **$TC_MPP5[1,3] = – 2** disables wear group number 2 of the magazine with number 1).
This applies even if there is only one disabled location in a wear group.

Note

Wear groups are only available for real magazines. The definitions for $TC_MPP5 do not affect the status of tools.

Activate wear group

System parameter $TC_MAP9 defines which wear group (magazine area) is active. To change to active wear group, the corresponding number is set in this parameter, thereby defining which wear group will be used to start the machining operation.

The default setting is 0, thereby ensuring compatibility with magazine data generated on older NCK software versions.

The wear group can also be set internally with a tool change or by the user via language commands/OPI active.

Disable wear group

If there are no more replacement tools in any location of the active wear group, the next wear group is activated and the old group disabled.

Machining is continued by activating the next group and searching for a suitable replacement tool.

The wear group is also disabled if one of the locations has been disabled via system parameter $TC_MPP5 (negative value).

Activate (internally)

Bit 0 of system parameter **$TC_MAMP3** can be set to determine how internal activation of a wear group will affect the status of the tools it contains.

Value 0: The tool status is not changed (default setting).

Value 1: When activated, one tool from each tool group included is set to "active". Any tools that have already been set to active are not reset.

Disable (internally)

Bit 1 of system parameter $TC_MAMP3 can be set to determine how internal disabling of a wear group affects the state of the tools it contains.

Value 0: The tool status is not changed (default setting).

Value 1: When a wear group is disabled all active tools are reset.

Note

For information about tool searches in wear groups, see Subsection 3.4.5.

3.1.7 Background magazine

Background magazines are not directly supported by the tool management. However, functions for background magazines can be activated by setting the system parameter selectively. System parameter $TC_MAMP2 – bit 7 can be used to set whether the tool search begins in the magazine last used for tool replacement (bit 7 = 0) or whether the search is carried out in the order defined by "Spindle to magazine" (bit 7 = 1).
This system parameter is allocated during magazine configuration (via start-up at the MMC) and saved as an INI file; $TC_MAMP2 – bit 7 is always preset to 0. For these reasons, the value of $TC_MAMP2 must either be modified in the INI file (prior to loading the magazine configuration) or overwritten per part program:
$TC_MAMP2=385 (bits 0, 7 and 8 set).

The assignment of "spindle to magazine" is set via system parameter $TC_MDP2[x,y]; the order corresponds to the order in which this variable is written. It is also preassigned via the magazine configuration:

Example for 4 magazines and one spindle:

$TC_MDP2[1,1]=0
$TC_MDP2[2,1]=0
$TC_MDP2[3,1]=0
$TC_MDP2[4,1]=0
... the first buffer (spindle) is assigned to magazines 1 to 4 above this; a tool search would therefore start in magazine 1, followed by magazine 2, etc. up to magazine 4.

This search order can be manipulated as follows by writing this parameter:

1. Delete assignment:
   $TC_MDP2[1,0]=0
   $TC_MDP2[2,0]=0
   $TC_MDP2[3,0]=0
   $TC_MDP2[4,0]=0
2. Re-assign in different order:
   $TC_MDP2[2,1]=0
   $TC_MDP2[3,1]=0
   $TC_MDP2[4,1]=0
   $TC_MDP2[1,1]=0
   ... which produces the search order Magazine 2, 3, 4, 1

The trigger criterion for changing the order of assignment can be the information in the change cycle that the new tool was found in another magazine. This can be read in the program via $A_TOOLMN[t], where “t” is the internal T number of the tool. The new tool is obtained via GETSELT. You must remember the previous foreground magazine.

3.2 Tool changes on box, chain and circular magazines

A tool change from a box or chain magazine is programmed differently from the tool change operation from a circular magazine.

The different methods of tool change used for these three magazine types are set channel-specifically via machine data MC_TOOL_CHANGE_MODE.

3.2.1 Prepare a tool change

Different methods of tool change can be programmed as a function of machine data (MD 22550) $MC_TOOL_CHANGE_MODE:

$MC_TOOL_CHANGE_MODE=0

-
  ```
  T=“Tool identifier”            ;Tool preparation and tool change with
                                 ;an NC language command (= within an
                                 ;NC block)
                                 ;NCK outputs command to PLC
  ```

If an error is detected during tool preparation, processing halts and the block with T identifier is read in.

After error rectification and NC Start, the block with T=identifier is interpreted again and the program run continued.

$MC_TOOL_CHANGE_MODE=1

- In one NC block
  ```
  T=“Tool identifier” M06        ;Tool preparation and tool
                                 ;change
                                 ;This programming line results in a
                                 command to the PLC
  ```
  Programming tool preparation and tool change in one block (**T**= “Tool identifier” **M06**) corresponds to setting TOOL_CHANGE_MODE = 0.
- In two NC blocks
  ```
  T=“Tool identifier”            ;Tool preparation
                                  ;NCK outputs a command to the PLC
  M06                            ;Tool change (the M code number is
                                 settable)
                                 ;NCK outputs a command to the PLC
  ```

Tool preparation and tool change are typically programmed in different blocks. Two commands are transferred to the PLC.

If an error occurs in T= “Tool ID”, an alarm is triggered. If the MD TOOL_CHANGE_ERROR_MODE (MD 22562) is set appropriately, the alarm is delayed until the associated tool change command M06 is interpreted in the program run. Only then is the alarm output. The operator can make corrections in this block.

Note

A D offset is activated with the tool change. If the D command is not programmed in the block containing the tool change command, the tool offset set in MD 20270: CUTTING_EDGE_DEFAULT is activated. If the value of the variable is –1 or greater than zero (selection of a specific offset), the alarm 17181: “D number for tool does not exist in NCK” might be generated.

If the value is 0 (offset deselected) or. –2 (old offset retained), there is no problem when determining the offset.

Empty spindle

Program commands T0 and M06 remove the tool from the spindle and return it to the magazine The spindle is then empty.

Possible problems in programming T / M06

$MC_TOOL_CHANGE_MODE=0; tool change with T address
The part program is processed up to the record T=“identifier”. The following problems can occur and are handled in the manner described:

- The tool data record is in the NCK but not assigned to a magazine location. The tool must be reloaded mechanically, if necessary, e.g. directly onto the spindle. The tool is assigned to the magazine location / spindle, for example, with the “Overstore” function $TC_MPP6[m,p] = T no. or by the MMC operation “Load to spindle”.
- The tool data record is not in the NCK:
  Create data record, e.g. via MMC operation, in NCK.
- Programming error in part program:
  Correct faulty NC block in the part program.
- Alarm 22067:
  The desired tool change is not possible. The specified tool group does not contain a “ready to use” replacement tool which could be loaded. The tool monitoring function may have set all potentially suitable tools to the “disabled” status.

The START key is pressed after the appropriate corrective action has been taken. The NC block with T = “Tool identifier” is interpreted again and the program run continues provided that the operator has responded correctly. If not, the alarm will be generated again.

3.2.2 General tool change sequence

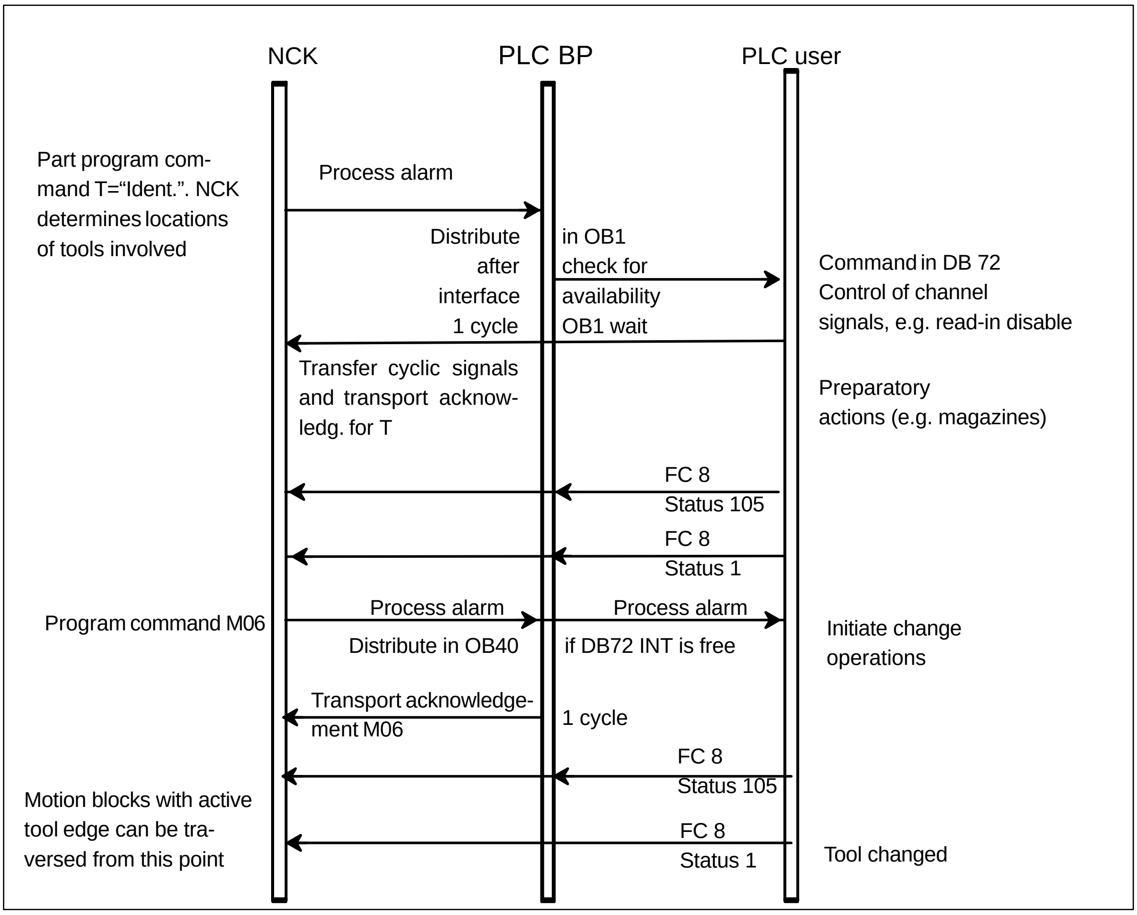

NCK
PLC BP
PLC user
Part program command T="Ident.". NCK determines locations of tools involved
Process alarm
Distribute after interface 1 cycle
in OB1 check for availability OB1 wait
Command in DB 72 Control of channel signals, e.g. read-in disable
Transfer cyclic signals and transport acknowledg. for T
Preparatory actions (e.g. magazines)
FC 8 Status 105
FC 8 Status 1
Program command M06
Process alarm
Distribute in OB40
Process alarm
if DB72 INT is free
Initiate change operations
Transport acknowledgement M06
1 cycle
FC 8 Status 105
Motion blocks with active tool edge can be traversed from this point
FC 8 Status 1
Tool changed

Fig. 3-4 Preparing and changing a tool

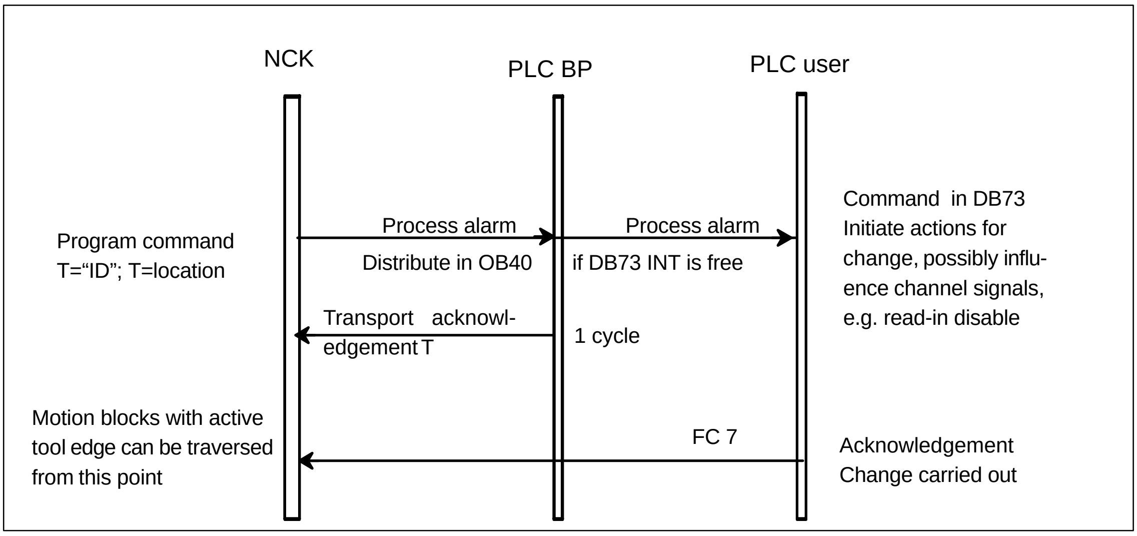

NCK
PLC BP
PLC user
Command in DB73 Initiate actions for change, possibly influence channel signals, e.g. read-in disable
Program command T="ID"; T=location
Process alarm
Distribute in OB40
Process alarm
if DB73 INT is free
Transport acknowledgement T
1 cycle
Motion blocks with active tool edge can be traversed from this point
FC 7
Acknowledgement Change carried out

Fig. 3-5 Change tool with T command

A tool change is triggered from the part program by the T command or by the M command.

1. The tool management in the NCK searches for a suitable tool (preparation) according to the tool search strategy and the data specified in the T call and,a t the same time, searches for an empty location for the tool to be changed.
2. The calculated data are stored in DB 72/73. The user program must react by making a new tool available.
3. If machine data MD 22550: TOOL_CHANGE_MODE is set to 1, the PLC executes the tool change with the “M06 command” in the part program and signals the completion of the change operation.
   If the machine data is set to 0, the tool data is changed and the desired offset become active when T or D are programmed. The PLC has the option of applying its own tool change strategy. It can choose its own empty location for storing the old tool.

Example

If, for example, in a tool change with a dual gripper, the old spindle tool is to be replaced in the magazine as “quickly” as possible, the PLC must check whether the location is suitable to accommodate the old spindle tool in terms of type and adjacent locations. The tool change procedure must then be communicated to the tool management via the PLC (FC8 block).

SW 5 and later

The new empty location search strategy “Replace new tool for old” is available as of SW 5. The TM function also checks whether the old tool can be moved to the location of the new tool (1:1 exchange).

Note

The tool change is an internal NCK operation executed in an interactive process with the PLC. The MMC only has the task of displaying data and facilitating data input.

Spindle and toolholder (SW 5 and higher)

Tool management can also be used for machines that have no spindle (e.g. punch presses or turrets). In this case the term “spindle” is replaced by “toolholder” according to the setting in MD 20124 TOOL_MANAGEMENT_TOOLHOLDER. If the MD setting is > 0, the spindle numbers $TC_MPP5 are interpreted as toolholder numbers.

Fixed location coding

If fixed location coding is selected for a tool, the tool will always be returned to the same location when it is replaced.

Variable location coding

Tools defined with variable location coding can be returned to any location for the appropriate tool size and location type in the magazine.

Automatic tool return to real magazine

1. An automatic tool return is initiated by the TM only if the tool is transported via several stations (status 105) after a T preparation command from the PLC and the T preparation command is finally acknowledged positively with status 1. The return of a preselected tool from the buffer can be suppressed by setting MD 20310: TOOL_MANAGEMENT_MASK, bit15 = 1.
2. If a tool change is interrupted because the control is switched off but the tool is already located in a buffer location (gripper), the next tool change must either return the tool in the buffer to the spindle or to the real magazine.
3. If several tools are located in the buffer the spindle tool is considered first. If there is no tool on the spindle, the order for return is in accordance with system parameter $TC_MLSR.

3.2.3 Select a tool and the cutting edge

Note

The T number and the M function are no longer transferred as an auxiliary function to the PLC when tool management is active.
Numbers are also permissible as tool names, e.g. “3” instead of T=“3” can be programmed simpler as T3.
The T number can be used for selection only if a tool with the T number as its identifier exists.

Example:
If you want to call a tool using T3, the tool must have the name “3”. A tool CANNOT be called via the internal T number as this is used exclusively by the NCK.

Select/deselect tool offset on Reset

The following machine data can be used to control the behavior on Reset:

- MD 20310: TOOL_MANAGEMENT_MASK bit14
- MD 20122: TOOL_RESET_NAME
- **MD 20110**: RESET_MODE_MASK
- MD 20130: CUTTING_EDGE_RESET_VALUE
- MD 20132: SUMCORR_RESET_VALUE

You can determine whether:

- deselect the active tool
- keep the active tool selected
- or select a specific tool (corresponding to MD 20122 TOOL_RESET_NAME).

If a new tool is selected and its data not yet available at the master spindle or master toolholder (or main spindle or main toolholder), a tool change is carried out on RESET or when the program ends. With this type of tool change (in a similar manner to block searches), the PLC is not capable of influencing the selection of the tool.

Select a tool at start of program

Machine data

- MD 20310: TOOL_MANAGEMENT_MASK bit14
- MD 20122: TOOL_RESET_NAME
- **MD 20112:** START_MODE_MASK
- MD 20130: CUTTING_EDGE_RESET_VALUE

can be set to define whether

- the tool on the main spindle or the main toolholder is selected again or remains selected
- a particular tool is selected (according to MD 20122: TOOL_RESET_NAME)

If a new tool is selected which in the data is not yet specified as being on the spindle, a tool change is performed when the program is started. In this type of tool change the PLC cannot influence the selection of the tool, exactly as for block search.

Tool rejection by the PLC

On a block search, selection on reset or program start, the tool is selected during preprocessing. In this case the PLC is not allowed to reject the tool.

Note

If bit 4 of machine data MD 20310: TOOL_MANAGEMENT_MASK is set, the PLC usually has the option of requesting another tool change preparation with new parameters, i.e. to reject the tool.

Communication between PLC and tool management

Communication during tool change between the PLC and the NCK is performed via the VDI interface. Tool change is triggered by the tool management in the NCK. The TM outputs commands to the PLC which acknowledges them either positively or negatively depending on the situation (see also Section 2.3).

Select a tool offset

Once the tool has been changed the following options are available for selecting the tool offset:

1. The offset number is programmed in the same block as the tool change command.
2. It is defined by the setting in MD 20270: CUTTING_EDGE_DEFAULT

   = 0 The offset is deselected (= D0).

   > 0 Number of offset selected after M06.

   = –1 The offset number of the old tool remains valid and is selected for the new tool after M06.

   = –2 The last selected offset remains valid until a D number is programmed.

Note

You will find detailed information about cutting edge and offset numbers in /FB1/ W1 – Tool offset.

3.2.4 Predecoding (preprocessing) and block execution (main run)

Sequence

The cutting edge geometry cannot be calculated until the tool management knows the tool that is actually to be used. Only the identifier is stated in the part program for tool change. Generally, the tool with the status “active” is then used. But if this is disabled, then one of the other twin tools, i.e. the replacement tool is selected instead. The precoding delays selection of the new offsets until it is clear which tool is to be used. Only then can precalculation of the blocks be restarted.

Tool change must have been completed before the path can be traversed with the tool offset of the new tool.

The block is split if the preprocessing run detects that a new edge of a new tool has been selected for the first time and tool preparation has been initiated, but not yet completed.

The following synchronization points exist between predecoding and block execution:

Example:

Programmed NC block:

```
N1D1 M06 Txx X100 Y100
```

Sequential blocks:

```
N1 Txx M06 end of block
N2 D1 X100 Y100
```

1.	Interpreter detects an offset selection (D number)
2.	It detects that a tool change has already been programmed although a tool has not yet been selected.
3.	Interpreter performs “block splitting”.
4.	Output of block N1: Block 1 contains a request from the run blocks to output their collected blocks and any M06, T numbers, ... that have been programmed.
5.	Output of block N2: Block 2 contains the rest, most importantly, any traversing information and D no., if one has been programmed.
6.	Tool management stops predecoding of a block until it is clear which tool is to be used.
7.	After receipt of the tool preparation acknowledgement, block 2 is processed, and the new T number is entered initially in the block so that a new contour calculation can take place.

Tool change in master spindle or master toolholder

Main run waits in synchronism with tool change block for transport acknowledgement

1. Main run waits in synchronism with tool change block for end of acknowledgement (if bit 5 or bit 6 of MD 20310: TOOL_MANAGEMENT_MASK is set) or
2. After a tool change in the main run, the NCK automatically performs synchronization with the end of the tool change in the first block in which an edge of the new tool is selected.

Note

The transport acknowledgement is an internal acknowledgement to an NCK command. It indicates to the NCK that the output command was accepted. When a new command is output to the PLC, the NCK waits for the acknowledgement of the previous command.

Tool already in spindle

If the programmed tool is already in the spindle, per default no command is sent to the PLC (the behavior can be influenced via the MD setting).

Tool change in a secondary spindle or secondary toolholder

1. Main run does not wait. There is no synchronization.
2. Main run waits in synchronism with tool change block for transport acknowledgement
3. Main run waits in synchronism with tool change block for end of acknowledgement.

Prepare to change tool in a main spindle

1. Tool management decides during the main run which tool is to be used (the active tool or a replacement tool). Until then, the preprocessor waits at the point in the program at which the compensation values of the new tool are to be considered for the first time.
2. The PLC can also decide which tool is to be used. In this case, the PLC can reject the proposed tool with a negative acknowledgement. If rejected by the PLC, the NCK selects a new, different tool (only if MD 20300: MC_TOOL_MANAGEMENT_MASK Bit5 = 1, see also FC 8 description, Section 4.9).
3. Even if the function “GETSELT(...,x)” is programmed, the preprocessor again has to wait until a decision has been made as to which tool is to be used.

Prepare to change tool in a secondary spindle

1. Predecoding does not wait. There is no synchronization.

Note

During a synchronization operation where the new offset is used or allowed for by the preprocessor, a “block split” must be performed. This ensures that a preprogrammed tool change T or M06 is actually performed and not collected in run blocks.

Unlike the STOPRE command, the preprocessor does not necessarily wait until all blocks have been processed, but only waits if tool selection has not taken place by the relevant time. The relevant time when programming offsets on the one hand and when programming GETSELT on the other.

3.2.5 Traverse axes while tool is being changed

After the tool change command M06 the axes can continue travel without having to wait for the tool change acknowledgement and, e.g., execute traversing blocks without tool offset. Only in the block with an offset selection (D no.) does processing stop until the PLC has reported completion of the tool change.

Requirements: MD 20270: CUTTING_EDGE_DEFAULT= 0 or –2

Example: Traversing blocks between tool change and cutting edge selection

```
N10  T=“Drill18”          ;Prepare tool change
N15  M06                  ;Change tool
N20  D0                   ;Select offset
N25  G00 X100 Z200        ;Traverse machine axes
N30  Y150 M79             ;Traverse machine axes
N35  G01 D1 X10           ;Activate tool offset.
                          ;Check whether tool has changed. Preprocessing stop
                          ;until tool change preparations are completed.
                          ;Main run waits until tool change is acknowledged
                          ;by PLC.
```

Preprocess stop is maintained until tool change preparation is completed. The main run waits at N35 (D1) until the tool change has been executed and acknowledged.

3.2.6 Tool change in the spindle for chain and box magazines

Spindle/buffer DB 72

Data block **DB 72** loads tools into the spindle. This data block also prepares the tool change. This data block has an interface for every spindle.

User data are located in every interface (order corresponds to the spindle number) as is the case for the loading/unloading points. The data block also contains additional data for the new tool. These data include location type, sizes, tool status and the T number assigned internally in the NC.

The buffer address of the spindle is contained in DB 72. DBW(n+16) and DBW(n+18) as the destination for the new tool. This position is communicated as the target position of the new tool in parameters “NewToolMag” and “NewToolLoc” when the tool change has been successfully completed. The target position for the old tool (DB72. DBW(n+24) and DBW(n+26)) is transferred to FC 8 in parameters “OldToolMag”, “OldToolLoc” together with parameter “Status = 1” after the change tool command has been executed.

Description of tool exchange in spindle

The tool in location 1, magazine 1 is to be loaded to the spindle (magazine no. 9998, location 1) and the tool in the spindle is to be returned to magazine 1 location 8.

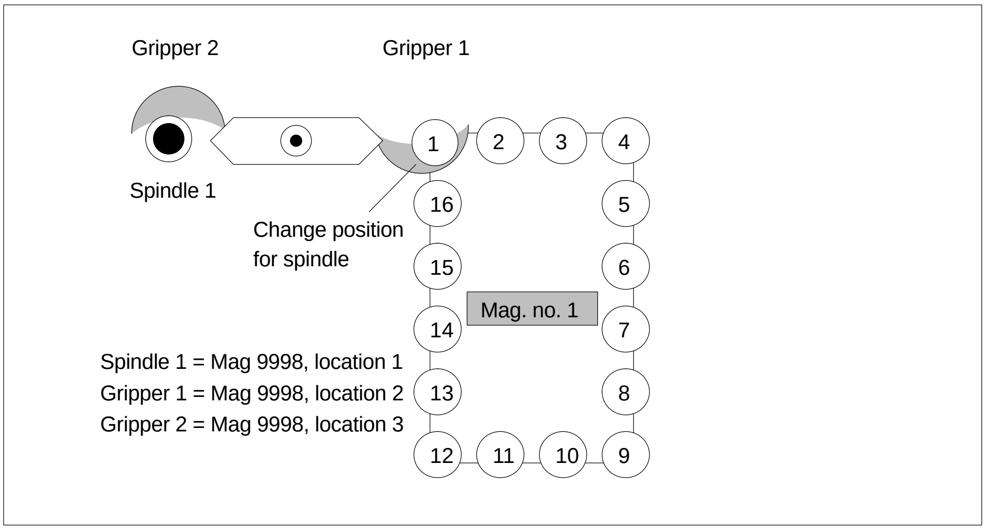

Gripper 2
Gripper 1
Spindle 1
Change position for spindle
Mag. no. 1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Spindle 1 = Mag 9998, location 1
Gripper 1 = Mag 9998, location 2
Gripper 2 = Mag 9998, location 3

Fig. 3-6 Load tool into spindle

The tool change in the spindle is split into two steps (for TOOL_CHANGE_MODE=1):

1. Prepare change: Search for new tool and move to the change position
2. Perform change: New tool into the spindle and old tool into magazine in correct location

1. Prepare change

Bit 2 is set in DB72.DBB n+0. During preparation for tool change, the current positions of the tools are passed to FC 8 in the relevant parameters. as soon as preparation is complete. “Status” = 1 is parameterized in FC 8 at the same time. This means that the “old tool” is still in the spindle and that the “new tool” is either still in the source magazine at the same location or has been placed in a buffer.

The following information is passed to FC 8:

- The new tool is in the change position, but is still located in magazine (NewToolMag = 1 and NewToolLoc = 1).
- The old tool is still in the spindle (OldToolMag = 9998 and OldToolLoc = 1).

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	2	DB 72 interface
TaskIdentNo	1	No. of active interface
NewToolMag	1	(n+20) mag no. of new tool
NewToolLoc	1	(n+22) location no. of new tool
OldToolMag	9998	(n+16) mag no. of old tool
OldToolLoc	1	(n+18) location no. of old tool
Status	1	Operation complete
Ready		Checkback from FC 8
Error		Checkback from FC 8

2. Change tool

If the preparation command has been correctly acknowledged with status = 1, the “Change” bit DB72.DBB n+0 bit 1 is set with the M06 command in the part program. The user parameters are also transferred again. All other values remain unaffected by the “Change tool” operation.

Two tools are involved in the tool change procedure. The old tool is in the spindle and the new tool is in the magazine. In this example, the tool is transported by grippers 1 and 2. Any change in the position of the tools must be communicated to the tool management with FC 8. FC 8 must be called twice for this purpose.

FC 8 call with status 105 “Change in progress”

The tool is removed from the magazine and the spindle with the dual gripper. The old tool is now in gripper 2 at location no. 3 and the new tool in gripper 1 in location no. 2. The resulting FC 8 call is as follows:

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	2	DB 72 interface
TaskIdentNo	1	No. of active interface
NewToolMag	9998	(n+16) mag no. of spindle
NewToolLoc	2	(n+18) location no. of new tool New tool now in gripper 1
OldToolMag	9998	Mag no. of old tool
OldToolLoc	3	Location no. of old tool Old tool now in gripper 2
Status	105	Change in progress
Ready		Checkback from FC 8
Error		Checkback from FC 8

Note

The operator uses FC 8 to notify the tool management of the new positions of the exchanged tools.
The tool management knows which is the new (called) tool and which is the old (spindle) tool.
The current positions are also known to the tool management. If these positions change, the tool management is only informed about this through FC 8.

Note

If T preparation and change signals are present at the same time, the tool call and change command (T and M) are programmed in one block.
When FC 8 is called in such a case, only the change and not the selection need be acknowledged.

FC 8 call with status 1 “Tool change complete”

While the gripper is moving the tools, the PLC can read the magazine location for the old tool (from the spindle) from DB72.DBW (n+24) and (n+26) and move the magazine to the change position. This position is location 8 in magazine 1 in this example. The tool change operation can now be ended mechanically through “insertion” of the tools. This change in the tool positions must also be communicated to the tool management with an FC 8 call with “Status = 1”. The new tool is placed in the spindle of magazine no. 9998, location no. 1 and the old tool in magazine no. 1 at location 8.

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	2	DB 72 interface
TaskIdentNo	1	No. of active interface
NewToolMag	9998	(n+16) mag no. of spindle
NewToolLoc	1	(n+22) location no. of spindle
OldToolMag	1	(n+24) mag no. of old tool
OldToolLoc	8	(n+26) location no. of old tool
Status	1	Operation complete
Ready		Checkback from FC 8
Error		Checkback from FC 8

If the dual gripper is to place the spindle tool in the magazine location of the new tool, the user must ensure that the magazine location is of the same size and location type as the spindle tool.
As of SW 5, a 1:1 replacement is supported here through appropriate setting of the search strategy by the tool management.

If this is the case, the transfer can be performed simultaneously (on the dual gripper in the spindle and in the magazine location at the change position).

FC 8 must be parameterized as follows.

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	2	DB 72 interface
TaskIdentNo	1	No. of active interface
NewToolMag	9998	(n+16) mag no. of new tool
NewToolLoc	1	(n+18) location no. of new tool
OldToolMag	9998	(n+20) mag no. of old tool
OldToolLoc	3	(n+26) location no. of old tool
Status	1	Operation complete
Ready		Checkback from FC 8
Error		Checkback from FC 8

3.2.7 Exceptional cases “T0”, empty spindle, multiple T selections

T0: Empty spindle

DB72.DBX(n+0).3 indicates that TO has been programmed. If T0 has been programmed to empty the spindle, DBW (n+20), DBW(n+22) – data for new tool – in DB72 are assigned the value “0”.

Parameters NewToolMag and NewToolLoc of FC 8 must then be set to “0”.

This applies to the preparation and to the change procedure.

Spindle is empty

The tool must be changed. This status is indicated by the fact that parameters OldToolMag and OldToolLoc are set to “0”.

In this case, FC 8 parameters OldToolMag and NewToolLoc must be set to “0” for tool preparation and change.

Multiple T selections

In the case of multiple T selections, it may not be possible to abort the program with a RESET.

The interruption response can be enhanced as follows:

- Cancel the read-in enable to prevent following blocks from being accepted in the main run.
- Then acknowledge with status 3 via FC 8 (the tool command is denied by the PLC).
- When the acknowledgement has been issued, the RESET can be activated for the channel.

3.2.8 Tool change with turret

Turret DB 73

DB 73 is the block used to “change” tools in the turret (i.e. by rotating the turret so that the required tool is in working position). This data block has an interface for every turret. The turrets are numbered using ascending magazine numbers. User data are located in every interface as is the case for the unloading/loading points. The data block also contains additional data for the new tool. These data comprise location type, sizes, tool status and the T number assigned by the NC.

When the tool change has been successfully completed the arrival of the new tool in its target is acknowledged via FC 7. For this purpose, the turret number of the loaded tool is passed to parameter “ChgdRevNo”.

FC 8 parameters	Values	Comment
Start		Starts task
ChgdRevNo	1	1st turret
Ready		Checkback from FC 8
Error		Checkback from FC 8

3.2.9 Number of replacement tools (SW 5.1 and later)

Machine data **MD17500: MN_MAXNUM_REPLACEMENT_TOOLS** can be set to select the maximum number of replacement tools.

Once the set threshold for the number of replacement tools has been reached it is no longer possible to:

- create a tool with ID (alarm) or
- assign a tool by renaming an already fully assigned group (alarm).

If the machine data is set to value 0, the number of replacement tools is not monitored (same behavior as software versions up to SW 4).

Alarms

For operation via the MMC, alarm 17192 is output as an indication as soon as the defined limit is violated.

If programming via a part program an additional interpreter alarm is triggered (e.g. 14020 if NEWT fails).

Note

Machine data MD 17500: MAXNUM_REPLACEMENT_TOOLS is limited (up to 600) only by the upper limit value set in machine data MD 18082: MM_NUM_TOOL.

3.2.10 Tool changing errors

If an error is detected by the NCK in the programmed tool preparation (e.g. no tool available, no free position in magazine) program processing is terminated with an alarm.

SW 5.1 and later

SW 5.1 and later allows the operator to assess and rectify various problems without terminating the program.

The following problems can be solved:

- The tool data record is not or not entirely in the NCK.
- The part program contains a programming error.
- No more replacement tools of the tool group in question are available (only applies when tool management is active).
- Alarm 22067 or 22069 is stored. The tool data record has been loaded into the NCK but is not assigned to a magazine location or the magazine of the tool is not accessible to the tool search (only applies when tool management is active). The tool must be reloaded “manually” (e.g. directly onto the spindle).

Note

“Illegal D number” can occur either if the part program contains an error or the data record for the D number in question is not loaded in the NCK.

Programming example

```
N10 ...
N100 T=“Drill”       ;NCK detects an error
N110 ...
N200  M06            ;if the tool change is programmed explicitly for
                     ;the tool change in the same program
N210 ...
```

Note

In many cases, **M06** is not programmed on the same level as the tool preparation, but in a subroutine, cycle or macro.

Bit 0 of machine data MD 22562: TOOL_CHANGE_ERROR_MODE determines the block at which the program must stop.

TOOL_CHANGE_ERROR_MODE, bit 0=0:

```
N10 ...
N100 T=“Drill”;                NCK detects an error,
                               program stops at this block
N110 ...
N200 M06
N210 ...
```

TOOL_CHANGE_ERROR_MODE, bit0=1:

```
N10 ...
N100 T=“Drill”;              NCK detects an error
N110...
N200  M06                    program stops at this block
N210 ...
```

The error is detected during tool preparation but is ignored by the NCK. The program continues and stops at M06. During normal program execution, tool preparation has been completed by this point. If an error has occurred, tool preparation can be repeated with the correct data.

The programming error (in block 100 in this example) is corrected by adding the offset to the tool change block:

```
N200 “T=Drill_1” M06
```

If the tool change (programmed with M06) is performed in a subroutine or cycle program, the error can be remedied by inserting an overwrote block (in the example).

3.2.11 Manual tools (retrofitting tools during machining)

Bit 1 in MD 22562: TOOL_CHANGE_ERROR_MODE can be set to select additional tools without magazine assignment during tool changes. The automatically selected tool must be inserted in the machine manually and removed again manually after machining.

Responsibility of the user

The user must make sure that

- the data record of the tool positioned on the spindle is actually in the NCK and
- that he places the tool that corresponds to the data record in the NCK on the spindle.

Tools which are loaded manually during machining are referred to as **“Manual tools”**.

Note

It is the responsibility of the user to ensure that the safety regulations are met via the PLC program.

Sequence

Internally, the NCK initiates an automatic sequence until the user can perform the tool change with a manual tool. The NCK searches for the selected tool and detects that a suitable tool is not available in the magazine. Once the NCK has established that a suitable tool is not available in the magazine, it scans the tools which are not assigned to any magazine. The tool with the active status is selected from these tools. If no active tool exists, the tool with the lowest duplo number is selected.

If no tool is found, a manual tool can be loaded. Manual tools are identified in the interface to the PLC (VDI) by **magazine location no. 1** in **magazine 9999**. The PLC can detect from this identifier that a manual tool is to be loaded. The PLC ensures that the machine is in a safe state in order to allow the user to perform the manual tool change.

Note

If the manual tool is loaded, alarm 17212 “Channel %1, Manual tool %2, duplo no. %3, Load to toolholder %4” is output. The alarm is confirmed by the tool change acknowledgement from the PLC.

Note

The PLC is not allowed to reject a manual tool preselected by the NCK (for tool rejection, see also MD 20310: TOOL_MANAGEMENT_MASK).

Block search, program testing

As regards block searches, there is no difference to a normal tool change. However, the corresponding alarms are not generated.

No change commands are output to the PLC during the block search. If a manual tool needs to be loaded when the NC is first started, this can be programmed by **magazine location 1** in **magazine 9999** and output of the corresponding alarm.

The tool and magazine data must not be modified in **program test mode** in the NCK. The data of a manual tool which is loaded when program testing is selected are therefore removed from the toolholder and stored internally. The stored manual tool is loaded back into the toolholder in response to PLC task “Return manual tool from magazine 9999, location 1”.

Note

Due to the technology of the internal memory, several toolholders with manual tools can exist during program testing mode.

Conditions

Only correction block problems that have occurred as the result of programming errors or incorrectly defined data in the NCK can be solved in connection with tool selection, tool change or offset selection.

Problems or errors that result from incorrect communication between the NCK tool management and the PLC cannot be remedied in this way. Errors of this type, however, only occur when a new PLC program is being installed on the machine and not during production.

3.2.12 Tool changes in NCK via synchronized actions (SW 5.1 and later)

Synchronized actions are statements programmed by the user, which are interpreted by the NCK in the interpolation cycle synchronous to machining of the part program. If the condition in the synchronous action is satisfied or no condition is defined, the assigned actions are activated synchronously with subsequent machining.

Reference: /FBSY/ Function Description Synchronous Actions

A tool change with active tool management is performed as follows:

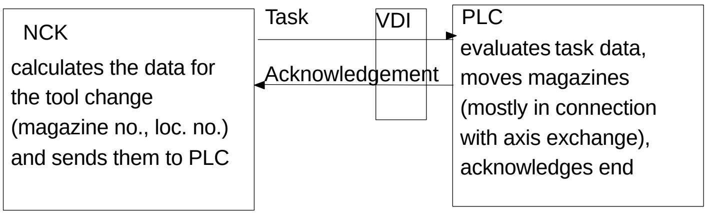

NCK
calculates the data for the tool change (magazine no., loc. no.) and sends them to PLC
Task
Acknowledgement
VDI
PLC
evaluates task data, moves magazines (mostly in connection with axis exchange), acknowledges end

Fig. 3-7 Principle of tool change – magazine positioning is among the functions performed by the PLC

This procedure is relatively time-consuming when applied to turrets. The necessary magazine axis and axis motion data are located in the NCK. It is therefore not necessary for the PLC to fetch a replacement axis from the NCK and move it (very time-consuming). Because of the asynchronism of the NCK and PLC signal traffic, the NCK can execute the axis movement without the need for additional communication with the PLC.

In SW 5.1 and later, the user himself can define which PLC data he makes available to and receives from the NCK (function 'Availability of PLC variables'). This function makes it possible to transfer tool change tasks as synchronized actions to the NCK.

In this solution, the synchronized action uses data from the NCK itself, which are first output to the PLC and then returned to the NCK. It has no direct control over the **NCK-PLC interface of the TM function**.

To optimize tool change times on circular magazines still further, the "Prepare tool" and "Change tool" tasks can be detected and setpoints read in synchronized actions.

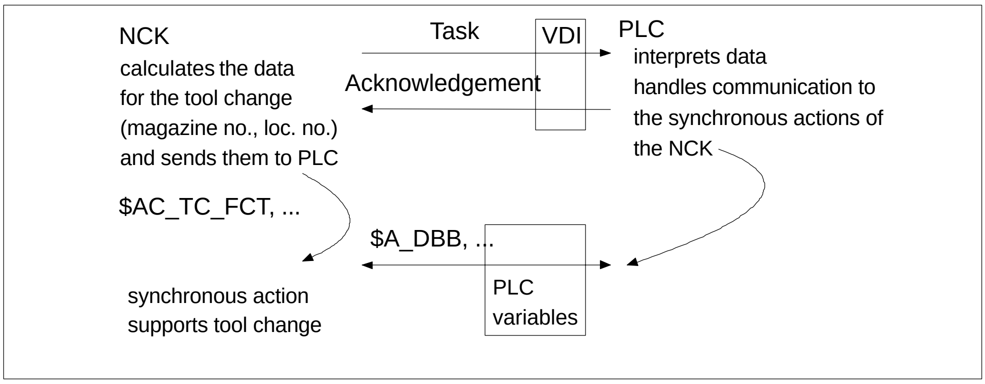

NCK
calculates the data for the tool change (magazine no., loc. no.) and sends them to PLC
$AC_TC_FCT, ...
synchronous action supports tool change
Task
Acknowledgement
VDI
PLC
interprets data handles communication to the synchronous actions of the NCK
$A_DBB, ...
PLC variables

Fig. 3-8 Principle of tool change with synchronized actions (NCK opens TM VDI interface via system parameters for synchronized actions)

The synchronized action acquires the necessary information for positioning the magazine by reading the interface data.

Description of function

TM NCK-PLC interface via the VDI

The NCK-PLC interface is defined in PLC documentation /FB/ Description of Functions, Basic Machine, P3 Basic PLC Program and Section 2.3.

The following data can be accessed via the NC language. The assignment between the system variables and interface data is also specified:

- Command/function number
  for motion, preparing change,
  change ON $AC_TC_FCT
- Acknowledgement status of PLC FC 8 $AC_TC_STATUS
- Programmed spindle number
  or toolholder number $AC_TC_THNO
  - Data for **new** tool:
    Internal NCK T number $AC_TC_TNO

- MagNo._From / LocNo._From $AC_TC_MFN / $A_TC_LFN
  MagNo._To / LocNo._To $AC_TC_MTN / $A_TC_LTN
- Data for **old** tool:
  MagNo._From / LocNo._From $AC_TC_MFO / $A_TC_LFO
  MagNo._To / LocNo._To $AC_TC_MTO / $A_TC_LTO

Existing communication mechanisms in the NCK are not affected. The NCK continues to transfer the data to the interface if a command for tool management is detected in the NCK. It continues to wait for the end or error acknowledgement for the command. The system parameters can only be defined and programmed when the tool management function is active. See also Section 3.8.5.

Note

The PLC user programmer or person programming the synchronized actions decides whether the **PLC** or a **synchronized action** receives the data.

Search for empty location

The system parameters **$AC_TC_MTO** and **$AC_TC_LTO** supply the values for the empty location for the old tool (of the spindle / tool to be removed from the toolholder).

Secondary conditions

It is only advisable to evaluate the parameters if the NCK has an open (not acknowledged with ’end’ or ’error’) command pending at the interface. The same rules must be followed as apply to the PLC user programs (see PLC description).

Tool preparation and tool change commands are registered particularly for block searches and are stored at the end of a block search ready for output.

However, it is not possible to output these commands directly to the PLC. The PLC must first be switched to a state in which tool preparation or tool changes are permitted.

This can be done by starting an ASUB routine which is executed at the end of the block search, but before the tool command is output to the PLC. The tool has thus been prepared by the time the tool command is output.

It does not make sense to start the synchronized action until the NCK outputs the tool command to the PLC, i.e. a synchronized action based on the read-in values of the system parameters above is not affected by a block search.

Note

Synchronized actions affect the interpolation cycle of the NCK. The interpolation cycle time may be insufficient for complex programs. You will then have to adapt the cycle time. If the process does not allow you to do this, you must program the synchronized action more efficiently.

Example 1

Illustration of a tool change with a circular magazine

The following table shows how tool change operations can be split between the NCK and PLC. As many tasks as possible should be assigned to the NCK. This applies, for example, to the movement of the turret along an NCK indexing axis. It saves time that the PLC needs for the axis transfer to request an axis from the NCK and then enable it again.

Default setting: Servo enable = 0 for turret:

Step	NCK	Synchronized action	PLC
1	Outputs command to the PLC on **part program command T...**		
2		**Reads** command and values of tool change command (as signaled to the PLC by the NCK)	**Accepts command** and distributes it to application interface (PLC basic program)
3		Calculates the position of the turret from these data and **initiates positioning** (via a travel command)	PLC **aborts clamping** of turret
4			PLC sets **controller enable** of the revolver to the NCK on release of clamping
5	Turret axis **positioned** (travel command)		
6	Turret axis **positioned** (travel command aborted)	Registers that target position is reached, then outputs e.g. **auxiliary function H** to the PLC	

Step	NCK	Synchronized action	PLC
7			Detection of help function H. **Initiates clamping** of the revolver
8			If clamping successful, then **Acknowledgement** of change command to NCK
9	Part program processing is continued		

Note

Axis positioning can be triggered internally via the NCK without further communication with the PLC because the axis movement is not performed until the PLC sets servo enable.

The PLC derives from auxiliary function H that the NCK has completed turret positioning (PLC is master).

Example 2

Principle of a programmed synchronized action

Requirement:

- Circular magazine number 1 and location numbers 1 to 12
- An indexing axis “REV” has been defined for the turret.
- Tool preparation and tool change by programming T (TOOL_CHANGE_MODE = 0)
- The programmed T value is the magazine location number
  (set via MD 20310: TOOL_MANAGEMENT_MASK; bit16; corresponds to T = location number).
- Magazine zero is at change position ($TC_MDP2[1,1]=0); i.e. the current magazine position is the magazine location number at the change position.
- Location number 1 in the buffer magazine is a ’toolholder location’ ($TC_MPP1[9998,1]=2)
- Master toolholder: $TC_MPP5[9998,1]=1 and MD 20124: TOOL_MANAGEMENT_TOOLHOLDER = 1
- No tool is yet active.

Magazine positioning to be performed by synchronized action.

Excerpt from part program:

```
...
N50 T6         ;Machining with tool in location 6; the suitable tool
               ;with T=111 is stored in this location
...
```

From this the NCK calculates the data for the command to the PLC. The command is output to the PLC.

The **static synchronized action** triggers on command number 4 and initiates axis positioning exactly once. The command to the PLC can be present for several IPO cycles. The PLC decides when the end acknowledgement for the command is output.

```
               ;very simple action: “detect command and position axis”
N111 ids=1     every (($ac_tc_fct==4) and ($ac_tc_lfn>0))
               ;do pos[x]=cac($ac_tc_lfn)
               ;fa[x]=10000.
...
N999 m30
```

To ensure that the synchronized action remains active beyond RESET it is defined as a static synchronized action. It can be activated after power-up of the NCK (usually by the PLC when the relevant program is started).

Note

The system parameters can only be defined and programmed with the active tool management function. If a parameter is read when no command is pending, value –1 is output.

3.2.13 Tool change cycle (workshop interface (ShopMill))

The tool change is initiated by a cycle for the workshop interface. This cycle is called by the ShopMill cycles in which a tool can be programmed.

Sequence

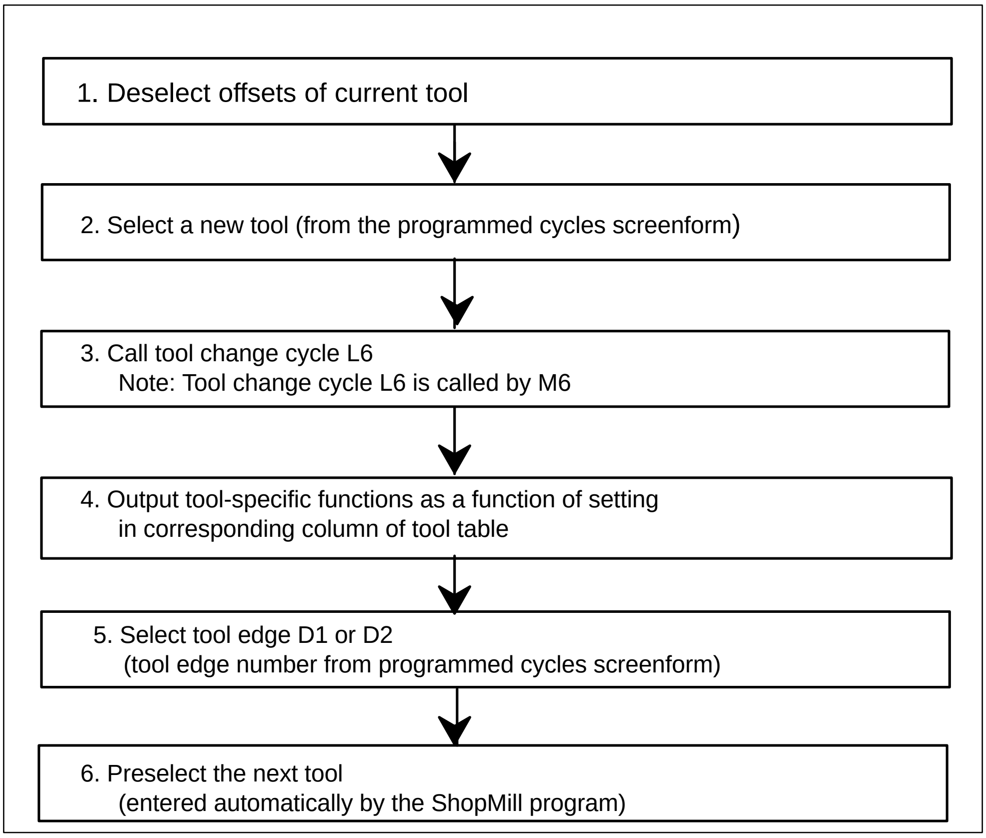

1. Deselect offsets of current tool
2. Select a new tool (from the programmed cycles screenform)
3. Call tool change cycle L6
Note: Tool change cycle L6 is called by M6
4. Output tool-specific functions as a function of setting in corresponding column of tool table
5. Select tool edge D1 or D2 (tool edge number from programmed cycles screenform)
6. Preselect the next tool (entered automatically by the ShopMill program)

Fig. 3-9 Workshop interface – tool change cycle

Example

Tool change cycles can be generated using sample cycle L6.BSP in toolbox\cycles\bsp in the Toolbox.

The following conditions must be fulfilled first:

- Redefinition of DEFINE M6 AS L6:
  This is stored in file SMAC.DEF in toolbox\cycles\define in the Toolbox.
- Definitions of M206 as an M function for tool changes:
  This is stored in toolbox\cycles\md machine data set CMM:8X0

```
PROC L6
...
;————————————————————————————————
Example of tool change cycle for machine manufacturer
;————————————————————————————————
DEF INT_WZ_IN_SP_WZ_VOR
;
STOPRE                                     ;Preprocessing stop
;
IF (NOT $P_SEARCH)                         ;If no block search
   _WZ_IN_SP=$TC_MPP6[9998,1]              ;Tool in the spindle
   GETSELT(_WZ_VOR)                        Preselected tool
   ;
   IF (_WZ_IN_SP<>_WZ_VOR)                 ;If another tool
      ;Position spindle:
      SPOS=...
      ;Approach tool change position
      SUPA D0 G0 G90 G40 G60 Z=...
   ENDIF
ENDIF

;
Load tool: Tool management and PLC
M206
STOPRE          ;Preprocessing stop
M17
```

References FBSP/ Description of Functions, ShopMill

3.2.14 Block search

Block search with calculation

On a block search, selection on reset or start, the tool is selected during predecoding. In this case the PLC is not allowed to reject the tool (see bit 4 in MD 20310) If it does, an alarm is generated. The block search must then be repeated. Use of the active tool can only be prevented from an external source (MMC, PLC).

In block search with calculation the program is generally put into a state where the selected block can be executed. With respect to the tool management function, this means that the tool that should be located in the spindle when the machining block is reached must now be loaded to it.

If another tool is located in the spindle a “’replace’’ command is initiated. Since the auxiliary functions are output together, signals “Prepare change” (DB72.DBX(n+0).2 and “Change tool” (DB.DBX(n+0).1 are applied jointly in such cases.

Example: $MC_TOOL_CHANGE_MODE=0

Tool “Drill1” is loaded in the spindle. The current tool programming of the new search target is T=“Drill2”.

NCK initiates the tool change. PLC must not intervene.

Note

Tool rejection by PLC: If bit 4 of machine data MD 20310: TOOL_MANAGEMENT_MASK is set, the PLC usually has the option of requesting another tool change preparation with new parameters, i.e. to reject the tool. This is not possible during block search. In this case, the machine data setting is ignored.

Note

Because the tool change is frequently performed using cycles, a “replace command” generated by the block search must be executed in an asynchronous subroutine (ASUB). Modal and static motion-synchronous actions remain valid at the beginning of the ASUB and are also effective in the asynchronous subroutine. If the asynchronous subroutine is not continued with Repos, the modified modal and static motion-synchronous actions in the main run remain operative.

Alternatively, a feed and read-in hold can be used to stop the NC part program running and generate the error message “incorrect tool in spindle after block search”.

Tool cannot be used

If the tool to be loaded is not located at the search destination, an attempt is made to enable a disabled tool. If no suitable tool is found, alarm 22068 is output. The alarm can only be cleared by a RESET.

If further tool changes are programmed, the disabled tool is not tagged for future block searches and the search operation is not interrupted. However, if an attempt is made to load the disabled tool on a start after the end of the block search, the NCK outputs alarm 22067. The program cannot be resumed. With SW 5.1 and later, the PLC can be used to control whether or not the disabled tool is loaded anyway.

3.2.15 Program testing

The “program testing” function can be used to traverse a program without axis motion.

All other data are determined and calculated. This means that, when a tool is called up, the tool management searches for the tools and transfers the corresponding values to the PLC interface.

The PLC must acknowledge these jobs without moving the magazine or changing a tool. Special measures are therefore required on the PLC.

The tool management acts in exactly the same way as it would when a program is running. In the case of tools without fixed location codes and acknowledgement, this can result in the PLC data indicating different tool locations to the actual mechanical locations in the magazine. This can be prevented by configuring FC 8 such that a fixed location is simulated for the duration of the program test rather than the calculated empty location applied as a parameter.

The old location of the tool is stored in the function block which handles PRT and returned again to this location in the software (data settings). Any existing spindle tool is also returned to the spindle in the software at the end of the program test or on a reset. This ensures that the magazine assignments in the software match the mechanical assignments after PRT.

For testing programs involving manual tools, see Subsection 3.2.11.

Example of how to adapt the PLC in test mode

The following example program can be used as a template for adapting the PLC to program testing mode. Only the first channel and a spindle are supported as tool change locations.

The tool is always changed directly on the spindle. The spindle is used as the tool change location (DB72). Access to the NCK/PLC interface (DB 21, 72) is symbolic. The standard UDTs (UDT 21, 72) are included for this purpose. These are stored on the basic program diskette and must be copied into the project and subsequently compiled.

The following must be entered in the symbol table:

Symbol	Address	Data type	Comment
Channel1	DB 21	UDT 21	
WstSp	DB 72	UDT 72	
WZW_VAR	DB 119	DB 119	To test tool change operation

All necessary variables are stored in the instance data block.

If program testing mode is **deselected**, no action occurs. The target positions suggested by tool management are confirmed by the PLC.

If program testing mode is **selected**, the target positions are defined by the PLC. These correspond to the source positions of the respective tools. The target position is only confirmed and saved by tool management on the first tool change. It is thus possible to undo the first tool change after program testing mode is selected.

Two asynchronous transfers are required for this purpose. The first one returns any tool present in the spindle to the magazine. The second asynchronous transfer is intended to return a tool which was loaded in the spindle before program testing mode back into the spindle.

Note

The PLC example is stored in the toolbox with SW 3.6 and later. The sample file WZV_PROG.AWL is packed in file WZV_BSP.EXE.

Program testing in SW 5 and later

Machine data $MC_TOOL_MANAGEMENT_MASK – bit 20 can be set such that the NCK does not output any tool change commands to the PLC in “Program test active” mode, but acknowledges them itself in such a way that no further tool motions are recorded in the data.

The disabling of tool change command outputs is selected intentionally as the default, even though this renders the software incompatible with earlier NCK versions.

As regards the tool used during program test mode, the following applies:
The tool status “active” can still be set and the “was in use” status is set. However, since the tool monitoring function is generally deactivated in test mode, this has no negative impact.

When **bit 20**, value **1** is set, generated commands are output to the PLC. Depending on how these are acknowledged by the PLC, this may cause tool/magazine data in the NCK to be altered. If the acknowledgement parameters for the “target magazine” are set to the values of the “source” magazine, then the tool is not moved and the data therefore not altered in the NCK.

Exception: The status of the tool activated in test mode can assume the “active” state.

Note

It must not be assumed (when the setting “no tool change commands to PLC” is selected) that the tool in the spindle or toolholder in “Program testing active” mode is actually the active tool.

3.2.16 Several spindles in one channel or TO unit

When using tool management and more than one spindle please note the following:

Two spindles in one channel

Only one tool offset can be active per channel. Spindle 1 defined as master spindle with $MC_SPIND_DEF_MASTER_SPIND = 1. Spindle 2 is a secondary spindle.

The master spindle is spindle no. 1 in each case.

Two channels each of which access the same TO memory have been set in the machine data. One spindle is assigned to each channel. Two spindles are assigned to one magazine in the machine configuration.

The master spindle is spindle no. 1 for both channels In order to change a tool in spindle no. 2 as well, the second spindle must be defined as master spindle in the second channel before the tool is changed. In the TM system, the spindle number is sent to the PLC. This number is determined from the extended address of T. If this is not programmed, the NCK assigns the master spindle number of the channel in which the program is running (Figure 3-9).

Each channel has its own master spindle

Two channels each of which access the same TO memory have been set in the machine data. One spindle is assigned to each channel.

Two spindles are assigned to one magazine in the machine configuration.

In each channel the assigned spindle is defined as the master spindle. Tool change is possible without making any additional definitions in the program.

3.2.17 Decoupling the tool management from the spindle number

The program must specify the location (spindle number on milling machines) at which the tool is to be changed before the tool management can insert a tool.

In SW 4 and earlier, this location was defined by the master spindle number (MD 20090: SPIND_DEF_MASTER_SPIND).

In SW 5.1 and later, machine data **MD 20124: TOOL_MANAGEMENT_TOOLHOLDER** can be set to determine whether a toolholder number must be assigned to define the location of the tool to be loaded instead of a spindle number. It is thus possible to use an identifier (spindle number or toolholder number) as appropriate for the application.

The following figures show which variable definitions you require for the following variants:

- Working with two spindles in two channels and one TO unit (standard function)
- Working with two spindles in one channel (standard function)
- Working with 2 toolholders in 2 channels (one TO unit)
- Working with two toolholders in one channel (standard function)

Working with spindle numbers

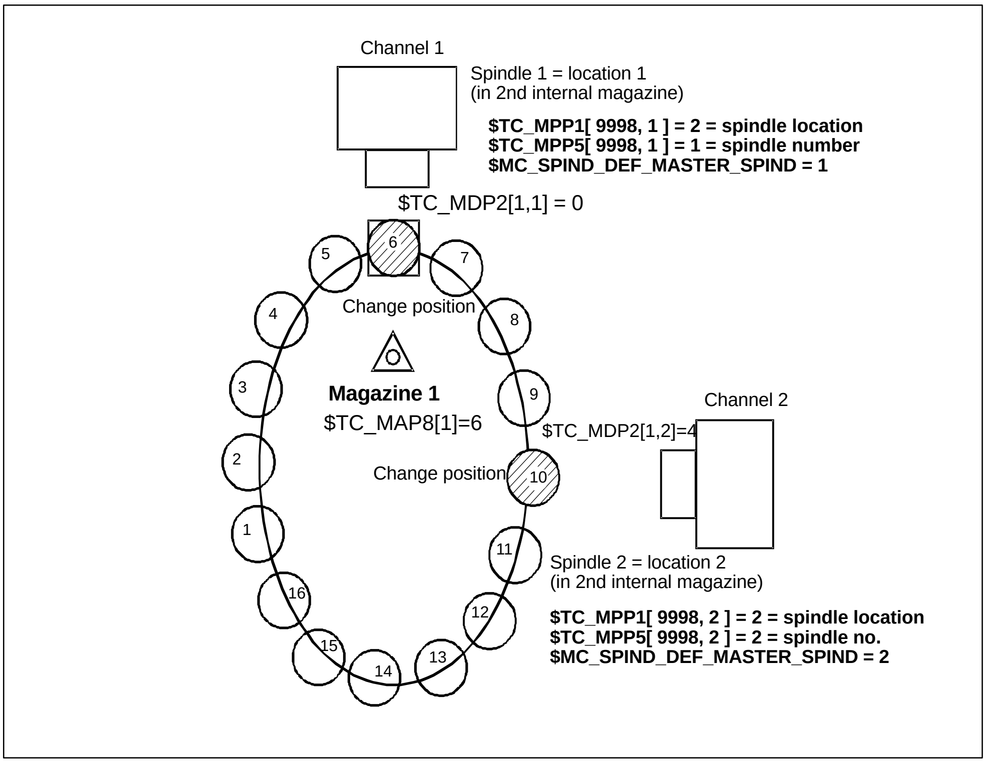

Channel 1
Spindle 1 = location 1
(in 2nd internal magazine)
$TC_MPP1[9998, 1] = 2 = spindle location
$TC_MPP5[9998, 1] = 1 = spindle number
$MC_SPIND_DEF_MASTER_SPIND = 1
$TC_MDP2[1,1] = 0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Change position
Magazine 1
$TC_MAP8[1]=6
Change position
Channel 2
$TC_MDP2[1,2]=4
Spindle 2 = location 2
(in 2nd internal magazine)
$TC_MPP1[9998, 2] = 2 = spindle location
$TC_MPP5[9998, 2] = 2 = spindle no.
$MC_SPIND_DEF_MASTER_SPIND = 2

Fig. 3-10 Two spindles in two channels and one TO unit)

Two channels are operating with the data of one TO unit (with one magazine). One spindle is defined in each channel.

Spindle 1 in channel 1 has been declared the master spindle with MD SPIND_DEF_MASTER_SPIND=1. Spindle 2 on channel 2 is the master spindle.

Both spindles must be assigned different numbers because the assignment of the spindle to the second internal magazine (buffer magazine) must be unique.

This assignment is made via $TC_MPP1 (spindle location) and via $TC_MPP5 (spindle number).

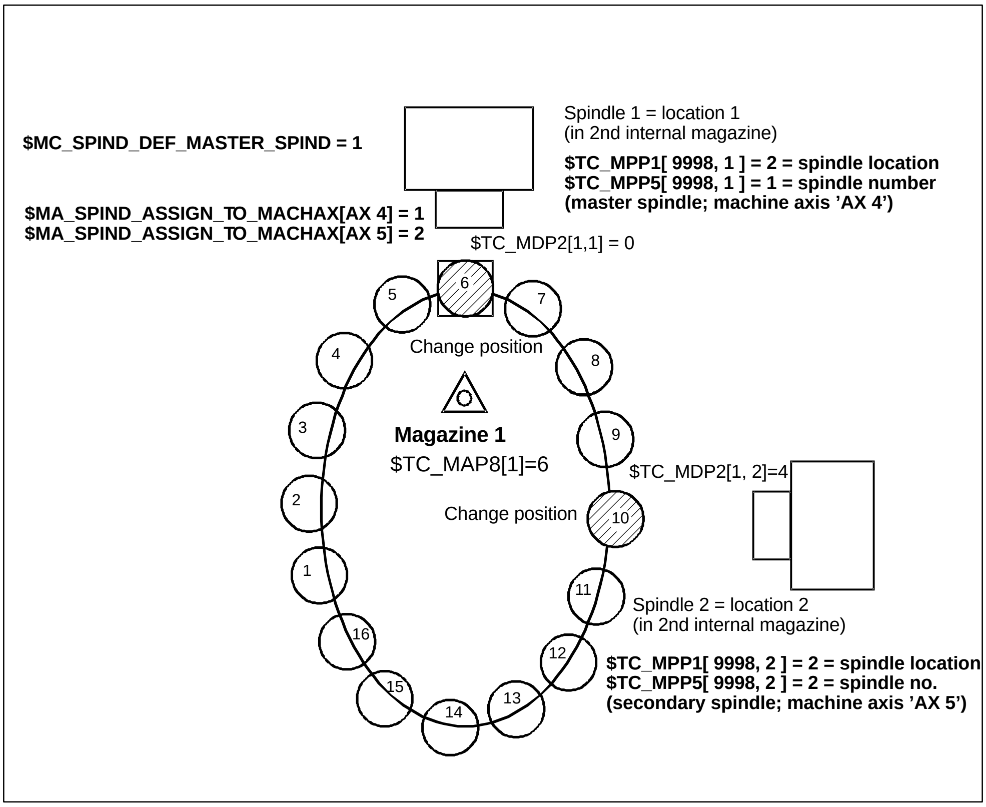

$MC_SPIND_DEF_MASTER_SPIND = 1
$MA_SPIND_ASSIGN_TO_MACHAX[AX 4] = 1
$MA_SPIND_ASSIGN_TO_MACHAX[AX 5] = 2
Spindle 1 = location 1
(in 2nd internal magazine)
$TC_MPP1[9998, 1] = 2 = spindle location
$TC_MPP5[9998, 1] = 1 = spindle number
(master spindle; machine axis 'AX 4')
$TC_MDP2[1,1] = 0
Change position
Magazine 1
$TC_MAP8[1]=6
Change position
$TC_MDP2[1, 2]=4
Spindle 2 = location 2
(in 2nd internal magazine)
$TC_MPP1[9998, 2] = 2 = spindle location
$TC_MPP5[9998, 2] = 2 = spindle no.
(secondary spindle; machine axis 'AX 5')
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Fig. 3-11 Two spindles in one channel

Two spindles of a single channel are operating with one magazine.

Spindle 1 defined as master spindle with SPIND_DEF_MASTER_SPIND = 1.

Spindle 2 is not a master spindle (secondary spindle).

References: **/PGA/**, “Programming Guide, Advanced”
(description of system parameters)

Example of a part program (for a channel with two spindles)

(Requirement: CUTTING_EDGE_DEFAULT=1;
i.e. D1 becomes active implicitly with tool change M6):

```
T=“Mill” M06   ;No address extension programmed –> the master
                     ;spindle is addressed, i.e. spindle 1 = value of machine
                     ;data $MC_SPIND_DEF_MASTER_SPIND.
                     ;The tool is changed on spindle 1.
                     ;The path is corrected
                     ;with the tool offsets
```

```
....
T2="Drill" M2=6       ;Address extension for secondary spindle has been pro-
                      ;grammed. The tool is changed in the PLC on the tool
                      ;interface for spindle 2.
                      ;The path is not corrected
...
SETMS(2)              ;Declares spindle no. 2 as master spindle
T="Mill_2" M6         ;No address extension programmed -> the
                      ;master spindle is addressed (spindle 2).
                      ;The tool is changed on spindle 2.
                      ;The path is corrected
                      ;with the tool offsets
....
T1="Drill_1" M1=6     ;Address extension for current secondary
                      ;spindle has been programmed.
                      ;The tool is changed on spindle 1.
                      ;The path is not corrected!
...
SETMS                 ;Declares the spindle defined by
                      ;$MC_SPIND_DEF_MASTER_SPIND to be the master spindle
T="Mill_3" M6         ;No address extension programmed -> the
                      ;master spindle is addressed (spindle 1)
                      ;Value of machine data $MC_SPIND_DEF_MA
                      ;STER_SPIND). The tool is changed on spindle 1.
                      ;The path is corrected
                      ;with the tool offsets.
```

Further example (starting conditions as above):

```
N10 SETMS             ;Declares spindle no. 1 as the master spindle
N20 T2=3
...
N50 M2=6              ;Address extension for secondary spindle has been pro-
                      ;grammed. The tool is changed in buffer
                      ;location 2.
                      ;The path is not corrected
...
N70 D3                ;The path is corrected with the offsets of the active
                      ;tool (activated before block N10)
                      ;with the tool offsets.
N80 SETMS(2)          ;declare spindle no. 2 master spindle
T3
M06
N90 D2                ;The path is corrected with the offsets of the active
                      ;tool (activated before block N10)
                      ;with the tool offsets.
```

Note

SETMS does not change the active tool. The new master spindle definition cannot be referenced until the subsequently programmed tool change.

Working with toolholder numbers

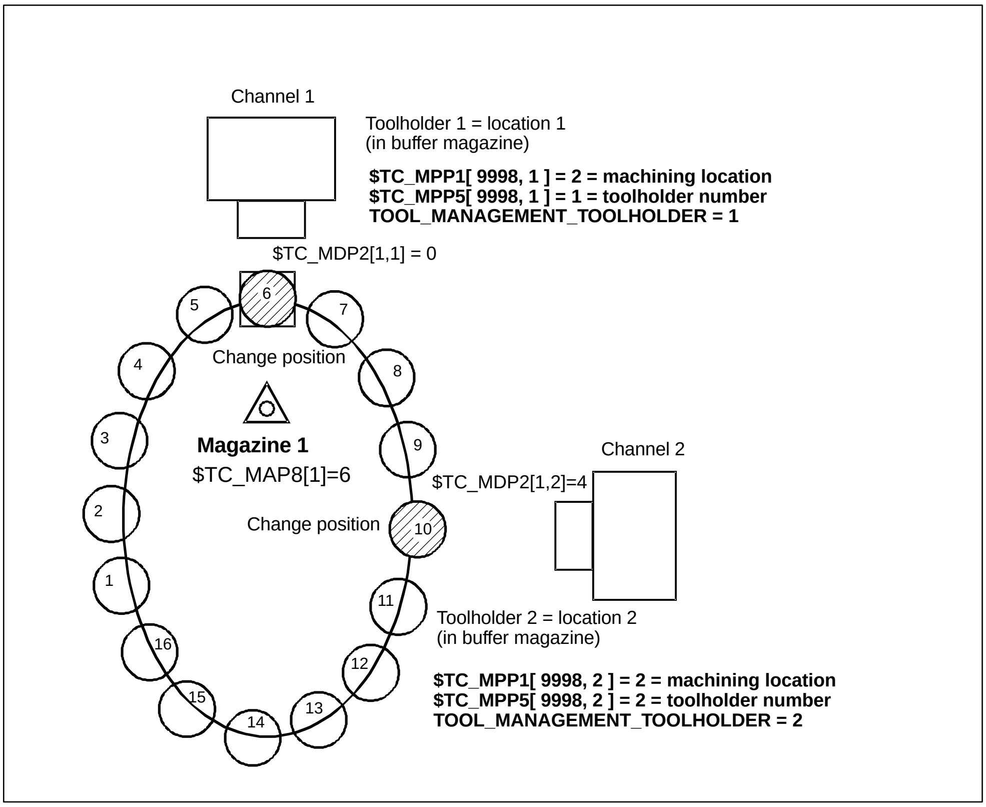

Channel 1
Toolholder 1 = location 1
(in buffer magazine)
$TC_MPP1[9998, 1] = 2 = machining location
$TC_MPP5[9998, 1] = 1 = toolholder number
TOOL_MANAGEMENT_TOOLHOLDER = 1
$TC_MDP2[1,1] = 0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Change position
Magazine 1
$TC_MAP8[1]=6
Change position
Channel 2
$TC_MDP2[1,2]=4
Toolholder 2 = location 2
(in buffer magazine)
$TC_MPP1[9998, 2] = 2 = machining location
$TC_MPP5[9998, 2] = 2 = toolholder number
TOOL_MANAGEMENT_TOOLHOLDER = 2

Fig. 3-12 Two channels with one toolholder each and one TO unit (the zero position is at the tool change position of toolholder 1)

Two channels are operating with the data of one TO unit (with one magazine). Tool change no longer requires that a spindle number be specified. The address extensions of T and M now refer to the setting in machine data MD 20124: TOOL_MANAGEMENT_TOOLHOLDER

Instead of ’spindle location’ the general term ’tool machining location’ is used (spindle is standard). If no address extension is programmed, the value in MD 20124: TOOL_MANAGEMENT_TOOLHOLDER is added as the extension.

TOOL_MANAGEMENT_TOOLHOLDER = 0

The previous function remains active (default).

A value greater than zero activates the new function.

TOOL_MANAGEMENT_TOOLHOLDER > 0

If a tool change to a buffer location of type 'tool machining location' is programmed with $TC_MPP5 = TOOL_MANAGEMENT_TOOLHOLDER, the offset data defined for that tool (TO unit) correct the path.

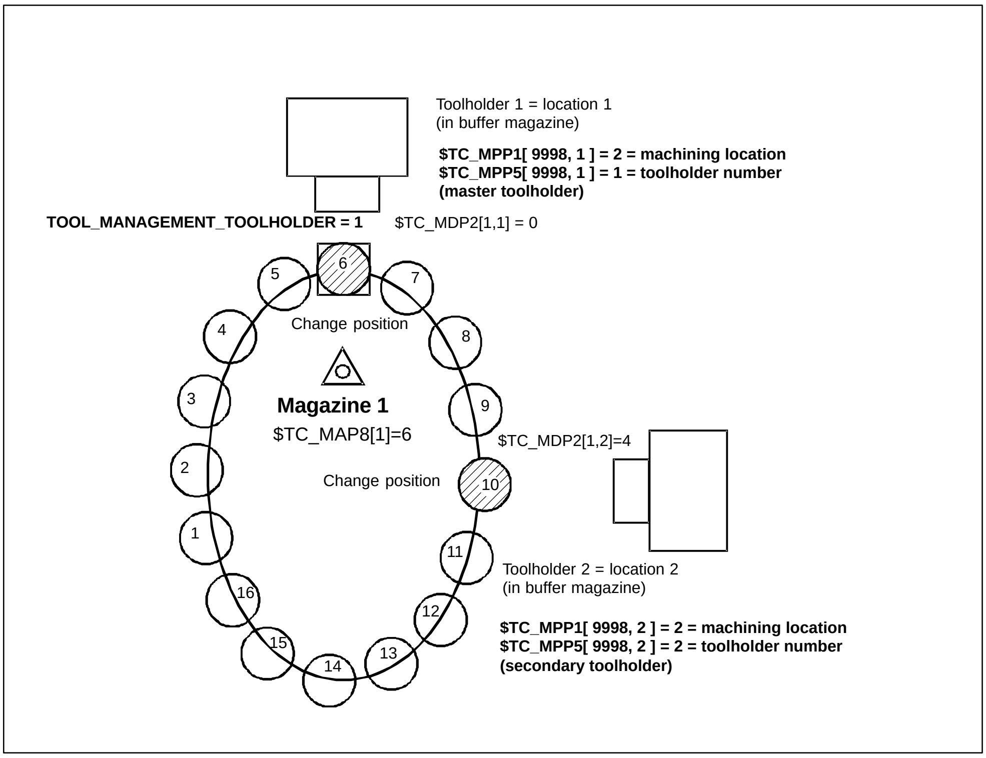

Toolholder 1 = location 1
(in buffer magazine)
$TC_MPP1[9998, 1] = 2 = machining location
$TC_MPP5[9998, 1] = 1 = toolholder number
(master toolholder)
TOOL_MANAGEMENT_TOOLHOLDER = 1
$TC_MDP2[1,1] = 0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Change position
Magazine 1
$TC_MAP8[1]=6
Change position
$TC_MDP2[1,2]=4
Toolholder 2 = location 2
(in buffer magazine)
$TC_MPP1[9998, 2] = 2 = machining location
$TC_MPP5[9998, 2] = 2 = toolholder number
(secondary toolholder)

Fig. 3-13 One channel with two toolholders (zero position is at tool change position of toolholder 1)

Two toolholders in a channel are operating with one magazine. Toolholder 1 has been declared the master via **TOOL_MANAGEMENT_TOOLHOLDER = 1**. Toolholder 2 is thus the secondary toolholder.

Programming example

Language command

SETMTH (toolholder number),

is used to declare different toolholders as the master toolholder.

```
T=“Mill” M6           ;No address extension programmed -> the master tool-
                      ;holder is addressed (toolholder 1 – value of machine
                      ;data $MC_TOOL_MANAGEMENT_TOOLHOLDER).
                      ;The tool is changed in buffer
                      ;location 1.
                      ;The path is corrected
                      ;with the tool offsets.

....
T2=“Drill” M2=6       ;Address extension for secondary toolholder has been
                      ;programmed.
                      ;The tool is changed in buffer
                      ;location 2.
                      ;The path is not corrected

...
SETMTH(2)             ;Declares toolholder 2 as the master toolholder
T=“Mill_2” M6         ;No address extension programmed -> the
                      ;master toolholder is addressed (toolholder 2).
                      ;The tool is changed in buffer
                      ;location 2.
                      ;The path is corrected
                      ;with the tool offsets.

....
T1=“Drill_1” M1=6     ;Address extension for secondary toolholder has been
                      ;programmed.
                      ;The tool is changed in buffer
                      ;location 1.
                      ;The path is not corrected!

...
SETMTH                ;Declares the toolholder set in $MC_TOOL_MANAGE
                      MENT_TOOLHOLDER to be the master toolholder
T=“Mill_3” M6         ;No address extension programmed -> the master tool-
                      ;holder is addressed (toolholder 1 – value of machine
                      ;data $MC_TOOL_MANAGEMENT_TOOLHOLDER).
                      ;The tool is changed in buffer
                      ;location 1.
                      ;The path is corrected
                      ;with the tool offsets.
```

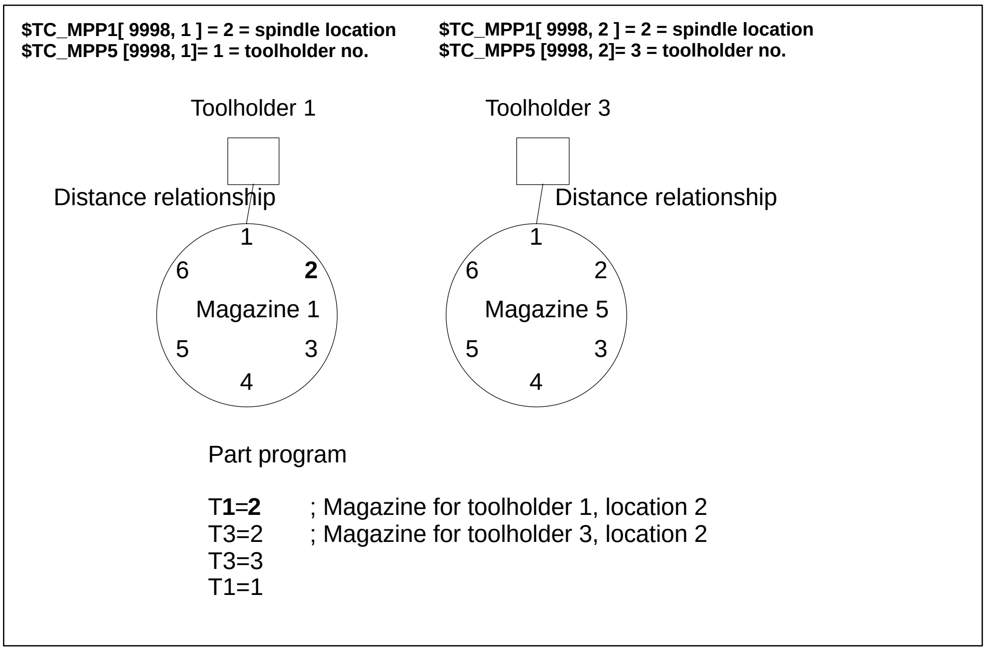
References: ***/PGA/***, “Programming Guide, Advanced”
(Description of System Parameters)

Note

SETMTH does not change the active tool. The new master toolholder definition cannot be referenced until the subsequently programmed tool change.

3.2.18 Several magazines in one channel or TO unit (SW 5.1 and later)

NC address T can be programmed with an address extension. The tool management function interprets the programmed address extension as a spindle number or toolholder number. The NC address T without programmed address extension then refers to the main spindle (master spindle).


$TC_MPP1[9998, 1] = 2 = spindle location
$TC_MPP5 [9998, 1]= 1 = toolholder no.
$TC_MPP1[9998, 2] = 2 = spindle location
$TC_MPP5 [9998, 2]= 3 = toolholder no.
Toolholder 1
Toolholder 3
Distance relationship
Distance relationship
1
6
2
Magazine 1
5
3
4
1
6
2
Magazine 5
5
3
4

Part program

```
T1=2     ; Magazine for toolholder 1, location 2
T3=2     ; Magazine for toolholder 3, location 2
T3=3
T1=1
```

Fig. 3-14 T="location" and several magazines in the same channel

The figure shows the procedure for using more than one magazine in a channel (when programming with T="location" this is usually a turret).

Note

The tool offset is calculated only for the toolholder that is assigned to the master spindle or master toolholder at the time of programming.

3.3 Search for tool

The tool search is triggered by the preparation command (T selection). The search begins for a tool to load in the spindle.

3.3.1 Strategies for tool searches

Tool search

The tools with the same identifier (name or Ident) but different duplo numbers are combined to form one tool group. The tool identifier is programmed in the part program with NC address T, i.e. only the tool group is defined during preparation.

In order to move a tool from a physical magazine to a spindle it must have the following characteristics:

- Tool status must be “enabled”
- Tool status must not be “disabled”
- Tool status must not be “tool change in progress”
- Tool must not already be assigned a spindle other than the requesting spindle
- Tool must be present in the magazine location (except for manual tools)
- This magazine must be linked to the requesting spindle via a distance relationship ($TC_MDP2)
- This magazine must not have the status “disabled”.

The explicit tool is requested at the time of the tool call. The request is made for a special toolholder (general toolholder); this is the number of the address extension of T. At this point in time, user interface DB 72 is written for the relevant spindle and must be evaluated by the PLC application program

The tool search strategy is defined by system parameter **$TC_MAMP2**. You can select the conventional search strategies with **bit0** to **bit2**. These strategies start searching at the magazine from which the loaded tool was fetched previously.

Extended tool search strategies

The existing tool search strategies are extended in SW 5.1 and later. As in earlier versions, the search strategy is defined via parameter **$TC_MAMP2**. The older strategies are selected via bits 0, 1 and 2. The additional functions can be activated via **bits 3, 4** and **5**.

By setting **bit 7** additionally, it is possible to make the strategies defined via **bits 0, 1 and 2** begin their search from the 1st magazine of the distance table (order in the distance table is defined via the programming order of $TC_MDP2). The default setting is **bit 7=0**. The search starts in the magazine from which the last tool changed was taken.

Note

Bit 3 = 1 to **bit 5 = 1** are relevant only when the monitoring function is active (defined via $TC_TP9). Otherwise they have no effect on the suitability check.

Activation

The following conditions must be met for the tool search strategies:

- The tool monitoring function must be active within the tool management system.
- The appropriate monitoring parameter values must be set for the cutting edges of the tools.
- The monitoring must be defined for the appropriate tool (system parameter $TC_TP9).

Note

If a monitoring type is activated for the tool with $TC_DP9, the current monitoring parameters are evaluated and, if necessary, the tool status set to ’disabled’ or ’prewarning limit reached’. An existing tool disable is not canceled, however, even if the monitoring function is deactivated for this tool.

3.3.2 Example of a tool search

Tool search operation

You wish to change the tool on a spindle.

The search sequence for the correct tool is as follows:

1. The control checks whether the tool which is called is already located on the spindle.
2. If buffer locations are linked to the spindle (see $TC_MSLR), the control checks whether a suitable tool is already located in one of these.

3. The tool search starts in the 1st magazine of the distance table ($TC_MDP2) according to the selected search strategy.(Applies only if bit 7 of $TC_MAMP2 = 1; otherwise, the search starts in the magazine from which the last loaded tool was fetched.)
4. If no tool is found in the first magazine, the distance table of the search is repeated in the next magazine.
5. If all the magazines that are linked to the spindle have been searched and no suitable tool found, the search is terminated with an alarm (22069 or 22068).

Any suitable tool with the programmed identifier found (not disabled) in one of the stages described above will be used.

3.3.3 Search in box magazines

Tool search strategy for box magazines

The special tool search strategy **“shortest distance”** is available for box magazines. The search strategy is set in system parameter **$TC_MAMP2**.

Definitions

The special tool search strategy **“shortest distance”** is defined as follows for box magazines:

- Location number with the smallest absolute value of the difference to the current magazine position.

The term **“current magazine position”** is defined as:

- the location number from which the previously loaded tool was taken.

Requirements

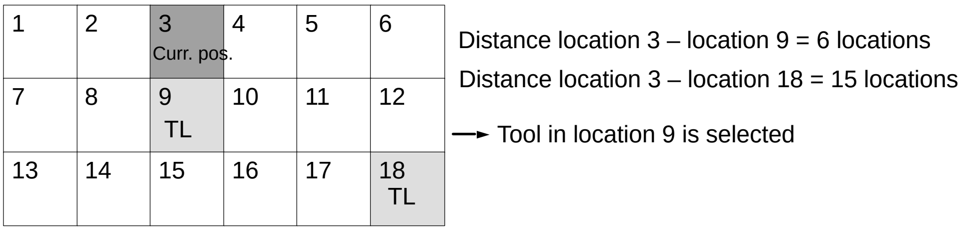
The search strategy can only be used if the box magazine is assigned a “current magazine position”. This is set in system parameter **$TC_MAP8**.

The NCK sets the current magazine position for box magazines. Because box magazines do not move, the magazine position is used as a formal value which is required for the tool search strategy.

Example

The machine tool has a box-type magazine with 3x6 locations (=18). The current position **$TC_MAP8** is location 3. Suitable tools are stored in locations 9 and 18.

The search strategy detects the tool at location 9, because the absolute value of the difference is only 6, compared with the difference of 15 to location 18.



1	2	3 Curr. pos.	4	5	6
7	8	9 TL	10	11	12
13	14	15	16	17	18 TL

Distance location 3 – location 9 = 6 locations
Distance location 3 – location 18 = 15 locations
→ Tool in location 9 is selected

Fig. 3-15 Search strategies in box magazine

3.4 Empty location search

3.4.1 Empty location search for a tool – from spindle to magazine

The T preparation command automatically searches for a suitable empty location for the spindle tool. The location in which the new tool is stored is still occupied at this time and **cannot** therefore be identified as an empty location (see also "Replace tool search strategy", Subsection 3.4.4)

Note

Generally in SW 5.2 and later, the empty location from which the current tool in the toolholder was taken is searched for.

Fixed location coding (up to SW 5.1)

The tool is "notified" of "its" magazine and magazine location as it loaded. After this, the location search only checks this location to determine whether it is still able to accept the tool. If a location type hierarchy has been defined this is taken into account. If the check is not successful, the search is terminated with an error code.

Fixed location coding is the most time-efficient method of finding an empty location for a tool.

Fixed location coding (SW 5.1 and later)

The definition for an empty location search for fixed location coded tools has been changed in SW 5.1 and later.

When searching for an empty location for fixed-location coded tools its previous location in the magazine is usually retained.

If the search for an empty location for a fixed-location-coded tool is started with a specific magazine number, that number is ignored. The old tool location is defined as an empty location.

But if this number is an internal magazine number (for a load or buffer magazine), the number is applied explicitly and the fixed-location code ignored. This occurs when tools are loaded and unloaded.

If a location search for a fixed location coded tool is initiated using a specific magazine number and magazine location number, the fixed location coding is ignored and the specified location checked as a suitable location for the tool. This method is used with the MMC function ’Restore’.

Variable location coding

Initially, the procedure for an empty location search is the same as that for a fixed location-coded tool. If this check fails, the search for a free location is continued. The search is performed according to the programmed search strategy ($TC_MAMP2). If the search cannot find an available location with the specified location type in this magazine, a new search operation based on the location type hierarchy (see Subsection 4.3.5) is started in the magazine. A location is interpreted to be a suitable location type if the “location type of the location” is greater than the “location type of the tool”, with the “greater than” relationship defined by the location type hierarchy. If no free location is found in this magazine, the search is continued in the next magazine (search strategy).

3.4.2 Search strategy for empty locations

Search strategy

The strategy according to which a search for a free location is made in the chain magazines of the TO unit can be defined in the magazine configuration. For magazines other than chain magazines, the default search strategy is applied (forwards search starting at the first location number).

Possible strategies are listed in the table.

$TC_MAMP2	Search strategies	Meaning
Bit 8 = 1 256	Forward search	The search is started from location no. 1 in ascending order.
Bit 9 = 1 512	Forward search	The search starts from the current location at the change position in ascending order.

Bit 10 =1 1024	Backward search	The search started backwards from the last location no.
Bit 11 = 1 2048	Backward search	The search starts from the current location in the change position backwards.
Bit 12 = 1 4096	Symmetrical search	The search begins at the tool change position in the current location (1st location on left, 1st location on right, 2nd location on left, 2nd location on right, etc.).

Definition of the current magazine position

The current magazine position in relation to the zero point is stored in $TC_MAP8. The value is automatically updated in response to the PLC acknowledgement of a command initiated by the TM function. If the magazine is moved without a task from the NCK, the user must adjust the actual position in $TC_MAP8 independently.

This can be done via a part program or by the PLC by writing $TC_MAP8 (selection via NC VAR selector block TM, variable magNoPlaces and assignment via PLC with FB 3, see Section 5.4.1).
Also via FC8 with parameters TaskIdent = 4, TaskIdentNo = channel no., status = 5, OldToolMag = 9998, OldToolLoc = 1. The current position is parameterized (referred to spindle) in NewToolMag and NewToolLoc.

3.4.3 Empty location search criteria

Criteria for empty location search

- Location type must coincide with location type of tool. Any defined hierarchy is taken into account.
- Tool size check
- Location must have the status “free”.
- Location must not be “disabled”
- Magazine must not be “disabled”

Magazine location type

The essential search criterion for the empty location search is the magazine location type. The magazine location type must match the magazine location type entered in the tool-specific data ($TC_TP). The magazine is searched. Each location is checked. If a suitable location is found the search is terminated.

If no suitable location is found a search is made for a magazine location type hierarchy for the magazine location type entered in the tool. If there is none, the next magazine is taken, if one exists.If there is a defined hierarchy, the search is repeated at the magazine that has just been searched. If this search is also unsuccessful, the search moves to the next magazine, assuming another one is available.

3.4.4 “Replace tool” search strategy (old for new; SW 5.1 and later)

With this search option, the magazine location of the ’new’ tool (tool to be loaded) is made available as the empty location for the ’old’ tool (tool to be unloaded).

It is not necessary for the “new” tool to be stored in the magazine location. It only needs to have been loaded (it may be located on a gripper, for example). If the location in question is not suitable for the “old” tool, then another appropriate empty location is sought.

Description of function

The new empty location search strategy is preset via **bit 13** of the existing bit-coded system parameter **$TC_MAMP2**.

Conditions

With this strategy, the NCK checks a magazine location that is normally occupied by the “new” tool or tagged as “reserved for tool from buffer” at the time the check is performed. This location is defined as an empty location for the “old” tool if the check gives a positive result.

If the new or old tool is coded with a fixed location, the strategy is not used – the fixed coding has priority.

Note

The PLC program must execute the tool movements required to change a tool in the correct sequence:

- Remove “new” tool from magazine location
- Take “old” tool to the magazine location

Otherwise the machine or tool might be damaged.

The empty location search strategy only works as part of the tool change programmed in the part program. It cannot be utilized by the PI services (e.g. TMFDPL, TMFPBP) for empty location searches (see Subsection 5.12.5).

Example

This strategy is especially suitable for use with grippers and tools of the same type (same size and same location type or type that is compatible with defined location type hierarchy).

The already existing system parameter $TC_MAMP2 has been provided with an additional setting option for this new empty location search strategy.

Bit	Value	Meaning
0		
...		Tool search strategy
7		
8		
...		Empty location search strategy
13		The magazine location of the “new” tool to be loaded is transferred to the “old” tool to be unloaded and vice versa. Method is applicable only if the **tool sizes and location types of the tools are identical** or are compatible in terms of the location hierarchy. The location of the “new” tool is recorded as the empty location for the “old” tool, even if the “new” tool is still stored in this location when the check is made. The tool transportation sequence must be arranged such that the “new” tool is removed from the magazine location before the “old” tool is taken to it. This order is vital to prevent the risk of any damage to the machine which might result from mechanical tool transportation motions. The type of empty location search is determined via bits 8 to 12. The exchange cannot work if the “old” tool is not assigned to any magazine location. The tool search strategy is then determined via bits 8 to 12.

3.4.5 Tool search in wear group (SW 5.1 and later)

If “Wear group” function is used:

With existing tool search strategies, the search applies only to the active wear group, i.e. searches within a tool group take only those tools located in the magazine locations of the active wear group into account.

Tools in magazine locations with wear group number 0 are also checked for suitability.

If no replacement tools are available, all $TC_MPP5 parameters of the are negated and all locations thus blocked individually. $TC_MAP9 is also negated (wear group disabled). All active tools are reset if this response has been configured via $TC_MAMP3 (bit1 = 1).

The next wear group is called ($TC_MAP9 is assigned the number of the next wear group that can be activated).

If no further groups are available the search is terminated with an alarm. In such a case, the disabled tools should be replaced, if necessary. In order to enable the wear groups again, the wear group numbers of the magazine locations must be reset to values > 0.

Search strategies

There are two search strategies for finding the next available wear group for activation:

- Starting from the lowest magazine location number, the replacement tools are searched through location by location according to the way they are sorted internally (time-optimized search). The wear group that is the subject of the search is found by searching for the first tool that is assigned to a wear group that can be activated.
- A search is made for the wear group with the lowest enabled wear group number (the first that can be activated).

Search in several magazines

The magazine definition for a machine defines whether the search is to be performed in one or several magazines.

If the search is conducted in several magazines while several wear groups are being used, always make sure that a wear group can only ever be assigned to one magazine.

The search is conducted acc. to the following priorities:

1. The search is performed in a magazine according to the configuration and strategy.
2. The search is performed in the active wear group.
3. The set tool search strategy is applied.

Activation

In order to work with wear groups, the magazine locations must be assigned to wear groups via system parameter **$TC_MPP5** and the function must be activated via the machine data.

In addition, the number of the wear group with which machining is to commence must be assigned to system parameter **$TC_MAP9** of the magazine to be selected (value > 0).

When the machine is configured, **$TC_MAMP3** is set to define how the tool status must change in the event of switchover from one wear group to the next ("tool status remains unchanged" is default setting).

Example: Tool searches in wear groups (SW 5.1 and later)

$TC_MAMP3 = 3 – Change "active" status of tools

Aim

- The tools must be set to "active" when a wear group is activated.
- When a wear group is disabled all the tools contained in that wear group are also to be disabled.

Requirement

- Circular magazine number 1 (6 locations)
- The magazine is to be divided into two parts:
  – Locations 2 and 3 from wear group 1.
  – Locations 4, 5, 6 and 1 form wear group 2.
- $TC_MAP9 = 1 (wear group 1 is "active")

Assignment to the wear group is achieved by:

```
$TC_MPP5[1,2] = 1
$TC_MPP5[1,3] = 1
$TC_MPP5[1,4] = 2
$TC_MPP5[1,5] = 2
$TC_MPP5[1,6] = 2
$TC_MPP5[1,1] = 2
```

The tools with T=10 and T=11 are assigned to wear group 1. As wear group 1 was activated, tools T=10, 11 were thus also set to "active" (via $TC_MAMP3, bit0=1).

Note

Language command SETTA (see Subsection 5.8.18) can also be used to set the tools to active.

Tool assignments:

```
$TC_MPP6[1,2] = 10 ;T=10 has identifier "TL1"/duplo no.=1 TL status "active"
$TC_MPP6[1,3] = 11 ;T=11 has identifier "TL2"/duplo no.=1 TL status "active"
$TC_MPP6[1,4] = 12 ;T=12 has identifier "TL1"/duplo no.=2
$TC_MPP6[1,5] = 13 ;T=13 has identifier "TL2"/duplo no.=2
$TC_MPP6[1,6] = 14 ;T=14 has identifier "TL1"/duplo no.=3
$TC_MPP6[1,1] = 15 ;T=15 has identifier "TL2"/duplo no.=3
```

$TC_MAMP2 = 1

The active tool is to be searched for. If none is available, the next possible tool is to be located.
This tool search strategy is a secondary function of the check for the number of the active wear group, i.e. the search for a tool with “active” status takes only those tools into account that are stored in magazine locations with the number of the currently activated wear group.

T=“TL2”

Tool group “TL2” consists of tools T=11, 13, 15.

T=11 is positioned in a location of the active wear group (no. 1) and is “active”.
The tool search result is thus T=11.
Machining is continued. T=11 is disabled during machining.

T=“TL1”

Wear group 1 is still active. T=10 is identified as active and suitable.

T=“TL2”

The tool group of identifier “TL2” now has no active tool (has been disabled) and a new tool has not yet been set to “active”. This step is not taken until “TL2 is reprogrammed. The tools of the group are examined. In the locations of wear group 1, which is still active, there is no tool with identifier “TL2” or any other suitable tool.

This condition causes the next wear group (2) to be activated. Wear group 1 is now no longer the active wear group. The status of the tools in wear group 1 has been reset (not “active”), as configured by **$TC_MAMP3, bit1=1**.

The tool search is now centered on wear group 2. Its tools were set to “active” when the wear group was activated (one tool from each tool group in the wear group because setting of $TC_MAMP3, bit0=1).

The turret is now assigned as follows:

```
$TC_MPP6[1,2] = 10 ;T=10 has identifier “TL1”/duplo no.=1.
                  Tool status “not active”

$TC_MPP6[1,3] = 11 ;T=11 has identifier “TL2”/duplo no.=1.
                  Tool status “disabled”

$TC_MPP6[1,4] = 12 ;T=12 has identifier “TL1”/duplo no.=2, tool status “ac-
tive”

$TC_MPP6[1,5] = 13 ;T=13 has identifier “TL2”/duplo no.=2, tool status “ac-
tive”

$TC_MPP6[1,6] = 14;T=14 has identifier “TL1”/duplo no.=3
```

```
$TC_MPP6[1,1] = 15;T=15 has identifier “TL2”/duplo no.=3
```

In the example **T=13** is now taken as the next available tool “WZ2”.

Note

The tool search generates an alarm only when no more replacement tools in the tool group with the specified identifier can be found **and** no other wear group can been activated.

Control behavior

Control behavior on Power On, Mode group change, Reset, Block search and REPOS is described below.

Configuration of $TC_MAMP3, bit0=1 (internal activation)

During Power On, the NCK checks whether the value of $TC_MAP9 is > 0, i.e. whether a wear group has been selected. In this case the tools of that wear group are checked again and the value for $TC_MPP5 of each location in question is set to positive. In addition, the status of the tool in the location is set to “active”.

Configuration of $TC_MAMP3, bit1=1 (internal disabling)

On Power On, the NCK checks whether $TC_MAP9 is negative, i.e. a wear group has been disabled. In this case the tools of the disabled wear group are checked again and the value $TC_MPP5 of the location negated. The “active” status of the tool in the location is reset.

Machine/option data

The following bit has been added to the existing bit-coded machine data **MD 18080: MM_TOOL_MANAGEMENT_MASK**:

Bit	Value	Meaning
6		0The **wear group** function is **not** available. Parameters $TC_MAMP3 and $TC_MAP9 are **not** programmable. $TC_MPP5 is **not** defined for magazine locations of location type 1.
	1	The **wear group** function is available. Parameters $TC_MAMP3 and $TC_MAP9 are programmable. Wear groups can be defined. $TC_MPP5 contains the wear group number for location type 1.

3.5 Load a tool

When a tool is loaded, it is taken to its magazine location and the associated data entered. Tools can be loaded via the spindle or a load magazine.

With the MMC 103, tool data can be transferred from the tool catalog, tool cabinet or via a code carrier system (see Section 2.8 and 3.12). In SW 3.2 or later, the tool data can be entered directly into the magazine list with the MMC.

- Manual loading only
- Empty location search
- Load current location (location at the loading/unloading point)

3.5.1 Loading sequence

The loading operations supply the magazines with tools and write data to the relevant data areas of the TM system (magazine list with tool data, offset memory). Various methods of loading are available depending on the magazine configuration (load magazine yes/no) and the data flow (when and from where are tool data written to the relevant data areas).

The loading method is mainly relevant to the MMC. As regards the TM system on the NCK, only the result is important, e.g. that the tool is in the magazine and enabled for use after transfer of all its data.

Reference: /BA/ Operator’s Guide

Loading is a channel-specific operation which can be executed while the part program is running. System variable $TC_MAP3 = 16 (enabled for loading) must be programmed if tools are to be loaded during part program runs.

There are two basic loading methods:

Free loading

With this method, the user can specify a magazine location to which the tool must be loaded.

Prompted loading

With prompted loading via the MMC, the location is determined by the TM using an empty location search (see Subsection 3.4.2).

3.5.2 Tool data

The MMC 102/103 offers various options for loading and unloading the data of a tool and for storing the data.

These options can be used either individually or in parallel by the user.

When a tool is unloaded, the data can

- stay on the NCK (tool list)
- be written to code carrier (floppy, ext. hard disk, etc.)
- be stored in the tool cabinet (int. hard disk).

The tool data can be fetched again from these “data carriers” on loading. Tool data can also be entered directly by the user into the magazine list and/or the tool list.

Note

The type of data backup can be defined by access rights in the PARAMTM.INI. file.

Master data can be stored in the tool catalog. Other enabled functions, such as interactive programming, can access tools which are defined here.

Selecting a tool for loading

- Select tool from tool catalog (new tool)
- Select tool from tool cabinet (particular tool data)
- Enter tool data directly in the magazine list (MMC 102/103)
- Select tool from the tool list (tool offset memory)
- Read in tool data via a code carrier system (see Section 3.12)

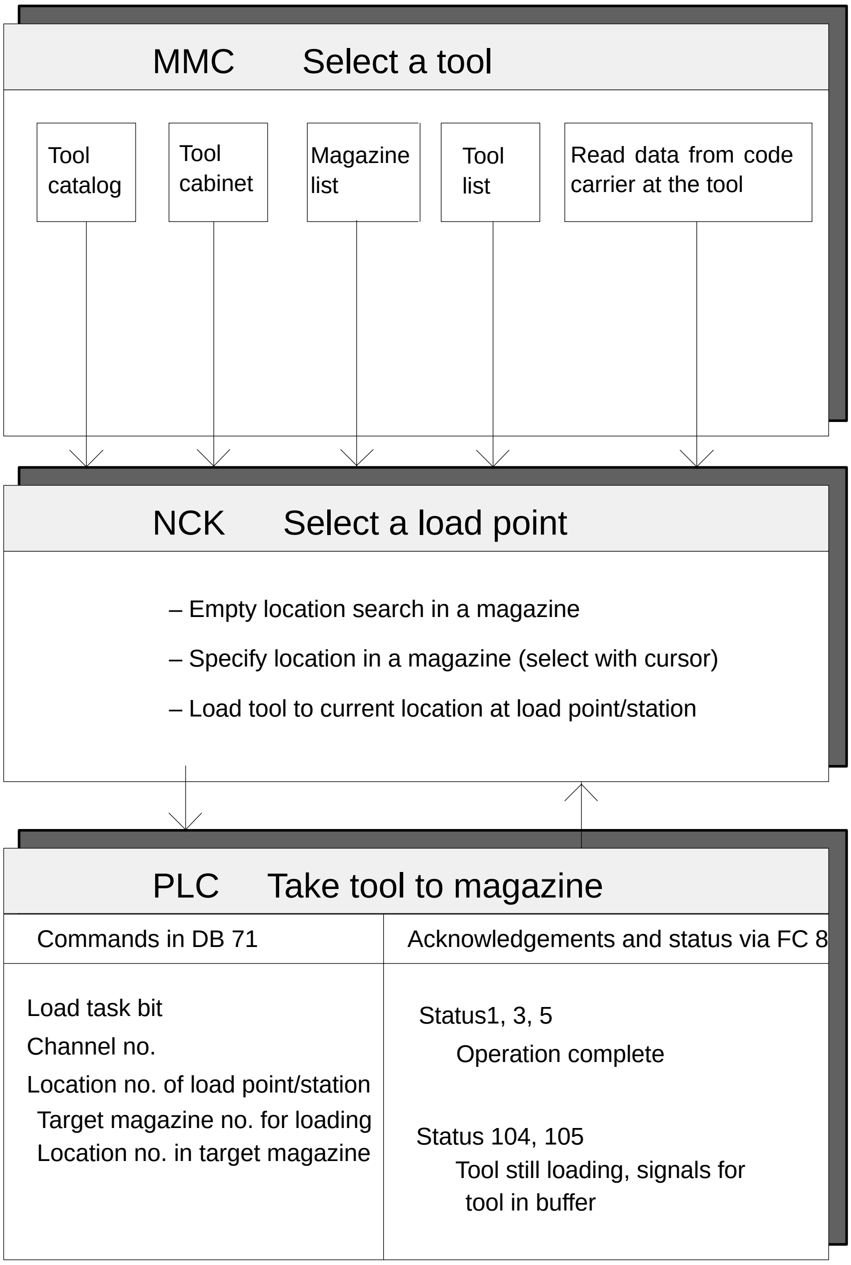

MMC Select a tool
Tool catalog
Tool cabinet
Magazine list
Tool list
Read data from code carrier at the tool
NCK Select a load point
– Empty location search in a magazine
– Specify location in a magazine (select with cursor)
– Load tool to current location at load point/station
PLC Take tool to magazine
Commands in DB 71
Acknowledgements and status via FC 8
Load task bit
Channel no.
Location no. of load point/station
Target magazine no. for loading
Location no. in target magazine
Status1, 3, 5
Operation complete
Status 104, 105
Tool still loading, signals for tool in buffer

Fig. 3-16 Loading-related functions of MMC, NCK and PLC

3.5.3 Select magazine location for loading

Find location in magazine

There are 3 possible ways of selecting an empty location:

- Initiate an empty location search (softkey)

- Input desired location number in magazine list (cursor)
- Move the desired empty location manually to the load magazine and then load this location with softkey “Current location”.

3.5.4 Functions of PLC during loading

Loading sequence

During load operations, the PLC is controlled from the NCK via magazine and location numbers. It receives the request to move the magazine to the appropriate load magazine for tool loading.

When a tool is loaded, the target address is the magazine and the loading location for the tool (DB71. DBW (n+24) and (n+26)). FC 8 receives this target address as parameters **“NewToolMag”** and **“NewToolLoc”** and “Status = 1” once the load operation has been successfully completed. Parameters “OldToolMag” and “OldToolLoc” must be set to zero. The no. of the active interface identifies the load magazine (location no.).

The loading procedure is performed as follows:

1. The PLC is requested to load the tool. The information is transferred to the PLC in DB 71.

   Example:

   Data in DB71 for load operations via the 2nd interface,
   (location 5 in magazine 1 must be loaded from load magazine 2)

   ```
   DB71.DBX0.1     =1        ;Interface 2 active
   DB71.DBX34.0    =1        ;Command: Load
   DB71.DBW50      = 9999    ;Magazine no. of load magazine
   DB71.DBW52      =2        :Location no. of load magazine
   DB71.DBW54      = 0       ;Magazine no. for unloading
   DB71.DBW56      = 0       ;Location no. for unloading
   DB71.DBW58      =1        ;Target magazine no. for loading
   DB71.DBW60      = 5       ;Target location no. for loading
   ```

2. The PLC must now move “location 5” from “magazine no. 1” (in which tool must be loaded) to “load magazine 2” and execute the load operation.
3. When the tool is in the magazine, the user program must call FC 8. This notifies the tool management that the tool has been loaded.

Example of FC 8 call on loading

FC 8 parameters	Values	Comment
Start	1	Starts task
TaskIdent	1	DB 71 interface
TaskIdentNo	2	No. of active interface
NewToolMag	1	Mag. no. 1
NewToolLoc	5	Location no. 5
OldToolMag	0	During loading = 0
OldToolLoc	0	During loading = 0
Status	1	Operation complete
Ready		Checkback from FC 8
Error		Checkback from FC 8

Problems during loading

A tool cannot be loaded. Check the following:

- Is the location type correct?
- Is a suitable empty location available?
- Has the number of tools enabled in the NCK (MD18082) been reached?
- Does the tool size contain a “0”, e.g. “1011”? (This is illegal.)

Alarms on the operator panel:

- No suitable empty location available
- “Create tools” command cannot be output to the NCK

3.5.5 Load tools via a part program

T number

The data required for a tool can also be loaded via a part program.

There are two methods of obtaining the T number that addressed the data. You can:

- assign the T number yourself or
- allow the NC to assign a T number
  (via command NEWT(...), see Subsection 5.8.8).

A T number derived in this way can now be used to address further data. Otherwise the T number can be assigned by the user (see example below)

Example

```
DEF INT TNr
TNr=NEWT(“test”,1)
$TC_TP3[4711]=2               ;Size left
$TC_TP4[4711]=2               ;Size right
$TC_TP5[4711]=1               ,Size top
$TC_TP6[4711]=1               ;Size bottom
$TC_TP7[4711]=2               ;Location type
$TC_TP8[4711]=2               ;Tool status
$TC_TP9[4711]=0               ;Monitoring mode
$TC_TP10[4711]=0              ;Replacement TL strategy
$TC_TP11[4711]=0              ;TL inf

$TC_DP1[4711]=120             ;TL type:
                              ;(all the necessary offset data can be sup
                              ;plied here)
$TC_MPP6[MagNo,LocNo]=4711    ;Tool with T number 4711 is written/loaded to
                              ;the location
```

The tool described here also occupies adjacent locations. These are automatically reserved for/assigned to the tool by the TM system (see Subsection 4.3.3).

It is also possible to delay assignment of a tool to a location, in which case the command $TC_MPP6 is not required. After execution of the part program the tools are contained in the tool list and can be loaded at a later time.

3.5.6 Retroload tool data

When tool data are “retroloaded” this means that the compensation data are not entered or loaded until after the tool loading operation.

Procedure

- The tools are already located in the magazine both mechanically and in the software, i.e. the assignment “tool –> location” has been made.
- There are either no tool compensation data in the NC or they are no longer up to date.

The offset data are assigned via the part program, i.e. the existing data are overwritten. To do this, the internal T number of the tools in question must be determined in the “retroload” program if it is not already known.

The internal T number is the tool number used by the NC. It is unique and describes a tool. All parameters of this tool are addressed via this T number.

The T number is either assigned by the user during loading or it is assigned by the NC (if the user does not make an entry).

If the T number is known to the user (e.g. defined at the measuring station by the entries made), he can refer to this number in the retroload program.

If the T number is not known, it must be defined for every tool to be reloaded and then assigned a variable. This means less effort for the user and thus less error-prone.

Create the retroload program

The tool is measured at a measuring station and the measured data stored. For this purpose, the tool must already be defined, i.e. by both an identifier ("Drill 12 mm" or "Mill 23" below) and the relevant duplo number. (The combination of tool identifier and duplo number uniquely defines the tool). The command GETT(...) is programmed at the beginning of every to determine the internal T number of the tool which is then stored as a variable ("TNo") in this case) (see Subsection 5.8.10). The data required for the tool are written and then the entire program is transferred to the NC where it is processed.

Only the variables for which data are entered have to be written. The first tool in the next retroload program contains all the data, the second tool only contains the relevant data.

If the T number is defined during loading, it does not have to be derived in the retroload program because in this case the data can be assigned directly.

For a tool "1" with length L1, the program would look like this:

```
$TC_DP1[1,1]=120;     ;Tool type

$TC_DP3[1,1]=4711;    ;Length1
```

Program for retroloading tool offset data

```
DEF INT Tno                  ;Definition of variable TNo
tl1:
TNo=GETT ("Drill 12mm",1)
if TNo==-1 goto tl2
$TC_DP1[TNo,1]=120           ;Tool type
$TC_DP2[TNo,1]=0
$TC_DP3[TNo,1]=4711          ;Length1
$TC_DP4[TNo,1]=0
$TC_DP5[TNo,1]=0
$TC_DP6[TNo,1]=24            ;Radius
$TC_DP7[TNr,1]=0
$TC_DP8[TNr,1]=0
$TC_DP9[TNo,1]=0
$TC_DP10[TNo,1]=0
$TC_DP11[TNo,1]=0
```

```
$TC_DP12[TNo,1]=0
$TC_DP13[TNo,1]=0
$TC_DP14[TNo,1]=0
$TC_DP15[TNo,1]=0
$TC_DP16[TNo,1]=0
$TC_DP17[TNo,1]=0
$TC_DP18[TNo,1]=0
$TC_DP19[TNo,1]=0
$TC_DP20[TNo,1]=0
$TC_DP21[TNo,1]=0
$TC_DP22[TNo,1]=0
$TC_DP23[TNo,1]=0
$TC_DP24[TNo,1]=0
$TC_DP25[TNo,1]=0
$TC_MOP1[TNo,1]=0
$TC_MOP2[TNo,1]=0
$TC_MOP3[TNo,1]=0
$TC_MOP4[TNo,1]=0
tl2:                          ;Next tool
TNo=GETT (“Mill23”,2)
if TNo==-1 goto error         ;Possible error routine if tool is not
                              ;available
$TC_DP1[TNo,1]=120
$TC_DP3[TNo,1]=4712
$TC_DP6[TNo,1]=25
Error:;                       ;Error
:
:
M17
```

3.6 Unload a tool

On unloading, the tool is removed from the magazine and the magazine list. You can:

- unload manually or
- unload the current location (location at the loading/unloading point)

The unloading sequence is as follows:

1. Selecting a tool for unloading
   To do this, place the cursor on the tool in the magazine list or, on the MMC 102/103, on the tool in the tool list and select softkey “Unload”.
2. Select unload point.
3. Move tool to unload point (by means of user PLC program).
4. Save or delete tool data.

Reference: /BA/ Operator’s Guide

3.6.1 Data backup during unloading

On unloading, the particular tool data are removed from the magazine list.

The following options are available for backing up the particular tool data:

1. Back up tool data on a code carrier
2. Back up tool data in tool list (TO memory)
3. Back up particular tool data in tool cabinet

It is still possible to delete the tool data without saving them.

Note

Data can be backed up on the MMC 103 in the following different ways:

- From the tool list
- From the tool cabinet or
- From the tool catalog

3.6.2 Functions of PLC during unloading

During unload operations, the FC 8 receives the identifier of the load/unload point as the target address of the tool (DB71.DB(n+16) and DBW(n+18), basic address “n” is included in the interface list). This target address is sent to FC 8 as parameters “**OldToolMag**”, “**OldToolLoc**” and “Status” = 1 once the tool has been successfully unloaded. The “NewToolMag” and “NewToolLoc” parameters must be assigned the value zero.

Unloading sequence

Unloading is controlled via DB71. The unloading sequence is as follows:

1. The PLC receives the command to unload the selected tool. The information is transferred to the PLC in DB71. Example of unloading data in DB 71 for the 2nd interface. Location 7 of magazine no. 1 must be unloaded to load magazine 2.

```
Example:

DB71.DBX0.1= 1            ;Interface 2 active
DB71.DBX34.1= 1           ;Command: Unload
DB71.DBW50= 9999          ;Magazine no. of unload point
DB71.DBW52= 2             ;Location no. of unload point
DB71.DBW54= 1             ;Magazine no. for unloading
DB71.DBW56= 7             ;Location no. for unloading
DB71.DBW58= 0             ;Target magazine no. for loading
DB71.DBW60= 0             ;Target location no. for loading
```

2. The PLC must now move “Location 7” of “Magazine no. 1” (from which the tool must be unloaded) to “Load/unload point 2” and then unload the tool.
3. As soon as the tool has been taken out of the magazine, FC 8 must be called by the user program. This signals the tool management where the tool has been transported.

Example: FC 8 call on unloading

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	1	DB 71 interface
TaskIdentNo	2	No. of active interface
NewToolMag	0	During unloading = 0
NewToolLoc	0	During unloading = 0
OldToolMag	9999	Mag. no. 9999
OldToolLoc	2	Location no. 2
Status	1	Operation complete
Ready		Checkback from FC 8
Error		Checkback from FC 8

The PLC user program then has to move the magazine to the correct unloading point and execute unloading. If the tool arrives at the unload point or station via the buffer (gripper, loader, etc.), each position change must be notified to the NCK via FC 8 with status 104, 105. Status “1” is not set via FC 8 until the tool is in the specified unload point/station. The unloading operation is now complete.

Position for unloading (with OP030 and MMC 103)

When a magazine is being **positioned** at a load magazine, the target address is stored in DB71.DBW(n+16) and DBW(n+18). This target address is passed to FC 8 as parameters “NewToolMag” and “NewToolLoc” and “Status” = 1 once the magazine has been successfully positioned. Parameters “OldToolMag” , “OldToolLoc” must be set to 0.

The magazine and magazine location to be positioned are stored in DB71.DBW(n+20) and DBW(n+22). “Positioning” in this sense refers only to the magazine positioning of a free or occupied location at a load/unload station. The number of the active interface identifies the load magazine (location no.).

Example: Position for unloading

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	1	DB 71 interface
TaskIdentNo	2	No. of active interface
NewToolMag	**9999**	**Mag. no. 9999**
NewToolLoc	**3**	**Location no. 3**
OldToolMag	0	During positioning = 0
OldToolLoc	0	During positioning = 0
Status	1	Operation complete

Note

The function Positioning to unload can only be triggered from operator panel OP 030 in SW 3.2.

3.7 Relocate, find and position tools

3.7.1 Relocate (task from TM system)

The target address for **relocation** is the magazine and location for the tool to be relocated (DB71.DBW(n+24) and DBW n+26). The tool source address is stored in DB71.DBW(n+20) and DBW(n+22). The target address is passed to FC 8 as parameters "NewToolMag" and "NewToolLoc" and status = 1 when relocation has been successfully completed. Parameters **"OldToolMag"** and **"OldToolLoc"** must be set to zero because the tool management recognizes the location of the old tool.

Reference: /BA/ Operator's Guide

Example of relocating a tool

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	1	DB 71 interface
TaskIdentNo	1	Channel no. for TM
NewToolMag	2	/New magazine no.
NewToolLoc	17	New location no.
OldToolMag	**0**	**Old mag. no. not used**
OldToolLoc	**0**	**Old location no. not used**
Status	1	Operation complete

Note

If the relocation operation is configured with status = 6, the magazine location for the tool is reserved in the buffer. As in the case of overwrote from PLC, this status can only be activated if the relocation is from a real magazine location to a location in the buffer.

3.7.2 Relocation by the PLC

Task from PLC

The PLC can also request the tool management to relocatea tool. This is done by notifying a new location for the tool to the tool management. The FC 8 block (TaskIdent := 4) is called with the following parameters:

• Old magazine no.	(OldToolMag)
• Old location no.	(OldToolLoc)
• New magazine no.	(NewToolMag)
• New location no.	(NewToolLoc)

1. Example

Relocation by PLC

The tool in magazine no. 1, location no. 5 is to be relocated to magazine no. 2, location no. 17. The PLC takes responsibility for ensuring that the location type is correct for the transfer. This example for an FC 8 call does not include a check-back signal to the tool management for intermediate positions of the tool.

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	4	Task from PLC
TaskIdentNo	1	Channel no. for TM
NewToolMag	2	New magazine no.
NewToolLoc	17	New location no.
OldToolMag	1	Old mag. no.
OldToolLoc	5	Old location no.
Status	1	Operation complete

2. Example

Relocation by PLC

Example: The tool is to be relocated from mag. no. 1, location no. 5 via grippers 3 and 4 to mag. no. 2, location no. 17.
FC 8 must be called up 4 times in this procedure. Only the important parameters are listed. All other parameters are as for the example above.

The tool is transported in 4 steps:

1. Move from magazine, location 5 to gripper 3 (location no. 4)

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	4	Task from PLC
TaskIdentNo	1	Channel no. for TM

FC 8 parameters	Values	Comment
NewToolMag	9998	New magazine no.
NewToolLoc	4	New location no.
OldToolMag	1	Old mag. no.
OldToolLoc	5	Old location no.
Status	1	Operation complete

2. Move from gripper 3 to transfer location 2, location no. 6

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	4	Task from PLC
TaskIdentNo	1	Channel no. for TM
NewToolMag	9998	New magazine no.
NewToolLoc	6	New location no.
OldToolMag	9998	Old mag. no.
OldToolLoc	4	Old location no.
Status	1	Operation complete

3. Move from transfer location 2, location no. 6 to gripper 4, location no. 5

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	4	Task from PLC
TaskIdentNo	1	Channel no. for TM
NewToolMag	9998	New magazine no.
NewToolLoc	5	New location no.
OldToolMag	9998	Old mag. no.
OldToolLoc	6	Old location no.
Status	1	Operation complete

4. Relocate from gripper 4, location no. 5 to magazine 2, location 17

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	4	Task from PLC
TaskIdentNo	1	Channel no. for TM
NewToolMag	2	New magazine no.
NewToolLoc	17	New location no.

FC 8 parameters	Values	Comment
OldToolMag	9998	Old mag. no.
OldToolLoc	5	Old location no.
Status	1	Operation complete

Relocation by PLC with location reservation TaskIdent 5

When a tool is transferred from a magazine location to a buffer with initiation from the PLC, it can be useful to reserve the magazine location.

You can do this with Task Ident 5.

The magazine location is now reserved when a tool is transferred to a buffer.

Note

TaskIdent 5 may be programmed only for a tool transfer (magazine 0 –> buffer location). Otherwise an error message is output, even though the tool is transferred.

Reservation “ Z ” is automatically reset when the tool is transferred from the buffer back to the magazine.

3.7.3 Find and position (with OP 030 only in SW 3.2)

With a find and position operation, a traversing task is sent to the PLC by the tool management. Bit 3 in DB71.DBB(n+0) “Position at load magazine” is set. The magazine no. and location no. are transferred (as the target) in parameters DB71.DBW n+20 and n+22 during positioning.

The PLC then has to move this location to the load magazine. The number of the load magazine is entered in DB71.DBW (n+18) or determined by the number of the interface. If the PLC has moved the magazine location to the load magazine, FC 8 must be called and the operation acknowledged with status 5 “Position changed”.

Example:

Location 5 in magazine 1 (source) must be moved to the load magazine 2 (target).

DB71.DBX0.1	=1	Interface 2 active	
DB71.DBX34.3	=1	Initiate positioning	(n+0)
DB71.DBW50	=9999	Magazine no. of load magazine	(n+16)
DB71.DBW52	=2	Location no. of load magazine	(n+18)

DB71.DBW54	=1	No. of magazine to be positioned	(n+20)
DB71.DBW56	=5	No. of location to be positioned	(n+22)

Parameters “OldToolMag” and “OldToolLoc” in FC 8 are not required for positioning because only the PLC requires the information for traversing the magazine. The PLC has to execute the positioning task and acknowledge it with an FC 8 call as follows:

Example of FC 8 call for positioning

FC 8 parameters	Values	Comment
Start		Starts task
TaskIdent	1	DB 71 interface
TaskIdentNo	2	No. of active interface
NewToolMag	0	During unloading = 0
NewToolLoc	0	During unloading = 0
OldToolMag	9999	Mag. no. 9999
OldToolLoc	2	Location no. 2
Status	5	Operation complete
Ready		Checkback from FC 8
Error		Checkback from FC 8

3.8 Tool monitoring (workpiece count, tool life, wear)

3.8.1 Monitoring types

Number of workpieces

The workpiece counter must count all the tool cutting edges that are used to produce a workpiece. If the number changes, the monitoring data of all tool cutting edges involved must be updated. It should be remembered that the machine may have several spindles and that different tool cutting edges can be used simultaneously.

Tool life

The tool life monitor operates only on the tool edge that is currently in use.currently in use. As soon as the path axes are moved (except for G00), the time monitoring data for the tools loaded in the toolholder or spindle are updated. If the monitoring time for a tool cutting edge is running during machining, the tool is blocked as a whole.

Wear

The wear monitoring function is available only if the “Tool monitoring” system has been enabled via machine data, in the same way as for time and wear monitoring. The wear monitoring must also be enabled via a machine data. The wear parameters of the cutting edge correspond to the local offsets (total offset parameters), see Subsection 3.9.4.

Tool life, workpiece count and wear

The monitoring type is defined for the tool when it is loaded. The setting of system variables MD $TC_TP9 can be changed any time subsequently to alter the monitoring type.

The tool management performs monitoring for tool life, workpiece count with prewarning limit and degree of wear, or sum offset monitoring.

All types of monitoring can be active for different tools in operation simultaneously. If values have been entered for several types of monitoring, all monitoring counters are decremented.

The monitoring counter that triggers the tool status change depends on the tool Parameter $TC_TP9 (= type of monitoring):

- $TC_TP9 = 0 –> No monitoring
- $TC_TP9 = 1 –> Time-monitored tool
- $TC_TP9 = 2 –> Workpiece-count-monitored tool
- $TC_TP9 = 4 –> Wear-monitored tool
- $TC_TP9 = 8 –> Sum offset

Several monitoring functions can be activated simultaneously for one tool (SW 5.1 and higher). The only mutually exclusive types are wear and sum offset.

If the monitoring criterion (tool life/workpiece count and wear) for a tool that is currently located in the spindle expires, it remains in use. Machining is not automatically interrupted to replace the tool with a replacement tool. The tool is not disabled until the next time it is selected. Since it is no longer “available”, a search is made for a replacement tool and the replacement tool is then loaded into the spindle. The tool change must be organized by the PLC or NC cycle.

The monitoring counters count from a set value > 0 down to zero. When a counter has decremented to a value of ≤ 0, the limit value is reached. When the cutting edge (of maximum 12) of a tool has reached its limit value, the whole tool is set to status “disabled”.

A **G** then appears next to the status for the tool in the magazine table.

Prewarning limit reached

When a cutting edge of a tool has reached its prewarning limit the whole tool is given the status “Prewarning limit reached”.

A **V** then appears next to the status for the tool in the magazine table.

A message is output simultaneously to inform the operator that a replacement tool may be required. If an operator action sets a monitoring counter from zero or the prewarning limit back to a value > 0 or > prewarning limit, the tool status changes automatically to reflect the change in the data. This allows the operator to abort a “disabled” status which has occurred as the result of a monitoring limit being reached.

If the tool has several cutting edges, all cutting edges must be outside the monitoring limits.

Tool monitoring alarms

When the prewarning limit or the monitoring limit of a tool is reached, one of the alarms 6010, 6011, 6012, 6013 (abort clear acknowledgement condition) is output for information.

When NC language command SETPIECE(...) (see Subsection 5.8.11) or PI command _N_TMPCIT (= change workpiece counter) is programmed, several tools may reach a limit value, causing several alarms to be generated).

No alarm is output if a limit value is reached as the result of data manipulation via Variable services.

Check monitoring status

During a program run, the programmed tool change command (e.g. “M06” for milling) can be used without a T call to check whether a monitor has responded. If it has, a replaced tool is sought and a tool change request output.

Enabling memory and function

At least bits 0 and 1 (3) must generally be set in machine data

- MD 18080: MM_TOOL_MANAGEMENT_MASK and
- MD 20310: TOOL_MANAGEMENT_MASK.

This prepares the memory for the monitoring data and enables the function.

Enabling tool life monitoring

To implement tool life monitoring, the spindle (toolholder) or spindles which require this type of monitoring must also be specified in channel-specific MD 20320: TOOL_TIME_MONITOR_MASK. This machine data is bit-coded.

Example: MD 20320: TOOL_TIME_MONITOR_MASK

• Value = 1	Spindle number 1 only
• Value = 2	Spindle number 2 only
• Value = 3	Spindle numbers 1 and 2 only

Refer to Chapter 8 for machine data.

3.8.2 Tool life monitoring

Monitoring of tool cutting edge

The tool life monitor operates only on the tool edge that is currently in use. The spindle (toolholder) must have been activated beforehand (MD 20320: TOOL_TIME_MONITOR_ MASK = spindle no.).

If MD 20124: TOOL_MANAGEMENT_TOOLHOLDER > 0, the toolholder number is selected in MD 20320 rather than the spindle no.

Tool life	The time is entered with 1 minute resolution up to SW 5.1 and can be entered on loading or set in the program with $TC_MOP2=500. The tool life is decremented internally in milliseconds and displayed in milliseconds. Data backup during unloading takes place in a matter milliseconds with SW 5.1 and higher.
Disabled	If the remaining tool life is ≤ 0, the tool is set to “disabled”. After the next tool change it is no longer used.
Monitoring from NCK	The residual tool life is decreased whenever one of the 3 path axes is traversed at machining feedrate (e.g. G01). G00 traversing blocks are not “counted”.
Monitoring from PLC	The user can start and stop the time monitor using PLC signal “Time monitor active” (DB 21DBX1.3). This method of control can be selected via machine data 20310.
Prewarning limit	Input when tool is loaded or via part program with $TC_MOP1=50. When the prewarning limit has been reached, the tool is assigned the status “Prewarning limit reached” (display in the magazine list).
Special case, limit values	The tool life of a tool expires while it is in use. If this disabled tool is reprogrammed by a tool change (e.g. M06 without T word) a check is made to see whether the monitoring time has expired. If yes, a replacement tool is used.

$A-MONIFACT factor

By entering a channel-specific factor which is set before a tool is used for the first time, it is possible to monitor the different degrees of tool wear resulting from machining different types of workpiece material. The value is multiplied by the current time unit before the time value of the cutting edge is decremented. The write operation is performed synchronously with the main run. For further information, see Subsection 5.8.24.

Start and stop the tool life decrementation

Tool life monitoring runs when geometry axes do not traverse with **G00** (default). The user can start and stop the time monitor using PLC signal “Time monitor active” (DB 21 DBX 1.3).

The active control mode is set via a machine data, i.e. MD 20310: TOOL_MANAGEMENT_MASK bit17. The default setting (bit17=0) means that traversing blocks other than G00 activate the time counter.

Time monitoring hierarchy

The combination of system parameter $A_MONIFACT and function ’Program testing active’ produces the following nested time monitoring structure:

Machine data MD 20310: TOOL_MANAGEMENT_MASK defines whether monitoring is controlled via G00 or a PLC signal. Tools on spindles activated in machine data MD 20320: TOOL_TIME_MONITOR_MASK are time-monitored.

The VDI signal “Program testing active” activates or deactivates the currently valid time monitoring function, i.e. ’Program testing active’ has higher priority than the current time monitoring.

When time monitoring is active, the real time (as defined by the internal clock) is multiplied by the factor $A_MONIFACT and the result subtracted from the current time count of a tool edge mounted on the spindle.

3.8.3 Workpiece count monitoring

Changing the number of workpieces

The number of workpieces can be changed:

- Operation at the MMC, HMI
- With a part program command (SETPIECE)
- PI service (TMPCIT) via PLC or MMC-OEM

Workpiece counter per spindle

Every spindle has a “memory” for the cutting edges used on it. With program command SETPIECE (1) the workpiece counter for the cutting edges that are used on the main spindle is decremented by 1. The workpiece counter of each spindle can be addressed individually.

The workpiece counter must count all the tools that are used to produce a workpiece. It should be remembered that the machine may have several spindles and that different tools can be used simultaneously.

If a tool is located on the main spindle with an offset number D>0 during a count, this is stored in the “memory” when the next block is loaded during the main run, and then included in the next count.

The cutting edge of a tool is only counted once per spindle.

The programmer of the part program who programs **SETPIECE** can program the parameter as a function of the material.

SETPIECE (factor * no. of workpieces)

Like the factor for time monitoring, this function allows a workpiece count that depends on the process, the workpiece material or other factors.

Workpiece counting can be deactivated via channel DB DBX29.5.

Monitoring from the NCK	When the workpiece counter has reached the prewarning limit this is displayed in the magazine list. When the workpiece count has reached zero the tool is disabled. The next time the tool is called, the replacement tool is inserted.
Set workpiece counter	Entered during loading or via part program with e.g. $TC_MOP4=500.
Decrement number of workpieces	The number of workpieces must be decremented at the relevant point in the part program with the NC command SETPIECE (x, y) (e.g. SETPIECE(1) –> workpiece counter for main spindle tools is decremented by 1). The function for updating the workpiece count is activated via PI command from the PLC program.
Disabled	When the workpiece count has reached zero the tool is disabled.

Prewarning limit	Entered during loading or via part program with e.g. $TC_MOP3=50. When the prewarning limit has been reached, the tool is assigned the status “Prewarning limit reached” (display in the magazine list).
Special case, limit values	It is not possible to activate a workpiece count for any number of cutting edges simultaneously! If the monitoring function has been enabled and activated in the MD, all spindles can simultaneously monitor the cutting edges for the number of workpieces = “Number of cutting edges in the TO area” (=MD). An edge of a tool is only counted once per spindle.

3.8.4 Wear monitoring (SW 5.1 and later)

The wear monitoring function is available only if the “Tool monitoring” system has been enabled (via machine data, see Subsection 8.1.2).

The wear monitor must also be enabled via machine data (MD 18080: MM_TOOL_MANAGEMENT_MASK; bit5).

Definition

$TC_TP9 = 4; Wear monitoring is active for the tool.

$TC_TP9 = 8 can be set to select the “Sum offset” monitoring function if this is required. For bit assignments, see Section 5.3.

$TC_TP9 = 4

The wear parameters for a tool edge are defined with system parameters $TC_DP12, ..., $TC_DP20.
These are assigned directly to the edge geometry values
TC_DP3, ... , $TC_DP11.

$TC_DP10 and $TC_DP11 describe “angles”. The other parameters stand for the tool edge lengths and radii.

Only these values are included in the monitoring, i.e. wear parameters $TC_DP19 and $TC_DP20, which are analogous to parameters $TC_DP10 and $TC_DP11, are not taken into account. For bit assignments, see Subsection 5.2.1.

Note

Wear monitoring does not monitor each individual value but only the highest value of the maximum seven wear parameters ($TC_DP12, ..., $TC_DP18).

$TC_TP9 = 8

Equivalents of the cutting edge wear parameters can be found in the **sum offset parameters**.

Analogous to the wear parameters, the following parameters for other local offsets of the tool edge are monitored:

- $TC_SCP13, ... $TC_SCP21
  First sum offset of cutting edge (if defined)
- $TC_SCP23, ... $TC_SCP31
  Second sum offset of cutting edge (if defined)
- $TC_SCP33, ... $TC_SCP41
  Third sum offset of cutting edge (if defined), etc.

Wear monitoring does not monitor each individual value but only the highest value of the maximum seven sum offset parameters.

($TC_SCP13, ..., $TC_SCP21, $TC_SCP23, ..., $TC_SCP31, ...)

Most tool geometries are described by a subset of the named data records.

If a parameter is changed (written), the NCK then checks whether the new value is higher than any of the other parameters and, if necessary, this value is subtracted from the wear setpoint. The result is the new wear actual value.

The actual wear runs from the positive setpoint towards zero like the other monitoring values.

Monitoring parameters (SW 5.1 and later)

• $TC_MOP15	Wear setpoint or sum offset value
• $TC_MOP5	Wear prewarning limit or sum offset prewarning limit
• $TC_MOP6	Actual wear value or actual sum offset

The physical quantity of the new monitoring parameters is “Length”. The unit is the same as for the wear values.
Wear monitoring can be deactivated via the channel DB DBX29.6.

The signal acts only on changes to the wear data that occur in NC program processing. If these data are changed via the OPI (e.g. MMC operation), the PLC signal is suppressed.

3.8.5 Signals to and from the PLC (SW 5.1 and later)

Previously, an alarm message was output as soon as the prewarning limit or limit value was reached. Alarms **6410** and **6411** are output when the prewarning limit is reached and **6412** and **6413** when the limit value is reached. Alarms 641^0 and 6412 are triggered via the OP interface and alarms 6411 and 6413 via the NC program. The alarm texts identify the affected tool via the tool ID, duplo number and (as of SW 5.3) D number.

With effect from SW 5.1, the following information is returned to the channel interface for one DB1 cycle (internal T numbers):

- Prewarning limit reached (channel DB.DBD348)
- Limit value reached (channel DB.DBD352)

This information is indicated by a strobe signal in channel DB.DBB344.

The data are written by the NCK as soon as the alarm in question occurs.

VDI signal “T number of new replacement tool” – Channel DB21–DB30

If a tool is **switched** to “active” status when it is found during a tool change search in the NCK, this status change is interpreted as “initial selection of the replacement tool”.

This process status change is output to the PLC via the T number of the replacement tool.

An operator action which alters the tool status does not change the signal.

VDI signal “Last replacement tool in tool group”, channel DB.DBD360

If a tool is found as the result of a tool search in the NCK during a tool change, and no other replacement tool for the programmed spindle/toolholder is available at this point, this is interpreted as “last replacement tool in tool group”.

This process status change is output to the PLC via the T number of the replacement tool.

An operator action which alters the tool status does not change the signal.

Note

This function increases the main processor time requirement in the NCK at the point of tool selection for tool groups with several tools.

The following function must also be enabled via **MD 20310: TOOL_MANAGEMENT_MASK**. It is activated by setting **bit18=1**.

Disabling monitored tools – PLC-controlled via VDI signal

In earlier versions, a tool has assumed the status “disabled” as soon as the actual value of the active monitoring function reaches the value zero. A tool currently performing a machining operation but which is set to “disabled” remains in operation until the next tool change is performed. After that the tool can no longer be used.

With SW 5.1 and later, the PLC can also determine when a disabled tool can no longer be used, i.e. when the “disabled” status is taken into account in the tool search.

- When **VDI signal “Do not disable tool”** is set to 1
(channel DB. DBX29.7 = 1) the NCK does not take the tool status “disabled” into account during the tool search.
- When **VDI signal “Do not disable tool”** is set to 0
(channel DB. DBX29.7 = 0) the NCK does take the tool status “disabled” into account during the tool search.

The bit is channel-specific.

“Search for active tool” strategy

This search strategy can ensure that a machining operation is not performed with different tools from the same tool group.

When the tool is disabled, a monitoring function and the set VDI signal “Do not disable tool” ensure that the status “active” is **not** canceled.

This **tool is therefore assigned** the states “**active**” and “**disabled**”.

If the required machining operation is terminated without a tool change, the status of all disabled tools must be checked. A new PI service (_N_TMRASS, in PLC TMRASS, see Subsection 5.12.5) is available for this purpose; this allows you to cancel the “active” status for all disabled tools (e.g. by PLC program at end of program).

Other tool search strategies

A disabled tool can also be used for the other tool search strategies if the VDI signal “Do not disable tool” (channel DB. DBX29.7 = 1) is set. The tool selected solely depends on the search strategy.

This **search strategy** therefore **has priority** over **VDI signal “Tool disable ineffective”** when tools are selected. Both the last tool to be disabled or any other disabled tool can be selected.

Another tool which is not disabled might also exist, but is not selected because of the search strategy!

TO unit active in several channels

If a TO unit is assigned to several channels (tool and magazine data are “visible” in several channels), the setting of the channel-specific VDI signal “Tool disable” is active in every channel.

3.8.6 Monitoring data for setpoints (SW 5.1 and later)

In earlier SW versions, monitoring data have included the actual value and the prewarning limit for the variables to be monitored.

When the actual value reaches the value zero, the tool is disabled. Until now, the original starting value of the actual value has no longer been available in the NCK.

As from SW 5.1, this value is now available in the NCK, i.e. every monitored value receives a new data item – the setpoint. The setpoint is defined as a system parameter and OPI variable (TS).

$TC_MOP11

$TC_MOP11 is the time setpoint
($TC_MOP1=prewarning limit time)

$TC_MOP13

$TC_MOP13 is the workpiece setpoint value
($TC_MOP3=prewarning workpiece count)

Reset to setpoints

Resetting the actual values of wear and sum offset “fine” means that all the parameters for wear and sum offset used for monitoring are set to zero.

Boundary conditions for tool monitoring (SW 5.1 and later)

New system parameters are being defined. This means that for the same number of cutting edges more non-volatile memory is used in the NCK than in SW 4.

The monitoring function “Wear monitoring” must be enabled via a machine data. The default value is “not active” so that no additional memory is needed (corresponds to more than 20 KB non-volatile memory for 1000 cutting edges).

Activation (SW 5.1 and later)

The monitoring function must be enabled via machine data
MD 18080: MM_TOOL_MANAGEMENT_MASK.

Tools can be individually named for the different defined monitoring functions (time, workpiece count, wear, sum offset).

Wear monitoring is performed automatically by the NCK when the user changes the cutting edge offsets.

Control behavior (SW 5.1 and later)

Control behavior on Power On, Mode group change, Reset, Block search and REPOS is described below.

The VDI signal ’Activate program test’ has no effect on wear monitoring as new wear values are only entered during machining and not while a program is being tested (if wear values are not altered by the machining program itself).

3.9 Variants of D number assignments

D numbers can be organized in a number of different ways in the NCK:

3.9.1 Relative D no. for each T – default

D numbers ranging from 1 to max. 12 are available for every T = “identifier” (with TM) or for every T number (without TM). These D numbers are directly assigned to tool cutting edges.
Every D number = tool edge number has its own offset data record ($TC_DPx[t, d]).
D0 is the offset deselection code.

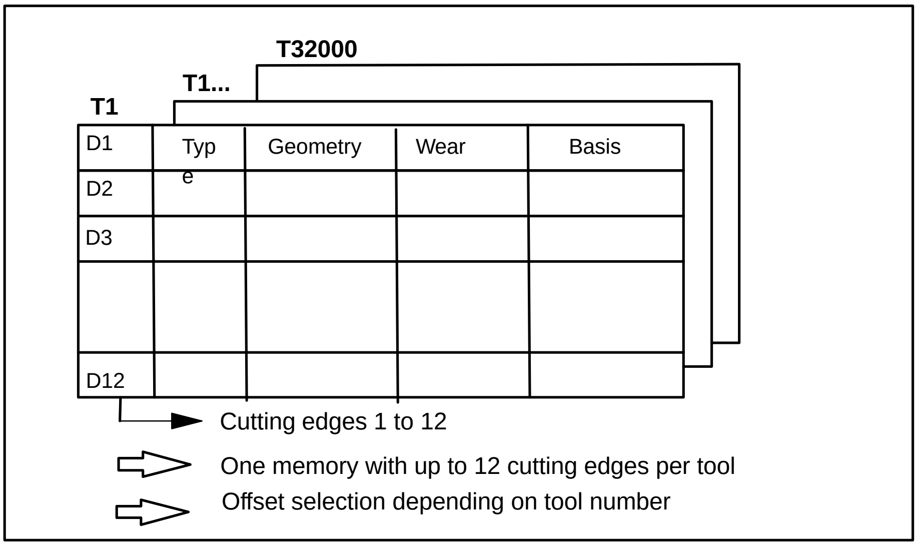

T32000
T1...
T1
D1
Typ
e
Geometry
Wear
Basis
D2
D3
D12
Cutting edges 1 to 12
One memory with up to 12 cutting edges per tool
Offset selection depending on tool number

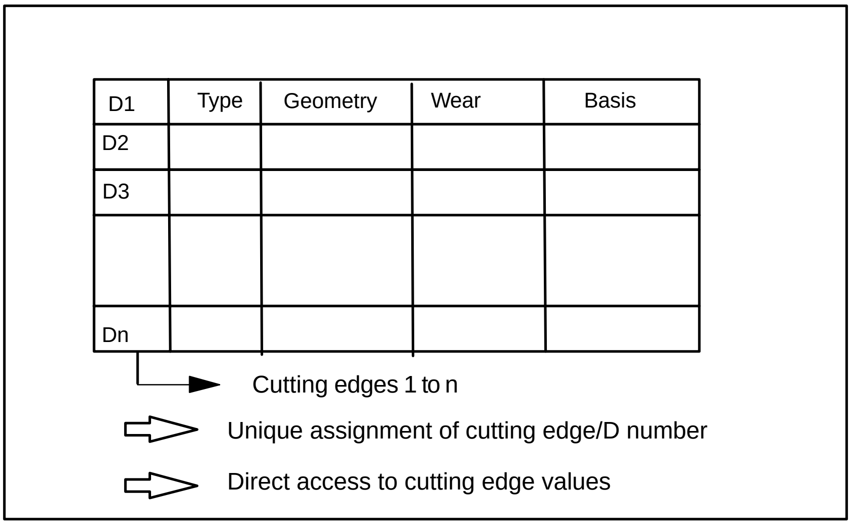
Fig. 3-17 Layout of the tool offset memory

3.9.2 Absolute D no. without reference to T number (flat D no.)

In systems without tool management, it is possible to select mutual independence of D and T numbers as an alternative to Subsection 3.9.1.
The user defines the relationship between T number, cutting edge and offset via a D number.
The D numbers range from 1 to 32000. D0 is the offset deselection code.

Note

With this type of tool offset, the T number is always output to the PLC with extended address (= spindle or toolholder no.).



D1	Type	Geometry	Wear	Basis
D2				
D3				
Dn				

Cutting edges 1 to n

Unique assignment of cutting edge/D number

Direct access to cutting edge values

Fig. 3-18 Layout of the tool offset memory

3.9.3 Free selection of D numbers for every T

In SW 5.1 and later, D numbers can be freely assigned to tool edge numbers in systems with and without TM. As described in Subsection 3.9.1, a tool “T” can have a maximum of 12 edges. The upper limit of the usable D numbers is limited by a machine data.
This assignment option is an extension of the process described in Subsection 3.9.1.
This setting makes additional program commands available; these can be used to check for unique assignment of D numbers to T numbers or identifiers.
The same D numbers are assigned to the edges of duplo tools (identical identifier).

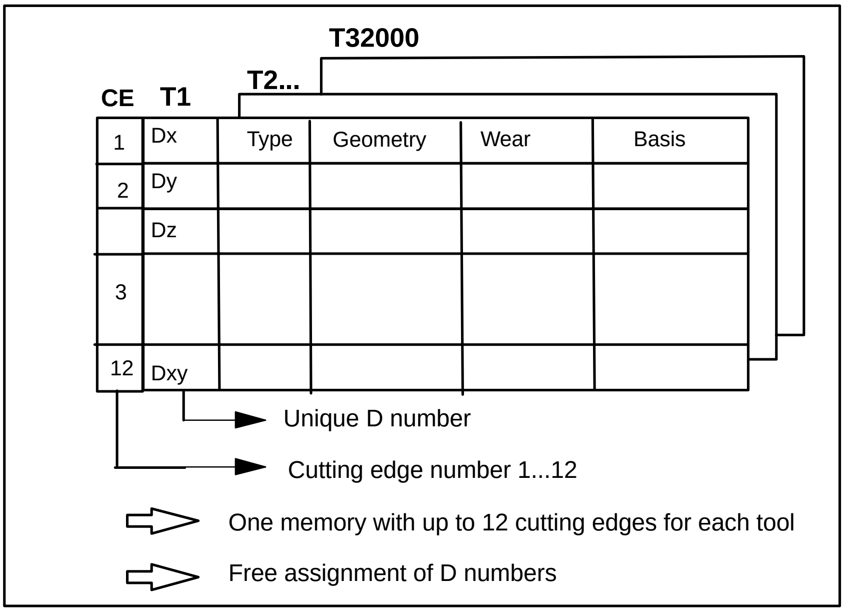

T32000
T2...
CE
T1
1
2
3
12
Dx
Dy
Dz
Dxy
Type
Geometry
Wear
Basis
Unique D number
Cutting edge number 1...12
One memory with up to 12 cutting edges for each tool
Free assignment of D numbers

Fig. 3-19 Layout of the tool offset memory

Note

Universal system support (tool cabinet, code carrier) is not available for this function.

Machine data for free (unique) assignment of D numbers

$MN_MAX_CUTTING_EDGE_NO= Maximum permissible D number

Example:

$MN_MAX_CUTTING_EDGE_NO=1 A maximum of 1 offset (D1) can be defined per tool.

$MN_MAX_CUTTING_EDGE_NO=9999 Allows unique D numbers to be assigned to tools:
T1 with D1, D2, D3
T2 with D10, D20, D30
T3 with D100, D200, D300

$MN_MAX_CUTTING_EDGE_PER_TOOL= Assignment of tool edges per tool

Example:

$MN_MAX_CUTTING_EDGE_PER_TOOL=1 Only for tools with 1 cutting edge

$MN_MAX_CUTTING_EDGE_PER_TOOL=12 Up to 12 cutting edges per tool (previously 9).

Unique use check (CHKDNO)

D numbers assigned within the NCK are checked for unique use with NC language command **CHKDNO**. The D numbers of all tools defined within a TO unit may be used only once. This does not include replacement tools. See also Subsection 5.8.1.

Check within the magazine (CHKDM)

Like CHKNO, the NC language command **CHKDM** checks the D numbers with the NCK for unique use when the TM is active. This check function can be restricted to individual magazines. See also Subsection 5.8.2.

D number to T number (GETACTTD)

NC language command **GETACTTD** allows the D number of the active tool in the tool group to be found using the T number when the TM system is active.
This option can be used only if the D numbers in the TO unit in question are uniquely assigned. See also Subsection 5.8.3.

GETDNO, SETDNO during setup

NC language commands **GETDNO** and **SETDNO** allow the offset number D for a specific cutting edge number CE to be read and written.

GETDNO (T, CE) : Read D number for cutting edge CE of tool T

SETDNO (T, CE, D): Set D number for cutting edge CE of tool T

$TC_DPCE[T, D]=...: Assign cutting edge number CE to offset number D

Example:

Rename cutting edge CE=3 from D2 to **D17**

– In the following initial situation:
Internal T number: 1
D number: 2
Tool 1 cutting edge with:

```
$TC_DP2[1, 2]=120          ;Tool length T1, D2: 120 mm
$TC_DP3[1, 2]=5.5          ;Tool radius T1, D2: 5.5 mm
$TC_DPCE[1, 2]=3           ;Cutting edge number T1, D2: 3
(programming: T1,...D2)
```

– variable definition:

```
DEF INT DNoOld, DNoNew=17
DnOld=GETDNO (1, 3)        ;Value 2 is read in DnOld
SETDNO (1 ,3, DNoNew)      ;New D no. is assigned to the cutting edge
```

– is used to assign new D value 17 to edge CE=3
  $TC_DP2[1,**17**]=120
  $TC_DP3[1, **17**]=5.5
  $TC_DPCE[1, **17**]=3

3.9.4 Local offsets (sum offsets)

Local offsets are a generalized form of wear. They are an integral component of the tool cutting edge data. The sum offset parameters refer to the geometric data of a cutting edge.

Local offsets can be used generally, i.e. with active/inactive TM; with flat D number function.

To meet the requirements of special machine operating modes, the relevant machine data can be set to divide the local offsets into the following categories:

– Local fine offsets

– Local coarse offsets = setup offset

The purpose of the setup offset is to allow the operator to set values prior to the machining operation. These values are stored in their own memory in the NCK, the operator can access the local fine offsets via the MMC. “Fine” and “Coarse” local offsets are added internally in the NCK and then applied like the sum offset itself.

Several local offsets can be defined for each D number. Machine data define the absolute number of local offsets, the maximum number of localoffsets per cutting edge and specify which additive offsets are active after the end of program or when the RESET key is pressed.

Applicable only when TM system is active:

Machine data 18104 can be set to define which sum offset must be operative if a tool is assigned the “active” status in the part program in the coarse of a programmed tool change:

– “Fine” tool offset values of tool cutting edges remain unchanged or

– “Fine” tool offset values of tool cutting edges are set to “0”.

The function is enabled by setting bit 8 = 1 in machine data $MN_MM_TOOL_MANAGEMENT_MASK.

DL programming of total/setup offset

The sum offset is always programmed relative to the active D number by means of programming command
DL =“n”

This activates the sum offset with relative number “n” with reference to the active D number, i.e. the sum offset “n” is added to the wear of the active D number.

The sum offset is deselected with command
DL = 0

Configuration of total/setup offset

$MN_MM_KIND_OF_SUMCORR, bit 4=0

Equals default setting, i.e. only one sum offset data record per DL number. In this case, the term “sum offset” merely refers to the data represented by $TC_SCPx.

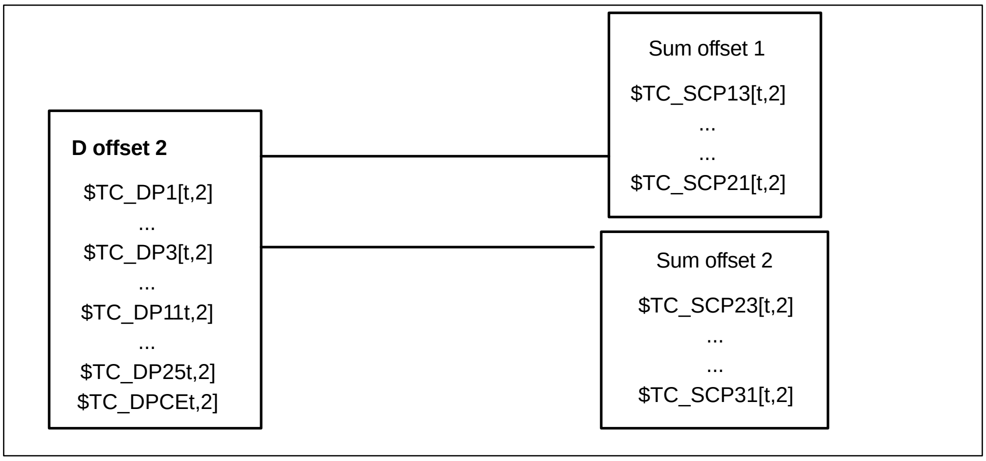


Fig. 3-20 $MN_MM_KIND_OF_SUMCORR, bit 4=0

The following is programmed with the data in Fig. 3-20 (tool with T=t is active in this example):

```
D2          ;Cutting edge offsets

            ;i.e. $TC_DP3,...$TC_DP11 + wear ($TC_DP12,...$DP29)
            + adapter dimension

            ...
DL=1        ;Sum offset 1 is added to the existing offsets of D2
            ;i.e. $TC_SCP13,...$TC_SCP21

            ...
DL=2        ;Sum offset 1 is no longer added to offset D2,
            but sum offset 2 instead
            ;i.e.  $TC_SCP23,...$TC_SCP31

            ...
DL=0        ;Deselection of sum offset; only the data of D2 are still
            effective
```

$MN_MMKIND_OF_SUMCORR, bit 4=1

Setup offsets are available. The general term “sum offset” refers to a combination of the “fine” sum offsets, represented by $TC_SCPx, and the sum offset, represented by $TC_ECPx. There are two data records for each DL number. The sum offset equals the product of the corresponding components $TC_SCPx + $TC_ECPx.

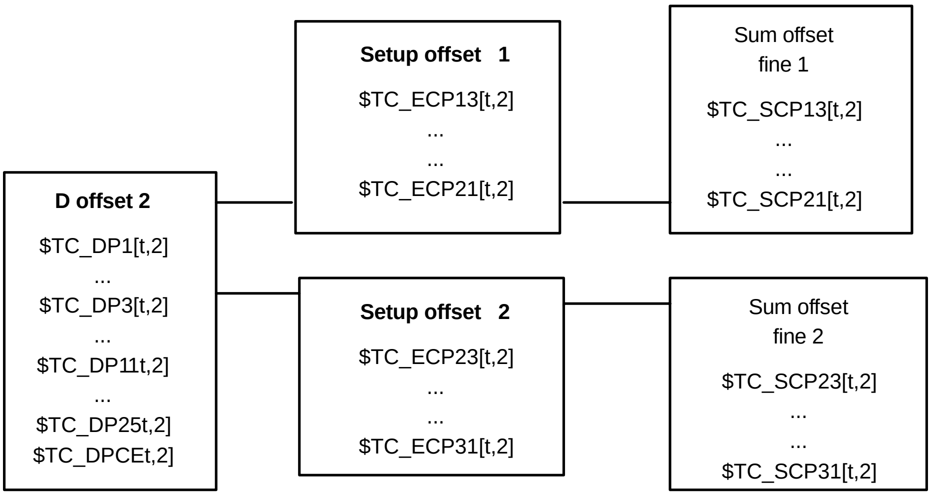


Fig. 3-21 $MN_MM_KIND_OF_SUMCORR, bit 4=1

The following is programmed with the data in Fig. 3-21 (the tool with T=t is active in this example):

```
D2              ;Cutting edge offsets
                ;i.e. $TC_DP3,...$TC_DP11 + wear ($TC_DP12,...$DP29)
                + adapter dimension

                ...
DL=1            ;Sum offset 1 is added to the existing offsets of D2
                ;i.e. $TC_ECP13 + $TC_SCP13 ,...$TC_ECP21 + $TC_SCP21

                ...
DL=2            ;Sum offset 1 is not longer added to offset D2,
                but sum offset 2 instead
                ;i.e.  $TC_ECP23 + $TC_SCP23,... $TC_ECP31 + $TC_SCP31

                ...
DL=0            ;Deselection of sum offset; only the data of D2 are still
                effective
```

The new NC language command DELDL can be used to delete local offsets from cutting edges (see Subsection 5.8.7).

3.10 Adapter data (SW 5.1 and later)

The standard data record for tool offsets contains parameters **$TC_DP21, $TC_DP22** and **$TC_DP23** via which dimensions (length1, length2 and length3) of an adapter can be entered. These data are defined offset specifically.

Application

If tool management is active the additional adapter data can also be assigned to specific magazine locations.

This function is used for adapters that are fixed to a magazine location for a long period and used by different types of tool.

In individual cases, it is also possible to use identical adapters on several magazine locations. To do this it makes sense to define and store the adapter data records separately from the magazine locations.

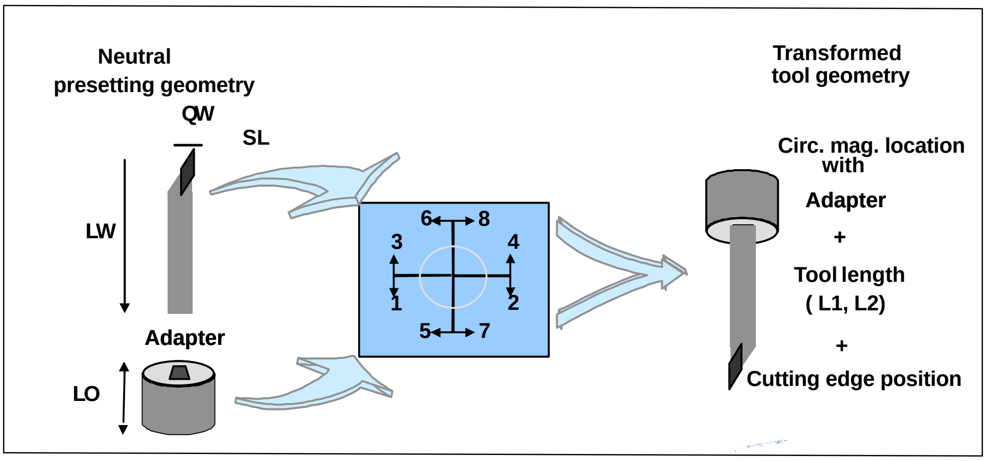


Fig. 3-22 Adapter transformation

Adapter transformation

Adapter data “adapter transformation” (SW 5.1 and later) allows fixed orientation of the tool on the adapter or orientation of the adapter including its tool with reference to the machine.

This function can be used as an alternative to the previous one. If adapter data are used, parameters $TC_DP21, $TC_DP22 and $TC_DP23 have a different reference and are therefore only formally part of the cutting edge data record in the NCK.

3.10.1 Description of function

The adapter data function must be enabled via machine data (MD18104: MN_MM_NUM_TOOL_ADAPTER).

Bit 7 must be set in MD 18080: MM_TOOL_MANAGEMENT_MASK to activate the setting.

Definitions

Two types of definition can be set in the machine data for adapter data:

- One adapter data record is assigned to each magazine location as standard.
- Adapter data records can be defined independently of magazine locations. The magazine locations are then assigned as an additional step.

The magazine location is the reference point for adapter **and** tool. Both are assigned to the magazine location.

The following points must be included when programming D numbers in the part program:

- The offset must be assigned to a real tool.
- The tool is assigned to a magazine location.
- It is possible to assign an adapter to the magazine location, for which a transformation (orientation) of the tool it contains can be defined.

This makes it possible to calculate the work offset uniquely and correct the tool path.

If a sum offset is programmed, their values refer to the active D offset.

3.10.2 Activation

Requirements

- In order to use the magazine-location-oriented data, machine data **MD 18104: MM_NUM_TOOL_ADAPTER** must be set to a value other than zero.
- Adapter data records must be defined.
- If the values of the machine data are greater than zero the adapters must be linked to the magazine locations or assigned to them (can be automated via the MMC or using a cycle).

As a result, the adapter data including the defined transformations are always taken into account for the tool located on the magazine location in question. The work offset is calculated including the transformation and the adapter data.

The offset data can then be displayed as follows:

- Geometry values of the tool (parameters $TC_DP3,...DP11); identified as neutral default geometry
- Non-transformed work offset (transformation of the total of the values of tool geometry, wear, sum offset, base dimensions and adapter)
- Transformed work offset (total of the values of tool geometry, wear, sum offset, base dimensions of adapter).

The quantities to be transformed can be selected via machine data. The mode of transformation of the sum offset can be set.

Magazine-location-related adapter data records

Create new

MM_NUM_TOOL_ADAPTER = –1:

One magazine location and one adapter data record are created. The specified values are put into the adapter data record which is automatically linked to the magazine location.

It is not possible to create a new free adapter at this point. The adapter numbers are assigned automatically (1 ... max. number of available magazine locations).

Delete

If an adapter data record is linked to a magazine location (MM_NUM_TOOL_ADAPTER = –1), it cannot be deleted.

User adapter data records

Create new

MM_NUM_TOOL_ADAPTER > 0:

Adapter data can be created by the user with a write operation to a non-existent data record.

$TC_ADPTi[n] = value; i = T, 1 2, 3, ..., n (number of the adapter)

If data record n does not yet exist and the maximum number of adapter data records that have already been defined is less than the value of MD 18104: **MM_NUM_TOOL_ADAPTER**, a new adapter data record is created and assigned the specified value.

The value “value” is assigned to parameter i. Parameterizing rule: 0<n≤32000. Index value 0 is reserved.

Note

The adapters must be assigned explicitly to the magazine locations if MM_NUM_TOOL_ADAPTER > 0.

Delete

If MD 18104: MM_NUM_TOOL_ADAPTER is set to a value of > 0, the adapter data can be deleted as required provided that they are not assigned to a magazine location.

```
$TC_ADPTT[n] = –1
```

Adapter data record n is deleted and the memory “freed”.

Deleting an assigned adapter data record:

The assignment to the magazine location must be undone first. You can only do this if the magazine location is empty. An alarm is output if deletion fails.

Proceed as follows:

- Remove the tool from the magazine location (unload, re-store).
- Remove the adapter from the magazine location.
- Delete the adapter data record (with $TC_ADPTTi[n] = –1).

Adapter data record n is deleted and the memory “freed”.

Deleting all adapter data records

If **MM_NUM_TOOL_ADAPTER > 0** you can delete the adapter data if they are not assigned to a magazine location:

```
        $TC_ADPTT[0] = –1
```

All unassigned adapter data of the TO unit are deleted. If you want to delete assigned adapters, you must first undo the assignment of those adapters to magazine locations. An alarm is output if deletion fails.

Read/write adapter data

You can modify adapter data whenever you want to even if that adapter is assigned to a magazine location and/or a tool is located in the magazine location with the adapter.

Magazine location assignment/decoupling

If **MM_NUM_TOOL_ADAPTER > 0** an adapter record must be assigned to a magazine location explicitly:

```
$TC_MPP7[m,p] = "adapterno."
```

Adapter number "adapterno." is assigned to magazine location p of magazine m. An existing assignment is undone if 'adapterno.' = 0.

Note

Assignment/decoupling is only possible if there is no tool in the magazine location.

Example of an Adapter transformation

A turning tool with lengths L and Q is described below.

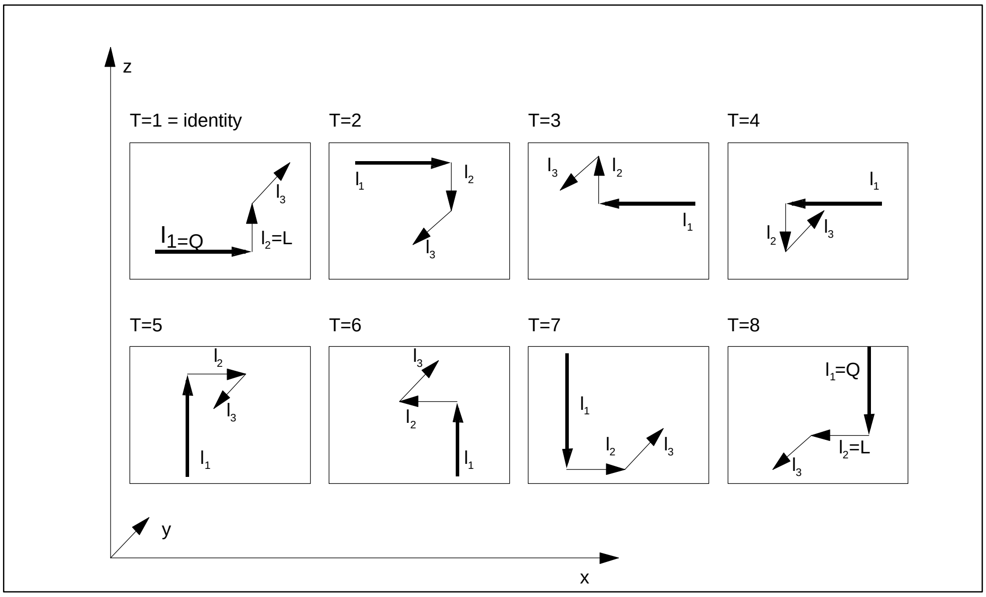


Fig. 3-23 The 8 defined transformations (T = 1...8) for the adapter with G 18 and for a turning tool. The diagram shows the assignments of tool lengths l_1, l_2 and l_3 to geometry axes x, y and z.

Transformations for numbers 1 to 8 are defined. Number 1 is the identity and not the transformation of the input data.

Other transformations can be implemented. The available transformations are designed initially for turning tools. These are typically defined by **Q=l_1=$TC_DP3** and **L=$l_2$=$TC_DP4** .

The transformation numbers correspond to the transformations given in the table. As a rule:

Length1$_t$, length2$_t$, length3$_t$ = f(length, length2, length3) = f(l_1,l_2,l_3) = f(Q,L,l_3)

Transformation number	Length1$_t$ length2$_t$ length3$_t$ transformed values			Transformation with ref. to plane G18
1	$+l_1$	$+l_3$	$+l_2$	—
2	$+l_1$	$–l_3$	$–l_1$	180° about x
3	$–l_1$	$+l_3$	$–l_2$	180° about z
3	$–l_1$	$+l_3$	$–l_2$	180° about z
4	$–l_1$	$–l_3$	$+l_2$	180° about x, z
5	$+l_3$	$+l_1$	$–l_2$	90° about y, 180° about z
6	$+l_3$	$–l_1$	$+l_2$	90° about y
7	$–l_3$	$+l_1$	$+l_2$	–90° about y
8	$–l_3$	$–l_1$	$–l_2$	–90° about y, 180° about z

l_1, l_2 and l_3 are working offsets of the tool prior to transformation with or without adapter (depending on machine data settings). They are assigned to the geometry axes during compensation.

Note

In turning, L and Q are also used to describe a tool. In the above table, l_1 corresponds, for example, to variable Q (or x direction) and l_2 to variable L (or z direction), assuming the plane G18 is selected (default setting for turning machines).

As a standard, activation of an offset is calculated as follows:

Offset = D offset + x_i (e.g. wear, sum offset)

```
Length1  =    $TC_DP3 + x_i
Length2  =    $TC_DP4 + x_{i+1}
Length3  =    $TC_DP5 + x_{i+2}
Radius1  =    $TC_DP6 + x_{i+3}
```

The adapter transformation then acts on the transformed tool offset values and is added to the transformed offset values.

The transformation number of the adapter causes a transformation of the tool (the cutting edges) located in this adapter (orientation according to the transformation number).

Work offset = f(offset)+adapter dimensions of magazine location

```
aLength1   =   Length1_t + $TC_ADPT1
aLength2   =   Length2_t + $TC_ADPT2
aLength3   =   Length3_t + $TC_ADPT3
aRadius1   =   Radius1
```

Depending on the programmed plane selection G17, G18, G19, these values are added to the geometry axes.

G17, G18, G19 – plane selection (declarations)

The following declarations apply to the assignment between the tool length parameter of the tools and the geometry axes (different for turning and milling tools):

Machining plane	**System parameters for tool length definition**		
	$TC_DP3($l_1$)	**$TC_DP4($l_2$)**	**$TC_DP5($l_3$)**
G17 Milling Turning	Z Y	Y X	X Z
G18 Milling **Turning**	Y **X**	X **Z**	Z **Y**
G19 Milling Turning	X Z	Z Y	Y X

Transformation of tool point direction

The tool point direction described by system parameter $TC_DP2 is also transformed.

Transformations for the tool point direction are performed as shown in the table below:

Transformation number	**Cutting edge position**								
	1	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	1	2	3	4	5	6	7	8	9
2	2	1	4	5	7	6	5	8	9
3	4	3	2	1	5	8	7	6	9
4	3	4	1	2	7	8	5	6	9
5	1	4	3	2	6	5	8	7	9
6	4	1	2	3	8	5	6	7	9
7	2	3	4	1	6	7	8	5	9
8	3	2	1	4	8	7	6	5	9

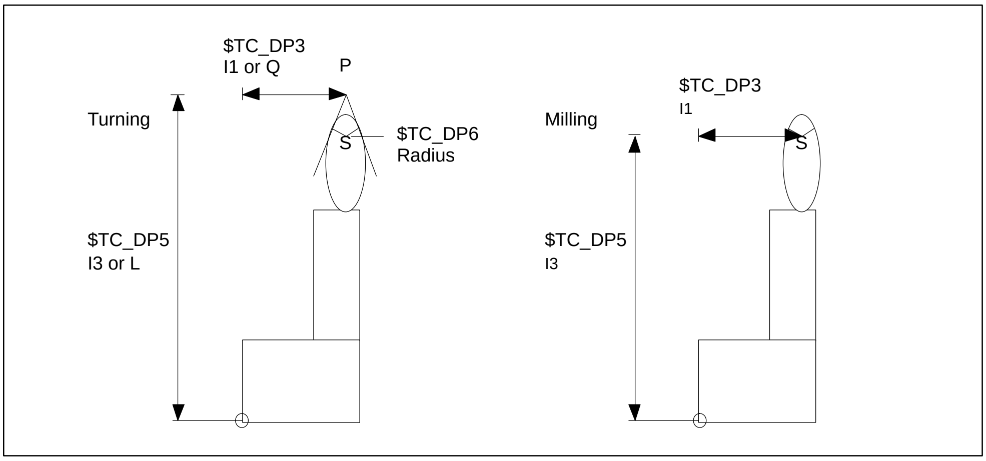


Fig. 3-24 Turning and milling tools – correlation between tool point direction and radius compensation

Turning tool geometries (I_1, I_3 or L, Q) are described with reference to point of contact P on the workpiece. However, the center point of cutting edge S with reference to the tool nose radius must be known for radius compensation.

This center point can only be precisely calculated if the tool point direction is known. Point S can thus be derived from point P.

The position of the tool in the workpiece coordinate system is described via the tool point direction (values 1 ... 8). Tool point direction 9 corresponds to S = P.

Note

The tool point direction is only used for turning tools because their geometry is described with reference to P and not with reference to S as is the case for milling tools.

Parameters $TC_DP21 ... 23 and $TC_ADPT

If the function “adapter” is active there are no more edge-specific data for the “base adapter dimension”.

In order to keep cycles that operate with adapters compatible, the following rules are laid down:

If a tool is positioned in a magazine location with an adapter and access to the adapter data is given via system parameters $TC_DP21...23, the adapter parameters of the location can be read and written.

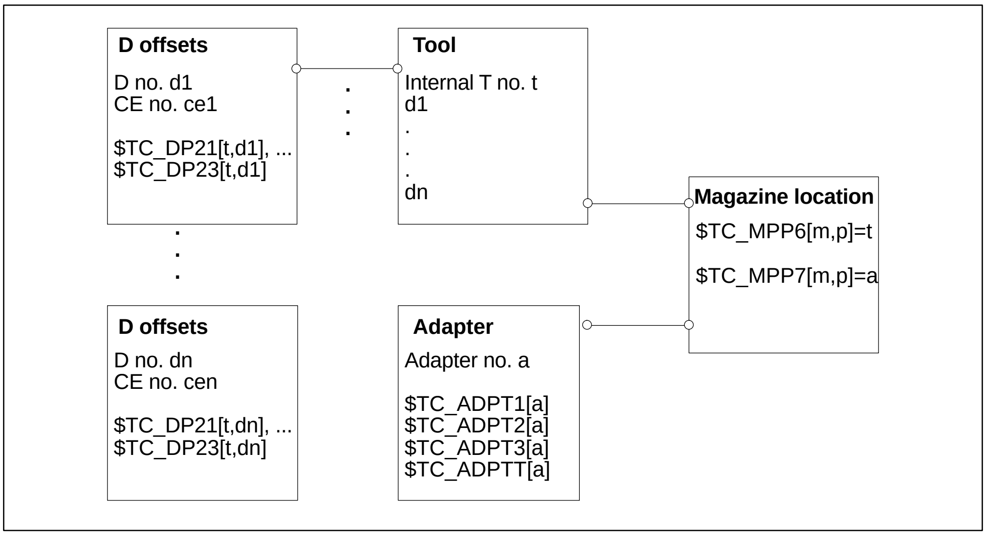


Fig. 3-25 $TC-DP21, ...23 – contents when “Adapter” function is active

Requirement:

- Tool t
- Magazine location p
- Magazine m
- Adapter a
- Tool with D offsets d_1, ... d_n

The adapter is assigned to the magazine location. If, for example, parameter $TC_DP21[t,$d_1$] is read or written in the part program, then parameter $TC_ADPT1[a] of the adapter is actually accessed by the program, i.e. the same machine data is accessed for all d_1, ... d_n.

If the assignment of the tool to the magazine location is released or the adapter is removed from the magazine location, no more data can be assigned to the parameters. A read operation returns the value 0, a write operation does not change the data (nor does it generate an alarm).

Transformed and non-transformed offset values

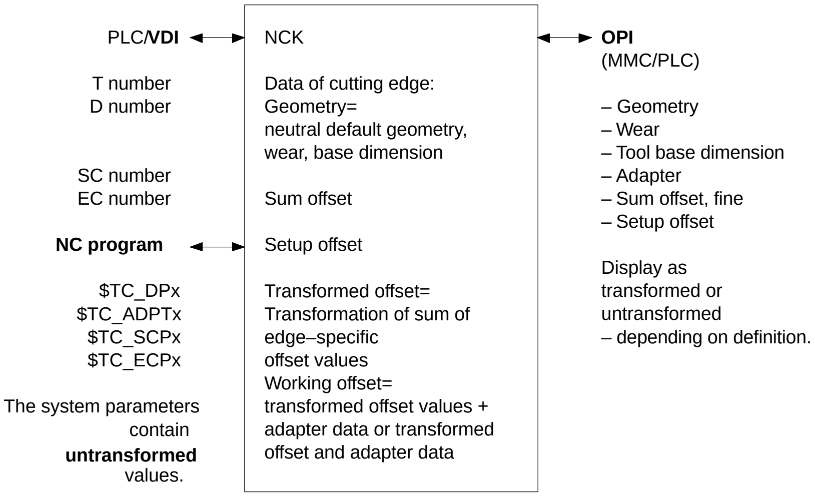
The values included in the path offset are usually the transformed work offsets.

It can generally be said that the data that describe a tool are subject to transformation. The transformation of the adapter is communicated to the tool (orientation in which it is positioned in the adapter). The adapter data themselves are not transformed.

Data transfer to the NCK

You must decide how you want to transfer the data to the NCK.

- You can transfer the data via the part program by programming the system parameters $TC_... They are defined as non-transformed values.
- You can transfer the data via the OPI interface using variable services. In this case, the data can be transferred either as transformed or non-transformed values.



PLC/**VDI**

T number
D number

SC number
EC number

NC program

$TC_DPx
$TC_ADPTx
$TC_SCPx
$TC_ECPx

The system parameters contain **untransformed** values.

NCK

Data of cutting edge:
Geometry=
neutral default geometry,
wear, base dimension

Sum offset

Setup offset

Transformed offset=
Transformation of sum of
edge–specific
offset values
Working offset=
transformed offset values +
adapter data or transformed
offset and adapter data

OPI
(MMC/PLC)

– Geometry
– Wear
– Tool base dimension
– Adapter
– Sum offset, fine
– Setup offset

Display as
transformed or
untransformed
– depending on definition.

Fig. 3-26 Geometry of a tool edge and applied offsets

Conditions

When using the function (magazine-location-oriented) “adapter data” the user must ensure that the old data records of all the data records with edge-specific adapter data are adapted to the requirements of the new function.

However, using the edge-specific adapter parameter definition described ($TC_DP21,...23), it is ensured that all old data are converted to the adapter data function by the NCK.

Edge-specific data “Base/Adapter dimension” do not exist with respect to the “Adapter data” function. These data are of no significance if the adapter is defined magazine location specifically.

The function “adapter data” is better suited to the applications of an adapter because it defines the adapter as part of the magazine location and not as part of the tool or cutting edge.

Examples for assigning adapter data

Example 1

Requirement:

- MM_NUM_TOOL_ADAPTER = –1
- MM_NUM_MAGAZIN_LOCATION = 20
- One chain with 16 locations, magazine number = 1
- Two grippers
- One spindle
- One loading/unloading point.
- Assignment

When creating the 20 locations in all, 20 adapters should be assigned, i.e. exactly one adapter assigned to each location.

Note

It does not matter if the real locations are not actually fitted with an adapter. Preassigned adapter data have no effect on the offset. When equipping a location with a real adapter make sure that the appropriate values are assigned to the adapter data.

The transformation number of the adapter in location 3 of the chain magazine (no. 1) is to be changed to the new value 8:

```
$TC_ADPTT[$TC_MMP7[1,3]] = 8 ;$TC_MMP7 contains the number of the adapter in
                              ;the new magazine location
```

Once adapter data records have been automatically generated and assigned, operations such as undoing an assignment, renewed definition of an assignment and deletion of an adapter data record are possible.

Example 2

Requirement:

- MM_NUM_TOOL_ADAPTER = 4
- MM_NUM_MAGAZIN_LOCATION = 20
- One chain with 16 locations
- Two grippers,
- One spindle
- One loading/unloading point.

There are 4 different adapter geometries in this case. Adapters must be configured for the chain only.

Assignment

These locations (20 in total) are initially created without adapters. Locations 1 to 4 of the chain are equipped with adapters of the same geometry (here adapter 1). 4 chain locations are to be equipped with adapters with the same geometry.

First, you must define the 4 adapter data records. Now you assign them:

```
$TC_MPP7[1,1] = 1,          $TC_MPP7[1,13] = 4,

$TC_MPP7[1,2] = 1           $TC_MPP7[1,14] = 4,

$TC_MPP7[1,3] = 1           $TC_MPP7[1,15] = 4,

$TC_MPP7[1,4] = 1,          $TC_MPP7[1,16] = 4

...
```

In this way you can assign one adapter data record to several magazine locations.

Note

If you want to delete an adapter data record with a multiple assignment you must make sure that you first undo all the adapter assignments.

3.11 Power failure while tool command is in progress

If a power failure occurs during an action requested by tool management, defined strategies are executed by the PLC or special part programs, in order to establish a defined and consistent status on the machine and the tool management system. These strategies are machine-specific. SINUMERIK controls support the following measures:

Backed up data

The tool and machine data are backed up. The tool attached to the spindle (= magazine location) is identified by the location and the tool block.
(This information is available even without tool management).

Control of data initialized via “Power On”

The following data are set to zero:

- Tool status “Tool change in progress”
- Magazine status “Motion is active”
- Magazine location status “Reserved for tool to be loaded”
- PI command status with reference to magazine operations such as “Active motion”.

Requirements of manufacturer configuration

The PLC must send the last unacknowledged FC 7 or FC 8 prior to power failure (READY did not change to TRUE before power connection) back to the NC when the supply is restored. The function “Asynchronous transfer” exists for tool transfer in FC 8.

The PLC initiates a copy operation of tool data from one location to another without a task from the TM system. For example, relocation of tool data from gripper to magazine if the tool needed to be returned manually to the magazine when the tool change operation was aborted.

Changes in position of the tools involved must be communicated via FC 8. The NC then updates the data for this tool in the tool management.

Further strategies may be necessary, e.g. if a tool change was interrupted. Tools stored in the buffer must be returned to the magazine for this purpose.

3.12 Code carrier

3.12.1 Function of the code carrier system

A link to a tool identification system is made available in the interactive tool loading and unloading dialog on the SINUMERIK 840D. This system allows tool data to be read and written from the tool code carrier rather than manually input.

It is important to remember that particular tool data can only be stored on the code carrier **or** on the MMC when they are unloaded.

When a tool is loaded, the MMC reads its data record from the code carrier and sends it to the tool list in the NCK. The tool data can be edited (offset data, ...) in the same way as the data for tools selected from the tool catalog.

In an existing production line, tools with code carriers may already have been used. The data will be stored in a format suitable for a particular machine control system. When tools of this type are used in combination with a SINUMERIK 840D control, the data formats must be converted to allow the same tool to be used on machines operating with different control systems and thus different data formats.

In addition to the data of the SINUMERIK 840D; user data (Section 3.10 and Subsection 4.10.2) can also be stored on the code carrier and processed via the load/ unload dialog.

The "Tool management data distributor" function block package is available for connection of code carriers via PLC Documentation describing special settings for every code carrier system used is stored on the appropriate installation floppy disk.

3.13 Load / unload tools via PLC with PLC/TOOLMAN data distributor

A PLC-TOOLMAN data distributor is available for connection on code carrier systems to the PLC. Use catalog NC 60 for ordering.

3.14 User data

In addition to the data described here, the machine manufacturer can utilize his own specific tool management data.

The new Siemens user data can be configured only by those with OEM_HIGH rights and are not described here. The associated machine data are mentioned in Chapter 8, but not described in detail.

Tool and cutting edge data

The user can define additional tool and/or cutting edge data during installation. Memory is allocated in the part program memory for this purpose.

The following machine data must be set:

- MD 18080: MM_TOOL_MANAGEMENT_MASK bit2=1
- MD 20310: TOOL_MANAGEMENT_MASK bit 2=1
- MD 18094: MM_NUM_CC_TDA_PARAM (number of parameters)
- MD 18096: MM_NUM_CC_TOA_PARAM (number of parameters)

Note

If the above settings have not been made no softkeys for cutting edge or tool user data appear.

Display screen forms

Depending on the number of defined user data screen forms are displayed in which the user can enter his data. These data are only maintained by the tool management and must be evaluated by the user in the part program (see also Chapter 5).

Defining name and unit

Names and units for these user data can be defined in file PARAMTM.INI (C:\USER\..) in areas [ToolParams] and [ToolEdgeParams]. The latter applies only to the input and display of user data on the MMC (see also Subsection 4.5.3).

Example of the use of user data:

- Max. spindle speed
- Coolant yes/no

- Max. cutting rate

Free user variables

Additional data can be transferred to the PLC on a tool change with the user variables ($P_VDITCP[x]). These data can then be processed in the PLC program. The user variables must be programmed in front of prepare change command T in the part program for this purpose.

Data transfer to PLC user interface DB72 or DB73 is then initiated with the programmed tool change preparation command. Up to three user variables can be transferred simultaneously per tool change. Data cannot be transferred from the PLC to NC by this method. The value format is DINT.

See also Chapter 5.

3.15 PLC description

3.15.1 Interfaces

The interfaces in the PLC consist of data blocks that are updated by the basic program. It is here that tasks such as Load tool or Prepare tool change with source and target are stored for each tool. The tool no. (internal no. allocated by the NCK on loading), tool size and tool status are transmitted additionally over the interfaces for a spindle or turret.

If the position of the tool changes (e.g. from magazine to gripper...), the new positions must be transferred to the tool management on the NCK. Two function blocks, **FC 7** (TM_REV) and **FC 8** (TM_TRANS) are provided for this purpose; these must be called by the PLC programmer and initialized with the correct parameters.

If a magazine or a turret is not driven by an auxiliary axis, the shortest direction of rotation can be calculated with **FC 22** (TM_DIR) and the positioning time thus optimized If it is positioned via an auxiliary axis of the 840D, then **FC 18** is available for the same purpose.

Start-up of tool management function

The tool management (TM) in the PLC is set up when the tool management is installed in the MMC and the NCK “Tool management” option activated. Before start-up of the PLC part of the tool management can begin, block FC 6 (part of the basic program) must be loaded in the PLC. This block is called up by the basic program and must not additionally be called in the user program. FC 8 TM_TRANS (transfer block) and FC 7 and, if necessary, FC 22 TM_DIR (direction selection) must also be loaded and called by the user program.

When installation is complete, the data blocks listed below are set up for the user (tool management user interfaces) and another data block is set up for the tool management FCs the next time the PLC is booted. The lengths of the data blocks arise from the start-up parameters of the tool management (see table below). The following data blocks are available:

Overview of data blocks

Block number	Length in bytes	Meaning
DB 71	4 + 30 bytes * B	Interface for loading/unloading points
DB 72	4 + 48 bytes * W	Interface for spindle as change position
DB 73	4 + 44 bytes * R	Interface for tool turrets as change position
DB 74	Length depends on configuration	Internal data block for tool management

B	= Number of load magazines
W	= Number of spindles as change positions
R	= Number of turrets

DB 71 to DB 74 use approximately 550 bytes with simple configurations of magazines, buffers and loading/unloading points.

Note

If new PLC data have been “generated”, data blocks DB 71 to DB 74 must be deleted in the PLC and the PLC then (cold) restarted. The DBs are then set up for the new configuration.

One interface (data record) in a data block exists for each loading/unloading point, spindle and circular magazine. The data blocks are assigned to the different tasks (see Chapter 9).

DB 71

DB 71 performs the **Load and Unload, Position** and **Relocate** functions. The relocate and position at buffer functions are generally performed on the 1st Interface in DB 71.

DB 72

DB 72 is the interface for loading tools into the spindle. This change procedure also includes preparation of the tool.

DB 73

DB 73 is the interface for tool changes with a circular magazine.

DB 74

Data block DB 74 is an internal tool management data block used for communication control. You must not write to this DB.

All the interfaces mentioned here contain the source and target positions of the tools involved in the procedure in question.

FC 6 is called in the basic program for communication between the NCK and the PLC when tool management is active. This block informs the user interfaces (DB 71 to DB 73) if a tool management function is activated via the part program or operator input.

Interfaces within DB 71 to DB 73

A bit field for the active and passive status of each interface is contained in bytes 0 and 1 of each of the data blocks (DB 71 to DB 73). DBX 0.0 represents the 1st interface, DBX 0.1 the second, etc. A total of 16 interfaces can be addressed. If one of these bits is set to the value = 1 by the tool management, the associated interface is activated. If set to 0, the interface may not be processed by the user.

Principle of interfaces DB 71–73

No. 8	No. 7	No. 6	No. 5	No. 4	No. 3	No. 2	No. 1
No. 16	No. 15	No. 14	No. 13	No. 12	No. 11	No. 10	No. 9
1st interface							
2nd interface							
....							
15th interface							
16th interface							

If the value = 1, the user must evaluate the commands at this interface (see Section 9) and initiate the necessary actions (e.g. position magazines, change tools, etc.). Once these actions have been initiated the programmer can also write to this interface (e.g. to store the current positions of the tools involved in the action, or to enter status bits that he has assigned, or to cancel the bit “Prepare change”). Each modification of the tool positions and/or status information (see FC 8 for description of the status information) via an interface task requires that FC 8 is called with these values.

Note

Once FC 7/8 has been started, it can be reset after a READY signal or error signal.

Tasks from NCK tool management

Tasks	Interface	Acknowledgement	Applications, special features
Load Section 3.5	DB 71	FC 8, TaskIdent = 1, TaskIdentNo = interface no.	NewToolPlace = target position for tool in requested magazine location, status = 1, OldToolPlace = 0
Unload Section 3.6	DB 71	FC 8, TaskIdent = 1, TaskIdentNo = interface no.	OldToolPlace = target position for tool in requested load magazine, status = 1, OldToolPlace = 0
Relocate	DB 71	FC 8, TaskIdent = 1, TaskIdentNo = 1	NewToolPlace = target position for tool in requested magazine location, Status = 1, OldToolPlace = 0

Positioning	DB 71	FC 8, TaskIdent = 1,
		TaskIdentNo = interface no.
			Positioning on load magazine according to interface no.
			Optional positioning on interface 1, status = 5
			NewToolPlace = LMG or BUF
			OldToolPlace = 0

Prepare change for tool in spindle	DB 72	FC 8, TaskIdent = 2,
		TaskIdentNo = interface no.
			Position NewTL at change point, OldTL remains in spindle. Finally status 1 so that change command can be output.
			OldToolPlace = BUF (spindle)
			NewToolPlace = Location NewTL

Change to spindle	DB 72	FC 8, TaskIdent = 2
		TaskIdentNo = interface no.
			OldTL is unloaded (gripper or directly into magazine), NewTL is loaded to spindle. Status 1 required to ensure part program execution continues.
			NewToolPlace = BUF (spindle)
			OldToolPlace = Location OldTL

Tasks	Interface	Acknowledgement	Applications, special features
Without NCK Command: Return OldTL to magazine		FC 8, TaskIdent = 4 TaskIdentNo = channel	The OldTL may need to be transferred asynchronously to the location specified in the Prepare Change command to move the tool from the gripper to the magazine.
Change with turret	DB 73	Normally FC 7, or FC 8, TaskIdent = 3, TaskIdentNo = turret no.	When turret has finished swiveling, FC 7 is called with turret no. as parameter ChgdRevNo.

LMG:	Load magazine
BUF:	Buffer
TL:	Tool
NewToolPlace:	FC 8 Parameter NewToolMag, NewToolLoc
OldToolPlace:	FC 8 Parameter OldToolMag, OldToolLoc

Changes to tool positions without an NCK task

Tasks	Acknowledgement	Applications, special features
Asynchronous transfer	FC 8, TaskIdent = 4, TaskIdentNo = channel for this tool	Required to notify of tool position changes (e.g. in case of voltage interruption, tool change return transport, manual control of turret)
Asynchronous transfer with location reservation for tool transport to BUF	FC 8, TaskIdent = 5, TaskIdentNo = channel for this tool	
Asynchronous transfer without location reservation for manual control of tool turret	FC 8, TaskIdent = 4,	

Further interfaces in the channel interfaces for the ToolMAn function (SW 5 and higher):

DBD 348	T number of tool prewarning limit
DBD 352	T number for tool limit value
DBD 356	T number of new replacement tool
DBD 360	T number of last replacement tool

See also Section 9.4.

Change bits in DBB 344

This information can be evaluated within one OB 1 cycle on the basis of a change bit. The PLC can deduce appropriate action from this information.

Other signals are as follows:

	Channel DB	
Tool missing	DBX 317.7	From NCK
Do not disable tool	DBX 29.7	To NCK
Deactivate wear monitoring	DBX 29.6	To NCK
Deactivate workpiece counter	DBX 29.5	To NCK
Activate time monitor	DBX 1.3	To NCK

3.15.2 Definitions of acknowledgement status

Magazine identifier

The location of a tool in the magazine is identified by a magazine identifier and a location identifier. In a real magazine (chain, turret, etc.), the position of the tool is identified by the magazine number and the location within the magazine assigned during start-up.

If the tool is located in a buffer, the “magazine identifier” is the constant 9998 and the location identifier corresponds to the buffer number assigned during start-up.

In a load magazine, the “magazine identifier” is the constan t 9999. The location identifier is the same as the load magazine number assigned during start-up. In this case, load magazine number = 1 has a special status. Load magazine = 1 (spindle) is for manual loading/unloading and also the interface for tool relocation.

Status 1–6

Status data 1 to 6 cause termination of the command. If one of these status data is passed to FC 8, the “active bit” of the interface defined in FC 8 is reset to “0”. The action is then terminated.

Status 103–105

When one of these data is transferred to FC 8, the “active” bit of the relevant interface remains at “1”, indicating the need for further processing by the user program in the PLC (e.g. continuation of magazine positioning). This status data is usually used to transmit a change in position of one or both tools without terminating the action. For a list of the status information for block FC 8 see

References: **/FB**/ P3, “Basic PLC Program”, Chapter 4

Synchronization

There are various methods by which the PLC and NCK can be synchronized (see Section 3.2.12). The two devices are forced to synchronize by bits 5, 6, 7 and 8 and, in SW 5.1 and later, bit 19 as well of machine data 20310: TOOL_MANAGEMENT_MASK. During internal communication between the PLC and NCK, the devices wait for each command to be acknowledged.

We distinguish between two types of acknowledgement:

- Transport acknowledgement
- End acknowledgement

Transport acknowledgement

Internal acknowledgement to an NCK command. The transport acknowledgement notifies the NCK that the command output by the PLC basic program has been accepted. Before a new command is output, the system checks whether the previous command was accepted. If this is not the case, the output cannot take place. The NCK waits for the acknowledgement before a new command is output.

End acknowledgement

Status check-back of PLC to an accepted NCK command. Error-free termination is indicated by status value = 1 and abnormal termination by status value = 3.

Output of the command

Synchronization of the NCK and PLC is implemented in three steps:

- The interpolation task of the NCK has prepared a command and outputs it to the internal NCK image of the VDI interface.
- The internal NCK image of the VDI interface is transferred to the VDI in the same cycle.
- The PLC basic program accepts the command from the VDI interfaces.

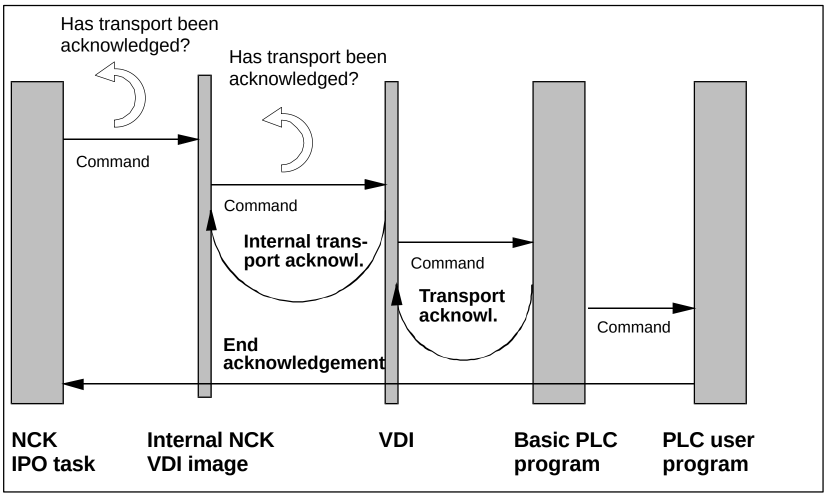


Fig. 3-27 Transport and end acknowledgements

Acknowledgement of output commands

While the output command is being executed, acknowledgements are returned from the PLC basic program and from the VDI.

- The PLC basic program outputs the transport acknowledgement to the NCK after accepting the command.
- An internal transport acknowledgement is output within the NCK after the transfer of the internal VDI image.

The PLC user program can only process one command at a time. It determines the command processing time. If the NCK outputs commands faster than the PLC user program can process them, the NCK is switched to wait mode.

The NCK can also output commands which do not originate from the part program over the interface. These include PI services, which are overlayed over the part program processing asynchronously.

Command termination

Depending on how bits 5 – 8 of MD 20310: TOOL_MANAGEMENT_MASK are set, the command output is deemed to be terminated at different times:

- If bit 5 (or bit 6 for secondary spindle) of MD 20310 is set, the command output is terminated when the internal transport acknowledgement and the transport acknowledgement have arrived. The command is accepted by the PLC basic program.
- If bit 7 (or bit 8 for secondary spindle) of MD 20310: TOOL_MANAGEMENT_MASK means that the command output has just been completed If the final acknowledgement has been received from the PLC.

- If the bits are not set, the command output is deemed to have terminated when the NCK has output the command to the NCK-internal VDI image.

Note

From the perspective of the tool change command, the block change can take place as soon as the NCK has output the command.

Setting bit 19 in combination with bits 5–8 of MD 20310: TOOL_MANAGEMENT_MASK prevents block changes until the necessary acknowledgements have arrived.

3.15.3 Diagnosis of NC-PLC communication

It is possible to record the NCK-PLC communication as part of the tool change function in a file.

Requirements

- **Bit 13** of machine data **MD 20310: TOOL_MANAGEMENT_MASK** must be set.
- There must be free user memory space on the NC for saving the data. This applies both to the **SRAM** (passive file system) and to the **DRAM** with approx. **4KB each**. The number of files in the file system must be below the maximum number of files.

Example of procedure

1. Start an NC program with the following pre-history:
   No tool is present in spindle 1 (magazine no./location no. = 9998/3). Changing with M06 is set. A “milling” tool is present in magazine 2 at location 1 and has the internal T number 1.

   The following is programmed in the NC part program:

```
T=Mill
; Acknowledgement by PLC with FC 8 and status 105 received
; Acknowledgement by PLC with FC 8 and status 105 received (not dis-
played)
T=Mill
; Command with same contents
M06
; The command is not mentioned in the list below
; Acknowledgement by PLC with FC 8 and status 105 received (not dis-
played)
T0 M06
; Acknowledgement by PLC with FC 8 and status
M30
```

2. By activating the RESET key, the recordings residing in an internal circular buffer are included in the file **_N_TCTRA'xx'_MPF,** with 'xx'= channel number 01, 02..., which is created in the passive file system in the _N_MPF_DIR directory. In the current configuration, up to 25 communication procedures can be recorded. If more procedures are recorded in the circular buffer, the oldest data are overwritten. Up to 25 entries can also be made in the _N_TCTRA'xx'_MPF file. More entries cause the file to be deleted and another one created. This means that at program end, after activating the RESET key, you need to save the current diagnostics file in the case of longer diagnostics procedures.

3. Evaluation of the log file.
   In the file**_N_TCTRA01_MPF** a communication process is shown as follows:

- The command **from NC to PLC**
  *T*00001 N:N10 CMD:00002
  NewTool: from M: 00002 P: 00001 to M: 09998 P: 00003
  TNo: 00001 spindle: 00001
  OldTool: from M: 00000 P: 00000 to M: 00000 P: 00000
  Meaning:
  – T00001 = Number of communication process, in this case "1"
  – N:N10 = Block number in part program (if present), here N10
  – CMD:00002 = Command output by the NCK, in this case "2"
  – NewTool = The tool to be loaded
  – OldTool = The tool to be unloaded (from toolholder, a buffer location)
  – TNo = The internal NCK T number of the tool to be loaded
  – Spindle = The spindle no. (toolholder no.) of tool to be loaded
  – M = Magazine number
  – P = Magazine location number
  i.e. a prepare command (CMD:00002) is output by the NCK in the above example. The new tool with T no.=1 is to be moved from location 2/1 to location 9998/3. There is no old tool. The magazine addresses in this case are equal to zero.

- **Acknowledgement** of the NC command by the PLC
  T00002 N: ACK:00002 St: 00105
  NewTool: from M: 00002 P: 00001 to M: 09998 P: 00001
  OldTool: from M: 00000 P: 00000 to M: 00000 P: 00000
  Meaning:
  – ACK = Acknowledgement command from the PLC
  – St = Acknowledgement status from the PLC

- No output of commands with the same contents
  If machine data settings specify that the NCK does not output consecutive commands with the same contents (dummy tool change, dummy tool preparation), this is recorded as follows in the diagnostics file:
  T00012 N:N20
  i.e., only the number and the block number are entered.

- **Result** of the above program (T0 – M6 – M30)
  (content of recording file):(contents of the log file):

  ```
  T00007N:N10 CMD:00005
  NewTool: from M: 00000 P: 00000 to M: 00000 P: 00000 TNo: 00000
  spindle: 00001
  OldTool: from M: 09998 P: 00003 to M: 00002 P: 00001
   T00008 N: ACK:00005 St: 00001
  NewTool: from M: 00000 P: 00000 to M: 00000 P: 00000
  OldTool: from M: 09998 P: 00003 to M: 00002 P: 00001
  ```

Explanation:

- T00007 –> T0 M6 outputs the command 00005
- No new tool is changed, i.e. the addresses of the new tool are equal to zero; *TNo: 00000*
- There is a tool on the spindle with the address 9998/3. This is to be moved back into magazine 2/1.
- T00008 –> The PLC acknowledges the command with 5 and status = 1 and leaves the suggested motion tasks.

List of values and meanings for CMD and ACK

CMD	Explanation
1	A tool is transported from ... to .... Load, unload, change, position
2	Tool change is to be prepared (setting MD 22550 = 1)
3	Tool change is to be carried out (setting MD 22550 = 1)
4	Tool change is to be prepared and carried out (setting MD 22550 = 0)
5	Tool change is to be prepared and carried out (setting MD 22550 = 1)

ACK	Explanation
1	Tool is/was transported. Load, unload, change, position FC 8 – Parameter TaskIdent = 1
2	Tool is/was prepared (setting MD 22550 = 1) FC 8 – Parameter TaskIdent = 2
3	Tool change is/was carried out (setting MD 22550 = 1) FC 8 – Parameter TaskIdent = 2
4	Tool change is/was prepared (setting MD 22550 = 0) FC 8 – Parameter TaskIdent = 3
5	Tool change is/was carried out (setting MD 22550 = 1) FC 8 – Parameter TaskIdent = 2
7	Terminate interrupted TOOLMAN command DB10.DBX105.0=1

ACK	Explanation
8	Tool was transported. If a tool is present at the source address, its data are transported to the target address. Otherwise, only the current magazine position is changed. If the tool transport is from a real magazine, the location to which the source address points is reserved. FC 8 – Parameter TaskIdent = 5
9	Tool transported. If a tool is located at the source address, its data are transported to the target address. Otherwise only the current magazine position is changed. FC 8 – Parameter TaskIdent = 4

3.15.4 Function blocks

Overview of function blocks

Block number	Meaning
FC 6	Block in basic program for tool management
FC 7	Transfer block for tool change with turret
FC 8	Transfer block for tool management, called in response to position and status changes
FC 22	Direction selection for shortest path

Cyclic block FC 6

This block is integrated in the basic program and is automatically triggered when tool management is activated.

Transfer block FC 7, tool change with turret

For description of block, see
References: /FB/,P3, “Basic PLC Program”

Transfer block FC 8

For description of block, see
References: /FB/,P3, “Basic PLC Program”

Direction selection FC 22 TM_DIR

For description of block, see
References: /FB/,P3, “Basic PLC Program”

Other PLC services

For more complex PLC user program tasks, the system offers other PLC services in addition to the FCs mentioned above for tool management control purposes. These services are available in FB 2, FB 3 and FB 4, FB 7 (read and write variables or PI services). These FBs are described in the basic PLC program description in Chapter 4. The tool management PI services (program instances) are also described in Chapter 4 of the basic PLC program description in connection with FB 4 and FB 7. The tool management variables are described in the lists in the section on variables. (See also NC-Var selector help.)

3.16 Workshop interface (SW 5.3) (ShopMill)

The workshop interface (Shopmill) can be used with MMC 100.2 for milling machines.

MD 9414: TM_KIND_OF_TOOLMANAGEMENT is set to select the tool management variant. For workshop tool management, MD is set to 1.

Tool list

The tool list shows all tools (including offset data) that are stored as a tool data record in the NC, irrespective of whether or not they are assigned to a magazine location. The tool list includes the most commonly used tool types to which geometric and technological data can be assigned.

Tool location coding

Display machine data 9672: CMM_FIXED_TOOL_PLACE can be programmed to define whether all tools have a fixed or variable location coding.

- With a fixed location coding (MD 9672=1), a tool is permanently assigned to a particular magazine location. This setting can be used on machines with disk-type magazines.
- In the case of variable location coding (MD 9672=0), the tool can be returned to a different magazine location than its original location. This variant is suitable for machines with chain magazines. Individual tools can be set to “fixed location coding” in the “Tool wear” form on the operator interface.

Tool wear list

The wear data (length and radius/diameter) to be applied in relation to edge 1 or edge 2 are defined in this list. The following monitoring modes can also be selected for a tool.

- Monitoring of effective operating time (service life)
- Monitoring of number of tool load operations
- Additional tool status data (disable tool, tool in fixed location, oversized tool)

Load/unload and sort

Display machine data 9651: CMM_TOOL_MANAGEMENT can be set to

- value **2** to configure the TM system without the Load/Unload and Sort softkeys.
- value **4** to configure the TM with the above softkeys.

When a tool is loaded, it is taken to a magazine location.

When it is unloaded, it is removed from the magazine.

Tools can be sorted according to magazine location, name and type in the tool list and tool wear list.

Manual tools

Manual tools are included in the tool list, but not stored in the magazine. They must be attached to the spindle by hand.

Other functionalities

- Load station for loading and unloading tools set via MD 9673: CMM_TOOL_LOAD_STATION
- Display tools as a diameter or radius via MD 9663: CMM_TOOL_DISPLAY_IN_DIAM

3.16.1 Key data of the workshop interface (ShopMill)

Term	**Data / range**
Possible channels	Channel 1
Active magazine configurations per channel	1

Term	Data / range
Supported magazine types	Circular, chain, disk
Total number of magazines	max. 30
Number of possible spindles	1
Possible load magazines	Either no. 1 or 2
Total number of magazine locations	max. 600
Total number of tools	max. 250
T no.	1–32000
Programming the tools in the NC program using an identifier (name) with 17 alphanumeric characters	e.g. T = “Mill_32”
Duplo no.	1–99
Number of edges per tool	D1 and D2
Location type definition	no
Adjacent location consideration in half locations	1-dimensional, 1 half location
Location coding	fixed or variable
Strategy for tool search	can be set (programmed) via system variables
Strategy for empty location search:	can be set (programmed) via system variables (refers to the current location at the change position)
M06 command for tool change	M code, settable via MD, channel-specific
Tool change with M06 or T command	Settable via MD, channel-specific
Wear monitoring	for every cutting edge
Wear monitoring according to tool life	Resolution msec
Wear monitoring according to number of workpieces	counter
Access to data via NC program	system variables
Automatic decoding stop until tool is selected.	yes
T = Location no.	no

3.16.2 Supported scope of functions

Tool types

– 120	End mill
– 200	Twist drill
– 220	Centering tool
– 710	3D probe
– 711	Edge probe

– 110	Cylindrical die-sinking mill
– 111	Ball end mill
– 121	Shaft mill with corner rounding
– 155	Conical mill
– 156	Conical mill with corner rounding
– 157	Conical die-sinking mill

Tool parameters

- Magazine location/magazine number
- Tool type
- Tool name
- Duplo number
- Geometry length 1
- Geometry radius
- Wear length 1
- Wear radius
- Type of tool life monitoring
- Tool life
- Workpiece count
- Tool status: Tool disabled
- Tool status: Tool oversized (right and left half locations)
- Tool status: Tool in fixed location
- Rounding radius
- Angle for conical milling tools

Supported magazine parameters

- Magazine location disabled

Notes

Start-Up 4

Magazines, buffers and load magazines

Before you begin to start up the machine, you must decide which tool management components you wish to utilize. To reduce tool change times, it is advisable to use synchronized actions. Please refer to Subsection 3.2.12) for details of synchronous actions.

The buffers (spindle, holder, gripper, ...) and load magazines must also be defined. All buffers are combined in one magazine with magazine number 9998 and all loading points are combined in a magazine with number 9999.

For information about the machine configuration, see Section 2.4.

The following equipment must also be started up:

- **NCK**
  Start-up via machine data of the tool management system. Programming and start-up of a tool change cycle.
- **MMC**
  – OP 030
  – MMC 100
  – MMC 103
  Configuring of screen forms in INI file (paramtm.ini). Setup of magazines and magazine configuration via the MMC interface.
- **PLC**
  The user interfaces (DB71–DB73) are automatically set up in the correct number and length during runup, parameterization of FC 8 and creation of machine program.

Start-up sequence for tool management

1. Set start-up machine data
2. Start-up TOOLMAN: Define all magazines, including buffers and loading magazines
3. Generate PLC data
   (the NC and PLC start up together on the next power-up)
4. Define location types

5. Create and load the magazine configuration
6. Active the tool management on the MMC 102/103 (enter paramtm.exe in regie.ini) (see Subsection 4.5.1)
7. Use paramtm.ini to adapt tool (MMC) (see Subsection 4.5.1)
8. Restart (reboot) the MMC, NCK and PLC
9. Create and load the PLC program
10. Perform dry run

4.1 Enter the machine data

General machine data

To start up the tool management function, it is necessary to set machine data for structuring the memory, assigning channels to TO units, etc. Memory in the battery-backed RAM is also required. When “memory influencing” MDs are changed, this memory area is also changed. The data must therefore be backed up first.

All machine data must be set that influence the memory area for tool management:

Order for releasing memory using the machine data

Tool management option bit

MD 18080: MM_TOOL_MANAGEMENT_MASK
Activation of memory for tool management

Definition of number of magazines and magazine locations

MD 18084: MM_NUM_TOOL_MAGAZINE
Maximum number of magazines which NCK can manage (min. 3 magazines). Buffer and load magazines must be added!

MD 18086: MM_NUM_MAGAZINE_LOCATION
Number of magazine locations which NCK can manage
Add buffer and load locations!

Definition of tools and tool edges

MD 18082: MM_NUM_TOOL
Number of tools to be managed by the NCK

MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA
Number of edges in the NCK, tool offsets per TOA block

Options for configuring additional user data for magazines, magazine locations, tools and tool edges

MD 18090: MM_NUM_CC_MAGAZINE_PARAM
Number of additional magazine data $TC_MAPCx[n] are generated

MD 18092: MM_NUM_CC_MAGLOC_PARAM
Number of additional magazine location data $TC_MPPCx[n,m] are generated

MD 18094: MM_NUM_CC_TDA_PARAM
Number of additional tool-specific data per tool $TC_TPPCx[t] are generated

MD 18096: MM_NUM_CC_TOA_PARAM
Number of additional data per tool edge $TC_DPCx[t,d] are generated

MD 18098: MM_NUM_CC_MON_PARAM
Number of additional monitoring data per tool edge $TC_MOPCx[t,d] are generated

Channel-specific machine data

Enabling of channel-specific functions for tool management

MD 20310: TOOL_MANAGEMENT_MASK
Channel-specific activation of tool management

Definition of spindle number for tool life check

MD 20320: TOOL_TIME_MONITOR_MASK
Activation of tool life monitoring function for spindle (toolholder no.) specified in this data

Tool change, turret or spindle

MD 22550 TOOL_CHANGE_MODE
New tool offset with M06 function

MD 22560 TOOL_CHANGE_M_MODE
M06 function for tool change

Selecting the cutting edge after tool change

MD 20270 CUTTING_EDGE_DEFAULT
Initial setting of tool edge without program

Definition of tool with which tool offset is to be selected as a function of MD 20110 and MD 20112 during power-up and reset

MD 20122: TOOL_RESET_NAME
Definition of tool length compensation selection

Definition of the active toolholder number

MD 20124: TOOL_MANAGEMENT_TOOLHOLDER
Definition of the active tool holder no.

Assignment of TO units to channels

MD 28085: MM_LINK_TOA_UNIT
Assignment of a TO area to a channel (default = 1)

Definition of tool length compensation selection after power-up/reset

MD 20110 RESET_MODE_MASK
Definition of control initial setting. Relevant bit = 0:
The current value remains valid.

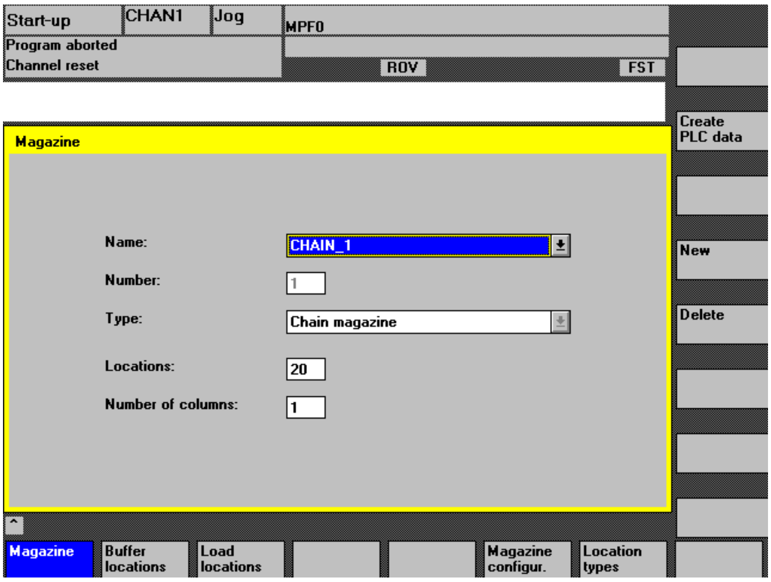
Note

Bits 0–3 of machine data 20310: TOOL_MANAGEMENT_MASK and 18080: MM_TOOL_MANAGEMENT_MASK must always be set identically.

4.2 Enter the magazine data with MMC 102/103

4.2.1 Real magazines

Entering the magazines display



Name: CHAIN_1

Number: 1

Type: Chain magazine

Locations: 20

Number of columns: 1

Fig. 4-1 Start-up: Entry of the magazine data

Machine	Parameter	Program	Services	Diagnosis	**Start-up**		Connect

Machine data	NC	User views	PLC	Drive/ Servo	MMC		**Tool management**

Magazines	**Buffer**	**Load locations**			**Magazine configuration**	**Location types**	

Magazines

Magazines are defined with the appropriate data or existing magazines displayed in the screen in Fig. 4-1 .

Name Enter or select the name of the real magazine.

Number Display of the magazine serial no.

Type Select a magazine type from the window using the EDIT key (chain magazine, circular magazine, box magazine)

Locations Enter or display number of magazine locations

Number of columnsThe “number of columns” is required for considering adjacent location.

No input is required for chain magazine and circular magazine (always 1).
For a box magazine, the number of columns must be entered.

Example of box magazine with 3 columns:

X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X

Number of locations must be divisible by the number of columns.

Create a new magazine

1. Press softkey **New**
2. Enter magazine name with up to 32 characters in length (the magazine no. is assigned according to the input sequence).
3. Select magazine type:
   – Chain magazine
   – Circular magazine
   – Box magazine
4. Enter number of magazine locations
5. For box magazines the "number of columns" must be entered too.
6. Accept the data with softkey **OK**

Note

If the message "invalid value in magazine" appears, the number of locations and/or number of columns is incorrect. The "number of locations" value must be divisible by the "number of columns".
Example:
20 locations cannot be divided into 3 columns, but 21 locations can be.

Delete magazine

1. Select magazine name with the EDIT key
2. Press softkey **Delete**
3. The magazine is deleted without confirmation.

Note

A magazine can only be deleted if it is not assigned to any magazine configuration.

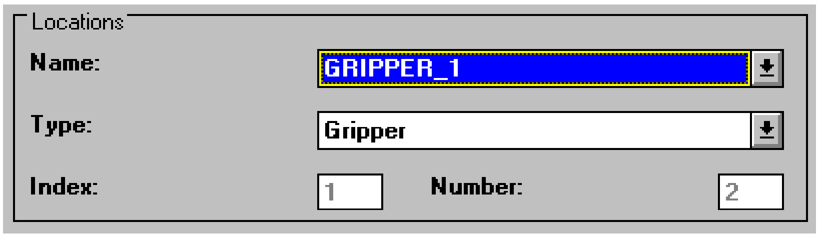
4.2.2 Buffers

Buffer spindle

The position on the spindle (toolholder) always has a distance of 0 at the change position. The location number at the change position therefore coincides with the location number output to the PLC (DBW(n+22) location no. ((source)).

The “Spindle” buffer with index 1 and spindle 1 in the NC are directly interrelated, i.e. the “spindle” buffer with index 1 must also be the 1st spindle in the NC, index 2 = 2nd spindle, etc.

Locations



Locations

Name: GRIPPER_1

Type: Gripper

Index: 1

Number: 2

Fig. 4-2 Locations

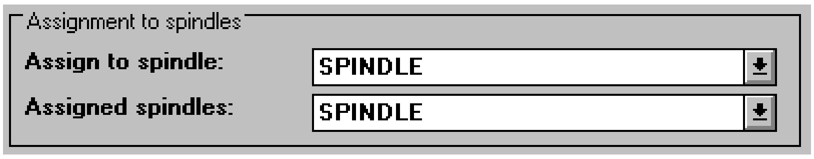
A name (up to 32 characters) must be entered for each buffer location.

Four different types of buffer can be selected (spindle/toolholder, gripper, transfer location, loader).

Display of the internal serial number by which the location is also addressed in the PLC.

The index counts the locations of a type.

Assignment to spindles



Assignment to spindles

Assign to spindle: SPINDLE

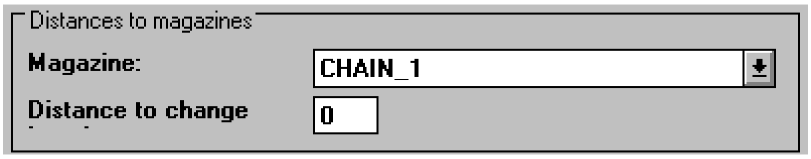
Assigned spindles: SPINDLE

Fig. 4-3 Assignment to spindles

This parameter assigns a buffer location (e.g. gripper) to a spindle. This value is needed if a tool stored in a buffer (e.g. gripper) is called. The tool management checks that the required tool is on its way to a spindle matching the specified assignment.

If the assignment is not correct, an alarm is output to indicate that the tool cannot be prepared for loading to the spindle.

Distances to magazines



Distances to magazines

Magazine: CHAIN_1

Distance to change 0

Fig. 4-4 Distances to magazines

Tool management functions can be utilized only if an assignment between magazine and buffer is defined during start-up.

This is done by defining “Distances to magazines”. If a magazine is selected and a value is entered here, an assignment is established. The number entered is not evaluated for the buffers (unlike the load locations). At least a “0” must be entered. No setting means that no tool can be transferred from this buffer.

If two spindles are supplied by one magazine, the distance from spindle 1 must be entered for the 2nd spindle in “Distances to magazines”.

Example

If, for example, tools are to be changed to “Spindle_1” from 2 magazines, the distance 0 to the buffer, “Spindle_1”, must be entered for both magazines. If no offset is entered, no tool change to the spindle can be performed from magazine_1 and magazine_2.

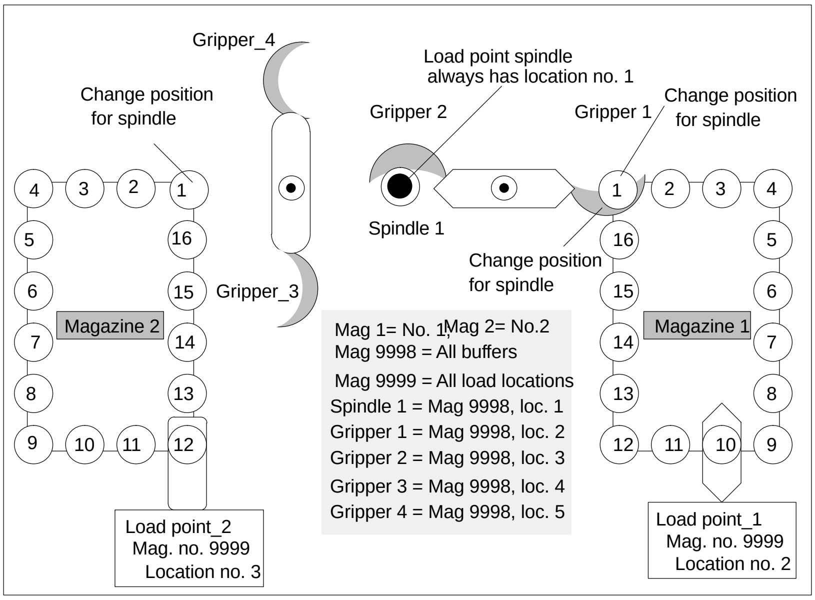

Gripper_4
Change position for spindle
Load point spindle always has location no. 1
Gripper 2
Gripper 1
Change position for spindle
Spindle 1
Change position for spindle
Gripper_3
Magazine 2
Magazine 1
Mag 1= No. 1, Mag 2= No.2
Mag 9998 = All buffers
Mag 9999 = All load locations
Spindle 1 = Mag 9998, loc. 1
Gripper 1 = Mag 9998, loc. 2
Gripper 2 = Mag 9998, loc. 3
Gripper 3 = Mag 9998, loc. 4
Gripper 4 = Mag 9998, loc. 5
Load point_2 Mag. no. 9999 Location no. 3
Load point_1 Mag. no. 9999 Location no. 2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Fig. 4-5 Example of a machine with buffer and load magazine

Distance between buffers and magazines

No.	Name	Offset from magazine
1	Spindle_1	Magazine_1 , Offset : 0 Magazine_2 , Offset : 0
2	Gripper_1	Magazine_1 , Offset : 0
3	Gripper_2	Magazine_1 , Offset : 0
4	Gripper_3	Magazine_2 , Offset : 0
5	Gripper_4	Magazine_2 , Offset : 0

Note

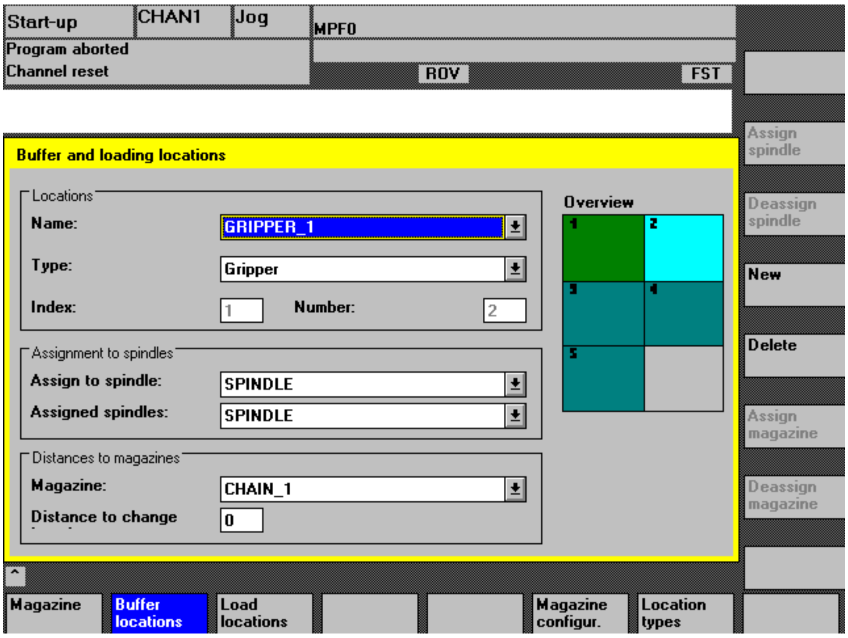
It is important to enter buffers in the correct sequence. The spindle must always be entered first. For each buffer a number is assigned internally by which the location is addressed from the tool management and the PLC.

Enter buffers

1. Press softkey **New** . The location number and the index is assigned internally and incremented.
2. Enter a name: e.g. **Gripper_1**
3. Select a type: Transfer location, **Gripper**, Loader, Spindle

4. Select the next window with the END key
5. Enter assignment to spindles, i.e. to which spindle the buffer (e.g. a gripper) can transfer a tool.
6. Press softkey **Assign spindle**
7. Select the next window with the END key
8. Enter distances to magazines. (i.e. to which magazine this buffer (e.g. a gripper) belongs from the mechanical point of view.)
9. Press softkey **Assign magazine**

Example for Gripper_1



Locations
Name: GRIPPER_1
Type: Gripper
Index: 1 Number: 2

Assignment to spindles
Assign to spindle: SPINDLE
Assigned spindles: SPINDLE

Distances to magazines
Magazine: CHAIN_1
Distance to change 0

Fig. 4-6 Start-up of buffer "Gripper_1"

The overview displays the number of all buffers graphically. The buffer just selected (no.) is also highlighted. Each "type" is displayed in a different color.

4.2.3 Load locations

Load locations

Load locations are locations which are required for loading the magazine. There are two types of load locations:

- Load magazine
- Load stations

Magazine no. 9999

All load locations are numbered 9999 in the load magazine.

Note

Load magazine 9999/1 is always automatically set for loading/unloading of the spindle (all spindles).

Load point

Load points are areas on the machine at which it is possible to load the magazine. These locations to be loaded are magazine locations and are moved to the load point. For example, on chain magazines there is a load point.

The spindle can also be defined as a load point. The load point is a location in magazine no. 9999 (load locations) and is addressed as such on the PLC. The load point (location) has an offset from the change position of the spindle. The load point is set up during start-up in the “Load locations” display.

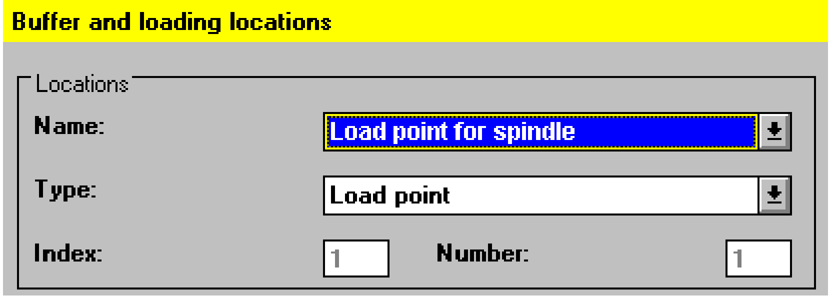
Load station

Load stations are not real magazine locations. Load stations are usual for box and cartridge magazines. If grippers, transfer locations or loaders transfer the tool into the magazine, these can also be defined as buffers during start-up.

Note

The terms “Load point” and “Load station” have been substituted in most cases by the term “Load magazine”.

Enter location data



Buffer and loading locations

Locations

Name: Load point for spindle

Type: Load point

Index: 1

Number: 1

Fig. 4-7 Entering the location data

A name (up to 32 characters) must be entered for each loading point.

It is possible to select either load point or load station.

Display of the internal serial number by which the location is also addressed in the PLC.

Display of the index number. The index counts the locations of a type.

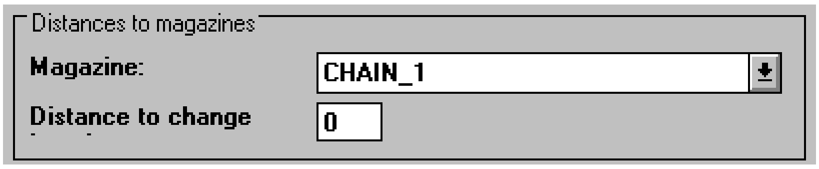
No.	Name	Type	Index
1	Load point for spindle	Load point	1
2	Load_station_1	Load station	1
3	Load_station _2	Load station	2

Load via the spindle

The load point “Load point for spindle” has number 1. This is a preset assignment and cannot be deleted. This “Load point for spindle” applies to all spindles, i.e. during start-up it is not necessary to define any further “Load points for spindles” for other spindles on the machine.

The spindle to be loaded is identified on the PLC via the target location parameter in the interface for “loading spindle in spindle”.

Distances to magazines



Distances to magazines

Magazine: CHAIN_1

Distance to change 0

Fig. 4-8 Distances to magazines

The distance setting creates a relationship between load positions and magazines. For example, if there are 2 magazines, 2 load stations and one load point, “Distances to magazines” must be set to define which magazine is to be loaded/unloaded by which load location. If no distance is entered, the tool management cannot assign load positions to magazines. Load points must also be assigned to the magazine with an entry in “Distances to magazines”. The setting value is critical in this case. The location at the load point is calculated using the value entered here. The value “0” can be entered for the “load point for spindle” because this distance is not evaluated.

The setting in “Distances to magazines” is also used to calculate the offset between load point/load station and the change position of the spindle.

Example:

Offsets of the load locations from other magazines.

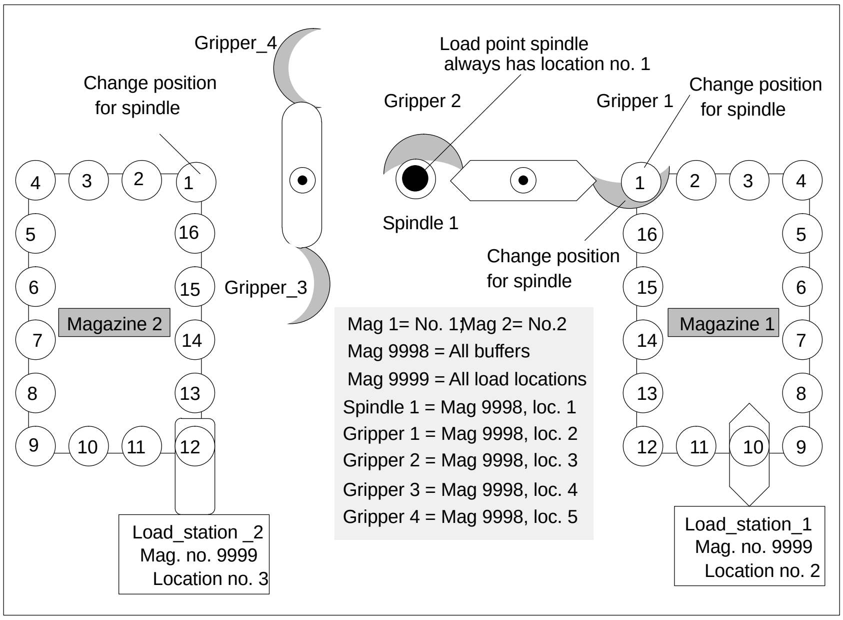


Fig. 4-9 Example of a machine with 2 magazines and 2 load points

Distance between load positions and magazines

No.	Name	Offset from magazine			
1	Load point for spindle	Magazine_1 , Offset : 0 Magazine_2 , Offset : 0			
2	Load_station_1	Magazine_1 , Offset : 9			
3	Load_station _2	Magazine_2 , Offset : 11			

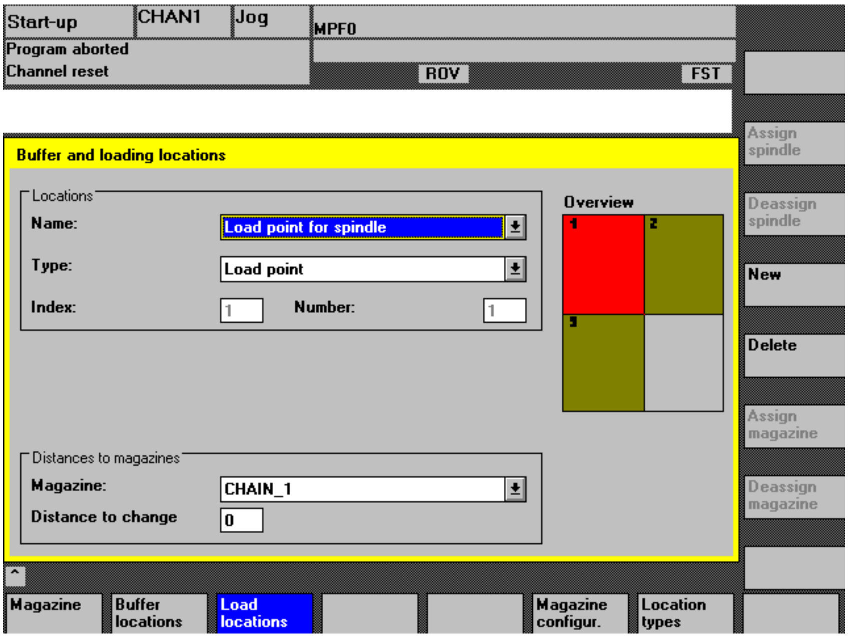
Both magazines can be loaded by the spindle. Load_station_1 is only assigned to Magazine_1 and Load_station_2 to Magazine_2.

Load locations

The data for the load locations (load points and load stations) are entered in the screen below. The load locations are treated internally like a separate magazine with number 9999.

Note

If the load location for spindles is not assigned a distance to the magazine, the load location does not appear as a dialog window during the load operation.



Start-up | CHAN1 | Jog | MPF0
Program aborted
Channel reset

Buffer and loading locations

Locations
Name: Load point for spindle
Type: Load point
Index: 1 Number: 1

Distances to magazines
Magazine: CHAIN_1
Distance to change 0

Fig. 4-10 Start-up of load location (within a magazine)

The entry sequence for the load locations is:

1. Enter the load point for the spindle
2. Enter other load locations
3. Example of load station

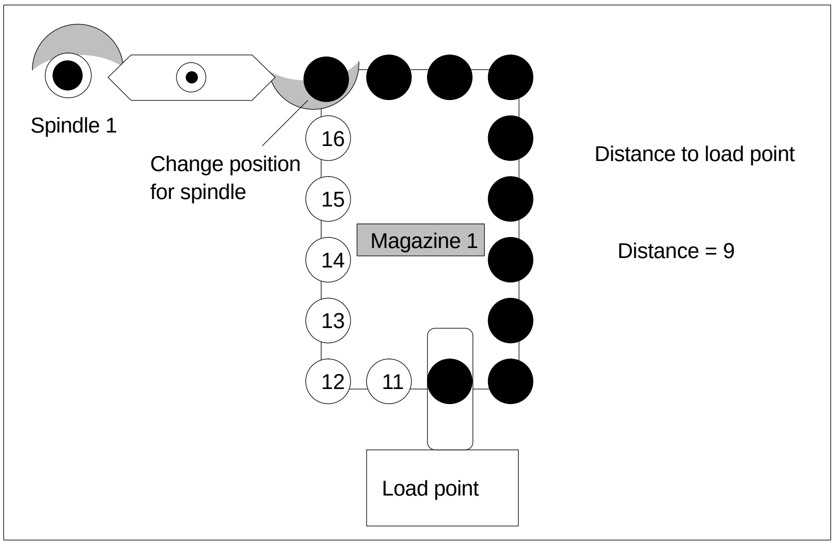


Fig. 4-11 Example of load location

Enter new load stations

1. Press softkey **New**
2. Enter a name: e.g. “**Load_1**”
3. Select a type: **Load station**
4. Select the next window with the END key
5. Enter distances to magazines, i.e. the distance between this load station and the change position on the spindle
6. Press softkey **Assign magazine**

Display overview

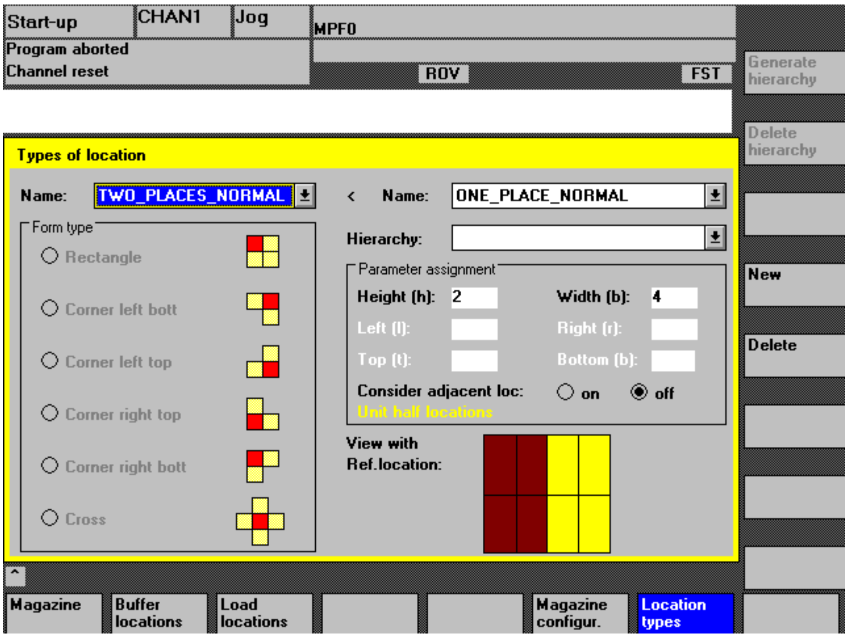
The overview displays the number of all load locations graphically. The number of the selected load point and selected load station is color-highlighted.

4.3 Create the magazine configuration with MMC 102/103

4.3.1 Enter location types

Entering a new location type

1. Press softkey **New**
2. Enter the name (max. 32 characters)
3. Select the **Parameterization** window with the END key
4. Enter the **height** and **width** of the location type, in half locations
5. Change to **Consider adjacent location** with the END key (see Subsection 4.3.3)
6. Select “Consider adjacent location” **on** or **off** with the cursor
7. Switch to **View with reference location** with the END key
8. Set the position of the reference location with the **Cursor**
9. Save with softkey **OK**



Types of location

Name: TWO_PLACES_NORMAL

Name: ONE_PLACE_NORMAL

Hierarchy:

Form type
○ Rectangle
○ Corner left bott
○ Corner left top
○ Corner right top
○ Corner right bott
○ Cross

Parameter assignment
Height (h): 2 Width (b): 4
Left (l): Right (r):
Top (t): Bottom (b):
Consider adjacent loc: ○ on ● off
Unit half locations

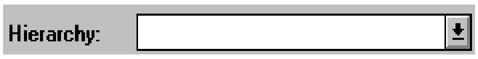
View with Ref.location:

Fig. 4-12 Defining and creating location types

With softkey **Delete**, the selected type of location is deleted if it has not yet been assigned to a magazine.

Select the name of the location (left of name:). E.g. Two_Locations_without_adj_loc and right of “Name:” the location that can go into this location (e.g. One_location_with_adj_loc).

Press softkey **Create hierarchy**

The name of the location type which may be changed over to the other location type is displayed in field  Hierarchy: .

The left-hand window “Name:” is opened with the END key.

Name: ONE_PLACE_NORMAL

If the cursor is positioned on a location type, any set hierarchy will be displayed in the “Hierarchy” field.

Softkey **Delete hierarchy** deletes the location type selected in the “Hierarchy:” field from the hierarchy.

Define location types

The location type defines the size and shape of the location in the magazine. A type must be assigned to every location. Tools of all types can only be transferred to locations of the appropriate type. A tool retains its location type during the entire time it is in the magazine. The location type is designated by a name and contains the following information:

- Name
- Hierarchy
- Parameterization
  (height, width, position of the reference location, consider adjacent location)
- View
  (Display of the assigned half locations and the reference location)

4.3.2 Parameterize a location

The number of half locations occupied by a tool in the magazine is defined when a location is parameterized. The four-digit number, e.g. 2 2 2 2, defines the half locations in the order left, right, top, bottom starting from a reference point.

View

Under the view you can define the position of the reference location (location in which the tool is really located) with the cursor.

Reference location

The reference location is the physical location in the magazine. It is used as a reference point for specifying the tool size and is required to calculate the magazine assignment. The size of the reference location is always displayed as tool size 1 1 1 1.

Normal size

A tool which occupies one magazine location exactly has tool size 1 1 1 1. It is referred to as a “normal sized tool” and parameterized with the following values: Height = 2, width = 2, left, right, top, bottom = 0.

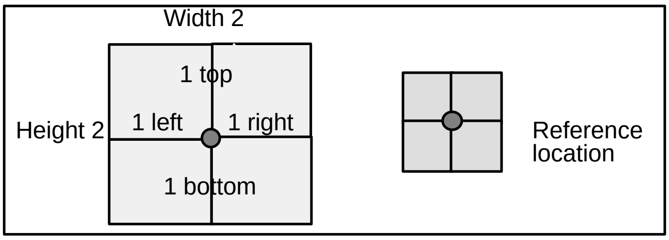


Fig. 4-13 Tool size 1111

Oversize for chain

In the chain magazine, 2 half locations on both the left and right are reserved for tool size 2 2 1 1.

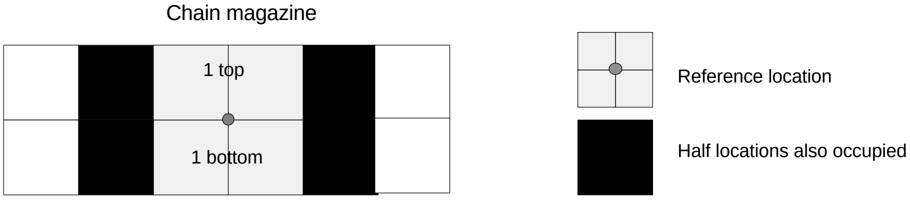


Fig. 4-14 Oversize for chain

Section of chain

Parameter settings for oversize for chain

Height (h):	2	Width (b) :	4
Left (l) :	0	Right (r) :	0
Top (o) :	0	Bottom (u) :	0

Oversize_1 for box

Two half locations are reserved in each direction for tool size 2 2 2 2 in the box magazine.

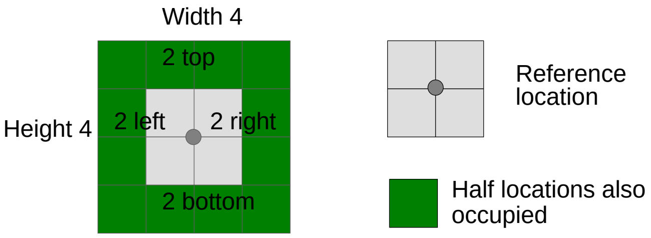


Fig. 4-15 Oversize_1 for box

Parameter settings for oversize_1 for box

Height (h):	4	Width (b) :	4
Left (l) :	0	Right (r) :	0
Top (o) :	0	Bottom (u):	0

Oversize_2 for box

Two half locations are reserved in each direction for tool size 2 2 2 2 in the box magazine. However, with this location type the half location in each corner is not used. Parameter settings for this type are different from type Oversize_1.

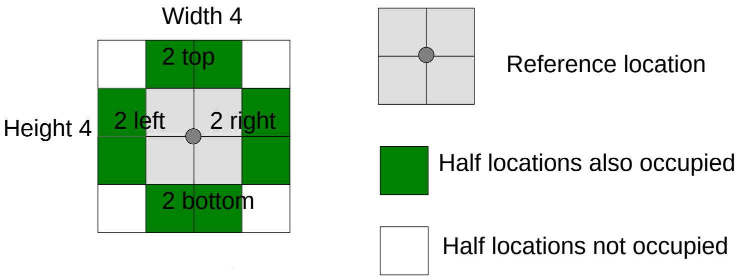


Fig. 4-16 Oversize_2 for box

Parameter settings for Oversize_2 for box magazine

Height (h):	4	Width (b)	: 4
Left (l) :	1	Right (r)	: 1
Top (o) :	1	Bottom (u)	: 1

The unoccupied half locations at the are defined as FREE by parameters Left, Right, Top and Bottom.

4.3.3 Consider adjacent location (function to ba activated by MD)

The information for considering an adjacent location is derived from the assigned location type and is used when a tool is loaded. The location calculation is performed in half locations. “Consider adjacent location” can only be applied in a real magazine, i.e. not in the two internal magazines (for load station, spindles, grippers, ...). The following must be taken into account in location calculation:

- Size of the tool in the half locations to left, right, top, bottom.
- On chain magazines and circular magazines, the check is only performed to the right and left, or from beginning to end.
- On box magazines, the check is made in all 4 directions.

Placing a tool in a location with consider adjacent location active

“Consider adjacent location” requires that the number of magazine locations and the type of magazine (box, chain, ...) are known at the time a tool is loaded.

If a tool which is so large that it occupies adjacent locations is placed in a magazine location, then the affected locations must be checked to ensure that they are empty and that the required half location is not already occupied (see Subsection 4.3.2). Occupation of a half location sets the location to “not free”.

Note

The first magazine location can be defined when the dimension of the associated magazine has been defined (total number of magazine locations).

Only when a parameter of one magazine location has been written, are all magazine locations determined by the dimension of the magazine also created.

If a tool is in a location with active consider adjacent location, the size of the tool and the magazine type must not be changed.

4.3.4 Special tools

When magazine location types are being defined to configure the magazine, it is possible to select whether “consider adjacent location” must be performed for locations of particular types. The user can thus define location types for his magazine in such a way that special tools can be inserted in locations of a particular type without risk of collision, thereby obviating the need for “consider adjacent location”. The following options are therefore given for the handling of special tools:

- Fixed location coding either by exactly one location of the appropriate type or by several locations through assignment of “fixed location coding” status to tools.
- Variable location coding
- Coding with or without consider adjacent location depending on setting for location type

Type classification of the magazine locations produces regions that are defined by the number of adjacent (contiguous) magazine locations.

4.3.5 Hierarchy of location types

To overcome the inflexible classification of magazine locations according to location type, locations can be arranged in ascending order, i.e. in a hierarchy. Several hierarchies of this type can be set up for one TO unit, but a location type can only belong to one hierarchy.

This hierarchy ensure that a tool that only requires a “small” location type can also be placed in a “larger” location type if no “small” locations are free.

For example, a hierarchy can be user to prevent a “normal-size” tool from being placed on an “over-sized” location. If a magazine were full, for example, this could mean that no space was available for an “oversized” tool.

Example:

If a tool is to be inserted in the magazine, the location type decides which locations are available. If there is a hierarchy for this location, the locations are allocated in accordance with this hierarchy.

Tool with location type B is to be placed in the magazine
The following location type hierarchy is defined: A < B < C

Procedure:

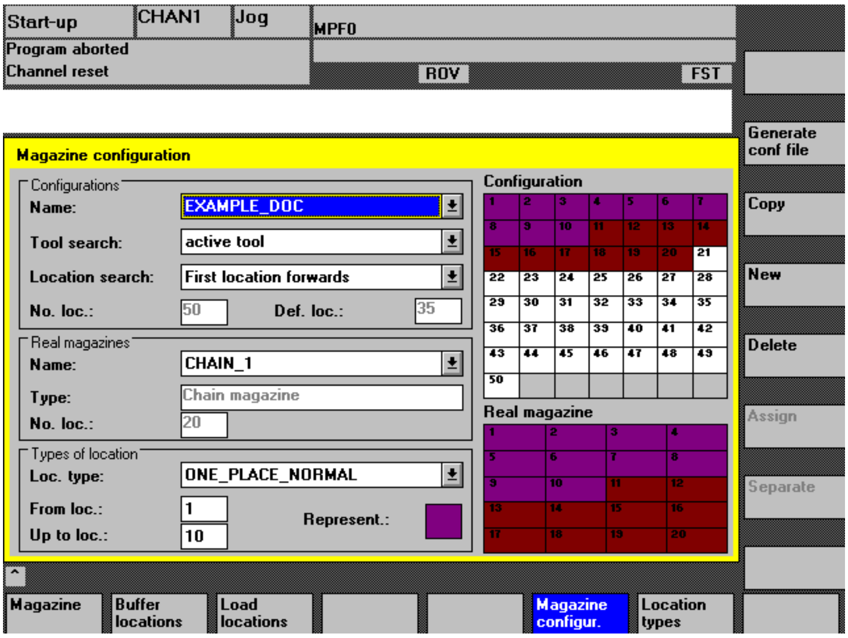
First of all, a check is made to see whether there is still a location of type B.
If not, the search continues for a location with type C.

4.3.6 Create a configuration

There is only **one** common magazine per channel for the configuration of the tool management. The composition of this magazine is determined by a magazine configuration. A magazine configuration can consist of one or several real magazines. Only **one** magazine configuration can be active per channel.

Sequence

- Enter a name for the magazine configuration
- Assign real magazines
- Assign location types for the real magazine.


Magazine configuration

Configurations
Name: EXAMPLE_DOC
Tool search: active tool
Location search: First location forwards
No. loc.: 50 Def. loc.: 35

Real magazines
Name: CHAIN_1
Type: Chain magazine
No. loc.: 20

Types of location
Loc. type: ONE_PLACE_NORMAL
From loc.: 1
Up to loc.: 10
Represent.:

Fig. 4-17 Magazine configuration

Machine	Parameter	Program	Services	Diagnosis	**Start-up**		Connect

Machine data	NC	User views	PLC	Drive/ Servo	MMC		**Tool man-agement**

Magazines	**Buffer**	**Load locations**			**Magazine configuration**	**Location types**	

Creating a new configuration

1. Press softkey **New**
2. Enter the name, e.g. Example_Docu (max. 32 characters)

3. Select a strategy for tool and empty location search (the selection menu is opened with the edit key)
   Tool search:
   Active tool or **shortest path**

   searching for empty location:
   First location forward
   The empty location search always starts at the first magazine location in the direction of ascending location numbers.

   Current location forward
   The empty location search starts at the current location in the direction of ascending location numbers. Depending on the reason for the empty location search, the current location is either the change position or the load point.

   Last location backward
   The empty location search always starts at the last magazine location in the direction of descending location numbers.

   Current location backward
   The empty location search starts at the current location in the direction of descending location numbers. Depending on the current position of the tool for which the empty location search is being performed, the current location is either the change position or the load point.

4. Press softkey **OK** (to create magazine configuration)
5. Select the next window with the END key (real magazines)
6. Select real magazines and press softkey **Assign**. These real magazines are then included in the magazine configuration. (Message: Assigning magazine.)
   After each assignment, the display of the magazine locations in the “No. loc:” box is refreshed.
   Softkey **Separate** can be selected to delete a magazine from the magazine configuration again.
7. Enter the location type for the selected real magazine.
   Place the cursor on the location type and select a location type (e.g. One_location_without_adj_loc). Now enter the magazine locations that you want to have this location type.
   Example: “From location:” 1 , “To location:” 10.
8. Press softkey **Assign**. The dialog text “Location type being assigned” appears as acknowledgement. The defined locations are thereafter displayed in the color for this location type.
9. Softkey **Separate** can be used to “undo” the assignments of locations.

Messages in response to input errors

If an incorrect value is entered when assigning the location types, e.g. dual assignment or overlapping of locations, the message “Magazine location occupied by location type” or “Invalid value in magazine location” appears.

Number of locations:

Displays the sum of all locations in the “real” magazines.

Defined locations:

The number of locations available for tools is indicated here. If all magazine locations are assigned the location type “A” (single location), the total number and the number of defined locations are identical. If there are any location types that occupy more than one location or if a magazine location has no location type assigned to it, the total number will not be the same as the number of defined locations.

Note

The tool management can only find **defined** locations such as empty locations. For this reason, type “A” at least must be assigned to the required locations.

Copy configuration

1. Press softkey **Copy**.
2. Enter the new name under “Copy magazine” and confirm with softkey **OK**.
3. If you want to close the copy window without copying, press **Abort**.

Separate

Softkey **Separate** can be pressed to remove selected magazines or location types from the assignment. Here, the important thing is the position of the cursor. If the cursor is on **Location type**, the selected location range is released. Before separation, the following prewarning is displayed: The location type will be deleted in the magazine assignment. Confirm the action with softkey **OK** or **Abort**.

If the cursor is positioned on **Real magazines**, then the real magazine will be deleted from the configuration. Before separation, the following prewarning is displayed: All data in the magazine assignment will be deleted. Press softkey **OK** or **Abort**.

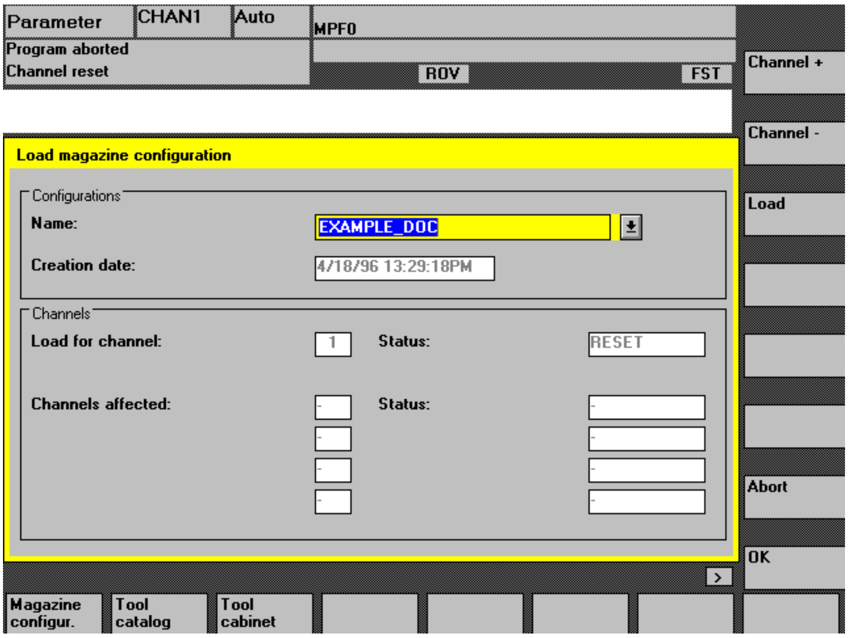
4.3.7 Create and load the configuration file

Softkey **Create conf. file** is selected, a configuration file of the currently selected configuration is created. To activate this configuration in the NCK, it must first be loaded. (An NC program is created that must be executed.)

Note

On versions up to SW 3.2, the function “Load conf. file” only works in the **Parameter** operating area in the menu Magazine configur.

Softkey **Create conf. file** displays the “Magazine Configurations Screen” containing a list of all magazine configuration files that have already been created. Select the file and channel of your choice with softkeys **Channel +/–** and then press softkey **Load**.


Parameter CHAN1 Auto MPF0
Program aborted
Channel reset

Load magazine configuration

Configurations
Name: EXAMPLE_DOC
Creation date: 4/18/96 13:29:18PM

Channels
Load for channel: 1 Status: RESET
Channels affected: - Status: -

Fig. 4-18 Load configuration file

The channel for which the load is to be performed must be in the reset state. This also applies to all channels involved in the same TO unit. Press NC start to load the configuration into the NC.

Note

The magazines, buffers and load locations are consecutively numbered in the sequence in which they are input.

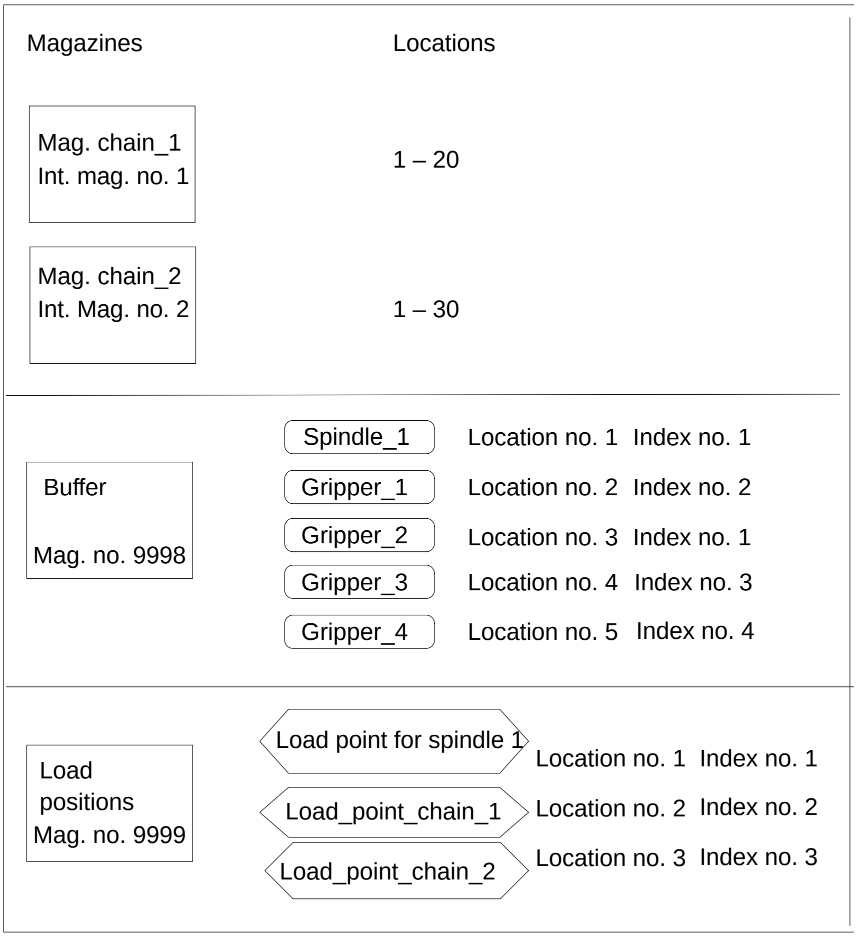


Fig. 4-19 Example of a machine configuration with MMC 102/103

Machine data for previous example:

MD 18080 = B	Activation of memory for tool management (bits 0–3 as for 20310),
MD 18082 = 100	Number of tools that the NCK is to manage. (inc. TL which are not in the magazine and those in the tool list.)
MD 18084 = 4	Number of magazines that the NCK can manage (add the buffer and load magazine to the real magazines)

MD 18086 = 58	Number of magazine locations that the NCK can manage (add the buffer and load locations to the real magazine locations)
MD 18090 – 18098	User data are not used in this instance.
MD 18100 = 200	Number of edges that the NCK must manage (e.g. one edge per tool)
MD 28085 = 1	Assignment of a TO unit to a channel (default = 1), channel 1=1, channel 2=2
MD 20310 = B	Channel-specific activation of tool management (bits 0–3 as for MD 18080),
MD 20320 = 1	Activation of tool life monitoring for the spindle specified here

A valid magazine configuration is active in the NCK once the configuration file has been loaded. This can be saved (TO.ini), for example, in the archive in the Services area under “Active NC data”, “Tool data”. The tool management variables are visible in the view of the TO.ini file.

4.4 Start-up of tool management with MMC 100

No graphic support for the tool management start-up process is available on the MMC 100. The start-up file for the magazine and PLC configurations must be created by the user himself. To active the start-up file for magazine configuring, it must be executed once by the NCK.

The start-up file can also be created and loaded into the NCK with the aid of the MMC 103 start-up tool.

There are several ways of creating the start-up file:

- Entering as a part program at the MMC 100 operator panel
- External creation on a PC with an ASCII editor without formatting.
- Loading the example from the tool box diskette and modifying it on the MMC 100 or on the PC.

MMC 100 supports up to 4 real magazines.

4.4.1 Structure of start-up files for MMC 100

Structure of the start-up file

- Delete old data
- Define type of search strategy
- Define real magazines

- Define buffer magazine
- Define load magazine
- Define locations of the real magazine
- Define locations of the buffer magazine
- Define spindle assignment (which buffer belongs to spindle)
- Define load magazine locations
- Define distances (offsets) from magazines (which spindle, gripper, load point belongs to which magazine)

Part program

The start-up file is a part program e.g. %_N_MAGKONF_MPF. The sample program %_N_MAGKONF_MPF is included on the toolbox diskette.

Brief description of the most important variables

Here, only variables which are important for the configuration file are described. For further descriptions of the system variables, see Section 5.4.

Magazine description data **$TC_MAP3**

$TC_MAP3[magazine no]=status of magazine

Default= 17 means: active magazine, enabled for loading screen form

Search strategy **$TC_MAMP2**

This form is divided into a right and left byte; the right byte describes the tool search and the left byte the empty location search for the spindle tool. A value must be entered for both strategies (see also Subsections 3.3.1 and 5.4.7)

Type of location **$TC_MPP1**

$TC_MPP1[Magazine no., location no.]= Type of location:

Default: value as kind of location

Type of location **$TC_MPP2**

$TC_MPP2[Magazine no., location no.]= Type of location

It is possible to enter any values as long as they match the tools that are to be loaded into the location. Buffers and load points have the value 0.

Consider adjacent location **$TC_MPP3**

$TC_MPP3[Magazine no., location no.]=Consider adjacent location on/off

For further information about “Consider adjacent location”, see Subsection 4.3.3

Location state **$TC_MPP4**

$TC_MPP4[Magazine no., location no.]= Location state (bit pattern)

Default=2	Location free

Location kind index **$TC_MPP5**

$TC_MPP5[Magazine no., location no.]=Location type index

When $TC_MPP1[Magazine no., location no.]=1 (location kind is magazine location) the location number is entered. For other location kinds, the index of the kind is incremented accordingly:

Example with 2 grippers with location kind 3

- The first gripper has location index 1
- The second gripper has location index 2

Distance between a change point, load point and a zero point

Offsets (distances) to the magazine

$TC_MDP2[Magazine no., buffer no.]
Offsets of the buffers from the magazine

For each buffer, a value must be entered here, at least a zero. The value is not evaluated at this point, it is only for assignment.

$TC_MDP1[Magazine no., load point no.]
Offsets of the load points from the magazine

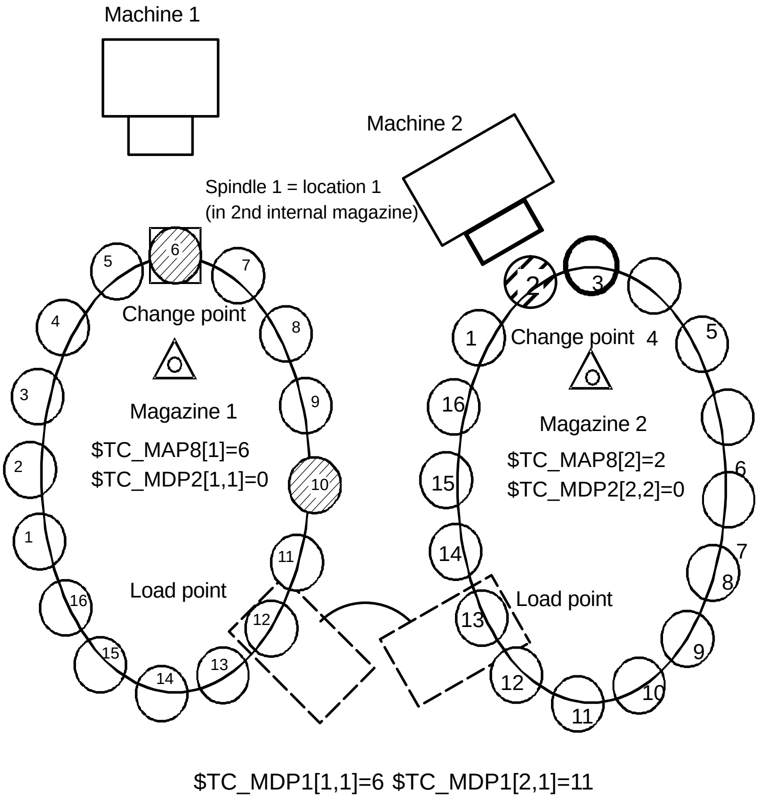


Fig. 4-20 Change position, load point, current position; magazine distance

The zero position in at the change point of the spindles so that the following applies: If location 1 is at the change point, the current magazine position = 1 = $TC_MAP8[x]

$TC_MDP**1**[1,1] = 6 **Distance** between location 1 of **load point** and **zero position** of magazine

$TC_MDP1[2,1] = 11 Distance between the same location from the zero position of magazine 2

$TC_MDP**2**[1,1] =0 **Distance** of location 1 of 2nd internal magazine (**spindle 1**) and the **zero position** of magazine 1

$TC_MDP2[2,2] = 0 Distance between the same location and the zero position of magazine 2

Assignment of magazine locations to spindles

$TC_MLSR [Location no. of buffer, location no. of spindle in buffer magazine]

This variable assigns buffers which have a link between a spindle and the magazines assigned to the spindle. It is thus possible to define which buffer, e.g. gripper, is allowed to change the tool in the spindle.

In Fig.4-9, for example, gripper 2 in location 3 can change the tool in the spindle in location 1 ($TC_MLSR[3,1]).

Example of a start-up file

System configuration:

- 1 chain magazine with 50 locations
- 3 buffer locations
- 2 load points

```
%_N_MAGKONF_MPF
;$PATH=/_N_MPF_DIR
N10 ;
N20 ;
N30;
N40 ;
----------------------------------
N50 ; Magazine configuration: MMC100
----------------------------------
N60 ;
N70 ;
N80 ; Delete old data
N90 ;
N100 $TC_MAP1[0]=0
N110 $TC_DP1[0,0]=0
N120 ;
N130 ; Configuration
N140 ;
N160 $TC_MAMP2=4097               ; Type of search strategy
N170 ;
N180 ; Magazines
N190 ; Real magazine with number [1]
N200 $TC_MAP1[1]=1                ; Magazine type (1: Chain, 3: Circular, 5:
Box)
N220 $TC_MAP3[1]=17               ; Magazine status (see also Configuring
Guide)
N230 $TC_MAP6[1]=1                ; Number of tiers in magazine
N240 $TC_MAP7[1]=50               ; Number of magazine locations
N250 ;
N260 ; Definition of buffer magazine (always number 9998)
N270 $TC_MAP1[9998]=7             ; Magazine type: 7: Buffer
N280 $TC_MAP3[9998]=17            ; Magazine status
N290 $TC_MAP6[9998]=1             ; Number of tiers
N300 $TC_MAP7[9998]=3             ; Number of locations
N310 ;
N320 ; Definition of load magazine (always number 9999)
N330 $TC_MAP1[9999]=9             ; Magazine type: 9: Load magazine
N340 $TC_MAP3[9999]=17            ; Magazine status
```

```
N350 $TC_MAP6[9999]=1             ; Number of tiers
N360 $TC_MAP7[9999]=2             ; Number of locations
N370 ;
N380 ; Locations of chain magazine
N390 ;
N400 $TC_MPP1[1,1]=1              ; Location kind
N410 $TC_MPP2[1,1]=2              ; Location type
N420 $TC_MPP3[1,1]=1              ; Consider adjacent location ON (OFF would
be 0)
N430 $TC_MPP4[1,1]=2              ; Location status (see also Configuring
Guide)
N440 $TC_MPP5[1,1]=1              ; Location kind index
N450 ;
N460 $TC_MPP1[1,2]=1
N470 $TC_MPP2[1,2]=2
N480 $TC_MPP3[1,2]=1
N490 $TC_MPP4[1,2]=2
N500 $TC_MPP5[1,2]=2
N510 ;
N520 $TC_MPP1[1,3]=1
N530 $TC_MPP2[1,3]=2
N540 $TC_MPP3[1,3]=1
N550 $TC_MPP4[1,3]=2
N560 $TC_MPP5[1,3]=3
N570 ;
N580 $TC_MPP1[1,4]=1
N590 $TC_MPP2[1,4]=2
N600 $TC_MPP3[1,4]=1
N610 $TC_MPP4[1,4]=2
N620 $TC_MPP5[1,4]=4
N630 ;
N640 $TC_MPP1[1,5]=1
N650 $TC_MPP2[1,5]=2
N660 $TC_MPP3[1,5]=1
N670 $TC_MPP4[1,5]=2
N680 $TC_MPP5[1,5]=5
N690 ;
.......
......
N3160 $TC_MPP1[1,47]=1
N3170 $TC_MPP2[1,47]=2
N3180 $TC_MPP3[1,47]=1
N3190 $TC_MPP4[1,47]=2
N3200 $TC_MPP5[1,47]=47
N3210 ;
N3220 $TC_MPP1[1,48]=1
N3230 $TC_MPP2[1,48]=2
N3240 $TC_MPP3[1,48]=1
N3250 $TC_MPP4[1,48]=2
```

```
N3260 $TC_MPP5[1,48]=4
8N3270 ;
N3280 $TC_MPP1[1,49]=1
N3290 $TC_MPP2[1,49]=2
N3300 $TC_MPP3[1,49]=1
N3310 $TC_MPP4[1,49]=2
N3320 $TC_MPP5[1,49]=49
N3330 ;
N3340 $TC_MPP1[1,50]=1
N3350 $TC_MPP2[1,50]=2
N3360 $TC_MPP3[1,50]=1
N3370 $TC_MPP4[1,50]=2
N3380 $TC_MPP5[1,50]=50
N3390 ; Locations of the buffer memory
N3400 $TC_MPP1[9998,1]=2               ; Location kind (in this case,
spindle)
N3410 $TC_MPP2[9998,1]=0               ; Location type: as BUF is 0 here
N3420 $TC_MPP3[9998,1]=0               ; Consider adjacent location OFF
N3430 $TC_MPP4[9998,1]=2               ; Location status
N3440 $TC_MPP5[9998,1]=1               ; Location kind index
N3450 ;
N3460 $TC_MPP1[9998,2]=3  ; Gripper 1
N3470 $TC_MPP2[9998,2]=0
N3480 $TC_MPP3[9998,2]=0
N3490 $TC_MPP4[9998,2]=2
N3500 $TC_MPP5[9998,2]=1
N3510 ;
N3520 $TC_MPP1[9998,3]=3  ; Gripper 2
N3530 $TC_MPP2[9998,3]=0
N3540 $TC_MPP3[9998,3]=0
N3550 $TC_MPP4[9998,3]=2
N3560 $TC_MPP5[9998,3]=2
N3870 ;
N3880 ; Spindle assignment             ; Spindle assignment
N3890 $TC_MLSR[2,1]=0                  ; 1st gripper (location 2) belongs to
spindle (location 1)
N3900 $TC_MLSR[3,1]=0                  := 2 Gripper (location 3) belongs to
spindle (location 1)
N3920 ; Load magazine locations
N3930 $TC_MPP1[9999,1]=7               ; Location type “load point” (for
spindle!)
N3940 $TC_MPP2[9999,1]=0               ; Location type (always 0 in this
case)
N3950 $TC_MPP3[9999,1]=0               ; Consider adjacent location
OFF!N3960 $TC_MPP4[9999,1]=2           ; Location status: Free
N3970 $TC_MPP5[9999,1]=1               ; Location kind index
N3980 ;
N3990 $TC_MPP1[9999,2]=7
N4000 $TC_MPP2[9999,2]=0
```

```
N4010 $TC_MPP3[9999,2]=0
N4020 $TC_MPP4[9999,2]=2
N4030 $TC_MPP5[9999,2]=2
N4040 ;
N4650 ; Offsets (distances)
; Distances to magazine
N4660 ;
N4670 $TC_MDP2[1,1]=0                    ; Spindle
N4680 $TC_MDP2[1,2]=0                    ; Gripper 1
N4690 $TC_MDP2[1,3]=0                    ; Gripper 2
N4700 $TC_MDP1[1,1]=0                    ; 1st load point
N4710 $TC_MDP1[1,2]=25                   := 2 Load point (distance 25 to ac-
tual position)
N4720 ;
N4730 ; End
N4740 ;
N4750 M30
```

Load and activate the start-up file

If the start-up file has been created on an external PC, it must be transferred to directory _N_MPF_DIR on the control system.

To activate the start-up file in the NC, it must be started as a part program and handled in the following way:

- Select the part program, e.g. _N_MAGKONF_MPF.MPF
- Start the program with NC start.

Create PLC data with MMC 100

The tool management function in the PLC is subdivided into:

1. One start-up section (program section in OB100, FB1)
2. Transfer section of NCK commands to PLC in OB40 or OB1 via block FC 6.
3. Acknowledgement of execution of NCK commands to NCK by block FC 7 (turret) or FC 8 (user program).
4. Direction selection for magazines (FC 22).

Data relating to start-up are stored from data word 64 onwards in DB 4. These must be written by the PLC user program. The number of magazines, load points, spindles and turrets are determined from these data and used to set up the tool management data blocks (DB 71 to DB 74) automatically. All interfaces that were activated before the power was last switched off are deleted and in DB 74 during start-up. The start-up routine is part of the basic program.

Address	Meaning	Data type
DBB 64	Number of magazines incl. BUF and load/unload stations	BYTE
<		
DBW 65 (70, 75, ...)	Magazine number	INT
DBB 67 (...)	Magazine type	BYTE
DBW 68 (...)	Number of locations	INT
>	Repetition with number of magazines in DBB 64	
Address dependent on number of magazines	Number of spindles	BYTE

4.5 Start-up of tool management for MMC 102/103

4.5.1 Activate tool management displays

In SW version 4 and later, the system directory structure has been modified to prevent user changes from being lost when the SW is upgraded. System directory “MMC 102” is a pure system directory in which “Read only” files are stored.

Customized settings which deviate from the supplied ini files are stored in user directory **user**. The latter contains only modifications to the appearance of the operator interface which can be made only via settings on the MMC interface itself.

As a general rule, the parallel directories to mmc2 contain only ini file entries which deviate from the original settings.

c:\user\regie.ini

To activate the tool management function on the MMC 102/103, the TaskConfiguration must be modified in file c:\user\regie.ini.
paramtm must be entered in line Task1 =name := **paramtm**.

```
;========================================
[TaskConfiguration]
;==========================================
; List of area applications. As in section ’SystemStartup’ a timeout
; value must be specified in milliseconds which the master control provides
; for start-up.
; Length of the list: Maximum 32 entries
; 1st bar
;==========================================
Task0 =name := machine, Timeout := 60000, TerminateTasks := rh
Task1 = name := paramtm, Timeout := 60000, TerminateTasks := rh
```

4.5.2 Configuring

All information which describes the user interface of the tool management is stored in the file **c:\user\paramtm.ini**. To edit this file, /MMC/DOS Shell must be selected and the file opened with command edit c:\user\paramtm.ini. The paramtm.ini file can also be created on an external PC and copied to directory c:\user.

National language components are parameterized in “language\patm_*.ini” in the section [BatchTools]. The asterisk “*” stands for the 2 letters of the language code, e.g. gr for German, en for English, sp for Spanish, nl for the Netherlands.

New list features as of SW 5.2

As of SW 5.2, the magazine, tool, work correction list and tool details include additional functions:

- Parameter settings for bitmaps in the list
- Modification of tool designations and duplo numbers in the lists
- New magazine list spanning several lines

4.5.3 Structure of file paramtm.ini

Directory c:\mmc2\user

Note

Please do not insert TAB characters in this file.
Do not exceed maximum file length of about 63 kbytes because all information higher 63 KByte will be ignored!
Since Version SW 5: Please read file \USER\PARAMINI.OUT after ini file is evaluated by tool management software.

A comment can be added at the end of entries with a semi-colon ";".

The column numbers in the magazine and tool lists range from 1 to 1000. The maximum number of columns in a list is 90. This maximum number of columns should not, however, be utilized since it will drastically reduce the display build time.

Please use ....<EndOfList> to identify the end of each list definition. This will improve the rate at which parameters are read in.

Example of syntax of a parameter line

"2=TC_TP2, 11, TC-TP2 ; Tool identifier"

The meanings of the individual syntax components are as follows:

Parameter line in the INI file for parameterizing a list column	Meaning
2	Number of entry
First TC_TP2	Shows data in relation to NC Programming Guide

Parameter line in the INI file for parameterizing a list column	Meaning
	New in SW 5: If setting “MultiLine=SINGLE” is used in the tool or magazine list, it is possible to specify the edge number by adding “@Ee” (edge number from 1 to max. number for each tool). This setting influences the following variables: cutting edge parameter TC_DPp@Ee ;cutting edge supervision parameter TC_MOPp@Ee ;cutting edge OEM parameter TC_DPCp@Ee ;assigned DNo TC_DPCE@Ee sum offset TC_SCPz@Ee set-up offset TC_ECPz@Ee if “@Ee” is not specified in these columns, the cutting edge = 1 and is thus compatible with older software versions. It is not necessary to specify “@Ee” if the setting “MultiLine=MULTI” is used for the magazine or tool list. The data of the current edge are automatically displayed.
11	Approximate width of columns “[GeneralSettingsForMagAndToolList]” for entries “ColumnWidthTwipsPerAlphaCharacter” and “ColumnWidthTwipsPerNumericCharacter”
Second TC_TP2	Column header text or key for column header text. If entry “ReadLanguageIni”=1 in the “[General]” area, the column header text is sought in files mmc2\language\patm_gr._ini, user\language\patm_gr.ini etc. in area “[ListColumnHeaderText]”, entry “TC_TP2” for this example
ToolIdent	with “;” begins a comment, “//” can also be used

Softkey texts [SoftKeysForMagAndToolList]

Magazine list	Tool list	Working offset list
[1_MagList] = M1	[1_ToolList] = T1	[1_ActLIST] = A1
[2_MagList] = M2	[2_ToolList] = T2	[1_ActLIST] = A1
[3_MagList] =M3	[3_ToolList] = T3	[1_ActLIST] = A1

Examples of a magazine list, a tool list and a working offset list are given below:

First display of the magazine list

```
[1_MagList]
; MultiLine=
; Values:  "SINGLE" (default) Single-line display
           "MULTI"  Multi-line display: The number of lines is determined by
           the existing data: With MagList and ToolList, this is the number
           of cutting edges, with ActLi this is the number of sum offset
           blocks DL.
MultiLine=SINGLE

; Fixed columns are located on the left of the grid – do not scroll – and
; cannot be changed.
NoOfFixedColumns=1

; Columns that have marks for the current data (current location,
; programmed tool, current tool) to be displayed.
; No marks are entered in the buffer display in the magazine lists.
; The current tool can be recognized indirectly from its magazine location
; "Spindle".
;ShowActLocationCol = 1
;ShowProgToolCol = 1
;ShowActToolCol =1

; Location no, MagazineLocationNumber
1= ToolInLocation, 3, ToolInLocation

; LocationStatus Disabled, Disabled
2= TC_MPP4_1, 1, TC_MPP4_1

; LocationStatus Free, Free (<_> assigned)
3= TC_MPP4_2, 1, TC_MPP4_2

; LocationStatus Handled, Reserved for tool in buffer
4= TC_MPP4_3, 1, TC_MPP4_3

; LocationStatus Moving, Reserved for assignment
5= TC_MPP4_4, 1, TC_MPP4_4

; LocationStatus Left, Assigned in left half location
6= TC_MPP4_5, 1,  TC_MPP4_5

; LocationStatus Right, Assigned in right half location
7= TC_MPP4_6, 1, TC_MPP4_6
```

```
; LocationStatus Top, Assigned in upper half location
8= TC_MPP4_7, 1,  TC_MPP4_7

; PlaceStatus Bottom, Assigned in lower half location
9= TC_MPP4_8, 1,  TC_MPP4_8

; # ;ToolIdent
10= TC_TP2, 11, TC_TP2

; DuploNo, DuploNo.
11= TC_TP1, 5, TC_TP1

; ToolNo ,T number
12= TC_MPP6, 5, TC_MPP6

; ToolsizeLeft
13= TC_TP3, 1, TC_TP3

; ToolsizeRight
14= TC_TP4, 1, TC_TP4

; ToolsizeTop
15= TC_TP5, 1, TC_TP5

; ToolsizeBottom
16= TC_TP6, 1, TC_TP6

; MagLocationType, ToolLocation_spec, Tool type of Pl
17= TC_TP7, 4, TC_TP7

; ToolState Active, Active tool
18= TC_TP8_1, 1, TC_TP8_1

; ToolState Permitted, Enabled
19= TC_TP8_2, 1, TC_TP8_2

; ToolState Disabled, Disabled
20= TC_TP8_3, 1, TC_TP8_3

; ToolState Measure, Measured
21= TC_TP8_4, 1, TC_TP8_4

; ToolState Warning limit, Prewarning limit reached
22= TC_TP8_5, 1, TC_TP8_5

; ToolState Change, Tool is being changed
23= TC_TP8_6, 1, TC_TP8_6

; ToolState Location, Fixed location-coded
```

```
24= TC_TP8_7, 1, TC_TP8_

; ToolState Used, Tool was being used
25= TC_TP8_8, 1, TC_TP8_8

; End of list. No search for higher entry numbers.
26= <EndOfList>
```

First display of the tool list

```
[1_ToolList]
NoOfFixedColumns=1
;ShowProgToolCol = 1
;ShowActToolCol =1

; List row number, List line number
1= NO, 4,  NO

; Magazine no., Magazine number
2= MagNo, 4,  MagNo

; Location no, MagazineLocationNumber
3=  ToolInLocation, 3,  ToolInLocation

; ToolIdent
4= TC_TP2, 11, TC_TP2

; DuploNo, DuploNo.
5= TC_TP1, 5, TC_TP1

; ToolNo ,T number
6= TC_MPP6, 5, TC_MPP6

; ToolsizeLeft
7= TC_TP3, 1, TC_TP3

; ToolsizeRight
8= TC_TP4, 1, TC_TP4

; ToolsizeTop
9= TC_TP5, 1, TC_TP5

; ToolsizeBottom
10= TC_TP6, 1, TC_TP6

; ToolState Active, Active tool
11= TC_TP8_1, 1, TC_TP8_1
```

```
; ToolState Permitted, Enabled
12= TC_TP8_2, 1, TC_TP8_2

; ToolState Disabled, Disabled
13= TC_TP8_3, 1, TC_TP8_3

; ToolState Measure, Measured
14= TC_TP8_4, 1,  TC_TP8_4

; ToolState Warning limit, Prewarning limit reached
15= TC_TP8_5, 1,  TC_TP8_5

; ToolState Change, Tool is being changed
16= TC_TP8_6, 1,  TC_TP8_6

; ToolState Location, Fixed location-coded
17= TC_TP8_7, 1,  TC_TP8_7

; ToolState Used, Tool was used
18= TC_TP8_8, 1,  TC_TP8_8

; MagLocationType, ToolLocation_spec, Tool type of Pl
19= TC_TP7, 4,  TC_TP7

; End of list. No search for higher entry numbers.
20= <EndOfList>
```

First working offset list

```
[1_ActList]
MultiLine=SINGLE
NoOfFixedColumns=1
NumLinesPerReq = 11
;ShowProgToolCol = 1
;ShowActToolCol =1

; List row number, List line number
1= NO, 4, NO

;# ;ToolIdent
2= TC_TP2, 11, TC_TP2

; DuploNo, DuploNo.
3= TC_TP1, 5, TC_TP1

; ToolNo ,T number
4= TC_MPP6, 5, TC_MPP6
```

```
; CuttEdgeNumber, Cutting edge number
5= CuttEdgeNo, 1, CuttEdgeNo

; Freely assigned DNo; assigned DNo
6= TC_DPCE, 6, TC_DPCE

; Magazine no., Magazine number
7= MagNo, 4, MagNo

; Location no, MagazineLocationNumber
8=  ToolInLocation, 3, ToolInLocation

; Loc. type of loc., LocationType, location-related
9= TC_MPP2, 3, TC_MPP2

;Wear group, LocationTypeIndex
10= TC_MPP5, 4, TC_MPP5

11= TC_DP1, 4, TC_DP1
12= TC_DP3, 11, TC_DP3
13= TC_DP4, 11, TC_DP4

14= TC_SCP13, 9, TC_SCP13
15= TC_SCP14, 9, TC_SCP14

16= TC_SCP23, 9, TC_SCP23
17= TC_SCP24, 9, TC_SCP24

18= TC_ADPT1, 11, TC_ADAPT1
19= TC_ADPT2, 11, TC_ADAPT2
20= TC_ADPT3, 11, TC_ADAPT3
21= TC_ADPT4, 4, TC_ADAPT4

; ToolState Active, Active tool
22= TC_TP8_1, 1, TC_TP8_1

; ToolState Permitted, Enabled
23= TC_TP8_2, 1, TC_TP8_2

; ToolState Disabled, Disabled
24= TC_TP8_3, 1, TC_TP8_3

; ToolState Measure, Measured
25= TC_TP8_4, 1, TC_TP8_4

; ToolState Warning limit, Prewarning limit reached
26= TC_TP8_5, 1,  TC_TP8_5
```

```
; ToolState Used, Tool was being used
27= TC_TP8_8, 2,  TC_TP8_8

; End of list. No search for higher entry numbers.
28= <EndOfList>
```

Softkey texts, tool size and location type for empty location search during loading

```
[SearchOfMagLocations]
1_SoftkeyText= EL1  ;                        Text for 1st softkey

; Half of location unit; left, right, top, bottom
1_ToolSizeLRTB=1,1,1,1

; Location type no for search of empty location
1_LocationTypeNo=1

2_SoftkeyText=EL2;                           Text for 2nd softkey

; Half of location unit; left, right, top, bottom
2_ToolSizeLRTB=1,2,1,1 ;half of location unit; left, right, top, bottom

; Location type no for search of empty location
2_LocationTypeNo=1

3_SoftkeyText=EL3                            Text for 3rd softkey

; Half of location unit; left, right, top, bottom
3_ToolSizeLRTB=2,2,1,1 ;half of location unit; left, right, top, bottom

; Location type no for search of empty location
3_LocationTypeNo=1

4_SoftkeyText=EL4                            Text for 4th softkey

; Half of location unit; left, right, top, bottom
4_ToolSizeLRTB=1,2,1,1 ;half of location unit; left, right, top, bottom

; Location type no for search of empty location
4_LocationTypeNo=2
```

Default settings

```
[DEFAULT_SETTINGS]

; Magazine list: Function load, data input directly in list:
; 0=default settings must be confirmed with picture “tooldata”,
;   if they are needed because of missing input
; 1=default settings will be taken without confirmation (except of ToolId-
ent)
; 2=default settings will be taken without confirmation (also ToolIdent)
DEFAULT_WITHOUT_CONFIRM=0

; Half locations: Range 1 to 7
TOOLSIZE_LEFT=1

; Half locations: Range 1 to 7
TOOLSIZE_RIGHT=1

; Half locations: Range 1 to 7
TOOLSIZE_UPPER=1

; Half locations: Range 1 to 7
TOOLSIZE_DOWN=1

; Tooltype, range 100 to 1000
TOOLTYPE=120

; Duplo number: Range 1 to 32000
TOOLDUPLO=1

; Ident: Max. length 27
TOOLIDENT=NEU

; Additive values, (0 = default):
; 1=active tool
; 2=allowed
; 4=disabled
; 8=measured
; 16=warning limit reached
; 32=changing
; 64=fixed place coding
; 128=has been used
TOOLSTATE=0

; Index of a defined location type
TOOLPLACESPEC=1

; 0=no monitoring (default)
; 1=monitor by service life
; 2=monitor by workpiece count
TOOLMONITOR_MODE=0

; 1=find next duplo (default)
; 2=find on shortest path
TOOLSEARCH_MODE=2
```

```
[TMMODES]
; 0=do not delete tool automatically
; when it is unloaded (magazine list only). (Default)
; 1=delete tool automatically, when it is unloaded (magazine list only)
DELETE_TOOL_ON_UNLOAD=0

; 0=do not handle edge parameters out of tool type (default)
; 1=handle edge parameters out of tool type (if not 0)
EDGE_PARAMS_OUT_OF_TOOLTYPE=1

; 0=display: left,right,top,bottom (default)
; 1=display: left,right
SHOW_TOOLSIZE_ONLY_LEFT_RIGHT=0

; True=display (default)
; False=do not display
; only used if SHOW_TOOLSIZE_ONLY_LEFT_RIGHT is 0 (or default)
SHOW_TOOLSIZE_COMPONENTS=left:=True, right:=True, top:=True, bottom:=True

; The ’Activate D check’ function refers to:
; –1=all magazines with distance reference to the spindle/toolholder (de-
fault)
;  1=only the current magazine
DCHECK_ACTIVATE=–1

; The “Activate D check” function can be executed automatically when
; the working offset list display is opened
; False=Function is activated only via softkey (default)
; True=Function is automatically executed when working offset list display
; is called
DCHECK_AUTO_ACTIVATE=False

; DB:  Original name from magazine configuration from database
;      No language-dependent texts
; DLL: Name = text from language-DLL + index; (default); example: Spindle1
;       Language-dependent texts
NameOfBufferLocationFrom=DB

; To display the “New tool edge” and “Delete tool edge” (multi-line list
; display only) functions in the basic tool/magazine list displays, softkey
; “Tool details” can be replaced by the new softkey “Data management”.
; Softkey ’Tool details’ ; is still stored with the same function-
; ality behind the new softkey if this option is selected.
; False=’Tool details’ remains (default)
; True=’Data management’ is activated
ACTIVATE_EDGE_MANAGEMENT_IN_LISTS=False
```

[CONTROL]

```
MDIList=pa\patm.mdi
ControlFile=pa\patm.zus
NewFormNames=1
ScreenTwips=1
```

User tool data

```
[ToolParams]
UserDataParamName1 = TC_TPC1
UserDataParamName2 = TC_TPC2
UserDataParamName3 = TC_TPC3
UserDataParamName4 = TC_TPC4
UserDataParamName5 = TC_TPC5
UserDataParamName6 = TC_TPC6
UserDataParamName7 = TC_TPC7
UserDataParamName8 = TC_TPC8
UserDataParamName9 = TC_TPC9
UserDataParamName10 = TC_TPC10
UserDataParamIO1 = <EndOfList>
```

User cutting edge data

```
[ToolEdgeParams]
[ToolEdgeParams]
; length 1
EdgeParamNameLLen1 = TC_DP3
; length 2
EdgeParamNameLLen2 = TC_DP4
; length 3
EdgeParamNameLLen3 = TC_DP5
; radius l1
EdgeParamNameRLen1 = TC_DP8
; radius l2
EdgeParamNameRLen2 = TC_DP9
; radius r1
EdgeParamNameRRad1 = TC_DP6
; radius r2
EdgeParamNameRRad2 = TC_DP7
; angle1
EdgeParamNameAng1 = TC_DP10
; angle2
EdgeParamNameAng2 = TC_DP11

UserDataParamName1 = TC_DPC1
UserDataParamName2 = TC_DPC2
UserDataParamName3 = TC_DPC3
```

```
UserDataParamName4 = TC_DPC4
UserDataParamName5 = TC_DPC5
UserDataParamName6 = TC_DPC6
UserDataParamName7 = TC_DPC7
UserDataParamName8 = TC_DPC8
UserDataParamName9 = TC_DPC9
UserDataParamName10 = TC_DPC10
```

Define access levels

```
[ACCESSLEVEL]
SKAVTIVTM=7            ; Activate TM in Application PARAM
SKMGLIST=7             ; Display Magazine List
SKTLLIST=7             ; Display Tool List
SKACLIST=7             ; Permit / Enable Work Correction List Display
SKTOOLLOAD=5           ; Permit Loading of Tools
SKTOOLUNLOAD=5         ; Permit Unloading of Tools
SKTOOLMOVE=7           ; Permit Movement of Tools in Magazine
SKSETTINGS=4           ; Enable SK Settings
SKFILFCT=4             ; Enable SK File Functions
SKNXTCHAN=7            ; Enable SK Next Channel
SKMAGCONF=4            ; Permit Configuring of Magazines
SKTOOLCAT=7            ; Enable / Permit Tool Catalog
SKTOOLCAB=7            ; Enable / Permit Tool Cabinet
SKSINCOMLD=7           ; Enable / Permit Load Tool from SINCOM (if code car-
                       ; rier installed)
SKCTORSINCOM=7         ; ... Code Carrier Functions or Load Tool from SINCOM
                       ;(if code carrier not installed)
SKMGLREPR1=7           ; Permit Selection of 1_MagList Display
SKMGLREPR2=5           ; Permit Selection of 2_MagList Display
SKMGLREPR3=5           ; Permit Selection of 3_MagList Display
SKNCTOOLDATA=7         ; Permit Reading Tool Data from NC or Data Management
                       ; (if ACTIVATE_EDGE_MANAGEMENT_IN_LISTS=True)
SKNCTOOLED=7           ; Permit / Enable Edge Data
SKNCTOOLSUPV=7         ; Permit / Enable Monitoring Data
SKNCTOOLDL=7           ; Permit / Enable DL Data
SKNCDETAILS=7          ; Permit Reading Tool Data from NC (if ACTI
                       ; VATE_EDGE_MANAGEMENT_IN_LISTS=True)
SKNCNEWTOOLED=6        ; Permit Creation of New Cutting Edges in NC
```

```
SKNCDELTOOLED=6       ; Permit Deletion of Cutting Edges in NC
SKNCDELTOOL=5         ; Permit Deletion of Tools in NC
SKTRAFO=7             ; Permit / Enable Toggle Transformed / Not Transformed
                      ; View of Edge Data
SKCHECKACTIVATE=6     ; Permit / Enable D-Check and Activate
SKMGBUFFER=7          ; Permit / Enable Display of Buffer
SKMGFIND=7            ; Permit / Enable SK Search & Posit.
SKMGLISTPOS=7         ; Permit / Enable Positioning
SKMGNEXT=7            ; Permit / Enable SK Next Magazine
SKTLNEWTOOL=6         ; Permit Creation of Tools in NC
SKTLLREPR1=7          ; Permit Selection of 1_ToolList Display
SKTLLREPR2=5          ; Permit Selection of 2_ToolList Display
SKTLLREPR3=5          ; Permit Selection of 3_ToolList Display
SKFINDPL1=7           ; Permit / Enable Find Location, User-Defined 1
SKFINDPL2=7           ; Permit / Enable Find Location, User-Defined 2
SKFINDPL3=7           ; Permit / Enable Find Location, User-Defined 3
SKFINDPL4=7           ; Permit / Enable Find Location, User-Defined 4
SKFINDPL=7            ; Permit / Enable Find Location
SKFINDPLACE=7         ; Permit / Enable Find Location, Load Tool List
SKACTPLACE=7          ; Enable SK Actual Location
SKLDTOOLDAT=7         ; Enable SK Tool Data in State Loading Tools
SKCONFLOAD=4          ; Permit Loading of Magazine Configuration
SKACLREPR1=7          ; Permit / Enable Selection of 1_ActList Display
SKACLREPR2=7          ; Permit / Enable Selection of 2_ActList Display
SKACLREPR3=7          ; Permit / Enable Selection of 3_ActList Display
SKDZERO=7             ; Permit / Enable SK Delete D-Numbers
SKDFIND=7             ; Permit / Enable SK Search for D-Numbers
ChangeToolTypeWithoutConfirmation=-1 ;Access level for allowing change of
tool type
                      WITHOUT confirmation dialog
                      ;From value -1: All users must confirm. (default)
                      ;To value 7:  No confirmation necessary.
ChangeToolSizeAndToolplace_spec=-1 ;Access level for allowing change of un
                      ;loaded tool of tool size and ToolLocation_spec data
                      ;From value -1: Noone is allowed to change (default)
                      ;To value 7:  Everyone is allowed to change
READ_GUD_LUD=7
WRITE_ZOA=7
READ_SYSVAR=7
EDIT_VIEW=7
```

4.5.4 Configure TM displays in file paramtm.ini

Settings can be made in file paramtm.ini to adapt the operator interface of the tool management function.

The following adjustments can be made:

- Change the structure and arrangement of lists
- Initialize certain values
- Protect or deactivate functions using access rights.

All functions and features of the MMC 102/103 tool management are illustrated in the sample file on MMC 102/103.

The functions required for the machine in question must be selected as part of the start-up process. You can initialize certain values and functions to make operation as easy and convenient as possible.

Examples for configuring the access rights

Example 1

- The tool data must be entered directly in the magazine list.
- When a tool is unloaded, the tool data are to be deleted automatically.
- The tool list function is not used.
- The tool catalog and tool cabinet functions are not used.

A configuration could be entered as follows:

```
...
[TMMODES]
...
DELETE_TOOL_ON_UNLOAD=1    The tool data are deleted automatically when
                           a tool is unloaded
....
[ACESSLEVEL]
...
SKTLLIST=2                 The tool list can only be activated by the
                           manufacturer’s password.
SLTOOLCAB=2                The tool catalog and cabinet SKTOOLCAT=2
                           can only be activated by the manufacturer’s pass-
                           word
....
```

Example 2

- The tool data are not deleted during unloading, but remain in the tool list (in NCK). The data can be used for loading tools.

- The tool catalog and cabinet functions are not used (data backup on MMC).

A configuration could be entered as follows:

```
...
[TMMODES]
...
DELETE_TOOL_ON_UNLOAD=0  The tool data are not deleted
                         when unloading
....
[ACESSLEVEL]
...
SKTLLIST=7               The tool list is always
                         active.
SLTOOLCAB=2              The tool catalog and cabinet SKTOOLCAT=2
                         can only be activated by the manufacturer’s password
....
```

Example 3

When a tool is unloaded, the tool data are stored on the hard disk (in the tool cabinet). The data are automatically deleted on the NCK when the tool is unloaded. The data stored in the tool cabinet can be accessed again when the tool is loaded.

The tool catalog and tool cabinet functions are used because the programs are created interactively by dialog programming.

A configuration could be entered as follows:

```
...
[TMMODES]
...
DELETE_TOOL_ON_UNLOAD=1  The tool data are deleted when unloading
....
[ACESSLEVEL]
...
SKTLLIST=2               The tool list can only be activated by the
                         manufacturer’s password.
SLTOOLCAB=7              The tool catalog and cabinet
SKTOOLCAT=7              are always active (not locked)
...
```

If access rights are allocated to a function and the current protection level is “less than” the level allocated, the softkey does not appear on the user interface and the function cannot be used.

This applies to all functions. For example, if the “tool cabinet” function is protected, the softkey is not displayed in the LOAD/UNLOAD functions.

Parameterization of bitmaps in lists

The display of the active tool, the programmed tool and the current location in the magazine list can be freely parameterized in SW 5.2 and later, i.e. bitmaps can be inserted in the parameterizable columns of individual lists. These bitmaps can be customized to suit user requirements and are created in programs such as Paintbrush. This view is activated in paramtm.ini.

The bitmaps for the current tools are red and the bitmaps for the programmed tools are green. The standard bitmaps described below reside in the directory “mmc2” (see /IAM/ Installation MMC, IM 3).

Standard bitmaps

Bitmap	Properties

Two arrow heads pointing to the right	TNo. <> 0;
	DNo./cutting edge no <> 0;
	DLNo. = 0

Arrow right	TNo. <> 0;
	DNo./cutting edge no <> 0;
	DLNo. = 0

Arrow head pointing to left	TNo. <> 0;
	DNo./cutting edge no = 0;
	DLNo. = 0

Dark green parallelepiped	Current location

User-defined bitmaps can be stored in the “user” directory. These can be displayed in the lists instead of the standard bitmaps.

Handling of lists

The columns of the lists in which the bitmaps are to be entered can be set for each list view. The width of the bitmaps is set in characters for the entire markings. The width of the column is automatically increased by the value set.

Bitmaps overwrite each other if they are activated in the same column and line. The marking at the top is for the current tool, below it the marking for the progammed tool and at the bottom, the marking for the current location. Hidden bitmaps are not output.

Note

In multi-line magazine and tool lists, the marking is entered in the cutting edge line when the current/programmed DNo./cutting edge no. <> 0. The same applies to the DL lines in working offset lists where DLNo. <> 0. Since only cutting edges can be displayed in the views of the working offset lists, the marking only appears if the current/programmed DNo./cutting edge no. <> 0.

The current magazine position is only marked in the magazine location views. Markings only appear in the normal magazine display and not in the buffer display.

Configuring the bitmaps

By default, the bitmaps are not entered in paramtm.ini and are not displayed. If the bitmaps are to be displayed in the lists, you will have to make some changes to the parameter file. One entry is required for each bitmap.

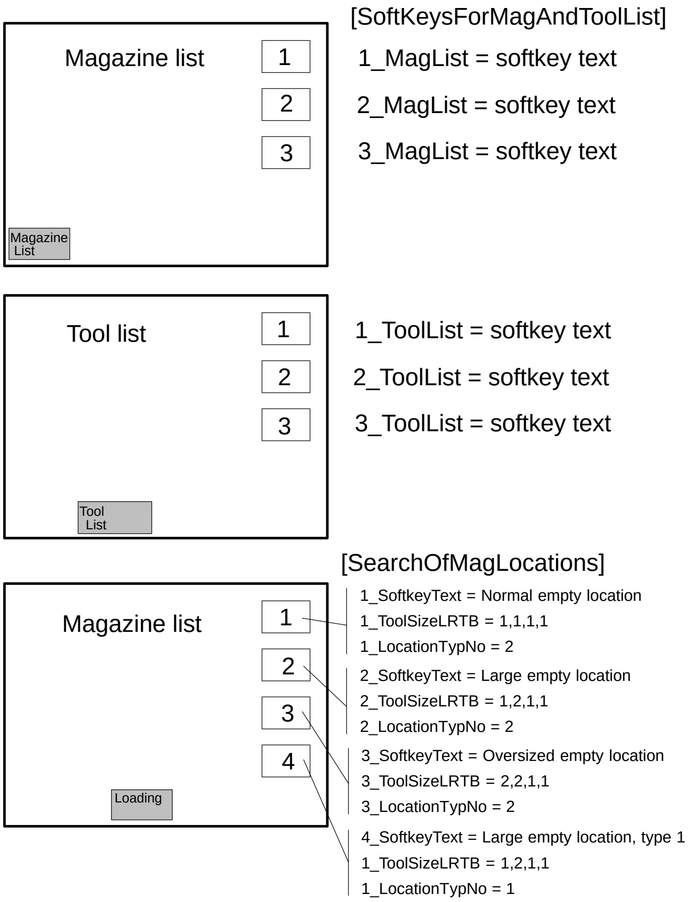
Entries in paramtm.ini:

```
[GeneralSettingForMagAndToolList]

;#| Width of bitmap display
;#| Unit: number of characters
WidthOfActBitmapsInCharacters=5

;#| Name of bitmap for the current tool /DNo./DL,
;#| with D<>0 and DL<>0
ActToolBitmap=paat.bmp

;#| Name of bitmap for the current tool /DNo.,
;#| if the current cutting edge is D=0.
ActToolZeroDBitmap=paatd0.bmp

;#| Name of bitmap for the current tool /DNo./DL,
;#| if the current DL=0.
ActToolZeroDLBitmap=paatdl0.bmp

;#| Name of bitmap for the programmed tool /DNo./DL,
;#| with D<>0 and DL<>0
ProgToolBitmap=papt.bmp

;#| Name of bitmap for the programmed tool /DNo.,
;#| if current cutting edge is D=0.
ProgToolZeroDBitmap=paptd0.bmp

;#| Name of bitmap for the programmed tool /DNo./DL,
;#| if current DL=0.
ProgToolZeroDLBitmap=paptdl0.bmp

;#| File name of bitmap for the current magazine location
ActLocationBitmap=paap.bmp
```

```
[1_MagList]
;#| Columns in which markings (bitmaps) are to appear
ShowActToolCol=1
ShowProgToolCol=1
ShowActLocationCol=1
```

4.5.5 Configuring instructions for paramtm.ini file

Input of softkey texts



```
[SoftKeysForMagAndToolList]
1_MagList = softkey text
2_MagList = softkey text
3_MagList = softkey text

1_ToolList = softkey text
2_ToolList = softkey text
3_ToolList = softkey text

[SearchOfMagLocations]
1_SoftkeyText = Normal empty location
1_ToolSizeLRTB = 1,1,1,1
1_LocationTypNo = 2
2_SoftkeyText = Large empty location
2_ToolSizeLRTB = 1,2,1,1
2_LocationTypNo = 2
3_SoftkeyText = Oversized empty location
3_ToolSizeLRTB = 2,2,1,1
3_LocationTypNo = 2
4_SoftkeyText = Large empty location, type 1
1_ToolSizeLRTB = 1,2,1,1
1_LocationTypNo = 1
```

Fig. 4-21 Softkeys

The displays stored behind softkeys 1 to 3 in the magazine and tool lists are defined in the file paramtm.ini. As the initial setting on selecting the tool management, those displays appear that have been configured for [1_MagList] and [1_ToolList].

Displaying the displays

Concealed fields can be made visible by scrolling with the cursor keys.

The serial number defined by the input sequence during start-up is displayed in the location type box rather than the name of the location. The screen that is displayed under the 1st vertical softkey in the magazine list is specified after vocabulary word [1_MagList] in file paramtm.ini.

Explanation on configuring a display

The columns must be numbered in ascending order. Gaps in the numbering are not permitted. To terminate the display, type a semicolon after the number. If columns are top be hidden, add a semicolon at the start of the line. The serial numbering must be retained.

Example

```
....
[1_MagList]
NoOfFixedColumns = 1  ;Fixed columns are located left on grid and do not
scroll.
;Syntax: Column = VarName_from_NCProgram, Length_in_characters, Text
;
;1=ToolInLocation,2,Pl  ;MagLocNumber, MagazinLocDataIndex,
;2=MagNo,3,Mag          ;Number of magazine in which tool is located
;2=NumCuttEdges,2,AS    ;NumberOfCuttEdges
2=TC_MPP4_1,1,P         ;LocationStatus, LocationStatus, Disabled
3=TC_MPP4_2,1,L         ;LocationStatus, LocationStatus, Free (<_> Occupied)
4=TC_MPP4_3,1,A         ;LocationStatus, LocationStatus, Reserved f TL i
Buffer
```

```
NoOfFixedColumns - Number of fixed columns on left
```

```
1= ToolNrLocation,2,PI
1=              Number of column
ToolInLocation Variable name from NC programming language
2,              Length in characters, also equals column width
PL,             Text to be displayed as column header
```

User data

The parameter name and the unit can be defined for the displays of the tool and cutting edge user data. The number of parameters that are displayed depends on the MD and the number of defined parameters.

[ToolParams]	Tool user data
[ToolEdgeParams]	User cutting edge data

Special characters

Special characters such as ü, ä, e' are entered in ANSI CODE so that they can be displayed in screen forms.

4.6 Display machine data on MMC 100 (SW 4 and later)

In SW 4 and higher, it is possible to initialize default settings and to allocate access rights to certain functions on the MMC100 (see Section 2.9).
A detailed description of the machine data is given in Subsection 8.1.1.

Additional user parameters

If additional user data (user parameters for cutting edge and/or tool data) were created via NCK machine data, these are displayed in further screen forms.

The data are administered but not evaluated by the tool management.

User texts

An application floppy with which the user can parameterize his files is supplied with the MMC 100 software.

The table shows the configurable texts which are stored in file pa.txt. The user can enter his own texts under “user text”.

User cutting edge data, texts

Name of text	User text		
T_EDGE_TEXT_1TT_EDGE_TEXT_1”	T_TM_OEM_CUT_TM_OEM_CUT	47”	72
...	...		
...	...		
T_10T_TM_OEM_CUTT_EDGE_TEXT_10”	“T_TM_OEM_CUTT_EDGE_TEXT_10”	47	72

User tool data, texts

Name of text	User text		
T_TM_OEM_TOOL_TEXT_1	“T_TM_OEM_TOOL _TEXT_1”	47	72
...	...		
...	...		
T_TM_OEM_TOOL _TEXT_10	“T_TM_OEM_TOOL_TEXT_10”	47	72

Display machine data as of SW 5.2

Cutting edge data error correction

In software version 5.2 and later, the use of WRITE_TOA_FINE_LIMIT and USER_CLASS_WRITE_FINE on cutting edge data can be controlled via MD 9449: WRITE_TOA_LIMIT_MASK.

The bits set in MD 9449 indicate whether the display machine data WRITE_TOA_FINE_LIMIT and USER_CLASS_WRITE_FINE are used on the cutting edge type. If the bits are not set, the FINE_LIMIT is not used.

Bit assignment for MD 9449

The bit assignment for MD 9449: WRITE_TOA_LIMIT_MASK is as follows:

Bit	**Application**	**Default value**
Bit 0	Cutting edge data (offsets), wear values	1
Bit 1	SC data (local offsets and their wear values)	1
Bit 2	EC data (local offsets and their setup values)	1

Default value for MD 9449 is 7. The FINE_LIMIT is applied to all data types.

Compatibility of the fine offsets for MMC 103 and MMC 100.2

Up to SW 5.2, the machine data WRITE_TOA_FINE_LIMIT and USER_CLASS_WRITE_FINE were used for the geometry, basic and wear parameters in the tool management for the MMC 103.

In the tool management for the MMC 100.2, these machine data are active only on the wear parameters of the cutting edges. As of SW 5.2, the two machine data are active only on the wear parameters of the cutting edges in the tool management for the MMC 103.

Changing the compatibility of the fine offsets

An entry in the paramtm.ini can restore the old response of the tool management for the MMC 103. MD WRITE_TOA_FINE_LIMIT and MD USER_CLASS_WRITE_FINE are then used for the geometry, basic and wear parameters again.

Entry in paramtm.ini:

```
[General]
; Application of
; $MM_WRITE_TOA_FINE_LIMIT and $MM_USER_CLASS_WRITE_FINE on geometry
; values and basic values of cutting edge data
UseFineLimitForToolGeoAndAdapt=False          ;Default
;UseFineLimitForToolGeoAndAdapt=True          ;
```

4.7 Special notes for start-up of the Workshop onterface (ShopMill)

Start-up of the workshop interface (ShopMill) corresponds to the MMC100 standard tool management. The following table documents only the deviations between the two processes:

NC-MD	Name	Value
18082	MM_NUM_TOOL	Max. of 250
18100	MM_NUM_CUTTING_EDGES	Max. of 2 MM_NUM_TOOL

The associated machine data are given in Subsection 8.1.1.

4.8 Start-up of OP 030 operator panel

The OP 030 tool management system supports the load/unload identifiers of the OEM application “SINTDI”.
The full scope of functions is available with no restrictions in NCK version 3.7 and later.

Display machine data

MM_TM_SINTDI
The default setting 0 means that the SINTDI identifier function is deactivated.
A numerical value higher than 0 identifiers the parameter from which the load/unload identifiers are read or written.

MM_TM_NUM_MAG

Default setting = 0
A numerical value of > 0 identifiers the magazine to be displayed first; if an illegal value is set, the first real magazine to be found is displayed.

MM_TM_UNLOAD_AND_DELETE
The default setting 0 means that tools are not erased from the TO memory when they are deleted, but only from the tool list.

MM_TM_LOAD_TOOL_NEW
The default setting 1 means that tools are marked immediately for loading in the tool list.

MM_TM_TOOL_STATE_DEF_VAL
The default setting is 2. Other values:

2	Enable
4	Disabled
8	Measure
64	Coded as fixed location

These values can also be combined (except for enable and disabled), i.e. 72 means Measure and Coded as fixed location. Each change applies to newly created tools until the tool status is changed.

Display machine data can be modified in file bd_op030.tea before the flash operation. A corresponding prompt is output during installation.

A distinction is made between the system software and the screen kit for installation purposes. In this case, screen kit means either the OEM variant or the development kit.

4.9 Start-up of PLC program

Overview

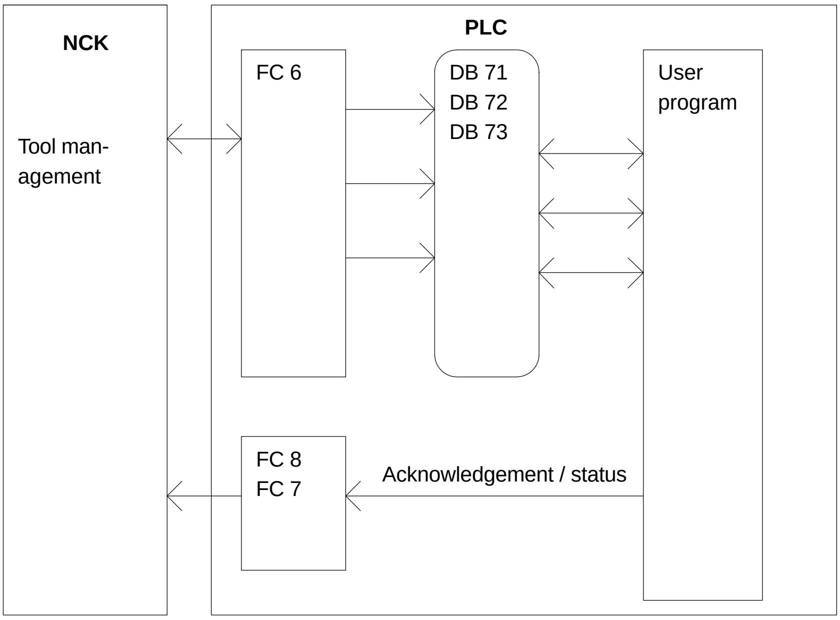


Fig. 4-22 Start-up of PLC program

FC 6 supplies the TM interfaces (data blocks DB71–DB73) with information for the new and old tools. The user has to process these active interface data in his user program and make sure that the tools (old and new) go into the correct positions (magazine, location). To ensure that the tool management (TM) always knows where the tool is currently located, each change in location of a tool must be communicated to the tool management via the FC 7 or FC 8 acknowledgement status.

4.9.1 Create PLC data

Once all the magazines, buffers and load points have been entered (for all channels/TO areas), the data must be sent to the PLC. Softkey **Create conf. file** transfers the data to the PLC. If the message: “Error on calling the NCDDE server” appears, the transmission is repeated.

Note

If new PLC data have been “generated”, data blocks 71–74 (and DB 77 in SW 5 and higher), must be deleted and the PLC restarted. The DBs are then set up for the new configuration.

Start-up of tool management

The tool management (TM) in the PLC is set up when the tool management is installed in the MMC (Create PLC Data) and the NCK “Tool management” option activated.

Before start-up of the PLC part of the tool management can begin, block FC 6 (part of the basic program) must be loaded to the PLC. This block is called up by the basic program and must not additionally be called in the user program.

FC 8 TM_TRANS (transfer block), or FC 7 for circular magazines, and, if necessary, FC 22 TM_DIR (direction selection) must also be loaded and called by the user program.

When installation is complete, the data blocks listed below are set up for the user (tool management user interfaces) and another data block is set up for the tool management FCs the next time the PLC is booted. The length of the data blocks is derived from the installation parameters of the tool management (softkey Create PLC data).

Sample sequence for tool change

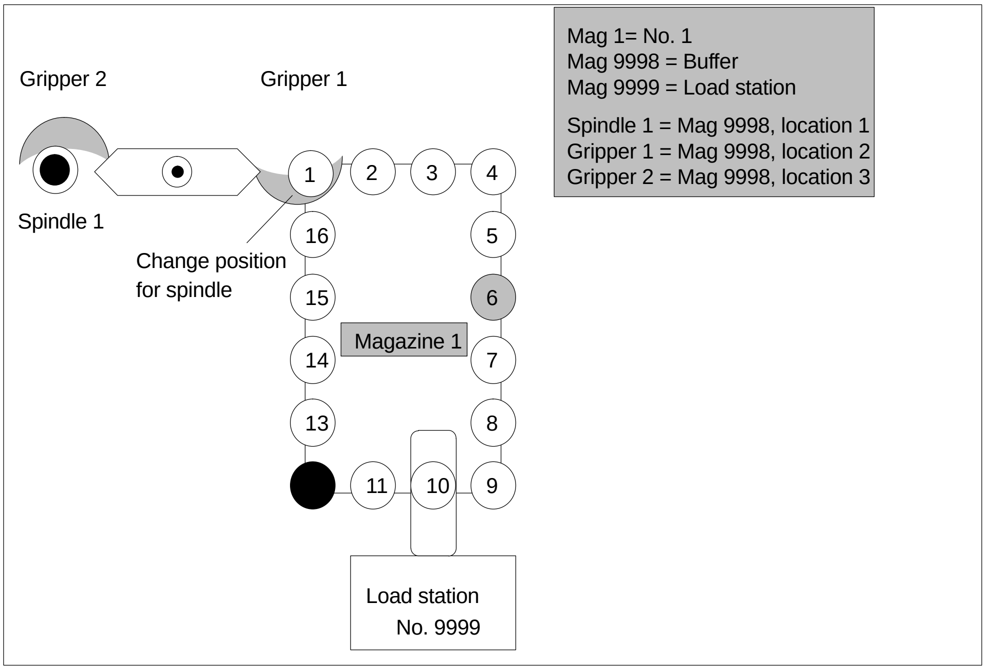


Fig. 4-23 Example of a magazine with gripper and load station

The tool “Drill120” is stored in location 6 while location 12 is reserved for the spindle tool to be replaced.

1. T=“Drill120” is programmed in the part program
   Output to the PLC:
   “PREPARE CHANGE” DBB(n+0) bit 2=1
   (move new tool from mag1, location 6 to mag9998, location 1 and old tool from mag9998, location 1 to mag1, location 12.
2. Location 6 is moved to the tool change position.
3. The tool in location 6 is transferred to gripper 1. “PREPARE CHANGE” DBB(n+0) bit2 is reset to zero by the user program. The new position (9998, 2) of the new tool (“Drill 120”) is communicated with status 1 via FC 8. The old tool remains in position 9998, 1. FC 8 resets bit 0.0 in DB 72.
   The magazine is moved to the change position with location 12 in order to receive the old tool.
4. M06 appears in the part program
   Output to the PLC: “CHANGE” DBB(n+0) Bit 1=1
   No new tool positions are entered in the interface on output of the M06 command. This can be carried out by the user program if the positions change.
5. The PLC user program executes the tool change to the spindle. The old tool is moved from the spindle to gripper 2. The new tool in gripper 1 is loaded on the spindle. FC8 acknowledges with status105 (position of new tool: 9998, 1; position ; position of old tool 9998, 3).

6. The (old) tool is returned from gripper 2 to magazine location 12. FC8 acknowledges with status 1 (position of new tool: 9998, 1; position of old tool 1, 12). The tool change operation is now complete. FC 8 resets bit 0.0 in DB 72.

Note

The timing of the tool change can be optimized by applying the following strategy for further processing in the part program:

Use status 1 with FC 8 in step 5 instead of status 105. The old tool is then returned to storage in step 6 with the asynchronous FC 8 transfer function (status 1, OldToolMag=9998, OldToolLoc=3, NewToolMag=1, NewToolLoc=12).

4.9.2 Description of test blocks

Overview of test blocks

Block number	Structure	Meaning
FC 40	Subprogram	Preparation of the data on a change with gripper via asynchronous transfer
FC 41	Block to be called in OB 1	Global functions (task control, command check, H decoder, ...)
FC 42	Subprogram	Supply of data for FC 8 if a task is active
DB 62	Data for active tasks Control parameters	
DB 63	Data for FC 22	
DB 64	Data for asynchronous transfer	

Test blocks for tool management

To test the tool management function, blocks FC 40, FC 41 and FC 42 and data blocks DB 62, DB 63 and DB 64 must be loaded to the PLC. FC 41 (without parameters) must also be called in the organization block 1. The following overall procedure is implemented by integrating these blocks.

1. The tool management function is activated (acknowledgement of tasks) by programming H9001 in the first channel (and deactivated with H9000).
   The system can also be activated by setting data bit DB62.DBX 15.7. The initial setting when the PLC is rebooted is H9000.
   The other functions can only be used once the system has been activated via H9001.

2. The direction selection function (FC 22) can be activated with the machine control panel (MCP) key above the rapid traverse override key (i.e. the normal MCP connected via FC 19 or FC 25). Data must be written to data block DB 63 (e.g. via the variable status) before the function is activated.

Structure of data block DB 63:

Input parameters

DBW 0 =	Magazine number
DBW 2 =	Setpoint position
DBW 4 =	Actual position
DBW 6 =	Offset for special positioning

Output parameters

DBW 8 =	Differential position (shortest path)
DBB 1 =	Rotation in CW direction == 1
DBB 1 =	Rotation in CCW direction == 1
DBB 1 =	Position reached
DBB 13 =	Error == 1

If an error (e.g. parameterizing error) occurs, the LED for the key lights up.

3. Every user interface (DB 71 to DB 73) is scanned for active status by block FC 41.
   If an interface is active, a transfer with new positions (usually target positions) and status information “1” (completed) is passed to the NCK immediately.
4. If H9003 is programmed in the first channel (equivalent to data block DB 62. DBX 15.6 set), the transfer operation described in paragraph 3 is not executed until the MCP key above the minus direction key is actuated.

This permits changes to the transfer values via status function. The function is deactivated via H9002 (default setting). The transfer values are provided in data block DB 62

Input parameters:

DBB 0	= Task identifier (1, 2, 3)
DBB 1	= Task number

(make changes only in DBW 2 to DBW 10)

DBW 2	= Magazine for new tool
DBW 4	= Location for new tool
DBW 6	= Magazine for old tool
DBW 8	= Location for old tool
DBW 10	= Status information (see description of FC 8)

Output parameters:

DBW 12	= Error

If an error occurs, the LED for the key lights up.

The following functions are implemented for command acknowledgement in DB 71, DB 72, DB 73:
Loading/unloading/relocation:
The required target positions are acknowledged with status 1 via FC 8.
Position:
The required target position is acknowledged with status 5 via FC 8 because the tool remains in the magazine.
Prepare change (spindle interface):
“New tool” remains at the original location,
“Old tool” remains in the spindle.
Special treatment is implemented for T0 or empty spindle.
Acknowledgement is with status = 1 via FC 8.
Change (spindle interface):
“Old tool” is transferred to allocated magazine location,
“New tool” is loaded into the spindle.
Acknowledgement is with status = 1 via FC 8.
Special treatment is implemented for T0 or empty spindle.
Change (turret interface):
Acknowledgement is via FC 7.
Acknowledgement with DB62.DBX 15.4 = 1 is optional via FC 8 with status = 1.

5. Values other than zero can be set in DB62.DBW 20 and DB62.DBW 22. DB62.DBW 20 means the spindle number and DB62.DBW 22 the buffer number of a gripper assigned to the spindle.

   It is thus possible to automatically allow for a gripper *located between a spindle and a magazine in the acknowledgement.*

   The following sequence is implemented (only for spindle as change position, M06 setting as change command):

The procedure for preparation is identical to “normal operation”.
The “New tool” remains in the magazine,
the “Old tool” remains in the spindle.
The “Old tool” must continue to machine.

On the change command:
“New tool” is loaded to spindle,
“old tool” is transferred to gripper.
An asynchronous transfer is used to move the “Old tool” to the suggested magazine location.
A manual acknowledgement is required for this purpose.

6. Asynchronous transfer
(changes in a tool location can be communicated without an NCK task)
DB 64 can be used to communicate a change in position of a tool to the tool management function in the NCK.
The position of the tool was changed by the PLC. Entries must be made in DB 64 (e.g. via variable status).
The asynchronous transfer can then be started with DB64.DBX 14.0 = 1.
The data in DB62.DBX 15.4 = 1 can be used to select the asynchronous transfer with location reservation.
This corresponds to TaskIdent = 5.
If value 0 is stored in the data specified above,
TaskIdent = 4 is activated.
Input parameters:

DBB 1	=	Associated NC channel number
DBW 2	=	Original magazine of tool
DBW 4	=	Original location of tool
DBW 6	=	Target magazine of tool
DBW 8	=	Target location of tool
DBW 10	=	Status information (see description of FC 8)
Only status	=	1 and Status = 5 are permitted

Output parameter:

DBW 12	=	Error

Note

If incorrect values are communicated from the NCK, the following error signals causing PLC stop are output and either displayed via the MMC or entered in the diagnostics buffer of the PLC.

Alarm 400604:
In function 4 the stated magazine is not a turret.
Remedy: Machine data (change with M06 command).

4.9.3 Delete pending tasks (SW 4)

During start-up, it is possible to terminate a communication sequence initiated by the NC and subsequently interrupted; this is done via the PLC task “Delete active task” (DB10.DBX105.0).

This function deletes active tool management tasks (see activating the NC). The NC tool management is reset according to defined specifications.

This function allows the operator to intervene directly, e.g. in order to remove a tool from the gripper if a tool change was to have taken place or if no acknowledgement was received from the PLC program.

Note

Please ensure that the data consistency in the NC remains.

Secondary conditions

The “Delete active task” function can be activated only if the NC is in the “Channel not active” state.

4.9.4 Start-up of ShopMill tool management in the PLC

Overview

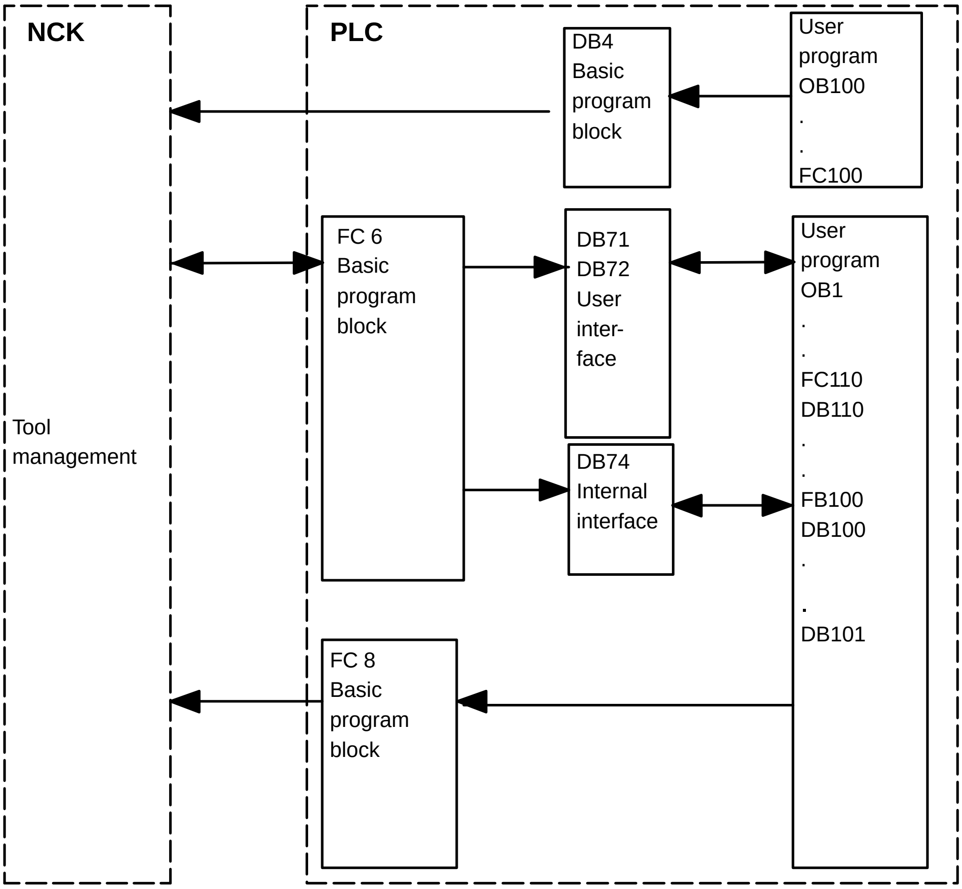


Fig. 4-24 Overview of ShopMill tool management

FC 6 supplies data blocks DB 71/72 with information for the new and old tools. This block is called up by the basic program and must not be called additionally in the user program.

So that the tool management always knows where the tool is currently located, each change in position of a tool must be communicated to the tool management via the FC 8 (transfer block). FC 8 (transfer block) is called by the user program (FB 110).

Data blocks DB 71/72 and 74 are set up automatically. The lengths of the data blocks are determined by the start-up parameters for tool management in DB 4. These are assigned by the user program (FC 100).

Start-up sequence

Requirements

- The MMC has been started up and a link set up to the NC.
- The NCK has been started up with the NC machine data for ShopMill.
- The standard basic program is loaded.

Execution

- Adapt and compile one of the following source files:

  – TM_WO_GR.STL (data transfer without dual gripper)
  – TM_W_GR.STL (data transfer with dual gripper)

  Source files TM_WO_GR.STL and TM_W_GR.STL contain the following blocks:

  – FC100 (block for tool management configuration)
  – FB 110, DB110 (data transfer blocks of ShopMill tool management). The data transfer blocks must be adapted to suit the specific features of the machine.

- Call blocks in OB1 and OB100:

  – Call FC100 in OB100 (before FB 1)
  – Call FB110 in OB1 (after FC 30)
  The blocks must be called in the correct order.

- Load blocks to the PLC

Description of blocks

FC 100

Block FC 100 transfers the configuration data of the tool management system to DB 4 and must be called in OB 100.

The configuration files are preset for 2 load points (DB 71) and one spindle (DB 72).

Parameter “RealMagLoc” of FC 100 (number of locations in real magazine) must be assigned when FC 100 is called so that FC 22 (direction selection) can be used.

FB 110

Block FB 110 controls data communication of the tool management system in normal operation. It must be called in OB 1. Data block DB 110 (instance DB) must be loaded.

The block contains the following functions:

- Acknowledge load/unload/relocate operations for the 1st load point
- Acknowledge load/unload operations for the 1st load point
- Acknowledge prepare/change for the 1st spindle
- Abort, i.e. acknowledge the above functions negatively

The acknowledgement of these functions can be enabled by the PLC via input parameters of FB 110, e.g. confirm load/unload operations via customer key.

Data transfer without dual gripper

FB 110 from the AWL source file TM_WO_GR.STL can be used for data communication without dual gripper.

In this case, the tool is changed from the magazine to the spindle in one step. The tool is loaded directly from the magazine to the spindle.

Input parameters of FB110:

Signal	Type	Default setting	Remarks
Prepare_IF1	BOOL	TRUE	Enable preparation for interface 1 (spindle 1)
Change_IF1	BOOL	TRUE	Enable tool change for interface 1 (spindle 1)
Load_IF1	BOOL	TRUE	Enable loading for interface 1 (load point 1)
Unload_IF1	BOOL	TRUE	Enable unloading for interface 1 (load point 1)
Swap_IF1	BOOL	TRUE	Enable relocation for interface 1 (load point 1)
Load_IF2	BOOL	TRUE	Enable loading for interface 2 (load point 2)
Unload_IF2	BOOL	TRUE	Enable unloading for interface 2 (load point 2)
Reset_IF	BOOL	TRUE	Abort for one of the functions mentioned above

Note

Machine data 9673: CMM_TOOL_STATION defines which interface is loaded or unloaded.

Data transfer with dual gripper

FB 110 from the AWL source file TM_W_GR.STL can be used for data communication with dual gripper.

In this case, the tool is changed from the magazine to the spindle in two steps. The tool is transferred from the magazine to the gripper first and from there to the spindle.

Inputs parameters of FB110 from TM_W_GR.STL:

Signal	Type	Default setting	Remarks
Prepare_IF1	BOOL	TRUE	Enable preparation for interface 1 (spindle 1)
Change1_IF1	BOOL	TRUE	Enable tool change for interface 1 (spindle 1)
Change2_IF1	BOOL	TRUE	Enable tool change step 2 (via gripper 1/2) for interface 1 (spindle 1)
Load_IF1	BOOL	TRUE	Enable loading for interface 1 (load point 1)
Unload_IF1	BOOL	TRUE	Enable unloading for interface 1 (load point 1)
Swap_IF1	BOOL	TRUE	Enable relocation for interface 1 (load point 1)
Load_IF2	BOOL	TRUE	Enable loading for interface 2 (load point 2)
Unload_IF2	BOOL	TRUE	Enable unloading for interface 2 (load point 2)
Reset_IF	BOOL	TRUE	Abort for one of the functions mentioned above

Signal description

The table below gives an overview of the data blocks that are used for data management.

DB 71	for loading/unloading points
DB 72	for spindle as change position

DB 73	for turret as change position
DB 74	Internal data block for tool management

If magazine, buffer or load position data are changed in the installation branch, then:

1. Press softkey “Create PLC data” and
2. Delete data blocks DB 71 to DB 74 and restart (cold) the PLC.

Example 1: Machine with disk-type magazine

Disk-type magazine with 16 locations. The tool is changed directly into the spindle.

NC machine data 18086 for ShopMill tool management is set:

MM_NUM_MAGAZINE_LOCATION = 22

Adapt the configuration file

Adapt configuration file TM_WO_GR.8X0 (configuration without dual gripper) at the places in **bold print** and load it to the NC.

```
%_N_TO_TMA_INI
CHANDATA (1)
;-----------------
;Magazine configuration
;-----------------
;Delete old data
;-----------------
$TC_MAP1[0]=0
$TC_DP1[0,0]=0
;Type of search strategy
;-----------------
$TC_MAMP2=257  ;Search for active tool from 1st location forwards
;-----------------
Magazine definition
;-----------------
;Real magazine
$TC_MAP1[1]=1  ;Magazine type (1: chain)
$TC_MAP3[1]=17 ;Magazine status (17: Active magazine, enabled for loading)
$TC_MAP6[1]=1  ;Number of tiers in magazine
$TC_MAP7[1]=16 ;Number of magazine locations

;Buffer magazine
$TC_MAP1[9998]=7              ;Magazine type (7: Buffer)
$TC_MAP3[9998]=17
$TC_MAP6[9998]=1
$TC_MAP7[9998]=1              ;Number of buffer locations (^: spindle)
;Load magazine
```

```
$TC_MAP1[9999]=9           ;Magazine type (9: Load magazine)
$TC_MAP3[9999]=17
$TC_MAP6[9999]=1
$TC_MAP7[9999]=2           ;Number of load point
;---------------------
;Locations of real magazine
;----------------------
;Location no. 1
$TC_MPP1[1,1]=1            ;Location kind (1: Magazine location)
$TC_MPP2[1,1]=1            ;Type of location
$TC_MPP3[1,1]=1            ;Consider adjacent location (1: ON)
$TC_MPP4[1,1]=2            ;Location status (2: Free)
$TC_MPP5[1,1]=1            ;Location kind index (1: Location no. 1)
.
.
;Location no. 16
$TC_MPP1[1,16]=1           ;Location kind (1: Magazine location)
$TC_MPP2[1,16]=1           ;Location type
$TC_MPP3[1,16]=1           ;Consider adjacent location (1: ON)
$TC_MPP4[1,16]=2           ;Location status (2: Free)
$TC_MPP5[1,16]=16          ;Location kind index (16: Location no. 16)
;-----------------------------
;Locations of buffer magazine
;------------------------------
;Spindle
$TC_MPP1[9998,1]=2         ;Location kind (2: Spindle)
$TC_MPP2[9998,1]=0         ;Location type
$TC_MPP3[9998,1]=0         ;Consider adjacent location
$TC_MPP4[9998,1]=2         ;Location status (2: Free)
$TC_MPP5[9998,1]=1         ;Location kind index (1: Location no. 1)
;------------------------------
;Locations of load magazine
;------------------------------
;1. Load station
$TC_MPP1[9999,1]=7         ;Location kind (7: Load point)
$TC_MPP2[9999,1]=0         ;Location type
$TC_MPP3[9999,1]=0         ;Consider adjacent location
$TC_MPP4[9999,1]=2         ;Location status (2: Free)
$TC_MPP5[9999,1]=1         ;Location kind index (1: Location no. 1)

:= 2 Load station
$TC_MPP1[9999,2]=7         ;Location kind (7: Load point)
$TC_MPP2[9999,2]=0         ;Location type
$TC_MPP3[9999,2]=0         ;Consider adjacent location
$TC_MPP4[9999,2]=2         ;Location status (2: Free)
$TC_MPP5[9999,2]=2         ;Location kind index (2: Location no. 2)
;--------------------------------------------------
;Distances between load points/buffers and real magazine
;--------------------------------------------------
```

```
$TC_MDP2[1,1] = 0          ;Spindle
$TC_MDP1[1,1]=0            ;1st load point
$TC_MDP2[1,2]=0            ;2nd load point

M17
```

Create TM blocks

Compile STL source TM_WO_GR.STL (data communication without dual gripper). Then load the generated blocks FC 100, FB 110, DB 110 to the PLC and call OB1 and OB 100.

Call FB 110 in OB 1

```
ORGANIZATION_BLOCK OB 1
VERSION: 4.4

VAR_TEMP
        OB1_EV_CLASS:       BYTE;
```

4.10 Start-up of code carrier

For details about code carrier system, see also Section 3.12.

4.10.1 Description of code carrier files

If your machine is provided with a code carrier for back-up of the tool data, you must start-up this system separately.

Start-up is performed as follows:

- DDE server (e.g. Balluf.exe, Bilz.exe, ...)

- ini file (e.g. Balluf.ini, Bilz.ini...)
  Set the appropriate "ini" file (e.g. Bilz.ini) to the "manufacturer-specific values" according to Operator's Guide supplied by manufacturer of code carrier system (format, check digit, coding, server parameters, reader station, code carrier capacity, ...).
- Create a conversion file (e.g. wkonvert.txt) for the data from code carrier.

The code carrier system is connected to the MMC via an RS232 C (V.24) interface, for example. The code carrier must be entered in file c:\user\mmc.ini or c:\user\mmc2.ini (SW 4 and later). Only systems for which a DDE server file is available may be entered. The server file is an EXE file and must be created for specific manufacturers for each code carrier system. The code carrier is activated in file c:\user\mmc.ini or c:\user\mmc2.ini.

DDE server

In SW 4 and later, the system directory structure has been modified to prevent user changes from being lost when the SW is upgraded. There is a clear distinction between

- the MMC system software and
- customer-specific expansions.

The following two distinct directories are provided for the code carrier system:

- **mmc2:**
  Pure system directory in which “Read only“ files are stored. The original software version is stored in this directory.
- **user:**
  User directory in which customized settings deviating from the original ini files are stored. This directory contains only modifications to the appearance of the operator interface that have been made by settings on the interface itself (e.g. language setting, file selection, view of File Manager,...). As a general rule, the parallel directories to mmc2 contain only ini file entries which deviate from the original settings in mmc2.

Note

SW 4.2 and later
None of the ini files in directory “mmc2” must be modified.

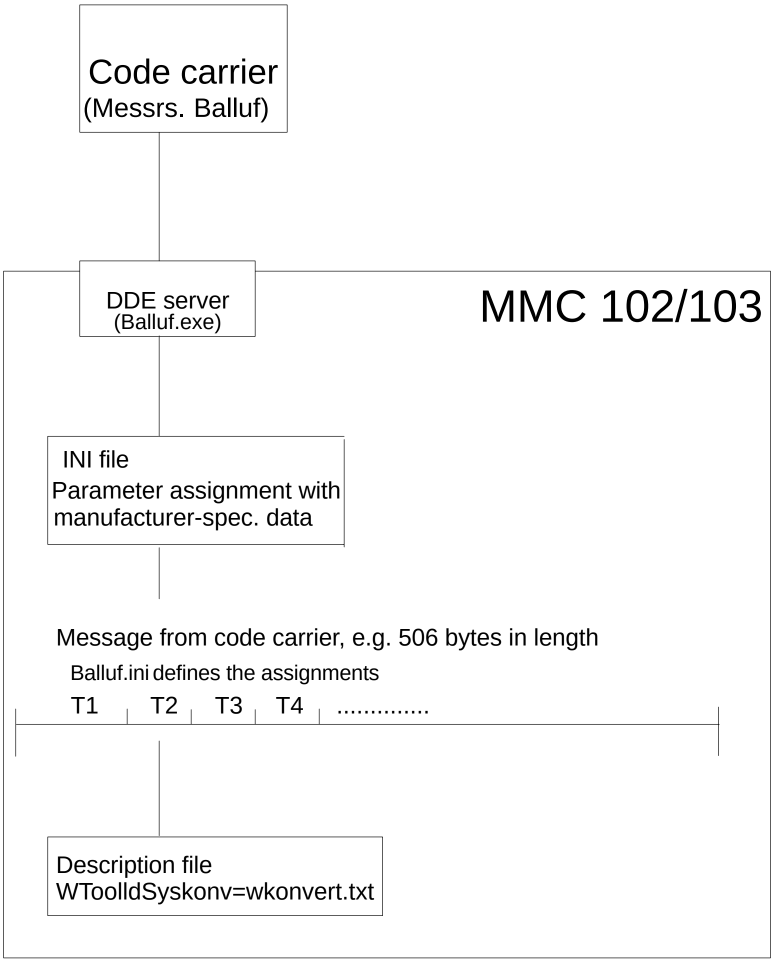

Code carrier
(Messrs. Balluf)
DDE server
(Balluf.exe)
MMC 102/103
INI file
Parameter assignment with
manufacturer-spec. data
Message from code carrier, e.g. 506 bytes in length
Balluf.ini defines the assignments
T1
T2
T3
T4
....
Description file
WToolIdSyskonv=wkonvert.txt

Fig. 4-25 Code carrier connection

ini file

An “ini file” exists for every “exe file”. Manufacturer-specific information about the code carrier system is entered in this “ini file”. The manufacturer’s manual provides a description of the parameters contained in the “ini file”.

e.g. Balluf.ini file:

```
; Description of parameters, see BALLUF TOOL DIALOG SYSTEM
; TDSi Operator’s Guide BA 200 for TDS Stations

[BIK parameters]
Format = T      ; = A        ASCII format
                ; = B        BCD format
                ;  =C        BCD format of data on data carrier
                ;  =T        Transparent format
```

```
Check digit =   ; =0          No CD
                ; =1          CD generated during writing + checked during
                              reading
                ; =2          Incorrect CD ignored during reading
                ;=4, 5, 6     As for 0, 1, 2 CD visible to user, included in trans-
                              mission

Code = 3        ;=0           4/12-bit coding
                ; =1          8-bit coding
                ; =2, 3       As for =0, =1, but without page structure
                ; =4, =5      As for =2, =3, but without entry and evaluation of a
                               CD per internal page

[Server parameters]
Data conversion=1             ; =0 None      =1 Intel-Hex
Trace=0                       ; =0 no Trace  =1 Trace on
Com=1                         ; = COM x interface, change connection to
                              ; serial interface on MMC 102/103 here if nec.

[Services]
Service=ToolIdentSystem

[Topics]
; Each topic specifies a reader station
Topic1=Unit1
Topic2=Unit2
Topic3=Unit3
Topic4=Unit4

[MISC]
MAX_CC_CAPACITY=506           ;Maximum code carrier capacity in bytes
```

4.10.2 Structure of description file

Description file

The data on the code carrier are stored in a specific order which is defined when the code carrier system is installed. In order for the tool management function to be able to read and write this flow of data, a conversion specification is available in the form of a description file containing precisely defined tool and cutting edge dialog data. Only these dialog data can actually be processed by the data management. All the other data on the code carrier must not be assigned to any dialog variables as otherwise they will not be processed. However, an OEM application can access these data.

The description file can be generated as an ASCII file and modified using a standard editor. The file name must be entered in mmc.ini with WToolIdSysKonv = **wkonvert.txt.**

Tool dialog data

The tool dialog data are defined as follows:

Dialog variable	Data type	Description
T1	String	Tool name, max. of 32 characters
T2	Integer	Duplo number
T3	Integer	Number of edges (max. of 9)
T4	Integer	Tool size left in half locations
T5	Integer	Tool size right in half locations
T6	Integer	Tool size top in half locations
T7	Integer	Tool size bottom in half locations
T8	String	Magazine location type, max. of 32 characters
T9	Integer	Status, bit combination consisting of: 1 = Active tool 2 = Tool is enabled 4 = Tool is disabled 8 = Tool is measured 16 = Warning limit reached 32 = Tool is being changed 64 = Tool is fixed-location coded
T10	Integer	Type of tool monitoring 0 = none 1 = tool life monitoring 2 = workpiece count
T11	Integer	Type of tool search

Cutting edge dialog data

Dialog variable	Data type	Description
C1	Integer	Sub-type
C4	Integer	Cutting edge position
		Geometry tool length compensation
C5	Double	Length 1
C6	Double	Length 2
C7	Double	Length 3
		Geometry tool radius compensation
C8	Double	Length 1
C9	Double	Length 2

Dialog variable	Data type	Description
C10	Double	Radius 1
C11	Double	Radius 2
C12	Double	Angle 1
C13	Double	Angle 2
		Wear tool length compensation
C14	Double	Length 1
C15	Double	Length 2
C16	Double	Length 3
		Wear tool radius compensation
C17	Double	Length 1
C18	Double	Length 2
C19	Double	Radius 1
C20	Double	Radius 2
C21	Double	Angle 1
C22	Double	Angle 2
		Base/adapter dimension tool length compensation
C23	Double	Base length 1
C24	Double	Base length 2
C25	Double	Base length 3
C26	Double	Clearance angle
C27	Integer	Overhead use
C28		Cutting edge number (not implemented to date)
C29	Integer	Tool life in minutes
C30	Integer	Prewarning limit tool life in minutes
C31	Integer	Number of pieces still to be machined
C32	Integer	Prewarning limit number of pieces still to be machined
C33	Double	Required tool life in minutes
C34	Integer	Required workpiece count
C35	Double	Wear prewarning limit
C36	Double	Wear
C37	Double	Required wear

Dialog variable	Data type	Description
C38	Double	Tool life in minutes
C39	Double	Tool life prewarning limit

Dialog variables C2, C3 and C28 are managed internally only.

Note

C38 and C39 may be used only as alternatives to C29 and C30.

User tool parameters and the new monitoring parameters are now also utilized for code carriers. The following new configuring variables are available for file wkonvert.txt:

A1 – A10: Tool OEM1 to OEM10 (see $TC_TPCx[t])
U1 – U10: Cutting edge OEM1 to OEM10 (see $TC_DPCx[t,d])
S1 – S10: Monitoring OEM1 to monitoring OEM10
(see $TC_MOPCx[t,d])

Data types

The data types of the data variables are defined:

- Integer: Value range –32768 to 32767
- Double: Floating point double precision
- String: String of ASCII characters

Keywords

The code carrier data are assigned to the dialog data in the code carrier description file. The description file can be generated as an ASCII file using a standard editor. The code carrier description file is divided into lines. Each line begins with one of the following **keywords**.

Inverted comma

The **’** (inverted comma) denotes the start of a comment. The following characters up to the end of the line are skipped.

Example:

```
’ This is a comment
```

Note

This form of inverted comma as a comment designation is used only in the description file for code carriers. The beginning of a comment is otherwise denoted by a semi-colon.

Datalen

DATALEN=CONST | VARIABLE 0x<separator>

The following data either have a constant (**CONST**) or variable (**VARIABLE**) data length. Data of a variable length are concluded with 0x<separator>.

Example:

```
DATALEN=VARIABLE 0x0A         ’ Variable data length, separator LF
```

DEFINE_KEY-WORD

DEFINE_KEYWORD=<keyword> <value><keyword> := any user keyword for marking a new section of data on the code carrier

<value> := “<string>” or 0x<hexvalue>

Definition of the keyword <keyword> with the value <value>

Example:

```
DEFINE_KEYWORD=DATA _OEM      “OEM”
DEFINE_KEYWORD=DATA_SIN840D   0x840D
```

<keyword>

A keyword defined by **DEFINE_KEYWORD** that identifies a new data section on the code carrier. The item following **<keyword>** in the code carrier description file must contain the value <value> defined by **DEFINE_KEYWORD**.

Item

Item<n>**=<**line**>**

<n> := Consecutive number of code carrier date, ascending from 1 without gaps
<line> := <(max.) length in bytes> <code carrier dataa format>code carrier data format> <dialogvariable>
<code carriergerdatenformat>code carrier data format> :
<dialogvariable> : Assignment of code carrier to dialog datum

If a user **keyword** is defined directly before **Item<n>**, <dialog variable> has the value **<keyword>**

Conversion specification for code carrier data <n>

Example:

```
Item1    32    ASCII    T3          ’ Convert tool identifier to/from
                                    ’ tool dialog data 3
```

BItem

BItem<n>**=<**line**>**

<n> := serial number of code carrier data within **block**<i>, ascending contiguously starting at 1

<line> := analog **Item**<n>

Conversion specification for code carrier data <n> within a block. If a tool dialog data T<n> is assigned to the code carrier data, the first value of the code carrier data in the block is assigned to the dialog data.

Example:

```
BItem1   1     BCD      C1, T2     ’ Convert subtype to/from cutting
                                   ’ edge dialog data 1 and tool dialog data 2
                                   ’ (1. value of block is relevant for T2)
```

Block

Block<n> < repeat specification >

<n> := serial number of the block, ascending contiguously, starting at 1
<repeat specification> := * <const> | * Item<n> | **CONTIGUOUS** BItem1

A block of data **BItem**<n> follows (up to keyword **End_Block<n>**), that is stored on the code carrier according to <repeat specification>.

Note

In the case of **Block**<n> **CONTIGUOUS** BItem1 the count variable BItem1 (cutting edge number) can also begin with a value > 1.

However, the following values must ascend contiguously.
When the code carrier data are written, the block data (cutting edge data) are preset with the basic values up to the starting value of the count variables (1st edge).

In the case of **Block**<n> * **Item**<n>, Item<n> must be defined before Block<n>.

Example:

```
Block1 * 2               ’ Repeat Block1 twice
Block1 * Item6           ’ Repeat Block1 by the number of times specified
by Item6
Block1 CONTIGUOUS BItem1  ’ Read Block1 repeatedly until the count variable
                         ’ BItem1 no longer supplies a value incremented
                         ’ by 1.
                         ’ Write Block1 as many times as defined by the
                         ’ value of the dialog variable assigned to BItem1.
End_Block<n>
```

End_Block

End identifier for a data block defined with **Block**<n>

Code carrier data formats

The following code carrier data formats are supported:
(cf. <code carrier data format> for **Item** /**BItem**)

Data format	Explanation
ASCII	ASCII character set
INT	16-bit integer (Intel format) • Value range –32768 <= INT <= +32767
FPX2	16-bit integer (SINUMERIK 850) • Value range –32768 <= FPX2 <= +32767 • Low byte to high address (reverse of Intel format)
BCD	• Binary coded decimal (with a sign and decimal point where appropriate) • Non-relevant decades are preset to 0 left-justified
BCD_SIN850	BCD with the following conventions (SINUMERIK 850): • Each BCD number fills 12 decades • Sign always in the 3rd decade Dekade • 0 = positive / 1 = negative • Non-relevant decades are preset to 0 • Value range: Max. of 8 digits with decimal point, max. of 9 digits without decimal point

Assignment between code carrier data and dialog data

The conversion specification for **Item<n>** or **BItem<n>** includes the assignment to none/one/several dialog variables, if necessary, with a conversion specification which is described in detail in this section.

The general conversion specification for **Item<n>** or **BItem<n>** is:

(B)Item<n>**=<**line**>**

<n> := serial number of code carrier data,
ascending contiguously

<line> := <(max.) length in bytes><code carrier data format><dialog variable>

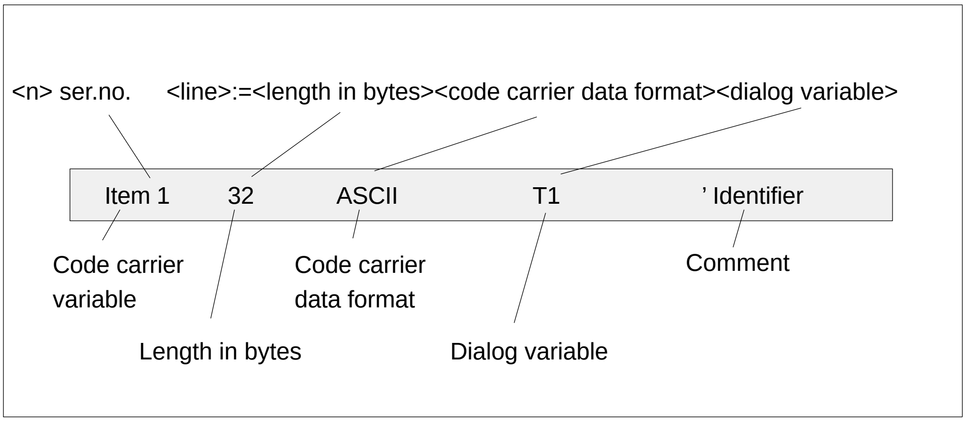


Fig. 4-26 Conversion specification

Dialog variable

<**dialog variable**> := <dvar1>[=(<uv>)] [, <dvar2>[=(<uv>)] [,<dvar3>[&<dvar4>]=(<uv>)] [, <dvarN>[=(<uv>)]

<dvar> := T<index> | C<index> | –
T = Tool data,
C = Cutting edge data,
index = Index within tool/cutting edge dialog data
– = No assignment to a dialog variable

< dvar1>&<dvar2>=<uv> : Conversion specification applies to <dvar1> and <dvar2>

uv := <arithm. Op1> [<arithm. Op2>] .. [<arithm. OpN>]
arithm. Op := +<const> | –<const> | *<const> | /<const>

Example:

```
T2=(*10), T3=(/100 + 10)
```

or

uv := <substitute1> [<substitute2>] .. [<substituteN>]
substitute := <const1> [, <const2>] ..
[, <constN>] **^** <constM>
or
<const1..const2> ^ <const3>
const1 = lower limit value,
const2 = upper limit value

Note

When converting the dialog variable to the code carrier variable on writing, if there are several left operands, the right operand is converted in the first left operands!

Example:

```
T2=(20..29 ^ 120 40,50 ^ 130)
```

The code carrier variable with the value 25 is converted to dialog variable T2 with the value 120 (read). Dialog variable T2 with value 120 is converted to code carrier variable with the value 20 (write).

or

uv := <Tetn>
Tetn := nth tetrad in byte sequence
Byte1, = Tet1 and Tet2
Byte2, = Tet3 and Tet4

Distribution of tetrads of code carrier variables (in BCD format) among dialog variables

Example:

```
T5=(Tet1), T6=(Tet2), T7=(Tet3), T8=(Tet4)
```

If the code carrier variable has value 0x1234, for example, dialog variable T5 is assigned the value 1, dialog variable T8 the value 4

or

uv := <compare>
compare := **<** <const> [INVSIGN] **|** **<=** <const> **|** **=** <const> **|** **>**
<const> **>=** <const>

Assignment of code carrier variable to a dialog variable according to the comparison result

INVSIGN

- reading: sign of the dialog variable
- writing: invert sign of the code carrier variable

Example:

```
C1=(<0 INVSIGN) , C2=(>=0)
```

Read:

a negative code carrier variable value corresponds to dialog variable C1, a positive value to dialog variable C2; dialog variable C1 is converted to a positive value.

Write:

Dialog variable C1 is multiplied by (–1). If the value is less than 0, the code carrier-variable is assigned the value of C1, otherwise of C2.

Note

Conversion specifications are only evaluated for dialog variables of data type “integer”.

4.10.3 Adapt file “mmc.ini”

c:\user\mmc.ini

The code carrier is activated in file **c:\user\mmc.ini** e.g. (WToolIdSys=**Bilz)**

```
...

[ToolMgmt]
; TM general !!! independent of code carrier
WDBaseName=wzaccess.mdb
; Name of the database for tools and magazines on MMC
; Must be stored in the data management under /WZV.DIR/WDAT.DIR

WToolIdSys=0 ;or Balluf; or Bilz
; Identifier for code carrier system
;0              means :“No code carrier active”
; Specify manufacturer name (only first 5 characters!)
; Balluf        means : code carrier from Messrs. Ballu active
; Bilz          means : Bilz code carrier active
                (only manufacturer supported up to P2.2)


WToolIdSysKonv=wkonvert.txt
; Name of the conversion file used for the code carrier format.
; The file is stored in directory c:\user.

[TIS]
; Tool Identification System
; EOT for code carrier data
TIS_EOT=0x2F2F
; The end character for data on the code carrier must be entered here.
```

4.10.4 Adapt manufacturer-specific “ini file”

c:\user\Bilz.ini

Set the manufacturer-specific ini file in the directory c:\user.

Only **“Balluf.ini”** is currently available.

```
                                                                        ;
;Description of parameters, see BALLUF TOOL DIALOG SYSTEM
;TDSi Operator’s Guide BA 200 for TDS Stations

[BIK parameters]
Format= T          ;=A ASCII format
                   ;=B BCD format
                   ;=C BCD format of data in data carrier
                   ;=T Transparent format

Check digit =      ;=0 No CD.
                   ;=1 CD generated during writing + checked during reading.
                   ;=2 incorrect check digit ignored during reading.
                   ;=4,=5,=6 like 0,1,2 CD visible to user,
                   ;included in transmission.

Code = 3           ;=0 4/12-bit coding.
                   ;=1 8-bit coding.
                   ;=2 =3 as =0, =1, but without page structure.
                   ;=4 =5
                   like =2, =3, but without entry and evaluation of a check digit per
                   internal page

[Server parameters]
Data conversion=1            ; =0 None       =1 Intel-Hex
Trace=0                      ; =0 no Trace   = 1 Trace on
Com=1                        ; = COM x interface, change connection to
                             ; serial interface on MMC here if nec.

[Services]
Service1=ToolIdentSystem

[Topics]
; Each topic specifies a reader station

Topic1=Unit1
Topic2=Unit2
Topic3=Unit3
Topic4=Unit4

[Misc]MAX_CC_CAPACITY=506     ;Maximum code carrier capacity in bytes
```

4.10.5 Create a conversion file

Name of the conversion file

Create a conversion file for SINUMERIK 840D. The file name must be entered in c.\user\mmc.ini when WToolIdSysKonv = **wkonvert.txt.**

Examples for SINUMERIK 840D

The name of the file is, for example, **wkonvert.txt**

Code carrier variable	**Length (bytes)**	**Data format**	**Dialog variable**	Comment
Item1	32	ASCII	T1	’ Identifier
Item2	3	BCD	T2	’ Duplo

Item3	2	BCD	T4=(Tet1),T5=(Tet2),T6=(Tet3),
			T7=(Tet4)	

’ Tool size: Left, right, top, bottom				
Item4	32	ASCII	T8	’ Location type
Item5	1	BCD	T9	’ Status
Item6	1	BCD	T3	’ No. of edges
Item7	1	BCD	T10	’ Type of tool monitoring
Item8	1	BCD	T11	’ Type of tool search
’Cutting edge data Block1 * Item6				
Bitem1	2	BCD	C1	’ Subtype, type
Bitem2	1	BCD	C4	’ Tool point direction
’ TL length comp.				
Bitem3	4	BCD	C5	’ Length 1
BItem4	4	BCD	C6	’ Length 2
BItem5	4	BCD	C7	’ Length 3
’TL radius comp.				
BItem6	4	BCD	C8	’ Length 1
BItem7	4	BCD	C9	’ Length 2
BItem8	4	BCD	C10	’ Radius 1
BItem9	4	BCD	C11	’ Radius 2
BItem10	4	BCD	C12	’ Angle 1
BItem11	4	BCD	C13	’ Angle 2
’Wear length compensation				

Code carrier variable	**Length (bytes)**	**Data format**	**Dialog variable**	Comment
BItem12	4	BCD	C14	’ Length 1
BItem13	4	BCD	C15	’ Length 2
BItem14	4	BCD	C16	’ Length3
’Wear radius compensation				
BItem15	4	BCD	C17	’ Length 1
BItem16	4	BCD	C18	’ Length 2
BItem17	4	BCD	C19	’ Radius 1
BItem18	4	BCD	C20	’ Radius 2
BItem19	4	BCD	C21	’ Angle 1
BItem20	4	BCD	C22	’ Angle 2
’Base dimension length compensation				
BItem21	4	BCD	C23	’ Base length 1
BItem22	4	BCD	C24	’ Base length 2
BItem23	4	BCD	C25	’ Base length 3
BItem24	4	BCD	C26	’ Clearance angle
BItem25	1	BCD	C27	’ Overhead use
BItem26	2	BCD	C29	’ Tool life in minutes
BItem27	2	BCD	C30	’ Prewarning limit tool life
BItem28	2	BCD	C31	’ Workpiece count
BItem29	2	BCD	C32	’ Prewarning limit tool life

Example for a SINUMERIK 850/880

The file name is, for example, **konv850.txt**

’File with conversion specification for SIN 850/880

DEFINE_KEYWORD=DATA_SIN840D 0x840D

’Additional SIN 840D data as from here

Code carrier variable	**Length (bytes)**	**Data format**	**Dialog variable**	**Comment**
’Tool data SIN 850/880				
Item1	2	FPX2	–	’ Magazine no.
Item2	2	FPX2	–	’ Magazine location no.
Item3	4	ASCII	T9=(0 ^ 2 10, 1000, 1100 ^ 4 100 , 110 ^ 16) ’ Identifiers	

Code carrier variable	Length (bytes)	Data format	Dialog variable	Comment
Item4	6	BCD_SIN850	–	’ T number
Item5	6	BCD_SIN850	T2	’ Duplo no.
Item6	6	BCD_SIN850	T4	’ Half location forwards
Item7	6	BCD_SIN850	T5	’ Half location backwards
’Cutting edge data SIN 850/880				
Block1 CONTINUOUS Bltem1				
Bltem1	6	BCD_SIN850	T3	’ Cutting edge number
Bltem2	6	BCD_SIN850	C1=(01..09^500 1 0..19^200 20..29^120 30..39^130) ’ Type’	
Tool offset				
Bltem3	6	BCD_SIN850	C5	’ Length 1
Bltem4	6	BCD_SIN850	C6	’ Length 2
Bltem5	6	BCD_SIN850	C10	’ Radius 1
’TL wear compensation				
Bltem6	6	BCD_SIN850	C14	’ Length 1
Bltem7	6	BCD_SIN850	C15	’ Length 2
Bltem8	6	BCD_SIN850	C19	’ Radius 1
Bltem9	6	BCD_SIN850	C29	’ Tool life
Bltem10	6	BCD_SIN850	C30	’ Prewarning tool life
Bltem11	6	BCD_SIN850	–	’ Approach identifiers
Bltem12	6	BCD_SIN850	–	’ Approach value

Example for additional data 840D

DATA_SIN840D Additional data SIN 840D

Code carrier variable	Length (bytes)	Data format	Dialog variable	Comment
’ Additional tool data SIN 840D				
Item8	2	BCD	<DATA_SIN840D> ’ Identifier SIN 840D additional data	
Item9	32	ASCII	T1	’ Identifier
Item10	2	BCD	T4=(Tet1),T5=(Tet2),T6=(Tet3), T7=(Tet4)	
’ Tool size: left, right, top, bottom				
Item11	1	BCD	T8	’ Location type
Item12	1	BCD	T10	’ Type of tool monitoring
Item13	1	BCD	T11	’ Type of tool search
Item14	1	BCD	T3	’ No. of edges
’ Additional cutting edge data SIN 840D’				
Block2 *Item11				
BItem1	1	BCD	C4	’ Tool point direction
’ Tool length compensation				
BItem2	4	BCD	C7	’ Length 3
’TL radius comp.				
BItem3	4	BCD	C8	’ Length 1
BItem4	4	BCD	C9	’ Length 2
BItem5	4	BCD	C11	’ Radius 2
BItem6	4	BCD	C12	’ Angle 1
BItem7	4	BCD	C13	’ Angle 2
’Wear length compensation				
BItem8	4	BCD	C16	’ Length 3
’Wear radius compensation				
BItem9	4	BCD	C17	’ Length 1
BItem10	4	BCD	C18	’ Length 2
BItem11	4	BCD	C20	’ Radius 2
BItem12	4	BCD	C21	’ Angle 1
BItem13	4	BCD	C22	’ Angle 2
’ Base dimension length compensation				
BItem14	4	BCD	C23	’ Base length 1
BItem15	4	BCD	C24	’ Base length 2
BItem16	4	BCD	C25	’ Base length 3

Code carrier variable	Length (bytes)	Data format	Dialog variable	Comment
BItem17	4	BCD	C26	’ Clearance angle
BItem18	1	BCD	C27	’ Overhead use
BItem19	2	BCD	C31	’ Workpiece count
BItem20	2	BCD	C32	’ Prewarning limit workpiece count

Programming 5

5.1 Overview of OPI and system variables

All the data required for the purpose of data management (e.g. to define a magazine or load a tool...) are stored in the NCK. These data can be read and written via part programs using system variables and via the PLC using FB 2 and FB 3. When configuring the machine, the user (machine manufacturer) must determine the most efficient method of reading and writing tool management data, i.e. in the PLC, the NC or in an ASUB.

Read and write access can generally be made to system variables.

When language commands are used, it may be necessary to program the “STOPRE” command.

The $TC variables do not produce a preprocessing stop.

Note

Additional information concerning OPI variables is given in the help file for the NC variable selector.

Overview

Fig. 5-1 displays an overview of all cutting edges and tool and magazine data ($TC_...) when the tool management function is active.

Note:

The sequence of system variables shown in the figure corresponds to the OPI numbering sequence.

Note

There are system variables for the OEM Siemens data. However, they are not described here because they are not significant at present.

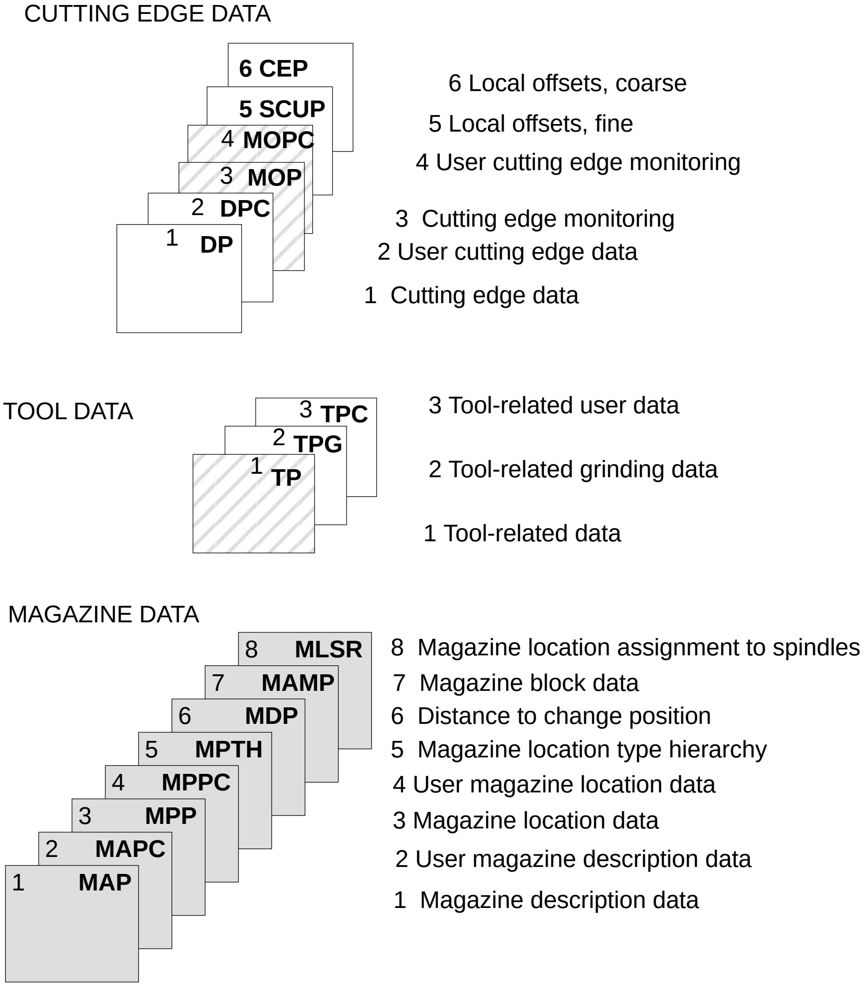


Fig. 5-1 Overview of cutting edge, tool and magazine data

The identifiers (DP,...TP,...MAP,...) are borrowed from the NC language. They are part of the names of the system parameters $TC_DP,...

Note

The gray data fields are only available if tool management is active.

Shaded data fields are available without TM function, but with monitoring function.

White data fields are available when the TM function is not active.

ADAPTER DATA
ADPT

Fig. 5-2 Adapter data

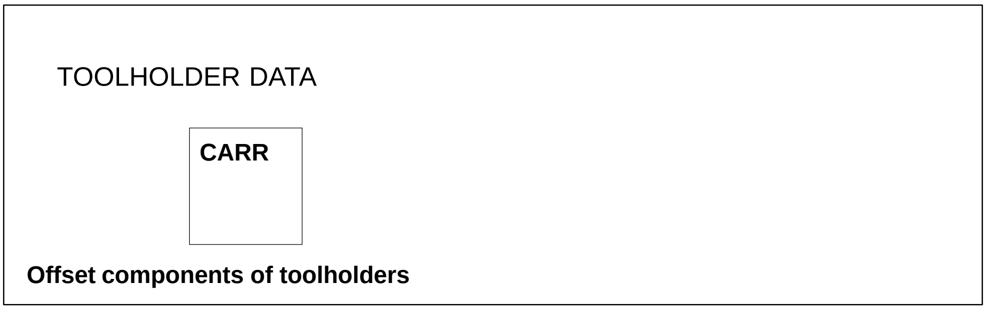


Fig. 5-3 Toolholder data

5.2 Cutting edge data

Cutting edge data

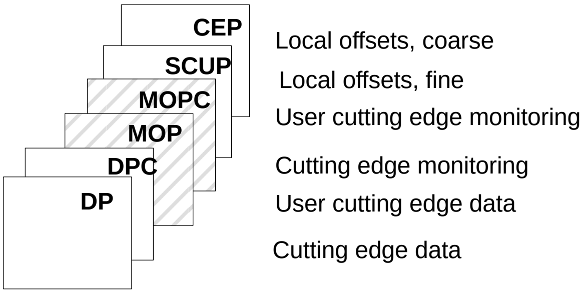


Fig. 5-4 Overview of cutting edge data

These data exist for every cutting edge that is created (D1 – D12). If tool management is active, optional cutting edge monitoring data is available in addition to the geometry and user data.

If the cutting edges are created via MMC, the D number is counted up from 1. If cutting edges are created via the NC program, the D numbers can be programmed with gaps, e.g. D1, D3, D6.

Note

Modifications to cutting edge data are displayed on the ShopMill TM operator interface only if they relate to the tool currently loaded in the spindle.

5.2.1 Cutting edge parameters

$TC_DPx[t,d]

Cutting edge parameters for geometry, technology and tool type.
Depending on the tool type, up to 25 cutting edge parameters can be programmed.

References: /FB/, Tool Compensation W1

x:	=	Parameter 1...25
t:	=	T number 1...32000
d:	=	Cutting edge number 1...12 (9 in SW 5 and earlier)

The maximum value of x is stored in the OPI variable numCuttEdgeParams in block Y.

OPI block TO

Calculation of line: (d–1)*numCuttEdgeParams+parameter no.

Calculation of column: T number

Tool offset parameters					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
$TC_DP1	INT	Tool type	edgeData	REAL	9999
$TC_DP2	Double	Cutting edge position	edgeData	REAL	0
$TC_DP3	Double	Geometry length 1	edgeData	REAL	0
$TC_DP4	Double	Geometry length 2	edgeData	REAL	0
$TC_DP5	Double	Geometry length 3	edgeData	REAL	0
$TC_DP6	Double	Geometry radius	edgeData	REAL	0
$TC_DP7	Double	Geometry corner radius (tool type 700; slotting saw)	edgeData	REAL	0
$TC_DP8	Double	Geometry length 4 (tool type 700; slotting saw)	edgeData	REAL	0
$TC_DP9	Double	Geometry length 5	edgeData	REAL	0
$TC_DP10	Double	Geometry – angle 1	edgeData	REAL	0
$TC_DP11	Double	Geometry – angle 2 for conical milling tools	edgeData	REAL	0
$TC_DP12	Double	Wear – length 1	edgeData	REAL	0
$TC_DP13	Double	Wear – length 2	edgeData	REAL	0
$TC_DP14	Double	Wear – length 3	edgeData	REAL	0
$TC_DP15	Double	Wear – radius	edgeData	REAL	0
$TC_DP16	Double	Wear – slot width b/rounding radius	edgeData	REAL	0
$TC_DP17	Double	Wear – projection k	edgeData	REAL	0
$TC_DP18	Double	Wear – length 5	edgeData	REAL	0
$TC_DP19	Double	Wear – angle 1	edgeData	REAL	0
$TC_DP20	Double	Wear – angle 2 for conical milling tools	edgeData	REAL	0
$TC_DP21	Double	Adapter – length 1	edgeData	REAL	0
$TC_DP22	Double	Adapter – length 2	edgeData	REAL	0
$TC_DP23	Double	Adapter – length 3	edgeData	REAL	0
$TC_DP24	Double	Clearance angle	edgeData	REAL	0

NCK identifier	Type	Description	OPI variable	Type	Default setting
$TC_DP25	DoubleE	1. The cutting rate value is stored here for Manualturn 2. A bit-coded value for various states of tools of type 1xx and 2xx are stored here for ShopMill.	edgeData	REAL	0
$TC_DPCE [t,d]	INT	System parameter of an offset data record with T=t and D=d containing cutting edge number CE (unique D no. or optional assignment of D no. to edge no.). Value range for permissible cutting edge numbers: 1 up to value of MD 18106.	–		0
$TC_DPH [t,d]	INT	H parameter (Y / extraCuttEdgeParams, Bit0=1	–		0

5.2.2 User cutting edge data

$TC_DPCx[t,d]

User cutting edge data
Up to 10 additional cutting edge parameters can be programmed for each cutting edge. Set with MD 18096: MM_NUM_CC_TOA_PARAM and enable with MD 18080 MM_TOOL_MANAGEMENT_MASK (set bit 2=1)

x:	=	Parameter 1...10
t:	=	T number 1...32000
d:	=	Cutting edge number 1...12 (9 in SW 5 and earlier)

OPI block TUE

Calculation of line:	(d–1)*numCuttEdgeParams_tu+parameter no.
Calculation of column:	T number

User cutting edge data					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
$TC_DPC 1	Double	CC_Cutting_Edge_ Parameter1	edgeData	REAL	0
...	Double	...	edgeData	REAL	0
$TC_DPC 10	Double	CC_Cutting_Edge_ Parameter10	edgeData	REAL	0

Note

The data are displayed in the tool management. Here you could store “Max. cutting rate”, for example, which is then evaluated in the part program.

5.2.3 Edge-related tool monitoring

$TC_MOPx[t,d]

Tool cutting edges are monitored according to tool life, workpiece count and/or wear.

x:	=	Parameter 1...15
t:	=	T number 1...32000
d:	=	Cutting edge number 1...12/D number

The maximum value of x is stored in the OPI variable numCuttEdgeParams in block Y.

OPI block TS

Calculation of line:	(d–1)*numCuttEdgeParams_ts+parameter no.
Calculation of column:	T number

Tool management monitoring data

NCK identifier	Type	Description	OPI variable	Type	Default setting
$TC_MOP1	Double	Prewarning limit tool life in min	data	REAL	0
$TC_MOP2	Double	Residual tool life in minutes	data	REAL	0
$TC_MOP3	INT	Prewarning limit for number of workpieces	data	REAL	0
$TC_MOP4	INT	Residual number of workpieces	data	REAL	0
$TC_MOP11	Double	Required tool life	data	REAL	0
$TC_MOP13	INT	Required workpiece count	data	REAL	0
$TC_MOP5	Double	Wear prewarning limit – or prewarning limit for local offset, fine	data	REAL	0

Tool management monitoring data					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
$TC_MOP6	Double	Actual wear – or actual local offset, fine	data	REAL	0
$TC_MOP15	Double	Required wear – or required local offset, fine	data	REAL	0

Formats of $TC_MOP1, $TC_MOP2 (SW 5.1 and higher)

The format of the two system parameters **$TC_MOP1, $TC_MOP2** has been changed from INT to Double to implement monitoring accurate to the nearest second.

5.2.4 User cutting edge monitoring

$TC_MOPCx[t, d]

Tool monitoring user data (edge-specific)

Up to 10 additional tool monitoring parameters can be programmed for each cutting edge. Set with MD 18098: MM_NUM_CC_MON_PARAM and enable with MD 18080 MM_TOOL_MANAGEMENT_MASK (set bit 2)

x: = Parameter 1...10
t: = T number 1....32000
d: = Cutting edge number 1....12

OPI block TUS

Calculation of line:	(d–1)*numCuttEdgeParams_tus+parameter no.
Calculation of column:	T number

Tool monitoring user data (edge-specific)					
NCK identifier	**Type**	**Description**	**OPI variables**	**Type**	**Default setting**
$TC_MOPC1	int	CC monitoring parameter	userdata	REAL	0
...	int	...	userdata	REAL	0
$TC_MOPC10	int	CC monitoring parameter	userdata	REAL	0

Note

These data are not supported by standard screen forms in SW lower than 3.x, i.e. they can only be accessed via the part program or PLC.

5.2.5 Local offsets, fine (sum offsets) (SW 5 and higher)

$TC_SCPx[t,d]

Local offsets – fine (the term “sum offsets” is also used frequently) comprise all the magnitudes of error which contribute to the total deviation between the actual workpiece and the desired workpiece dimension. The parameters of the local offsets refer to the geometric data of a cutting edge. DL stands for D Location.

x	=	Parameter for DL=1...DL=6
t	=	T number 1....32000
d:	=	Cutting edge number 1...12 / D number

OPI block TOS

Calculation of line: (d–1)*(maxnumEdgeSC*numParams_SC)+ ((EdgeSC–1)*numParams_SC)+parameter no.

Calculation of column: T number

Local offsets				
Name	**Type**	**Description**	**OPI variable**	**Type**
$TC_SCPx				
x = 13–21	Double	Can be activated with DL=1	edgeSCData	REAL
x = 23–31	Double	Can be activated with DL=2	edgeSCData	REAL
x = 33–41	Double	Can be activated with DL=3	edgeSCData	REAL
x = 43–51	Double	Can be activated with DL=4	edgeSCData	REAL
x = 53–61	Double	Can be activated with DL=5	edgeSCData	REAL
x = 63–71	Double	Can be activated with DL=6	edgeSCData	REAL
		Transformed, local offsets (fine), block TOST	edgeSCData	REAL

5.2.6 Local offsets, coarse (setup offsets) (SW 5 and higher)

$TC_ECPx[t,d]

The coarse local offsets (including setup offsets) can be set by the machine setter before the machining operation (see also $TC_SCUP).

x	=	Parameter for DL=1...DL=6
t	=	T number 1....32000
d:	=	Cutting edge number 1...12 / D number

OPI blocks TOE, TOET

Calculation of line: (d–1)*(maxnumEdge_SC*numParams_SC)+ ((EdgeSC–1)*numParams_SC)+parameter no.

Calculation of column: T number

Setup offsets				
Name	**Type**	**Description**	**OPI variable**	**Type**
$TC_ECPx			edgeECData	REAL
x = 13–21	Double	Can be activated with DL=1	edgeECData	REAL
x = 23–31	Double	Can be activated with DL=2	edgeECData	REAL
x = 33–41	Double	Can be activated with DL=3	edgeECData	REAL
x = 43–51	Double	Can be activated with DL=4	edgeECData	REAL
x = 53–61	Double	Can be activated with DL=5	edgeECData	REAL
x = 63–71	Double	Can be activated with DL=6	edgeECData	REAL
		Transformed setup offsets, block TOET	edgeECData	REAL

5.3 Tool data

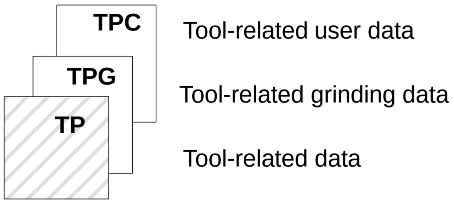


Fig. 5-5 Overview of tool data

Note

Modifications to cutting edge data are displayed on the ShopMill TM operator interface only if they relate to the tool currently loaded in the spindle.

5.3.1 Tool-related data

$TC_TPx[t]

General tool data
These data describe the tool in the magazine.
Programming of general tool data with tool management.

x: = Parameter 1...11
t: = T number 1...32000

OPI block TD

Calculation of line: T number

Calculation of column: Not applicable

Tool-related data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
$TC_TP2	String	Tool identifier	toolIdent	String	T no.
$TC_TP1	INT	Duplo number	duploNo	WORD	T no.
$TC_TP3	INT	Size to left	toolsize_left	WORD	1
$TC_TP4	INT	Size to right	toolsize_right	WORD	1

Tool-related data, tool management

NCK identifier	Type	Description	OPI variable	Type	Default setting
$TC_TP5	INT	Size at top	toolsize_upper	WORD	1
$TC_TP6	INT	Size at bottom	toolsize_down	WORD	1
$TC_TP7	INT	Magazine location type	toolplace_spec	WORD	9999
$TC_TP8	INT	Status Value 0 Not enabled Bit 0=1 Active tool (A) Bit 1=2 Enabled (F) Bit 2=4 Disabled (G) Bit 3=8 Measure (M) Bit 4=16 Prewarning limit reached (V) Bit 5=32 Tool being changed (W) Bit 6=64 Coded for fixed location (P) Bit 7=128 Tool has been used (E) Bit 8=256 Identifier for tools in buffer Bit 9= 1 Disabled, but ignore due to PLC signal “Do not disable tool” Bit 9=0 Do not ignore Bit 10 To be unloaded (SW5 and later) Bit 11 To be loaded (SW 5 and later) Bit 12 Master tool (SW5 and later)	toolState	WORD	0=Not enabled
$TC_TP9	INT	Type of tool monitoring Value 0 No tool monitoring Bit 0=1 Service life Bit 1=2 Workpiece count Bit 2=4 Wear monitoring active Bit 3=8 Wear monitoring, location-dependent Fine offset active The letters in brackets are the identifier in the MMC screen form.	toolMon	WORD	0
$TC_TP10	INT	Replacement strategy	toolSearch	WORD	0
$TC_TP11	INT	Tool info (not used)	toolInfo	WORD	0
$A_TOOLMN	INT	Magazine assignment tool	toolInMag	WORD	
$A_TOOLMLN	INT	Location assignment tool	toolInPlace	WORD	
$P_TOOLND	INT	No. of cutting edges	numCuttEdges	WORD	
–		Adapter no. assignment	adaptNo	WORD	

Tool-related data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
$A_MYMN	INT	Owner magazine for tool	toolMyMag	WORD in SW6 and later	
$A_MYMLN	INT	Owner magazine location for tool	toolMyPlace	WORD in SW6 and later	

Duplo no. and tool name

Just as the T number is sufficient for the unique identification of a tool, a tool is specified equally uniquely by its duplo number and tool name (identifier).

Consequently, only names with different duplo numbers can be contained within one TO unit. The write operations of $TC_TP1 and $TC_TP2 are checked for the above and rejected if collisions are found.

$TC_TP3 to TP6

Size in terms of half locations:
Size 1 means that the tool exactly completely occupies its own magazine location.
The maximum programmable size is 7.
Tools sizes cannot be specified completely freely (see Subsection 4.3.3).

$TC_TP7

The magazine location type cannot be changed if the tool is in a magazine location.

$TC_TP8

The tool status is described with system parameter **$TC_TP8**. This parameter is bit-coded. In other words, a particular state of the tool is assigned to each bit of this data.

A tool which is loaded into the toolholder (spindle, ...) is set to the “active” state by the NCK when it is selected.

The status “Has been in use” is set by the NCK if the tool is removed from a magazine location of the type “Spindle or tool holder”.

The tool status “Being changed” is always reset by the software during a warm restart. This status is also reset when a tool change or tool preparation has ended.

The tool status “Tool in buffer” (SW3.2 and later) ensures on the next tool change that a tool located in a buffer magazine which is not the spindle and is not required for the next machining operation is returned to the real magazine.

“To be loaded” status (bit 11)

Bit11 is set for tools which are not located in a magazine and are to be loaded. The following definitions are used:

- The status is maintained beyond Power On.
- It is included in the data back-up and rewritten when transferred back to the NCK.
- When assigning a tool to a real magazine the tool status is reset by the NCK (applies to locations of location type 1, i.e. not to internal magazines such as the load magazine, buffer magazine, etc.)

Bit	Value	Meaning
11	0	“Not to be loaded”
	1	**“To be loaded”**

“To be unloaded” status (bit 10)

This bit is set for tools that are located in a magazine and are to be unloaded. The following definitions are used:

- The status is maintained beyond Power On
- It is included in the data back-up and rewritten when transferred back to the NCK.
- When the tool is unloaded via an unload location the tool status is reset by the NCK.

Bit	Value	Meaning
10	0	“Not to be unloaded”
	1	**“To be unloaded”**

“Master tool” status (bit 12)

Bit 12 is set for tools that are to be permanently assigned to a magazine. This status is only set to provide information and has no effect on the NCK (e.g. does not disable a location). The user defines via the unload program whether the tool can be unloaded.

Bit	Value	Meaning
12	0	“Not a master tool”
	1	**“Master tool”**

Note

Take care when “manually” changing the tool status via the OPI during machining. This could undo any necessary internal changes in status by the NCK and result in incorrect machining.

5.3.2 Tool-related grinding data

$TC_TPGx[t]

Technology-specific grinding data
The default setting for grinding data is 0. Tools with **tool type 400 to 499** are always **grinding tools**, i.e. have these additional data which take up additional memory space. If a tool of type 400–499 is set to a value outside this range, then its loses its grinding-specific data – the associated memory is released again and can be used for other tools.

x: = Parameter 1...9
t: = T number 1...32000

OPI block TG

Calculation of line: T number

Calculation of column: Not applicable

Tool-related grinding data				
Name	**Type**	**Description**	**OPI VAR**	**Type**
$TC_TPG 1	INT	Spindle number	spinNoDress	REAL
$TC_TPG 2	INT	Chaining rule	conntectPar	REAL
$TC_TPG 3	Double	Minimum when radius	minToolDia	REAL
$TC_TPG 4	Double	Minimum wheel width	minToolWide	REAL
$TC_TPG 5	Double	Current wheel width	actToolWide	REAL
$TC_TPG 6	Double	Maximum speed	maxRotSpeed	REAL
$TC_TPG 7	Double	Maximum grinding wheel surface speed	maxTipSpeed	REAL
$TC_TPG 8	Double	Angle of inclination of inclined wheel	inclAngle	REAL
$TC_TPG 9	INT	Parameter number for radius calculation	paramNrCCV	REAL

5.3.3 Tool-related user data

$TC_TPCx[t]

User tool data
An additional 10 tool-specific parameters can be set up per tool. Set with MD 18094: MM_CC_TDA_PARAM and enable with MD18080 MM_TOOL_MANAGEMENT_MASK (set bit 2)

x: = Parameter 1...10
t: = T number 1...32000

OPI block TU

Calculation of line: T number

Calculation of column: Parameter number

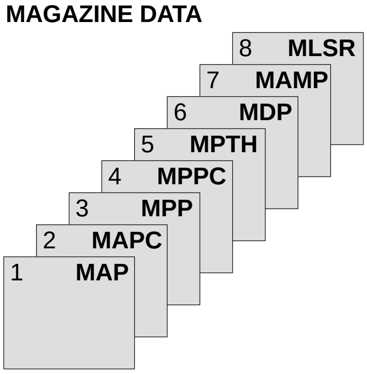
Tool-related data of OEM users				
NCK identifier	**Type**	**Description**	**OPI VAR**	**Type**
$TC_TPC1	Double		data	REAL
...	Double		data	REAL
$TC_TPC10	Double		data	REAL

Note

The data are displayed in the tool management. You could also store the tool states here, for example.

5.4 Magazine data

Magazine data



8 Magazine location assignment to spindles
7 Magazine block data
6 Distance to change position
5 Magazine location type hierarchy
4 User magazine location data
3 Magazine location data
2 User magazine description data
1 Magazine description data

Fig. 5-6 Overview of magazine data

5.4.1 Magazine description data

$TC_MAPx[n]

Magazine description data
These data identify the real magazine

x: = Parameter 1...10
t: = Magazine number 1...30, 9998, 9999

OPI block TM

Calculation of line: Magazine number

Calculation of column: Not applicable

Magazine description, data tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
		Magazine number	magNo	WORD	0
$TC_MAP2	String	Magazine identifier	magIdent	String	“ ”

Magazine description, data tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
$TC_MAP1	INT	Type of magazine **1 =** Chain **3** = Turret **5** = Box magazine **7** = Tool buffer magazine **9** = Load station magazine	magKind	WORD	0
$TC_MAP3	INT	Status of magazine Bit 0=1 Active magazine Bit 1=2 Disabled Bit 2=4 Magazine in load position Bit 3=8 Tool motion is active Bit 4=16 Magazine or tool may be moved. Enabled for loading	magState	WORD	2
$TC_MAP4	INT	Chaining to following magazine Magazine type = 1, 3, 5. For background magazines only	magLink1	WORD	–1
$TC_MAP5	INT	Chaining to preceding magazine Magazine type = 1, 3, 5. Reference (= number) to preceding magazine, backward chaining of background magazines	magLink2	WORD	–1
$TC_MAP6	INT	Number of tiers (box magazines only)	magDim	WORD	1
$TC_MAP7	INT	Number of columns (FIMag.) or number of locations (chain)	magNrPlaces	WORD	0
$TC_MAP8	INT	Current magazine position referred to tool change position	magActPlace	WORD	0
–			magCmd	WORD	
–			magCmdState	WORD	
–			magCmdPar1	WORD	
–			magCmdPar2	WORD	
$TC_MAP9	INT	Current wear grouping number	magWearCompoundNo	DINT	0
$TC_MAP10 (bit 0...7)	INT	Current tool search strategies of magazine (see $TC_MAMP2)	magToolSearchStrat	WORD	0
$TC_MAP10 (Bit 8...15)	INT	Current empty location search strategy of magazine	magPlaceSearchStrat	WORD	0

$TC_MAP3

The magazine status “Tool motion is active” is always reset during a warm restart.

A magazine that has the status “Tool motion is active” cannot be deleted.

Empty locations are not sought in magazines with the “disabled” status. If a disabled magazine is explicitly defined for the empty location search the process is aborted with an error message.

A tool that is located in a “disabled” magazine cannot be loaded to the spindle or toolholder.

$TC_MAP8

The current magazine position $TC_MAP8 is refreshed by the NCK every time the magazine is moved.

When the magazine configuration has been loaded, variable $TC_MAP8 is assigned the value zero. The position value is the number of the magazine location that is located at the zero position of the magazine. As a maximum, the magazine position can have the number of magazine locations in the magazine. Larger or negative values are rejected.

5.4.2 Magazine user data

$TC_MAPCx[n]

Magazine user data

Up to 10 user data can be additionally created for each magazine. Set in MD 18090 : MM_NUM_CC_MAGAZINE_PARAM and enable with MD18080: MM_TOOL_MANAGEMENT_MASK (set bit 2)

x:	=	Parameter 1...10
t:	=	Magazine number 1...30

OPI block TUM

Calculation of line:	Parameter number
Calculation of column:	Magazine number

Magazine description data, OEM user				
Name	**Type**	**Description**	**OPI VAR**	**Type**
$TC_MAPC1			userData	DINT
...			userData	DINT
$TC_MAPC10			userData	DINT

Note

These magazine user data are not supported by standard screen forms in SW lower than 3.x, i.e. they can only be accessed via the part program.

5.4.3 Magazine location data

$TC_MPPx[n,m]

Magazine location data
These data describe the magazine location.
x: = Parameter 1..7
n: = Physical magazine number 1..30, 9998, 9999
m: = Physical location number 1...32000
The maximum value of x is stored in OPI variable numMagPlaceParams in block Y.

OPI block TP

Calculation of line: (magazinLocNo–1)*numMagPlaceParams+parameter no.

Calculation of column: Magazine number

Magazine location data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
$TC_MPP1	INT	Location kind 1 = Magazine location 2 = Spindle, toolholder 3 = Gripper 4 = Loader 5 = Transfer location 6 = Load station 7 = Load point	placeData	WORD	0
$TC_MPP2	INT	Location type > 0: Location type for virtual location = 0: Every tool fits in this location 9999: Not defined	placeData	WORD	9999
$TC_MPP6	INT	T number of tool in this location	placeData	WORD	0
$TC_MPP3	BOOL	Consider adjacent location on/off	placeData	WORD	FALSE

Magazine location data, tool management					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
$TC_MPP4	INT	Location status Bit 0= 1 Disabled (A) Bit 1= 2 Free to hold a tool (occupied)(F) Bit 2= 4 Reserved for tool from buffer (G) Bit 3= 8 Reserved for new tool to be loaded (M) Bit 4= 16 Occupied in left half location (V) Bit 5= 32 Occupied in right half location (W) Bit 6= 64 Occupied in top half location (P) Bit 7=128 Occupied in bottom Half location (E) Bit 8 Left half location reserved Bit 9 Right half location reserved Bit 10 Top half location reserved Bit 11 Bottom half location reserved Bit 12 Wear group disabled (as of SW 5)	placeData	WORD	1
–		Reference phys. magazine (top right)	placeData	WORD	0
$TC_MPP5	INT	Location kind index (location kind numbering) or wear group number	placeData	WORD	0
$TC_MPP7	INT	No. of adapter in mag. location	placeData	WORD	0

Writing magazine location data

Points to be noted about writing magazine location data:
The first time one of the $TC_MPP... parameters is written all the magazine locations defined by magazine parameters are created with their default values (the memory for the locations is therefore “used up”), i.e. the magazine must have been defined by this time ($TC_MAP parameter).

$TC_MPP1 (location kind)

Magazines that do not have the type “internal” ($TC_MAP1 = 7 or = 9) may only be defiend as magazine locations of type “Magazine location” ($TC_MPP1 = 1).

Please note when writing the location status and number of the tool in this location that the following dependencies on $TC_MPP2 to $TC_MPP4 apply; these are checked during the write operation:

- If the location already contains a tool, the location type to be written must be checked against the tool location type.
- The “Free” state must only be written if no “Assigned” states are set or if no tool is in the location.
- The “Disabled” state can be set irrespective of the other states.
- If no tool is in the location, then the state “free” is automatically active, i.e. the state “Not free” cannot be set by the NC program, PLC or MMC.
- The “Occupied” states can only be set by the NCK as part of “Consider adjacent location” i.e. these states are ignored when the location is being written by the NC program, PLC or MMC.
- The state “Reserved for tool from buffer” is set when a tool is removed by the NCK from the real magazine during a tool change. This location is then not designated as “Free” for tools other than the tool removed.
- The states “Reserved for tool from buffer” and “reserved for new tool to be loaded” **of a location** are automatically reset when a tool is placed in this location.
- The states “Reserved for tool from buffer” and “Reserved for new tool to be loaded” of a **real magazine location** are automatically reset if a tool from this location is placed in a location in the load/unload magazine.
- The state “Reserved for tool from buffer” is reset during an empty location search if the tool for which the empty location is being sought is assigned a magazine location other than its previous real magazine location. The newly found empty location is assigned the state “Reserved for tool from buffer” and becomes the new owner of the tool being sought.

The magazine location state “Reserved for tool to be loaded” is always reset when the control system is restarted. If “Consider adjacent location” is active, reservations of adjacent locations are also considered.

The user need only familiarize himself with these rules if he wishes to define magazines directly on the PC program level. Data back-up is such that the rules are observed when data are imported to the NCK.

$TC_MPP5 (location kind index)

This data contains the spindle number for magazine locations of type “spindle” ($TC_MPP1) and is thus made known to the tool management.

The value cannot be changed for location type = 1 ($TC_MPP1; i.e. for all locations of the internal magazines), if a tool is at the location.

$TC_MPP6 (T no.)

- Tools can only be placed in magazine locations when both the tool and the magazine, plus its magazine locations, have been defined.

The tool may occupy only one magazine location!

Procedure:
This data establishes a relationship between the magazine block and the tool block.

- If it is already defined, then an attempt is made – subject to appropriate check procedure – to add it to the magazine location.
- If it is not yet defined, then an error has occurred.

Tests:

- The type of the tool to be placed must match the type of the location. If the type has not been set explicitly at the time of writing (default = 9999 = “Not defined”), then the tool is not placed.
- The state of the location must be “Free” and must not be “Disabled”.
- If the value T no. = 0 is programmed this means that the existing tool is removed from the magazine location.
  Caution: $TC_MPP6 = 0 also changes the state of the location: A tool can only be placed in a magazine location if the location does not already contain a tool. The old tool might first have to be removed with $TC_MPP6 = 0.

Note

Owing to this dependency of individual data, it is absolutely essential to type the T no. of the tool as the last data of a magazine configuration. If you do not keep to this sequence default values might be set which may result in unwanted data.

5.4.4 Magazine location user data

$TC_MPPCx[n,m]

Magazine location user data

Up to 10 user data can be additionally created for each magazine. Set for number of parameters in MD 18092 : MM_NUM_CC_MAGLOC_PARAM and enable with MD18080 MM_TOOL_MANAGEMENT_MASK (set bit 2)

x:	=	Parameter 1...10
t:	=	Magazine number 1...30
m:	=	Magazine location number 1...32000

OPI block TUP

Calculation of line: (m–1)*numMagLocParams_u+parameter no.

Calculation of column: Magazine number

Magazine location data, OEM user					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
$TC_MPPC1	INT		userplaceData	DINT	0
...	INT		userplaceData	DINT	0
$TC_MPPC10	INT		userplaceData	DINT	0

Note

These magazine user data are not supported by standard screen forms in SW lower than 3.x, i.e. they can only be accessed via the part program.

5.4.5 Magazine location hierarchy

$TC_MPTH[n,m]

Magazine location type hierarchy
The location types can be organized in a hierarchy by programming these system variables.
n: = Index of hierarchy, from 0...7
m: = Index within hierarchy n, location type 0...7

Magazine location types, see also $TC_TP7 and $TC_MPP2.

OPI block TT

Calculation of line: Number of location type+1

Calculation of column: Number of location hierarchy+1

Magazine data: Magazine location type hierarchy				
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**
$TC_MPTH[n,m]	INT	Magazine location hierarchy n: Hierarchy 0–7 m: Location type 0–7	placeType	WORD

If a tool must be loaded into the magazine, then the location type determines the availability of locations, i.e. $TC_TP7 and $TC_MPP2 must be defined.
If the location type of the tool is part of a location type hierarchy, then the location is assigned according to the hierarchy.
Several hierarchies of this type can be set up in one TO unit, but a location type can only be entered in one hierarchy.

5.4.6 Distance to tool change position

$TC_MDPx[n,m]

Distance from magazine zero

$TC_MDPx[n,m]=value

x:	=	1 :	Load magazine: Load points, load station (1st int. mag.)
		2 :	Buffer magazine: Spindle, gripper,..(2nd int. mag.)
n:	=		Magazine no. of real magazine
m:	=		Location no. of internal magazine (load point,..).
Value:	=		Distance in no. of locations

OPI block TPM

Calculation of line: (location no–1)*numPlaceMulti*numPlaceMultiPa rams+parameter no.

Calculation of column: Magazine number

Magazine data: Distance to change position				
NCK identifier	**Type**	**Description**	**OPI VAR**	**Type**
$TC_MDP1	INT	Distance between tool change position of magazine n and location m of 1st internal magazine (load magazine, 9999)	multiPlace	WORD
$TC_MDP2	INT	Distance between tool change position of magazine n and location m of 2nd internal magazine (load magazine, 9998)	multiPlace	WORD

Description

The current magazine position is required for tool change, loading and unloading. This position refers to the magazine zero defined by the machine manufacturer. This is usually at the change position.

The number of the location at the magazine zero must be specified during initialization. Otherwise, non-existent location 0 is taken to be the change position.

If the magazine is moved by a task, the current position is changed accordingly. The NC does not know how many positions the magazine has moved but knows the targets of the relevant commands. On the basis of the defined distance between and object (e.g. spindle 2) and the change position, the NC is able to update the current position.

Note:
In SW 5 and later, the value of the distance and the current magazine position is also evaluated for box magazines.

For empty location searches and tool searches, search strategies based on reference to the current magazine position convert the position contained in system parameter $TC_MAP8 to the change position, load point in each case at which the search is started. The change position, load point search reference is always specified as an internal NCK data in search tasks.

Note

Command $TC_MDP2[n,m]=9999 can be programmed to “undo” the relationship between spindle and magazine.

Example:

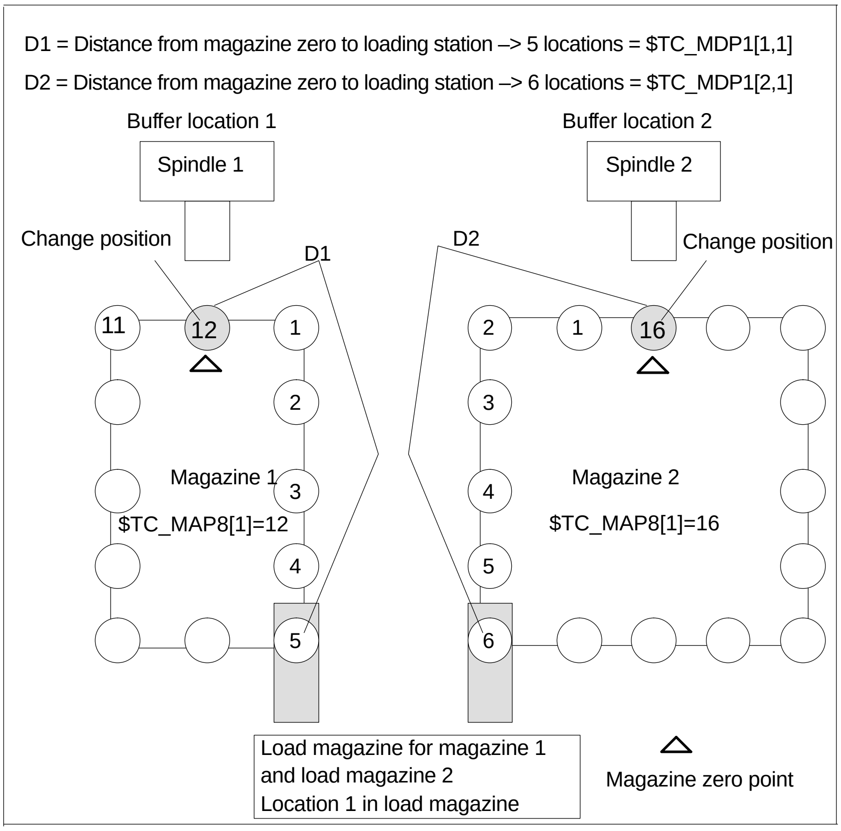

D1 = Distance from magazine zero to loading station –> 5 locations = $TC_MDP1[1,1]
D2 = Distance from magazine zero to loading station –> 6 locations = $TC_MDP1[2,1]
Buffer location 1
Buffer location 2
Spindle 1
Spindle 2
Change position
D1
D2
Change position
11
12
1
2
3
4
5
Magazine 1
$TC_MAP8[1]=12
2
1
16
3
4
5
6
Magazine 2
$TC_MAP8[1]=16
Load magazine for magazine 1
and load magazine 2
Location 1 in load magazine
Magazine zero point

Fig. 5-7 Distance to change position $TC_MDPx[y,z]=value

Normally, magazine zero is the change position of the spindle. The following statement therefore applies:

- If location 1 is located at zero position, the current magazine position = 1 ($TC_MAP8[1]).

Examples for programming the distance to the zero position:

$TC_MDP1[1,1] = 5	Distance between location 1 of the load station and the zero position of magazine 1
$TC_MDP1[2,1] = 6	Distance between the same location and the zero position of magazine 2
$TC_MDP2[1,1] = 0	Distance between location 1 of 2nd internal magazine and zero position of magazine 1
$TC_MDP2[2,2] = 0	Distance between location 2 of 2nd internal magazine and zero position of magazine 2

5.4.7 Magazine blocks

$TC_MAMPx

Magazine module data
x: = Parameters 1, 2, 3

OPI block TMC

Calculation of line: Not applicable

Calculation of column: Not applicable

Magazine block data, magazine check block					
NCK identifier	Type	Description	OPI variable	Type	Default setting
$TC_MAMP1	String	Identifier of the configuration magazine	magCBIdent	String	" "
		Number of load magazine	magBLMag	WORD	
		Number of buffer magazine	magZWMag	WORD	
$TC_MAMP2	INT	Type of tool search (bits 0...7) and type of empty location search (bits 8...15) Bit 0 Select the "active" tool, otherwise search for replacement tool with lowest duplo number Bit 1 Search for next replacement tool at the shortest possible distance from the current magazine position Bit 2 Select the "active" tool, otherwise replacement tool in $TC_TP10 Bit 3 Search for tool in group with the lowest actual value of the monitored quantity Bit 4 Search for tool in group with the highest actual value of the monitored quantity	magSearch		0

Magazine block data, magazine check block					
NCK identifier	**Type**	**Description**	**OPI variable**	**Type**	**Default setting**
$TC_MAMP2	INT	Bit 5 Consider only those tools with an actual value deviating by at least a factor of x between 0 and 1 of the setpoint Bit 7 1: Search for tool acc. to assignment sequence between “Spindle and magazine” (always from 1st magazine in distance table) 0: Start tool search in magazine from which last loaded tool was fetched	magSearch		0
		Bit 8=1 (256) Forward search from first location number starting with bit 9=1 (512) Forward search from current magazine position starting with bit 10=1 (1024) Backward search from last location no. starting with bit 11=1 (2048) Backward search from current magazine position starting with bit 12=1 (4096) Symmetrical search from current magazine position starting with bit 13=1 (8192) Replace tool, old against new (SW 5 and later)			
–		Number of locations in this TO without buffer memory, load magazines	magVPlaces	WORD	
–		all magazine locations, including the buffer memory, load magazines	magRPlaces	WORD	
–		Magazine command	magCBCmd	WORD	
–		Magazine command status	magCBCmdState	WORD	
–		Return parameter, magazine	magCMCmdPar1	WORD	
–		Return parameter, location	magCMCmdPar1	WORD	

Magazine block data, magazine check block

NCK identifier	Type	Description	OPI variable	Type	Default setting
$TC_MAMP3 (SW 5.1 and later)	INT	Handling of tools in a wear grouping (bits 0...7) Search strategies for wear groupings (bits 8...15) Bit 0=0 When a wear grouping is activated, the tool status remains unchanged Bit 0=1 When a wear grouping is activated, the tool status changes. A tool from every tool group is activated	modeWear-Group	WORD	0
		Bit 1=0 When a wear grouping is disabled, the tool status remains unchanged Bit 1=1 When a wear grouping is disabled, the tool status changes Bit 2=7 Reserved Bit 8=0 Find the next possible wear grouping Bit 8=1 Find the wear grouping with the next-highest group number that can be activated Bit 9...11 Reserved			
		Bit 12=0 Lowest possible duplo number Bit 12=1 Lowest possible magazine location number Bit 13...15 Reserved			

5.4.8 Assignment of buffers to spindles (SW 3.2 and later)

$TC_MLSR[x,y]

Assignment of buffer locations to spindles $TC_MLSR[x,y]

x: = Location no. in buffer 1... 32000
y: = Location no. of spindle in buffer magazine 1... 32000

No OPI block

Calculation of line: Not applicable

Calculation of column: Not applicable

NCK identifier	Type	Description	OPI variable	Type	Default setting
$TC_MLSR[x,y]	INT	Parameter for assigning magazine locations of buffer magazine to spindle	–	–	0

Programming sequence determines sequence in which tools are automatically returned

Note

The content value of the system parameter is not evaluated. The assignment is defined via indices x and y. To determine whether a particular assignment exists via the part program, a read operation must supply a value of zero. If magazine configurations created with a version earlier than SW 3.2 are being used, this system parameter must be defined additionally if the tool change operation involves buffer locations (e.g. gripper) other than the spindle and tool magazine. The NCK can find tools in buffers of this type only if the parameter has been assigned. The definitions that are made with this parameter, for example, allow the NCK to detect on power-up whether a tool change was interrupted on Power OFF and at which buffer location the tool is currently located.

5.5 Adapter data (SW 5 and higher)

$TC_ADPTx[n]

If machine data $MN_MM_NUM_TOOL_ADAPTER is set to a value = –1 or > 0, the adapter data are defined, deleted, read and written via the following variables

x: = Parameter 1...3, T
n: = Number of adapter

OPI block AD

Calculation of line: Length 1, 2, 3 = line 1, 2, 3, transformation = line 4

Calculation of column: Adapter number

Adapter data				
Name	**Type**	**Description**	**OPI VAR**	**Type**
$TC_ADPT1	Double	Adapter geometry: Length 1	adaptData	REAL
$TC_ADPT2	Double	Adapter geometry: Length 2	adaptData	REAL
$TC_ADPT3	Double	Adapter geometry: Length 3	adaptData	REAL
$TC_ADPTT[n]	Double	Adapter transformation number	adaptData	REAL

The adapter geometry values act on the geometry values of the cutting edge analogously to parameters $TC_DP 21, $TC_DP 22 and $TC_DP 23. These parameters are available only when the tool management is active.

Transformation numbers 1 to 8 can be programmed for the adapter transformation function. The parameter is available only when the tool management is active.

$TC_MPP7[m,p]	Number of adapter assigned to magazine location	
	Value=0	No adapter assigned to location
	Value>0	Number of assigned magazine

5.6 Toolholder data

$TC_CARRx

x: = Parameter 1...23

The maximum number of toolholders can be defined in machine data 18088: MM_NUM_TOOL_CARRIER. The value is divided by the number of active TO units. The resultant integer indicates how many toolholders can be defined per TO unit.

OPI block TC (currently not available)

5.7 Unassigned user variables

Unassigned user parameters

These programmable variables provide the user with three unassigned user parameters. These systems parameters are transferred to the PLC via the user interface with the T selection signal. The user can send additional tool management information to the PLC with these parameters. These parameters can be read and written from the NC program. They are not backed up and are set to “0” on Reset or end of program.

$P_VDITCP[x]

x: = Parameter 0..2

NCK identifier	Description	Format
$TC_VDITCP[0]	User parameter 0	int
$TC_VDITCP[1]	User parameter 1	int
$TC_VDITCP[2]	User parameter 2	int

Interface DB72, DB73

The user parameters are output in DB 72 and DB 73 on the tool management interface. They are only valid when the status of the interface is active. The format is DINT.

Example

```
$P_VDITCP[0]=12;          DB72.DBD(n+4) =12
or
$P_VDITCP[1]=33;          DB72.DBD(n+8) =33
or
$P_VDITCP[2]=2000;        DB72.DBD(n+12) =2000
```

T=“Tool”

The variables must be inserted before the T call in the part program if they are to be included in the transmission to the PLC for a tool.

5.8 NC language commands

See also table in Subsection 5.12.5.

5.8.1 CHKDNO – Uniqueness check on D number

The term “D number uniqueness” in this context (no replacement tools) means that the D numbers of all tools defined in a TO unit may be programmed exactly once => in other words, the D numbers in the TO unit are unique and absolute. When the tool management function is active, reference is made only to the possibility of assigning “unique” D numbers. The distinction is made on the basis of replacement tools that are generally present.

status = CHKDNO (T1, T2, D)

Parameters used:

Status	TRUE	D numbers have been assigned uniquely for the checked area
	FALSE	A D number is assigned more than once or the parameter assignment is invalid

The parameters are optional.

CHKDNO (T1,T2,) All D numbers of the specified tools are checked.

This function is available in SW version 3.4 for the MMC 102/103, but not until SW 4.1 for the MMC 100/100.2. For operator panel OP 030 this function is not implemented until SW 3.2.

D numbers of replacement tools

Replacement tools can be defined and used when tool management is active. The machining part program does not usually give any indication of whether replacement tools are available or not. The machining program usually addresses tools with T=’identifier’. (The programming of T=“location number” is referred back to T=“identifier” internally). The program otherwise only contains the actual programming of the offset (the D number). For that reason the D number for tool and replacement tool must be identical.

Example

Active tool and replacement tools for T=“drill_5mm”

- T no. = 10 with D numbers 1, 2, 3 (active)
- T no. = 11 with D numbers 1, 2, 3 (replacement)
- T no. = 12 with D numbers 1, 2, 3 (replacement)

Active tool and replacement tools for T="drill_3mm":

- T no. = 20 with D numbers 1, 2, 3 (active)
- T no. = 21 with D numbers 1, 2, 3 (replacement)
- T no. = 22 with D numbers 1, 2, 3 (replacement)

If no parameters are defined for **CHKDNO**, it establishes a collision of D numbers 1, 2 and 3 of "drill_5mm" with D numbers 1, 2 and 3 of "drill_3mm", but not between the D numbers of the active and replacement tools.

The collisions that occur are each displayed as alarms, e.g.:

- "Channel 1 D number 1 defined for tool T no. 10 and 20"
- "Channel 1 D number 1 defined for tool T no. 10 and 21"

In the case of a parameterizing error (specified T or D number is not defined in channel), the FALSE state is also returned.

If MAX_CUTTING_EDGE_NO <= MAX_CUTTING_EDGE_PER_TOOL, CHKDNO always returns the TRUE state, irrespective of the parameterization.

5.8.2 CHKDM – Uniqueness check within a magazine (as of SW 5)

When the tool management function is active, the CHKDM command checks existing data in the NCK for D number uniqueness within one or several magazines. It has the same functionality as CHKDNO. The parameters are optional.

state = CHKDM(Magno, Dno, TLHolderno.)

Result of check:

Value = TRUE	Checked D numbers are unique.
Value = FALSE	Check is not correct.

Meaning of parameters:

MagNo	Magazine number of magazine to be checked Omission of the parameter or setting it to zero means that all tools in the magazines linked to the spindle no. or toolholder no. specified in the 3rd parameter are checked.
Dnr	Reference D number for check Omission of the parameter or setting it to zero means that all D numbers in the specified magazine are checked for uniqueness.

Toolholder no.	Specifies the spindle no. or toolholder no. in relation to which magazines must be checked. Omission of the parameter means that the magazines to be checked are deduced from the distance table of the spindle location for the master spindle or master toolholder.

5.8.3 GETACTTD – Calculation of T no. for a unique D no. (as of SW 5)

When the TM function is active, this command is used (e.g. in relation to measuring cycle programs) to find the T number **of the active tool in the tool group** that is associated with a particular D number.

status = GETACTTD(Tno, Dno)

Dnr D number for which T number is to be found. The D number is not checked for uniqueness.

If the same D numbers have been defined in different tool groups of the same TO unit, the T number of the first tool group whose tools contain the specified number is found.

Tno Found T number

Status Result of search:

0: T number found, Tno assigned the value.

–1: No T number exists for the specified D number, Tno assigned the value 0.

–2: D number is not unique;
Tno assigned the value of the first D number to be found.

–3: The tool group does not contain any tools of the specified status or D number. Tno assigned the value 0.

–4: The tool group contains several tools of the specified status and D number.
Tno contains the value of the first tool to be found with the specified D number.

–5: Function could not be executed for other reasons.

5.8.4 GETDNO – Rename D numbers (as of SW 5)

Language command

d = GETDNO(t, ce)

can be programmed to read offset number d of cutting edge ce of the tool with T number t. If t or ce are parameters which have no data record, d=0 is returned. Any parameters violating the syntax rules will generate an alarm.

The command is only available if $MN_MAX_CUTTING_EDGE_NO > $MN_MAX_CUTTING_EDGE_PER_TOOL.
$MN_MAX_CUTTING_EDGE_NO <= $MN_MAX_CUTTING_EDGE_PER_TOOL returns GETDNO d=ce as the D number.

5.8.5 SETDNO – Rename D numbers (as of SW 5)

Language command

state = SETDNO(t, ce, d) can be used to set or modify offset number d of cutting edge ce of tool t. If t or ce are parameters which have no data record, state = FALSE is returned. Any parameters violating the syntax rules will generate an alarm.

> 0 must be specified for t, ce and d, d=0 cannot be set.

5.8.6 DZERO – Invalidate D numbers (as of SW 5)

Designates all D numbers of the TO unit as invalid. This is a support command for reset operations.
Offset data records designated as “invalid” are excluded from the check performed by language command CHKDNO. They can be made accessible by setting the D numbers again with command SETDNO.

5.8.7 DELDL – Delete additive offsets (as of SW 5)

This command deletes the additive offsets for the cutting edge of a tool (to release memory space). This operation deletes the defined wear values as well as setup offsets.

status = DELDL(t, d)

Explanation of parameters:

DELDL(t, d) All additive offsets of the cutting edge with D number d of tool t are deleted

DELDL(t)	All additive offsets of all cutting edges of tool t are deleted
DELDL	All additive offsets of all cutting edges of all tools in the TO unit are deleted (of the TO unit in the channel in which the command is programmed)

Status	0	The offsets have been successfully deleted
	–1	The delete operation has not been executed (if parameter was set for exactly one cutting edge) or not executed completely (if parameter was set for more than one cutting edge)

5.8.8 NEWT – Create a new tool

A tool can be created without specifying the T number with the function NEWT(...). The function returns an automatically generated T no. with which the tool can then be addressed. The 1st cutting edge is automatically created when a new tool is created. All offsets are set to “0”.

Return parameter = NEWT (“TL”, DUPLO_NR)

If it is not possible to create a new tool for any reason, the NEWT(...) function generates an alarm.

Specification of a duplo number is optional. It is generated in the NCK if it is not specified. (duplo no.= old duplo no. +1)

Example:

Create a new tool called drill with duplo no. 1

```
DEF INT DUPLO_NR
DEF INT T_NR
DUPLO_NR =1
T_NR = NEWT(“DRILL”,DUPLO_NR)
```

or alternatively

```
$TC_TP1[1] = “DRILL”          Identifier DRILL with T no. 1
$TC_TP1[1] = DUPLO_NR         Duplo no. = 1
```

This function is used to set up tools in a load program (load cycle).

5.8.9 DELT – Delete a tool

A tool can be deleted without specifying the T number with the function DELT(...). It is only possible to delete tools that have been unloaded.

DELT(“TL”, DUPLO_NR)

All tool-related data are set to 0 (user data, hierarchy data, ...).

Example:

DELT (“DRILL”, DUPLO_NR)

Function is used to delete tools in the part program.

5.8.10 GETT – Read T no.

The GETT function sends back the T number as the return value for the tool identifier and its duplo number.

Return parameter = GETT (“TL”, DUPLO_NR)

If no tool matching the tool identifier or duplo number is found, the value 1 is returned. Specification of the duplo number is optional.

If no duplo number is entered, the T number of the 1st tool from the group of tools with the specified identifier is returned.

Example:

T number determined for drill with duplo number

```
R10=GETT(“DRILL”, DUPLO_NR)                ;The T number is located
                                           in R10
$TC_TPx,[GETT(“DRILL”,DUPLO_NR)]=value     ;Write tool-related
                                           data
```

This function is used to reload tools via the part program.

5.8.11 SETPIECE – Decrement workpiece counter

With the SETPIECE function the user can update the workpiece count data of the tools involved in the machining process. All the tools that have been changed since the last time SETPIECE was activated are included in the update. The function is generally used for programming at the end of the NC part program to decrement the number of all tools involved in workpiece count monitoring.

Note

If a workpiece is in the spindle at the time when SETPIECE is called, it is also counted. As soon as an executable block is in the main run (interpolator) after SETPIECE, this tool is considered again for the next SETPOIECE call. The command is not active in the block search (with/without calculation). If the value for the quantity = 0, the tool noted in the internal table is deleted.

Programming

SETPIECE(x,y)

x := 0 ... 32000	Number of workpieces produced since last execution of the SETPIECE function.
y := 0...8	Spindle index, value 0 means index of main spindle (need not be programmed)

Example:

SETPIECE(1);	Workpiece counter of main spindle is decremented by 1
SETPIECE(1,1);	Workpiece counter of spindle no. or toolholder no. 1 is decrem ented by 1
SETPIECE(4,2);	Workpiece counter of spindle no. or toolholder no. 2 is decrem ented by 4

Example of SETPIECE with change command M06:

The tools involved in a tool (program) are to be decremented by the value 1.

```
T1              ;T1 is preselected (relative to main spindle)
M06             ;T1 is loaded
D1              ;D1 becomes active
T2              ;T2 is preselected
:               ;machining program
:
M06             ;T2 is loaded
D1              ;D1 of T2 becomes active
T3              ;T3 is preselected
;               ;machining program
:
:
M06
T0              ;preparation for clearing the spindle
:
M06             ;clear spindle
SETPIECE(1)     ;SETPIECE on all tools
M30
```

The counter must be decremented once per tool.

In this example, tools T1, T2, T3 are to machine a program. All 3 tools are monitored for workpiece count. The aim is to decrement tool T1 by the value 1, T2 by the value 2 and T3 by the value 3.
The command SETPIECE(0) must generally be programmed after the change, including the offset selection.

```
N500    T1
N600    M06
N700    d1
N800    setpiece(0)    ;previously noted tools for workpiece counting are
                       deleted
N900    t2
N1000   setpiece(1)    ;SETPIECE acts on T1
:                      ;machining commands
:
N1100   M06
N1200   d1
N1300   setpiece(0)    ;delete command for noted tools
N1400   t3             ;in this block, T2 is determined as “active tool and
                       entered in the table of noted tools
N1500   setpiece(2)    ;acts only on T2
N1600   M06
N1700   d1
N1800   setpiece(0)    ;delete command of noted tools
N1900   t0
N2000   M06
N2100   d0
N2200   setpiece(0)    ;delete command of noted tools, so no
                       tools are noted for SETPIECE
N2300   M30
```

5.8.12 GETSELT – Read the selected T no.

This function supplies the T number of the tool preselected for the spindle, allowing, for example, the offset data to be accessed before M06.

GETSELT (return parameter, x);

x: = 1–32 spindle number
x: = Index for main spindle

Specification of “x” is optional. If “x” is not specified the function refers to the main spindle.

Example:

```
T="DRILL"
...
...
GETSELT(R10)          ;Read preselected T no. for main spindle
```

This function executes a comparison in the tool-changing cycle to determine whether preselected tool is already loaded in spindle.

5.8.13 GETACTT – Read the active, internal T no.

This function offers the option of finding out the T number of the tool with the "active" status (a tool becomes active immediately before it is loaded into the toolholder) and "has been in use" from a tool group with identifier "name" using the parameter "TNo".

state=GETACTT(Tno,name)

The return parameter "status" indicates the success/failure of the call:

- 0 = Function successful; Tno. contains the desired value
- –1 = No tool matching the specified identifier exists; Tno. contains value = 0
- –2 = The tool group does not contain a tool with the desired status; TNo. contains value = 0
- –3 = The tool group contains several tools with the desired status; TTno. contains the value of the first tool with the desired status

GETACTT can have several meanings! It is always possible for several tools in one tool group to have the same status. The command will only function correctly if the user ensures that only one tool in the tool group has the required status.
The command does not initiate a main synchronization. It may be necessary to enter STOPRE before the call.

Example:

Tool group "Drills" contains three tools with the duplo numbers 1, 2, 3 and the T numbers 1, 2, 3:

```
def int Tno, status                ;Tool group "Drills" does not contain an
                                   ;active tool at first

status=GETACTT(Tno, "Drills")      ;status=–2, Tno=0
T="Drill"                          ;Prepare tool change; drill is assigned
                                   ;active status

status=GETACTT(Tno, "Drills")      ;status=0, Tno=1
M06                                ;Request tool change
T="Hugo"                           ;Prepare tool change; "Hugo" is assigned
                                   ;active status
status=GETACTT(Tno, "Drills")      ;status=0, Tno=1
D2                                 ;Tool change must now be complete; T no.=1
                                   ;now on master spindle
status=GETACTT(Tno, "Drills")      ;status=0, Tno=1
```

5.8.14 SETMS – Spindle can be declared the master spindle

SETMS(n) declares the spindle specified under n to be the master spindle. A spindle can also be defined as the master via a machine data.

When SETMS is programmed without a spindle name, the spindle programmed in the machine data used instead.

5.8.15 SETMTH – Set master toolholder number (SW 5 and higher)

In SW 5.1 and later, machine data **MD 20124: TOOL_MANAGEMENT_TOOLHOLDER** can be set to determine whether a toolholder number must be assigned instead of a spindle number in order to define the target location of a tool to be loaded. This language command can be used meaningfully only if the MD > 0.

Programming example

T=“Mill” M06	No address extension programmed –> this refers to the master toolholder, i.e. toolholder 1 (value of machine data TOOL_MANAGEMENT_TOOLHOLDER)**.** The tool change is performed in the buffer location with $TC_MPP5=1. The path is corrected with the tool offsets.
...	
T2=“Drill” ..M2=6	;Address extension for secondary toolholder has been programmed. The tool is changed in buffer location 2. The path is not corrected.
...	
SETMTH (2)	Declare toolholder 2 as **master toolholder**
T=“Mill_2” M06	No address extension programmed –> this refers to the master toolholder; i.e. toolholder 2. The tool is changed in buffer location 2. The path is corrected with the tool offsets.
...	
T1=“Drill_1” M1=6	;Address extension for secondary toolholder has been programmed. The tool change is performed in the buffer location with $TC_MPP5=1. The path is not corrected.
...	
SETMTH	Declare toolholder defined in TOOL_MANAGEMENT_TOOL HOLDER to be the master toolholder
T=“Mill_3” M06	No address extension programmed –> the master toolholder is meant, i.e. toolholder 1 (value of MD TOOL_MANAGEMENT_TOOLHOLDER). The tool is changed in buffer location 1. The path is corrected with the tool offsets.

Note

SETMTH does not change the active tool. The new setting for the master toolholder can only be taken into account as part of the subsequently programmed tool change operation.

5.8.16 POSM – Position magazine (SW 5 and later)

This NC language command enables you to initiate a magazine positioning operation to a particular location in an internal magazine (e.g. spindle, toolholder, load magazine), irrespective of how the location is assigned or the status of the tool it contains. The language command includes some of the functions of OPI PI service (see Section 5.12.5) _N_TMPOSM.

The full command is: **POSM (p, m, ip, im)**

Description of function

p Location number at which the internal magazine is to be positioned.

m Magazine number of the magazine to be moved.
This parameter is optional.
If it is not set, the location number refers to the magazine
contained in the distance table as the first magazine
for the specified internal location.

ip Location number of specified internal magazine (spindle location, load gazine, etc.)
The parameter is optional.
If it is not specified, the positioning operation refers to the main spindle location or the main toolholder location.

im Magazine number of internal magazine in relation to location number ip
to which the magazine must be moved. An internal magazine is either
a load or a buffer magazine.
This parameter is optional.
If it is not set, the command refers to the buffer magazine.

The magazine (number m) must be linked to the selected load and buffer magazine location by a distance relationship. Alarms are generated when incorrect parameters are specified (e.g. undefined location numbers).

Parameterizing example

Starting configuration:

- Magazine (magazine number = 1),
- Spindle (buffer magazine = 9998, location 1),
- Load magazine (load magazine = 9999, location 2).

Move from magazine 1, location number 4 to the spindle.

Command:

```
N100          POSM(4, 1, 1, 9998)
```

Command for traversal to load magazine:

```
N100          POSM(4, 1, 1, 9999)
```

Example with result check

At the outset we have a magazine as shown in the figure below.

Location 12 is to be positioned at the change position and the program must not be continued until positioning has been successfully completed (simplest case with only one magazine and one defined change position).

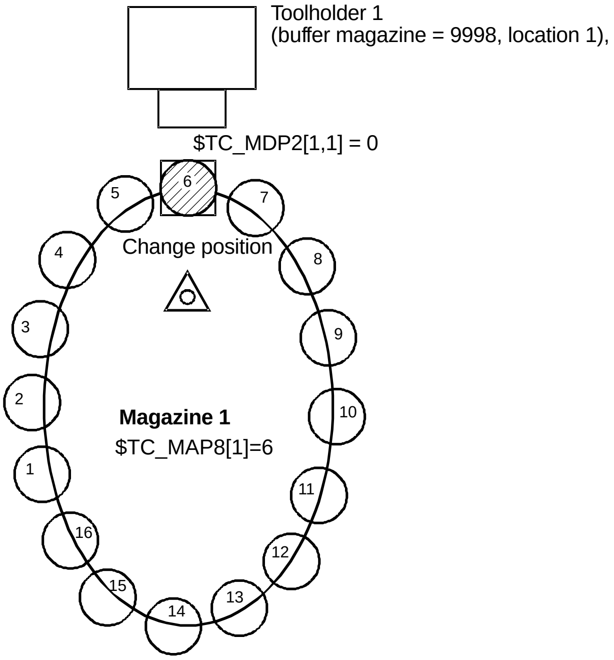


Fig. 5-8 Magazine positioning with check of positioning operation result

In this example, the magazine zero point is the location in front of toolholder 1. It is defined by parameter $TC_MDP2. Toolholder 1 is assigned to the master spindle of the channel.

```
N100 POSM(12)            ;Moves location 12 to the change position, any unprogrammed
                         ;parameters are set internally to POSM (12, 1, 1, 9998)
N200 wait:
N300 G4 F1               ;Wait time selected to suit conditions on the machine
                         :(exit jump may be necessary to allow reaction to position-
                         ;ing errors)
N400 if ( $TC_MAP8[1] <> 12 ) goto wait;
                         ;After execution of POSM(12), the current magazine position
                         must equal 12.
```

References: **/PGA/**, “Programming Guide, Advanced”
(description of system parameters)

Note

The language command POSM(...) is terminated without waiting for an acknowledgement from the PLC.

5.8.17 SETTIA – Dectivate tool from wear group (as of SW 5)

The SETTIA function cancels the “active” status for all active tools in the selected wear grouping. The command can still be used even if no wear groupings are defined or none are activated via MD settings at time the command is called.

SETTIA(status, mnr, vnr)

mnr = Magazine number in which tool(s) must be deactivated
vnr = Number of wear grouping in which tool(s) must be deactivated
status = Return parameter which can assume the following values:

- 0: Function has been executed correctly.
- –1: Function has not been executed because the selected magazines do not contain any active wear grouping.
- –2= Function has not been executed because the entered wear group number does not exist.
- –3: Function has not been executed because the entered magazine number does not exist.
- –4: Function has not been executed because wear grouping functionality has not been enabled via machine data.
- –5: Function has not been executed for some other reason.

When appropriately parameterized, the result of the SETTIA command is the same as for a change in the wear grouping in tool change operations with $TC_MAMP3, bit1=1.

5.8.18 SETTA – Activate tool from wear group (as of SW 5)

The SETTA function activates all tools that are not disabled in the selected wear grouping, but only activates 1 tool in a tool group. The command can still be used even if no wear groupings are defined or none are activated via MD settings at time the command is called.

SETTA(status, mnr, vnr)

mnr = Magazine number in which tool(s) must be activated.
vnr = Number of wear grouping in which tool(s) must be activated.

status = Return parameter which can assume the following values:

- 0: Function has been executed correctly.
- 1: Function has been executed, but another active twin tool has been detected.
- –1: Function has not been executed because the selected magazines do not contain any active wear grouping.
- –2: Function has not been executed because the entered wear group number does not exist.
- –3: Function has not been executed because the entered magazine number does not exist.
- –4: Function has not been executed because wear grouping functionality has not been enabled via machine data.
- –5: Function has not been executed for some other reason.

The result of the SETTA command is the same as for a change in the wear grouping in tool change operations with $TC_MAMP3, bit0=1 (see Subsection 3.4.5).

5.8.19 RESETMON – Language command for setpoint activation (SW 5.1 and later)

RESETMON(state, t, d, mon)

Set the actual value of the tool to the setpoint.

state Status of command execution.
Possible values are:
0 Command has been executed successfully
–1 The cutting edge with specified D number d does not exist.
–2 The tool with specified T number t does not exist.
–3 The specified tool does not have a defined monitoring function. This status is only possible if t is specified explicitly.
–4 Monitoring function is not active in the NCK, i.e. the command has not been executed.

t Internal T number
t = 0 Command applies to all tools.

t > 0 Command applies to this one particular tool.
t < 0 The absolute value of t is generated and the command applies to all twin tools of the specified tool.

d The D number of the tool (optional parameter).
If the parameter is not specified at all or is assigned the value 0, all D numbers or all cutting edges of the tool are processed.
d > 0 The command refers exactly to the specified D number.

mon Optional bit-coded parameter.
If the parameter is either not specified at all or assigned the value 0, all actual values of the active, tool-specific monitoring functions for the designation edge(s) are set to the setpoints.
mon > 0 The command applies precisely to the actual value of the specified monitoring type.
Possible settings are the positive values of system parameter **$TC_TP9** (1, 2, 4, 8) or the corresponding bit combinations when several monitoring types are activated

Note

No alarms are output explicitly. The user can perform his own error handling via the parameter **state**.

5.8.20 $P_TOOLEXIST – Determine existence of a tool

The system variable is read only.
The following applies to the function “Flat D numbers”: ’TRUE’ is returned for value t = 1, for all other values of t ’FALSE’ is returned.

Name	$P_TOOLEXIST[t]			
Meaning	If a tool with T no. = t exists, “TRUE” is returned. If no tool with T no. = t exists, “FALSE” is returned.			
Data type	BOOL		as of SW 4.2	
Value range	TRUE, FALSE			
Indices	Meaning			Value range
	The index specifies the T number			1–32000
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	x	–	–	–
Implicit preprocess stop	–	–		

5.8.21 $A_TOOLMN – Read magazine no. of tool

Note: The abbreviation TOOLMN stands for “**tool m**agazine **n**umber”. The name $A_TOOL was chosen to show the association with the existing system variables.

Name	$A_TOOLMN[t]			
Meaning	Returns the magazine number of the tool with T no. = t. If the tool is not assigned to a magazine, then 0 is returned. If the function Tool management is not active, –1 is returned. If no tool with T no. = t exists, –2 is returned. An alarm is output if the value range for the T number is violated.			
Data type	INT		as of SW 4.2	
Value range	1–32000			
Indices	Meaning			Value range
	The index specifies the T number			1–32000
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	x	–	x	–
Implicit preprocess stop	x	–		

5.8.22 $A_TOOLMLN – Read magazine location no. of tool

Note: The abbreviation TOOLMLN stands for “**tool m**agazine **l**ocation **n**umber”.

Name	$A_TOOLMLN[t]			
Meaning	Returns the magazine number of the tool with T no. = t. If the tool is not assigned to a magazine, then 0 is returned. If the function Tool management is not active, –1 is returned. If no tool with T no. = t exists, –2 is returned. An alarm is output if the value range for the T number is violated.			
Data type	INT		as of SW 4.2	
Value range	1–32000			
Indices	Meaning			Value range
	The index specifies the T number			1–32000
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action

Name	$A_TOOLMLN[t]			
	x	–	x	–
Implicit preprocess stop	x	–		

Note: It is not possible for $A_TOOLMLN[t]==0 and $A_TOOLMN[t]>0 to be valid, or $A_TOOLMLN[t]>0 and $A_TOOLMN[t]==0.

5.8.23 $P_TOOLND – Read number of cutting edges of tool

Note: The abbreviation TOOLND stands for “**tool n**umber of **D**s”.

Name	$P_TOOLND[t]			
Meaning	Returns the number of cutting edges of the tool with T no. = t. A tool always has at least one cutting edge. Default: If no tool exists with T no.=t, –1 is returned. The value 0 is rejected as an index error.			
Data type	INT		as of SW 4.2	
Value range	Default: –1, 1–9 Function “Flat D numbers”: –1, 1 – “Machine data value for maximum number of D numbers”			
Indices	Meaning			Value range
	The index specifies the T number			1–32000
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	x	–	–	–
Implicit preprocess stop	–	–		

“Flat D numbers” function (only when TM system is not active)

In cases when the function “Flat D number” is active, the procedure differs. Parameter t = 1 returns the total number of offset data records of the TOA unit. When t is set to other values, –1 is returned. –1 is also returned if there is no offset data record defined in the TOA unit.

5.8.24 $A_MONIFACT – Read factor for tool life monitoring

If different tool materials are to be machined with the same tool, it may be necessary to increase or reduce the time intervals for monitoring in order to detect the varying degrees of tool wear. The factor is set accordingly before the tool is used. The write operation is performed synchronously with the main run.

A channel-specific parameter, used to multiply the current time measurement, has been defined.

Setting a value = 0 deactivates the time monitoring function for all tools used on the channel via the part program (see Subsection 3.8.2).

Name	**$A_MONIFACT[t]**			
Meaning	Only relevant when time monitoring is active in the tool management. Factor for influencing the tempo for the time measurement for time-monitored tools. Values < 1 and > 0 slow down time measurement (the clock “runs slower”). Values > 1 speed up the time measurement (clock “runs faster”). Value 1 is active after the control has been powered up, after Reset and M30 (default) and corresponds to real time. Value 0 is also permitted and disables time measurement of all time-monitored tools that are operated on a time-monitored spindle on this channel. Note: You can cause the monitoring time to “run backwards” with negative values.			
Data type	REAL		as of SW 4.2	
Value range	Value range of type REAL			
Indices	Meaning			Value range
				–
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action

Name	$A_MONIFACT[t]			
	x	x	x	x
Implicit preprocess stop	–	x		

Tool life counter on monitor (SW 5.1 and higher)

If system parameter $A_MONIFACT is set accordingly, the tool life counter on the monitor can run at a speed other than real time. The time values of OPI block TS are converted on the interface for this purpose (see Section 5.8.26). The values are retained in the NCK. Those values are real-time values.

Read OPI:	The time values are divided by the current value of $A_MONIFACT and transferred.
Write OPI:	The time values output by the OPI are multiplied by the current value of $A_MONIFACT and stored in the NCK.

Example

The current values are specified (units in real time, i.e. normalized to $A_MONIFACT = 1).

Programmed tool life:	10 minutes (new data from SW 5.1)
Actual tool life:	2 minutes – the prewarning limit is reached within **one minute**
Prewarning limit:	1 minute

The values 10, 2, 1 are displayed on the screen.

$A_MONIFACT = 2 is programmed in the part program (clock runs faster). The actual tool life displayed on the monitor jumps and continues to run in real time. The programmed tool life and prewarning limit displayed also jump as soon as **$A_MONIFACT = 2** takes effect.

Programmed tool life	5 minutes (new data from SW 5.1)
Actual tool life:	1 minute – the prewarning limit is reached in **30 seconds**
Prewarning limit:	0.5 minutes

5.8.25 Further language commands (as of SW 5)

Name	$P_TOOLNO
Meaning	Active tool numbers T0 to T32000; when “flat D number” function is active, T can have eight digits

Name	$P_TOOLNO			
Data type	Integer		SW 2 and later	
Value range	1–32000			
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	X	–	–	–
Implicit preprocess stop	–	–	–	

Name	$P_TOOLP			
Meaning	Last programmed tool number (Ti) (without TM only)			
Data type	Integer		SW 5.3 and higher	
Value range	1–32000			
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	X	–	–	–
Implicit preprocess stop	–	–	–	

Name	$P_TOOL			
Meaning	Active tool cutting edge (Dx)			
Data type	Integer		SW 2 and later	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	X	–	–	–
Implicit preprocess stop	–	–	–	

Name	**$P_DLNO**			
Meaning	Active sum offset number DL=0–DL=max; max=value of $MN_MM_MAX_SUMCORR_PER_CUTTEDGE			
Data type	Integer		SW 5.3 and higher	
Value range	0–6			
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	X	–	–	–
Implicit preprocess stop	–	–		

Name	**$P_TOOLL[n]**			
Meaning	Active TL total length; n = 1...3			
Data type	REAL		SW 2 and later	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchr. action	Write in synchr. action
	X	–	–	–
Implicit preprocess stop	–	–	–	

Name	**$P_TOOLR**			
Meaning	Active radius			
Data type	REAL		SW 2 and later	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchr. action	Write in synchr. action
	X	–	–	–
Implicit preprocess stop	–	–	–	

Name	$P_TC			
Meaning	Active toolholder			
Data type	Integer		SW 5.3 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	X	–	–	–
Implicit preprocess stop	–	–	–	

Name	$P_TCANG[n]			
Meaning	Active angle of a toolholder axis; n = 1–2			
Data type	REAL		SW 5 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchr. action	Write in synchr. action
	X	–	–	–
Implicit preprocess stop	–	–	–	

Name	$P_TCDIFF[n]			
Meaning	Difference between calculated and applied angle of a toolholder axis in the case of graduated angle (Hirth tooth system)			
Data type	REAL		SW 5.3 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchr. action	Write in synchr. action
	X	–	–	–
Implicit preprocess stop	–	–	–	

Name	**$P_AD[n]**			
Meaning	Active tool offset; n = 1...25 => $TC_DP1...25; n = 26 => cutting edge number CE			
Data type	REAL		SW 2 and later	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	X	X	–	–
Implicit preprocess stop	–	–	–	

Name	**$AC_MSNUM**			
Meaning	Master spindle, return value 0: No spindle configured 1...n: Number of master spindle			
Data type	Integer		SW 3 and later	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	X	–	X	–
Implicit preprocess stop	X	–	–	

Name	**$P_MSNUM**			
Meaning	Master spindle 0: No spindle configured 1...n: Number of master spindle			
Data type	Integer		SW 5.2 and later	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchr. action	Write in synchr. action

Name	**$P_MSNUM**			
	X	–	–	–
Implicit preprocess stop	–	–	–	

Name	**$AC_MTHNUM**			
Meaning	Master toolholder Value=0 No master toolholder defined Value>0 Number of master toolholder			
Data type	Integer		SW 5 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	X	–	X	–
Implicit preprocess stop	X	–	X	–

Name	**$P_MTHNUM**			
Meaning	Master toolholder Value=0 No master toolholder defined Value>0 Number of master toolholder			
Data type	Integer		SW 5.3 and higher	
Value range				
Indices	Meaning			Value range
Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	X	–	–	–
Implicit preprocess stop	–	–		

5.8.26 Variables for subroutine replacement technique (as of SW 5)

TM language command	Functions	SW version
$C_T	Number of T word (without TM) for substitute subroutine for T (MD 10717)	5
$C_T_PROG	Bool variable: Contents in $C_T?	5
$C_TS	Programmed TL identifier (with TM) for substitute subroutine for T (MD 10717)	5
$C_TS_PROG	Bool variable: Contents in $C_TS?	5
$C_TE	Address extension of T word	5.3
$C_D	Number of D memory	5.3
$C_D_PROG	Bool variable: Contents in $C_D?	5.3
$C_DL	Number of DL memory	5.3
$C_DL_PROG	Bool variable: Contents in $C_DL?	5.3

5.8.27 Variables for tool change in synchronous actions

TM language command	Functions	SW version
$AC_TC_FCT	Function 1: Move (load, unload,...) 2: Prepare change 3: LOAD tool 4: LOAD tool (turret, without M06) 5: Prepare change and LOAD tool (with M06)	5
$AC_TC_STATUS	Acknowledgement status of PLC FC8	5
$AC_TC_THNO	ToolHolder or Spindle no.	5
$AC_TC_TNO	Internal T number	5
$AC_TC_MFN	New tool from magazine	5
$AC_TC_LFN	New tool from location	5
$AC_TC_MTN	New tool to magazine	5
$AC_TC_LTN	New tool to location	5
$AC_TC_MFO	Old tool from magazine	5
$AC_TC_LFO	Old tool from location	5
$AC_TC_MTO	Old tool to magazine	5
$AC_TC_LTO	Old tool to location	5

Access	Read in part program	Write in part program	Read in synchron. action	Write in synchron. action
	X	–	X	–
Implicit preprocess stop	X	–		

5.9 Conventions for programming data

5.9.1 Tool and cutting edge data

if a parameter for a cutting edge, tool or magazine that does not exist is written, a new cutting edge, tool or magazine is created.

Note

When a tool is created, all the cutting-edge-specific data of cutting edge D1 are created with it.
(DP, DPC, MOP, MOPC are initialized with “0”). The grinding-specific tool data ($TC_TG1...) are not created until one of the tool types ($TC_DP1) 400–499 has been programmed for one of the cutting edges of the tool.

Deleting data

When data are deleted the memory area is deleted with it and automatically re-enabled.

A tool can only be deleted if it is not involved in the current machining process. This applies both to tools selected or inserted with a “T” call and tools for which constant grinding wheel surface speed or tool monitoring is active.

Note

If tool management is active you must ensure that the tool being deleted is not assigned to a magazine location ($TC_MPP6). This assignment must be removed before the tool is deleted.

The grinding-specific tool data ($TC_TG1...) are not created until one of the tool types ($TC_DP1) 400–499 has been programmed for any of the cutting edges of the tool.

If the tool type is set from the current value taken from the range 400–499 to a value outside this range, the grinding data memory is enabled again, i.e. the grinding-specific data are lost.

Action	Program command	Description

Create a tool	Without tool management:
	$TC_DPx[y,z] = value;
		Create tool T if T does not yet exist!
		y = T number
		z = D number

	If tool management is active: T_NR = NEWT(“tool identifier”, duplo number) or $TC_TP1[y] = duplo number; $TC_TP2[y] = “tool identifier”;	y = T number

Create a cutting edge	$TC_DPx[y,z] = value;	Create cutting edge D = z if D = z does not yet exist!
		y = T number
		z = D number

Set tool data	If tool management is active: $TC_TPx[y] = value; or $TC_TPx[GETT(“DRILL”,DUPLO_NR)] = value; or $TC_TPCx[y] = value; $TC_TGx[y] = value;	y = T number Write tool-related user data Write tool-related grinding data
Set data of a cutting edge	$TC_DPx[y,z] = value; $TC_DPCx[y,z] = value $TC_MOPx[y,z] = value $TC_MOPCx[y,z] = value	Write offset data Write cutting-edge-related user data Write cutting-edge-related monitoring data Write CC (OEM) cutting edge monitoring data y = T number z = D number
Delete cutting edge data	Without tool management: $TC_DP1[0,0] = 0;	All tools of the channel are deleted, the memory is enabled.
	With tool management: $TC_TP1[0,0];	When deleting tools, the entries for the location data must also be corrected.

Action	Program command	Description
Delete tool data	Without tool management: $TC_DP1[y,0] = 0;	y = T is deleted, memory is enabled.
	With tool management: $TC_TP1[y] = 0; or $TC_TP1[GETT(“Tool identifier”, duplo number)] = 0; or DELT[“tool identifier”, duplo number]	All tool-related data are set to “0” (user data, hierarchy data, ...). When deleting a tool, the entries for the location data must also be corrected.
Delete data of all tools	Without tool management: $TC_DP1[0,0] = 0;	All the tools of the channel are deleted and the memory is enabled.
	With tool management: $TC_TP1[0,0] = 0;	When deleting tools, the entries for the location data must also be corrected.

5.9.2 Magazine data

Sequence for defining data

“Assign tool to a magazine location” establishes a dependency between the tool data and the magazine/magazine location data.

Example:

The tool contains the magazine location type for which it is intended. The magazine type contains its own magazine location type. If the tool has been assigned to the magazine location, as a rule the location type cannot be changed again as this can cause inconsistencies.

This results in the requirement that tools and magazines be loaded into the control by a special procedure and that definitions that determine the structure not be changed again during processing (these are e.g. magazine dimension, magazine location type, duplo no., tool name, ...). Not included are cutting edge data, magazine location status, tool status.

Load data

Because tools are linked to magazines via magazine location parameter $TC_MPP6, the following rules for correct definition of tools and magazines must be followed:

1. Load tool data
2. Load magazine data
3. Load $TC_MPP6 parameters (=> places tool in magazine location)

This sequence is maintained for data backup.

The grinding data of a tool cannot be written until tool type = “grinding tool” has been defined for at least one cutting edge.

The distance parameter ($TC_MDPx) and the buffer assignment parameter ($TC_MLSR) cannot be written until the magazines and their locations have been defined.

Delete data

A tool cannot be deleted while it is still contained in a magazine. The following sequence of operations must be followed when deleting:

1. Delete the magazine data (this removes tools from the magazine); or remove the tool explicitly from the magazine.
2. Delete tool data

In addition, a magazine cannot be deleted if it has status $TC_MAP3[i]= 8 (motion is active). The delete command is rejected for all magazines even if only **one** magazine is preventing the command from being executed.

Note

If a single tool is to be deleted it must first be removed from the magazine location with an unload operation and then it can be deleted.
Tools that are selected cannot be deleted! You can ensure that no tool is selected beyond a part program by programming T0 before the end of a part program independently of the settings in the machine data (see MD for selecting tools beyond the end of a program).

Action	Program command	Description
Create new magazine	$TC_MAPx[y] = value;	Value <>0 , y = magazine no. of a magazine which has not yet been created.
Delete a magazine	$TC_MAP1[y] = 0;	The data of the magazine and its magazine locations as well as any defined distances to change positions are deleted. The associated memory is enabled.
Delete a magazines and the tools contained in it	$TC_MAP6[y] = 0;	The data of the magazine and its magazine locations as well as any defined distances to change positions are deleted. Any tools contained in the magazine are also deleted. The allocated memory is enabled.
Delete all magazines	$TC_MAP1[0] = 0;	All data of all magazines of the selected TO area unit are deleted and the associated memory is enabled. The magazine data block is then empty.
Create new magazine location	$TC_MPPx[y,z] = value;	Value <>0 , y = location number not yet available. Before the data of the first location can be created, the associated magazine must be defined. Once the first parameter of the first magazine location to be created is written, all the magazine locations belonging to the magazine are created with default values according to the values for the number of rows and columns.
Set magazine location type hierarchy	$TC_MPTHx[y] = value;	
Set magazine distances (distance to change position)	$TC_MPTHx[y] = value;	
Delete magazine distances (distance to change position)	$TC_MDPx[y,0z]=0 $TC_MDPx[0,0]= 0;	Delete all defined distances of the magazine with the number "y", i.e. the magazine is no longer "seen" during a tool search and an empty location search. Delete all defined distances of all magazines in the TO unit.
Delete the assignment of buffer to spindles	$TC_MLSR[x,0]= 0; $TC_MLSR[0,0]= 0;	Delete all defined assignments of one buffer location with the number "x", i.e. location "x" is no longer "seen" during a tool search. Delete all defined assignments between buffers of the TO unit and spindles
Set magazine block data	$TC_MAMPx = value;	

5.9.3 Tool change

Programming the tool selection

Tool selection can be divided into 2 different steps:

1. Tool change preparation
2. Tool change execution

Steps 1–2 can be programmed separately or together in the NC program (see MD 22550 TOOL_CHANGE_MODE).

Examples

Tool change in one step: (turret)

Tx;	Make new tool x available and execute tool change

Tool change in two steps:

1. Tx;	Tool change preparation (selection of tool)
2. M06;	Tool change execution

Note

If tool management is active, a tool can only be selected with the tool identifier (name). If a T number is programmed, the number is used as the identifier (name). In this case the tool must have been assigned a T number as a name on loading.

Tool change with identifier:

T=“DRILL”;	A tool with identifier “DRILL” is sought.

Tool change with number as identifier:

T=“123”;	A tool with identifier “123” is sought. Alternatively, T123 can also be programmed

5.9.4 Cutting edge selection

Cutting edge selection after tool change

When a tool change has been completed, the tool cutting edge can be selected in one of the following ways

1. The offset number D is programmed.
2. The offset number D is not programmed and is preset by MD20270 CUTTING_EDGE_DEFAULT

= 0	No automatic cutting edge selection after M06.
> 0	Number of cutting edge selected after M06.
= –1	The cutting edge no. of the old tool remains valid and also selected for the new tool after M06.
= –2	The offset of the old tool remains valid and also selected for the new tool after M06.

Examples:

Tool selection with the following cutting edge selection
Cutting edge selection always refers to the tool that is changed with command M06.

T1 M06	Tool change – no D programmed; therefore offset selection accord ing to MD 20270
T5	Preselect tool
X .. Y.. Z...	Working with T1 and the offset from MD 20270
D2	Offset D2 from T1 !!!
M06	Tool change; T5 is loaded – offset selection according to MD 20270
T1	Preselect tool
X.., Y...	Working with T5 and the offset from MD 20270

When programming tool commands, main spindles and secondary spindles are programmed differently. Only tool offset values of the main spindle tool are taken into account by the geometry because only one active offset can be processed per channel. Processing of tool commands for a secondary spindle is only relevant for signal output to the PLC and the function GETSELT(...).

Spindle no. 2 = main spindle:

T2 = “DRILL”	
M2 = 06	
T1 = “MILL”	Preselect tool for secondary spindle
M1 = 06	Load new tool into secondary spindle
D1	Select cutting edge of “Drill” (main spindle)

Spindle no. 2 = main spindle:

T2 = “DRILL”	Select a tool for the main spindle. As an alternative, T= “Drill” could also be specified.
T1 = x;	Select a tool for a secondary spindle
M2 = 06	Change tool M06 could be programmed alternatively
D1	Select cutting edge of a tool with identifier “DRILL”

5.9.5 Tool transfer from program test mode (SW 4 and later)

In MD 20110 RESET_MODE_MASK, bit **3** , you can set whether the active tool and tool offset are to be taken

- (= 1) from the test program which was last terminated in test mode<HR or
- (= 0) from the program which was last terminated before the test program

was activated.

Requirements: Bits 0 and 6 must be set in MD 20110.

$P_ISTEST

The system variable $P_ISTEST is for checking from the part program whether a program test is active. The system variable returns the value TRUE when program testing is active.

5.10 Programming T=location number (SW 4 and later)

This function is only available when tool management is active. This type of programming is not only suitable for turrets, but for all other types of magazine.

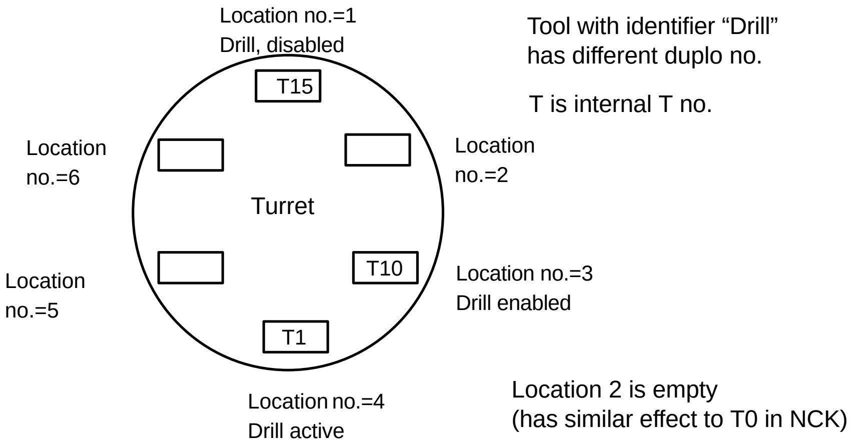


Fig. 5-9 Programming of T=location number

The programming method is set in MD 20310: TOOL_MANAGEMENT_MASK, bit16=1:

- T = "**x**" with **x** as the tool **identifier**
- Tx with **x as the location number** of the magazine containing the selected tool

When the function is active, **T1** selects the tool in location number 1 instead of the tool with identifier "1". The first magazine linked to the toolholder is accessed. The identifier of the tool in this location is then determined ("Drill").

The subsequent procedure is as if T="Drill" had been programmed. Which of the three tools from the "Drills" group is determined as the first step of the tool change process.

The set tool search strategy is used.

- When the strategy "Take the first available tool from the group" is applied, T10 from location 3 is loaded.
- When the strategy "Take the first tool with "active" status from the group" is applied, T1 is "loaded".

T15 at location number 1 cannot be used, because it is disabled. No alarm is generated if the programmed location does not contain a tool when the T=location programming method is used.

If more than one magazine is assigned to the toolholder, the programmed location number refers to the first magazine defined in the distance table.

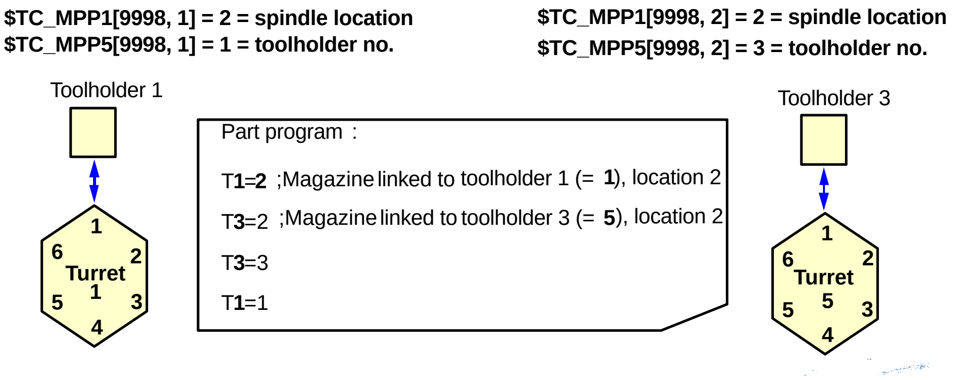
If the tools in the tool group are stored in different magazines of thetoolholder, the search procedure is the same as with the standard TM system.

Note

With the T=location function, T= “Drill” can be programmed alternatively
T = 1 ;Tool
T = “Drill” ;Tool with identifier Drill

5.10.1 Call several turrets with “T=location number”

Fig. 5-10 Working with T=location number in several magazines


$TC_MPP1[9998, 1] = 2 = spindle location
$TC_MPP5[9998, 1] = 1 = toolholder no.
$TC_MPP1[9998, 2] = 2 = spindle location
$TC_MPP5[9998, 2] = 3 = toolholder no.
Toolholder 1
Toolholder 3
Turret
1
2
3
4
5
6
1
Turret
1
2
3
4
5
6
5

```
Part program :
T1=2  ;Magazine linked to toolholder 1 (= 1), location 2
T3=2  ;Magazine linked to toolholder 3 (= 5), location 2
T3=3
T1=1
```

Fig. 5-11 T=location number as TM function on turning machines

It is possible to work in one channel or one TO unit using programming option “T=location number” and several magazines.

- NC address T can be programmed with address extension T1= ... .
- The TM then interprets this as a spindle number or toolholder number.
- T without address extension then refers to the main spindle.

5.11 Programming examples

Action	Program command	Description

Create a tool	DEF INT DUPLO_NR
	DEF INT T_NR
	DUPLO_NR = 7
	T_NR = NEWT(“DRILL”,DUPLO_NR)
		Create new tool called drill with duplo no. = 7. The automatically generated T number is stored in “T_NR”.

	T_NR = GETT(“DRILL”, DUPLO_NR) or $TC_TP2[1] = “DRILL” ; $TC_TP1[1] = DUPLO_NR	Determine the T number of tool “Drill” with duplo no. 7 that has already been created. In this case, the T number is defined by programming.
Tool data Read/write	$TC_DP1[GETT(“DRILL”, DUPLO_NR), 2] = 210	Write tool type for the 2nd cutting edge of tool “Drill”/ DUPLO_NR
	$TC_DP1[T_NR, 2] = 210	Write tool type for the 2nd cutting edge of tool “ T number ”
Select tool	T = “DRILL ” or:	If several tools match this name, the T number of the first possible tool is returned.
	T = GETT(“DRILL”, DUPLO_NR) or Tx	Determines T number for “DRILL” with duplo number = DUPLO_NR and selects it. Call with T no., e.g. T1,T2,T3,....
Delete a tool	$TC_TP1[T_NR,0]=0 or DELT (“DRILL”, DUPLO_NR) $TC_TP1[GETT(“DRILL”),0]=0 or alternatively: DELT(“DRILL”)	Tool with T_NR is deleted, tool “DRILL”, DUPLO_NR is deleted

5.12 Overview of the other OPI blocks in the tool management

The line need only be calculated if the OPI variable is followed by a field [ ]. The value of the line is otherwise 1.

5.12.1 Magazine directory data, MMC internal

OPI block TMV

Calculation of line:	Magazine number
Calculation of column:	Not applicable

NCK identifier	Description	OPI variable	Type
None	Number of magazines	numActMags	WORD
	Number of magazine	magVNo[ ]	WORD
	Magazine identifier	magVIdent[ ]	String

5.12.2 Tool directory data, MMC internal

OPI block TV

Calculation of line:	Serial no. of tools
Calculation of column:	Not applicable

NCK identifier	Description	OPI variable	Type
None	Number of tolls in TO area	numTools	WORD
	Last assigned T number for TM	TnumWZV	WORD
	T number	toolNo[ ]	WORD
	Tool identifier	toolIdent[ ]	String
	Duplo number	nrDuplo[ ]	WORD
	No. of cutting edges	numCuttEdges[ ]	WORD

NCK identifier	Description	OPI variable	Type
	Current magazine	toolInMag[ ]	WORD
	Current location	toolInPlace[ ]	WORD

5.12.3 Parameterization, return parameters TMGETT, TSEARC

OPI block TF

Calculation of line: No. of found tool

Calculation of column: Not applicable

NCK identifier	Description	OPI variable	Type
	Return: Found tools	resultNrOfTools	WORD
	Return: T numbers of found tools	resultToolNr[ ]	WORD
	Form for search criterion of PI TSEARCH (OPI block TD)	parMasksTD	WORD
	Comparison value for PI TSEARCH of variables of OPI block TD	parDataTD	WORD
	Comparison value for PI TSEARCH of variables of OPI block TD	parDataToolIdentTD	String
	Comparison value for PI TSEARCH of variables of OPI block TU	parMasksTU	WORD
	Comparison value for PI TSEARCH of variables of OPI block TU	parDataTU	REAL
	Comparison value for PI TSEARCH of variables of OPI block TO	parMasksTO	WORD
	Comparison value for PI TSEARCH of variables of OPI block TO	parDataTO	REAL
	Comparison value for PI TSEARCH of variables of OPI block TUE	parMasksTUE	WORD
	Comparison value for PI TSEARCH of variables of OPI block TUE	parDataTUE	REAL
	Comparison value for PI TSEARCH of variables of OPI block TS	parMasksTS	WORD
	Comparison value for PI TSEARCH of variables of OPI block TS	parDataTS	REAL

NCK identifier	Description	OPI variable	Type
	Comparison value for PI TSEARCH of variables of OPI block TUS	parMasksTUS	WORD
	Comparison value for PI TSEARCH of variables of OPI block TUS	parDataTUS	REAL

5.12.4 Working offsets

OPI block AEV

Calculation of line: Cutting edge number

Calculation of column: Not applicable

NCK identifier	Description	OPI VAR	Type
	Number of D numbers in block	numActDEdges	WORD
	D numbers	Dno[...]	WORD
	Internal T number	toolNo[...]	WORD
	Cutting edge number	cuttEdgeNo[...]	WORD
	Tool identifier	toolIdent[...]	STRING
	Duplo number	duploNo[...]	WORD
	Magazine	toolInMag[...]	WORD
	Location	toolInPlace[...]	WORD

5.12.5 PI services and language commands for TM function

FB 4 (PI_SERV) or FB 7 can be used to start program instance services (PI services) in the NCK area. Following a request via the PI service, a program section which performs a particular function (e.g. search for empty location in a magazine with active tool management) is executed in the NCK.

Reference: FB (Part 1), P3 Basic PLC Program

PI service	Functions	NC language command	SW version
MMCSEM	Semaphores for various PI services		
DELETO	Delete a tool	DELT(“TL”, Duplo)	

PI service	Functions	NC language command	SW version
DELECE	Delete a tool cutting edge	$TC_DP1[t,d]=0	
CREATO	Create a tool	NEWT(“TL”, Duplo)	
CRTOCE	Create TL specifying edge no.	$TC_DPx[t,d]	SW 5
TMCRTO	Create a tool	$TC_TPx[t]	
TMCRTC (not available in PLC)	Create a tool specifying edge no.	$TC_DPx[t,d]	SW 5
CREACE	Create a tool cutting edge	$TC_DP[t,d]=value	
CRCEDN	Create a new cutting edge	$TC_DPx[t,d]	
TMFDPL	Find empty location for loading	GETFREELOC	SW 6
TMMVTL	Ready a magazine location for loading, unload tool		
TMPCIT	Set incremental value for workpiece counter, decrement count by y	SETPIECE(SpinNo,y)	
TMPOSM	Position magazine location or tool	POSM (p, m, ip, im)	SW 5
TMFPBP	Find empty location acc. to properties		
TSEARC	Complex search using search screenforms	User cycles program	
TMRASS	Reset the active status		SW 5
TMGETT (not available in PLC)	Confirm T number for spec. tool identifier with duplo no.	GETT(“TL”, Duplo)	
	Read pre-selected T number	GETSELT(SpinNo)	
CHKDNO (not available in PLC) TMCHKD (not available in PLC)	Check the uniqueness of the D numbers of the tool data for the TO unit assigned to the executing channel. Parameters t1 and t2,d are optional.	status=CHKDNO (t1,t2,d)	SW 5
DZERO (not available in PLC)	Set the D numbers of all tools in the TO unit assigned to the channel to “invalid”. D numbers of this type are displayed with value 0 on the OPI. The invalid D number is generated in the NCK by assigning the value “old D number”+32000 to the D number.	DZERO	SW 5

PI service	Functions	NC language command	SW version
	Specify the associated internal T no.=t of the tool for offset no. D=d. The tool with the status “active” and “was in use” is fetched from the tool group.	status=GETACTTD (t,d)	SW 5
	Give the D no. for tool t and its edge ce	d=GETDNO(t,ce)	SW 5
	Set the D no. of tool t and its edge ce to value d	status=SETDNO (t,ce,d)	SW 5
	Read the active T no. and status	status=GETACTT (Tno,“TL”)	SW 4
	Delete command for all location-dependent/setup offsets of an edge, or a tool, if d is not specified	status=DELDL(t,d)	SW 5
SETTST (not available in PLC)	Set tool status to “active”	SETTA(Stat,m,vnr)	SW 5
SETTST (not available in PLC)	Set tool status to “not active”	SETTIAStat,m,vnr)	SW 5
CHKDM (not available in PLC)	Check unique D no. in magazine; m=magazine	CHKDM(m)	SW 5
Value of MD can be manipulated	Set toolholder no. (h=holder no.)	SETMTH(h)	
	Set master spindle (s=spindle no.)	SETMS(s)	
TRESMO (not available in PLC)	Tool life/quantity/wear setpoint activation	RESETMON	
TMAWCO (not available in PLC)	Activate a wear grouping	$TC_MAP9	SW 5

NC language commands

The NCK states are read using the following language commands.

Functions	NC language command	SW version
Active tool no. T	$P_TOOLNO	
Last programmed TL no. (without TM)	$P_TOOLP	
Active tool offset D	$P_TOOL	
Active tool length; n=1–3	$P_TOOLL[n]	

Functions	NC language command	SW version
Active toolholder	$P_TC	
Active angle of a TL carrier axis	$P_TCANG[n]	
Diff angle	$P_TCDIFF[n]	
Active radius	$P_TOOLR	
No. of edges for TL t	$P_TOOLND[t]	
Tool exists with number	$P_TOOLEXIST[t]	
Active tool offsets, n=1–25 (26, 27)	$P_AD[n]	
Active DL number	$P_DLNO	

Number of T word for replacement subroutine for T	$C_T	SW 5
Programmed TL identifier (with TM) for replacement subroutine for T	$C_TS	SW 5
Bool variable: Contents in $C_T?	$C_T_PROG	SW 5
Bool variable: Contents in $C_TS?	$C_TS_PROG	SW 5
1: Move (load/unload, relocate..); 2: Prepare change; 3:Load; 4: Load (turret, without M06); 5: Prepare change and load (with M06)	$AC_TC_FCT	SW 5
Acknowledgement status of PLC FC 8	$AC_TC_STATUS	SW 5
Toolholder or spindle number	$AC_TC_THNO	SW 5
New tool from magazine	$AC_TC_MFN	SW 5
New tool from location	$AC_TC_LFN	SW 5
New tool to magazine	$AC_TC_MTN	SW 5
New tool to location	$AC_TC_LTN	SW 5
Old tool from magazine	$AC_TC_MFO	SW 5
Old tool from location	$AC_TC_LFO	SW 5
Old tool to magazine	$AC_TC_MTO	SW 5
Old tool to location	$AC_TC_LTO	SW 5
Magazine number of tool t	$A_TOOLMN[t]	
Magazine location of tool t	$A_TOOLMLN[t]	
Lifetime factor for tool t	$A_MONIFACT[t]	
Master spindle	$AC_MSNUM	

Master spindle	$P_MSNUM	
Master toolholder	$AC_MTHNUM	
Master toolholder	$P_MTHNUM	

Please refer to Chapter 3 for explanations.

Data Backup 6

6.1 Backing up the NCK data

Reading in data via the RS232 interface is described in:

References: /BA/ Operator’s Guide

Complete backup

All the data of the active file system are output via file INITIAL.INI.

Tool data

All tool-specific data are backed up in file _N_TOx_TOA.

Magazine data

All magazine data are backed up in file _N_TOx_TMA.

Tool and magazine data

Tool and magazine data are backed up in file _N_TOx_INI.

The availability or non-availability of the data listed below is essentially determined by settings in the MD.

Note

Please make sure that the spindle is empty before backing up data. If it is no longer possible to replace the tool, for example during servicing, it is still possible to perform a backup. Alarm “22070 TO unit 000x, please load tool T=000x to magazine and repeat data backup”. The data are backed up correctly; however, you must make sure that this backup is used only for this machine, since the current states are also saved.

The format in the backup file is as follows:

1. Tool definitions
2. Magazine definitions
3. Parameters which set up a relationship between defined tools and defined magazine locations.

1. Tool definitions

$TC_TP1[i]	**Tool data**
...	
$TC_TP11[i]	
;	
$TC_TPC1[i]	CC user tool data
...	
$TC_TPC10[i]	
;	
$TC_DP1[i, j]	**Cutting edge data** (available with/without TM)
...	
$TC_DP25[i , j]	
;	
$TC_DPC1[i, j]	CC cutting edge data
...	
$TC_DPC10[i , j]	
;	
$TC_MOP1[i, j]	Monitoring data
$TC_MOP4[i , j]	
;	
$TC_MOPC1[i, j]	CC monitoring data
...	
$TC_MOPC10[i , j]	
$TC_TPG1[i]	Grinding (exists only for tools of type
...	’Grinding tool’ with/without TM)
$TC_TPG9[i]	
;	
$TC_TP1[i+1]	**Tool data**
...	
$TC_TP11[i+1]	
;	
$TC_TPG1[i+1]	Grinding
...	
$TC_TPG9[i+1]	
;	
...	
...	

2. Magazine definitions

$TC_MAMP1	**Magazine block parameter**
$TC_MAMP2	
;	
$TC_MPTH[n, m]	**Magazine location type hierarchy structure**
;	
$TC_MAP1[i]	**Magazine parameter**
...	
$TC_MAP8[i]	
;	
$TC_MAPC1[i]	CC magazine parameter
...	
$TC_MAPC10[i]	
;	
$TC_MPP1[i, j]	**Magazine location parameter**
...	
$TC_MPP5[i, j]	
;	
$TC_MPPC1[i, j]	CC magazine location parameter
...	
$TC_MPPC10[i, j]	
;	
$TC_MAP1[i+1]	**Magazine parameter**
...	
$TC_MAP8[i+1]	
...	
;	
$TC_MDP1[k, l]	**Magazine distances to spindles,**...
...	
$TC_MDP2[k, l]	
...	
;	
$TC_MLSR[k, l]	Relationship between buffer locations and spindles;
...	

3. Relationship between tools and magazine locations

$TC_MPP6[i, j]	Tool in magazine location
$TC_MPP6[i, j +1]	
...	
$TC_MPP6[i, j +J]	
$TC_MPP6[i+1, j]	
$TC_MPP6[i+1, j +1]	
...	
$TC_MPP6[i+1 j +J]	
;	
$TC_MPP6[i+I, j]	

```
$TC_MPP6[ i+I, j +1 ]
...
$TC_MPP6[ i+I, j +J ]
;
M17
```

The data in the magazine module are only backed up if at least one magazine location has been defined.

Note

Tool management data of tool management functions that are not available are ignored on write access operations to the active file system. No alarm is displayed.

An alarm (17020= ‘index error’) is output, however, when nonexistent tool management data are read.

This means that tool management data records (backup files) that have been generated in the NCK with a special tool management function configuration can be transferred to other SINUMERIK 840D controls that have different tool management functions. The permitted data are then “filtered out”.

6.2 Saving the PLC data

Save DB4 using the programming unit (S7). The type and number of magazines, load points, stations and spindles are stored here. The basic program uses this information to set up the interface blocks.

6.3 Data backup on MMC hard disk

The Access database from directory **Services \ Tool Management \ Tool Data \ WZACCESS.MDB** must be backed up.

This file contains all the MMC tool data, i.e.
– start-up data (configuration, buffer, load magazine)
– tool catalog, tool cabinet
– magazine configurations
...

Note

It is absolutely essential to execute a Power ON for both the MMC and NCK, e.g. by OFF/ON, before data are backed up to ensure that the database is not opened.

7 Secondary Conditions

Hardware

- The MMC 100 can also be used with SW 3.2 and later.
- MMC 101 and MMC 102/103, 8MB user memory
- OP 030 with MMC only
- PCU50 with OP 012

Software

The PLC blocks for tool management must be linked into the PLC from the “basic program” toolbox (FC 6, FC 7, FC 8, FC 22).

Options

The tool management option must be active.

M06 and T commands

When the tool management is active, the T numbers and the command M06 are not transferred to the PLC as an auxiliary function but to the tool management interface DB 71 to DB 73 in the PLC.

Notes

Machine Data 8

8.1 Machine data

8.1.1 Display machine data for MMC

No.	Name TM_DEFAULT	Description	Default setting	Max. value
9412	TOOLSIZE	Default setting for tool size	1111	7777
9416	TOOLTYPE	Default setting for loading, tool type	120	900
9417	TOOLSTATE	Default setting for loading, tool status	0	256
9418	SHOW_TOOL_SIZE	The tool size is specified in two or four-digit format in displays.	0	256
9419	DELETE_TOOL	Automatic deletion of tool data on unloading 0: No automatic deletion 1: Automatic deletion	0	1

No.	Name TM_DEFAULT	Description	Default setting
9250	SKMGLIST	Display of magazine list (horizontal)	7
9251	SKTLLIST	Display of tool list (horizontal)	7
9252	SKTOOLLOAD	Access rights for loading	7
9253	SKTOOLUNLOAD	Access rights for unloading	7
9254	TOOL_MOVE	Access rights for relocation	7
9255	SKMGLREPR1	Display of 1st magazine list (horizontal)	7

No.	Name TM_DEFAULT	Description	Default setting
9256	SKMGLREPR2	Display of 2nd magazine list (vertical)	7
9257	SKMGLREPR3	Display of 3rd magazine list (vertical)	7
9258	SKCNNEWTOOLE	Access rights: Create new cutting edge	7
9259	SKNCDELTOOL	Access rights: Delete tool	7
9260	SKMGBUFFER	Access rights: ON/OFF Buffer	7
9261	SKMGFIND	Access rights: Search	7
9262	SKMGLISTPOS	Access rights: Positioning	7
9263	SKMGNEXT	Access rights: Scroll to next magazine	7
9264	SLTLNEWTOOL	Access rights: Create a new tool	7
9265	SKMTLREPR1	Display of 1st Tool list (vertical)	7
9266	SKMTLREPR2	Display of 2nd Tool list (vertical)	7
9267	SKMTLREPR3	Display of 3rd Tool list (vertical)	7
9268	SKFINDPL	Access rights: Empty location softkey	7
9269	SKFINDPLACE	Access rights: Empty location softkey and display tool list	7
9270	SKACTPLACE	Access rights: Load current location	7
9271	SKLDTOOLDAT	Access rights: View and edit tool data (the tool data can be protected individually with machine data 9201, 9202 and 9209).	7

Display MD for workshop interface (ShopMill)

MD	MD identifier	Comment	Default
9414	KIND_OF_TOOLMANAGEMENT	Type of TM display	1
9651	CMM_TOOL_MANAGEMENT	TM variant	4
9652	CMM_TOOL_LIFE_CONTROL	Tool monitoring	1

MD	MD identifier	Comment	Default
9663	CMM_TOOL_DISPLAY_IN_DIAM	Display of radius/diameter for tools	1
9672	CMM_FIXED_TOOL_PLACE	Fixed location coding	0
9673	CMM_TOOL_LOAD_STATION	Number of load station	1
9674	CMM_ENABLE_TOOL_MAGAZINE	Magazine list displays	1

8.1.2 Memory settings for TM function

18080 MD number	**MM_TOOL_MANAGEMENT_MASK**	
Default setting: 0x0	Min. input limit: 0	Min. input limit: 0xFFFF
Change valid after: POWER ON	Protection level: 1/7	Unit: –
Data type: DWORD	Applies to SW 2	
Meaning:	Activation of the tool management memory with “0” means: No memory is allocated to the set TM data, the tool management function is not available. Bit 0=1: Memory for TM-specific data is made available, the memory-reserving MD must be set accordingly (MM_NUM_MAGAZINE_LOCATION, MM_NUM_MAGAZINE) Bit 1=1: Memory for monitoring data is made available Bit 2=1: Memory for user data (CC data) is made available Bit 3=1: Memory for “consider adjacent location” is made available Bit 4=1: Memory and function enable for PI service _N_TSEARC = “Complex search for tools in magazines” are made available. Bit 5=1: Wear monitoring active (SW 5 and later) Bit 6=1: Wear grouping available (SW 5 and later) Bit 7=1: Reserve memory space for adapters of magazine locations Bit 8=1: Memory for operational and/or setup offsets Bit 9=1: Tools in a turret no longer leave their turret location on tool changes (in display). This explicit memory reservation allows allocation of just the amount of memory required for the functionality. Example: Standard memory reservation for TM function: MM_TOOL_MANAGEMENT_MASK = 3 (bit 0 + 1=1) means TM and TM monitoring data are available MM_TOOL_MANAGEMENT_MASK = 1 means TM without TM monitoring function data	
Special case, errors...		

8.1.3 NC-specific machine data

17500 MD number	**MAXNUM_REPLACEMENT_TOOLS** Maximum number of replacement tools	
Default setting: –1	Min. input limit: –1	Min. input limit: 32
Change valid after: POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 5.1 and higher	
Meaning:	With active TM function only Value Meaning –1 Number of replacement tools is not monitored 0 No replacement tools can be defined, i.e. a maximum of one tool can be assigned to each identifier. n>0 Exactly m replacement tools may be assigned to one identifier, i.e. a total equal to n+1 tools. This data does not affect memory requirements, but merely has a monitoring function.	
Related to...		
References:	Description of Functions: Memory Configuration (S7)	

18082 MD number	**MM_NUM_TOOL** Number of tools that the NCK can manage	
Default setting: 30	Min. input limit: 0	Min. input limit: 600
Change valid after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	Here, the number of tools that the NCK can manage is entered. The maximum possible number of tools corresponds to the number of edges in the NCK. Non-volatile memory space is reserved for the number of tools.	
Special cases, errors,...		
Related to...	MM_NUM_CUTTING_EDGES_IN_TOA	
References:	Description of Functions: Memory Configuration (S7) Tool Compensation (W1)	

18084 MD number	**MM_NUM_MAGAZINE** Number of magazines that the NCK can manage	
Default setting: 3	Min. input limit: 0	Min. input limit: 32
Change valid after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	Number of magazines that the NCK can manage (active and background magazines). The non-volatile memory for the magazines is reserved with this MD. Important: One load and one buffer magazine are set up for each TOA unit in the tool management system. These magazines must be included here. Value = 0: The TM function cannot be activated because no data can be set up.	
Special cases, errors,...		
Related to...		
References:	Description of Functions: Memory Configuration (S7)	

18086 MD number	**MM_NUM_MAGAZINE_LOCATION** Number of magazine locations that the NCK can manage.	
Default setting: 30	Min. input limit: 0	Min. input limit: 600
Change valid after: POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	Number of magazine locations that the NCK can manage The non-volatile memory for the magazine locations is reserved with this MD. Important: The number of buffers and load points must also be included here. Value = 0: The TM function cannot be activated because no data can be set up.	
Special cases, errors,...		
Related to...		
References:	Description of Functions: Memory Configuration (S7)	

18088 MD number	**MM_NUM_TOOL_CARRIER** Maximum number of toolholders	
Default setting: 0	Min. input limit: 0	Min. input limit: 99999999
Change valid after: POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 4.1 and higher	
Meaning:	Maximum number of definable toolholders for orientatable tools in the TO area. The value is divided by the number of active TO units. The resultant integer indicates how many toolholders can be defined per TO unit. Example: 2 channels are active, with one channel per TO unit (=default). 3 holders must be defined in channel 1 and one holder in channel 2. The value to be set is 6 because 6/2 = 3.	
Special cases, errors,...		

18088 MD number	**MM_NUM_TOOL_CARRIER** Maximum number of toolholders	
Related to...		
References:	Description of Functions: Tool Offsets (S7)	

18090 MD number	**MM_NUM_CC_MAGAZINE_PARAM** Number of magazine data for user/compile cycles	
Default setting: 0	Min. input limit: 0	Min. input limit: 10
Change valid after POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	Number of magazine parameters (of type integer) that are available to the user or compile cycle. If this machine data is set, the amount of non-volatile memory required increases by sizeof(int)*max. number of magazines.	
Special cases, errors,...		
Related to...	MM_NUM_MAGAZINE	
References:		

18091 MD number	**MM_TYPE_CC_MAGAZINE_PARAM[n]** Type definition for magazine-related user data	
Default setting: 3	Min. input limit:	Min. input limit:
Change valid after: POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies to SW 5.2 and higher	
Meaning:	**The default settings for this machine data must not be altered.** Used to assign individual types to the parameters. The array index n can assume values between 0 and the setting in machine data MD 18090: MM_NUM_CC_MAGAZINE_PARAM. The possible values of the MD = 1, 2, 3, 4 and 5 stand for NC language types BOOL, CHAR, INT, REAL and STRING. Types FRAME and AXIS cannot be defined here. Type STRING must not be longer than 31 characters. Example: MD 18090: MM_NUM_CC_MAGAZINE_PARAM=1 MD 18091: MM_TYPE_CC_MAGAZINE_PARAM=5 Parameter $TC_MAPC1 can then be programmed to “UserMagazine”. The non-volatile RAM is used. Changing the value can, but does not necessarily, result in reconfiguration of the non-volatile memory.	
Related to...	MD 18201: MM_TYPE_CCS_MAGAZINE_PARAM	
References:		

18092 MD number	**MM_NUM_CC_MAGLOC_PARAM** Number of magazine location data for user/compile cycles	
Default setting: 0	Min. input limit: 0	Min. input limit: 10
Change valid after POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	Number of magazine location data parameters (of type integer) that are available to the user or compile cycle. If this machine data is set, the amount of non-volatile memory required increases by sizeof(int) ∗ max. number of magazine locations.	
Special cases, errors,...		
Related to...	MM_NUM_MAGAZINE_LOCATION	
References:		

18093 MD number	**MM_TYPE_CC_MAGLOG_PARAM[n]** Type definition for magazine location-related user data	
Default setting: 3	Min. input limit:	Min. input limit:
Change valid after: POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies to SW 5.2 and higher	
Meaning:	**The default settings for this machine data must not be altered.** Used to assign individual types to the parameters. The array index n can assume values between 0 and the setting in machine data MD 18090: MM_NUM_CC_MAGAZINE_PARAM. The possible values of the MD = 1, 2, 3, 4 and 5 stand for NC language types BOOL, CHAR, INT, REAL and STRING. Types FRAME and AXIS cannot be defined here. Type STRING must not be longer than 31 characters. Example: MD 18090: MM_NUM_CC_MAGAZINE_PARAM=1 MD 18091: MM_TYPE_CC_MAGAZINE_PARAM=5 Parameter $TC_MPPC1 can then be programmed to “UserMagazine”. The non-volatile RAM is used. Changing the value can, but does not necessarily, result in reconfiguration of the non-volatile memory.	
Related to...	MD 18203: MM_TYPE_CCS_MAGLOG_PARAM	
References:		

18094 MD number	**MM_NUM_CC_TDA_PARAM** Number of tool parameters for user/compile cycles	
Default setting: 0	Min. input limit: 0	Min. input limit: 10
Change valid after POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	Number of tool-specific data that can be created for each tool (of type integer) and are available to the user or compile cycle. If this machine data is set, the amount of non-volatile memory required increases by sizeof(double) ∗ max. number of tools.	
Special cases, errors,...		

18094 MD number	**MM_NUM_CC_TDA_PARAM** Number of tool parameters for user/compile cycles	
Related to...		
References:		

18095 MD number	**MM_TYPE_CC_TDA_PARAM[n]** Type definition for tool-related user data	
Default setting: 4	Min. input limit:	Min. input limit:
Change valid after: POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies to SW 5.2 and higher	
Meaning:	**The default settings for this machine data must not be altered.** Used to assign individual types to the parameters. The array index n can assume values between 0 and the setting in machine data MD 18094: MM_NUM_CC_TDA_PARAM. The possible values of the MD = 1, 2, 3, 4 and 5 stand for NC language types BOOL, CHAR, INT, REAL and STRING. Types FRAME and AXIS cannot be defined here. Type STRING must not be longer than 31 characters. Example: MD 18094: MM_NUM_CC_TDA_PARAM=1 MD 18095: MM_TYPE_CC_TDA_PARAM=5 Parameter $TC_TPC1 can then be programmed to “UserEdge”. The non-volatile RAM is used. Changing the value can, but does not necessarily, result in reconfiguration of the non-volatile memory.	
Related to...	MD 18205: MM_TYPE_CCS_TDA_PARAM	
References:		

18096 MD number	**MM_NUM_CC_TOA_PARAM** Number of TOA data for user/compile cycles	
Default setting: 0	Min. input limit: 0	Min. input limit: 10
Change valid after POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	Number of TOA data that are created for each tool (of type DOUBLE) and are available to the user or compile cycle. If this machine data is set, the amount of non-volatile memory required increases by sizeof(double) ∗ max. number of edges.	
Special cases, errors,...		
Related to...	MM_NUM_CUTTING_EDGES_IN_TOA	
References:		

18097 MD number	**MM_TYPE_CC_TOA_PARAM[n]** Type definition for cutting-edge-related user data			
Default setting: 4		Min. input limit:		Min. input limit:
Change valid after: POWER ON			Protection level: 2/2	Unit: –
Data type: DWORD			Applies to SW 5.2 and higher	
Meaning:	**The default settings for this machine data must not be altered.** Used to assign individual types to the parameters. The array index n can assume values between 0 and the setting in machine data MD 18096: MM_NUM_CC_TOA_PARAM. The possible values of the MD = 1, 2, 3, 4 and 5 stand for NC language types BOOL, CHAR, INT, REAL and STRING. Types FRAME and AXIS cannot be defined here. Type STRING must not be longer than 31 characters. Example: MD 18096: MM_NUM_CC_TOA_PARAM=1 MD 18097: MM_TYPE_CC_TOA_PARAM=5 Parameter $TC_DPC1 can then be programmed to “UserEdge”. The non-volatile RAM is used. Changing the value can, but does not necessarily, result in reconfiguration of the non-volatile memory.			
Related to...	MD 18207: MM_TYPE_CCS_TOA_PARAM			
References:				

18098 MD number	**MM_NUM_CC_MON_PARAM** Number of monitoring data for user/compile cycles			
Default setting: 0		Min. input limit: 0		Min. input limit: 10
Change valid after POWER ON			Protection level: 2/2	Unit: –
Data type: DWORD			Applies to SW 2 and higher	
Meaning:	Number of monitoring data that are created for each tool (of type integer) and are available to the user or compile cycle. If this machine data is set, the amount of non-volatile memory required is increased by sizeof(int) ∗ max. number of cutting edges.			
Special cases, errors,...				
Related to...				
References:				

18099 MD number	**MM_TYPE_CC_MON_PARAM[n]** Type definition for monitoring-related user data	
Default setting: 3	Min. input limit:	Min. input limit:
Change valid after: POWER ON	Protection level: 2/2	Unit: –
Data type: DWORD	Applies to SW 5.2 and higher	
Meaning:	**The default settings for this machine data must not be altered.** Used to assign individual types to the parameters. The array index n can assume values between 0 and the setting in machine data MD 18098: MM_NUM_CC_MON_PARAM. The possible values of the MD = 1, 2, 3, 4 and 5 stand for NC language types BOOL, CHAR, INT, REAL and STRING. Types FRAME and AXIS cannot be defined here. Type STRING must not be longer than 31 characters. Example: MD 18098: MM_NUM_CC_MON_PARAM=1 MD 18099: MM_TYPE_CC_MON_PARAM=5 Parameter $TC_MOPC1 can then be programmed to “UserEdge”. The non-volatile RAM is used. Changing the value can, but does not necessarily, result in reconfiguration of the non-volatile memory.	
Related to...	MD 18209: MM_TYPE_CCS_MON_PARAM	
References:		

18100 MD number	**MM_NUM_CUTTING_EDGES_IN_TOA** Number of cutting edges per TOA module	
Default setting: 30	Min. input limit: 0	Min. input limit: 600/1500 (SW 5 and later)
Change valid after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 1 and higher	
Meaning:	Number of possible cutting edges in the TOA area. The TOA area is the sum of all TOA modules (if tool management is active, also of all magazine blocks) in the NCK.	
Special cases, errors,...		
Related to...		
References:	Description of Functions: Memory Configuration (S7)	

18102 MD number	**MM_TYPE_OF_CUTTING_EDGE** Method of D programming	
Default setting: 0	Min. input limit: 0	Min. input limit: 1
Change valid after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 4.1 and higher	
Meaning:	Value Meaning 0 No “flat D number management” active 1 D numbers are programmed directly and absolutely Values 2 and 3 not yet enabled	
Special cases, errors,...		
Related to...		
References:	Description of Functions: Tool Offsets (W1)	

18104 MD number	**MM_NUM_TOOL_ADAPTER** Tool adapter in TO area	
Default setting: 0	Min. input limit: –1	Min. input limit: 600
Change valid after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 5 and higher	
Meaning:	Contains the number of tool data records available in the NCK. This function can only be used if magazine locations are available in the NCK. I.e. the tool management function must be active. In order to activate the setting, bit 7 (=0x80) must be set in MD $MN_MM_TOOL_MANAGEMENT _MASK. Adapter data records and the tool-edge-specific base/adapter data records are mutually exclusive. I.e. when adapter data are defined, parameters $TC_DP21, $TC_DP22, $TC_DP23 and their values are available in the NCK. Value Meaning –1 Every magazine location is automatically assigned an adapter, i.e. the number of adapters made available internally corresponds to the number of magazine locations set in MD $MN_MM_NUM_MAGAZINE_LOCATION. 0 Adapter data cannot be defined. Edge-specific parameters $TC_DP21, $TC_DP22 and $TC_DP23 are available in cases where adapters are utilized outside the active TM. >0 Number of adapter data records. These allow adapters to be defined independently of magazine locations. An additional step following definition of the data assigns the adapters to magazine locations.	
Related to...	MD 18080: MM_TOOL_MANAGEMENT_MASK MD 20310: TOOL_MANAGEMENT_MASK MD 18084: MM_NUM_MAGAZINE MD 18086: MM_NUM_MAGAZINE_LOCATION	
References:		

18105 MD number	**MM_MAX_CUTTING_EDGE_NO** Maximum value of D number	
Default setting: 9	Min. input limit: 1	Min. input limit: 32000
Change valid after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 5 and higher	
Meaning:	The maximum number of D numbers per cutting edge is unaffected by this MD. The monitoring of D number assignments associated with this setting applies only to newly defined D numbers. If MM_MAX_CUTTING_EDGE_NO is > MM_MAX_CUTTING_EDGE_PERTOOL, additional memory will be required. In this case, the “unique D numbers” function can be used.	
Related to...	MD 18106: MM_MAX_CUTTING_EDGE_PERTOOL	
References:	Description of Functions: Tool Offsets (W1)	

18106 MD number	**MM_MAX_CUTTING_EDGE_PERTOOL** Maximum number of cutting edges (D offset) per tool (per T number)	
Default setting: 9	Min. input limit: 1	Min. input limit: 12
Change valid after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 5 and higher	
Meaning:	A value of 1 can be set if only tools with 1 cutting edge are to be used. This will avoid the problem of assigning more than one cutting edge to the tool when data are defined. If MM_MAX_CUTTING_EDGE_NO is > MM_MAX_CUTTING_EDGE_PERTOOL, additional memory will be required. In this case, the “unique D numbers” function can be used.	
Related to...	MD 18105: MM_MAX_CUTTING_EDGE_NO	
References:	Description of Functions: Tool Offsets (W1)	

18108 MD number	**MM_NUM_SUMCORR** Total number of sum offsets in NCK	
Default setting: –1	Min. input limit: –1	Min. input limit: 9000
Change valid after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 5 and higher	
Meaning:	A setting of –1 means that the number of sum offsets equals the number of cutting edges * number of sum offsets per cutting edge. Non-volatile memory is reserved.	
Related to...	MD 18100: MM_NUM_CUTTING_EDGE_IN_TOA MD 18110: MM_MAX_SUMCORR_PER_CUTTEDGE	
References:	Description of Functions: Tool Offsets (W1)	

18110 MD number	**MM_MAX_SUMCORR_PER_CUTTEDGE** Maximum number of sum offsets per cutting edge	
Default setting: 1	Min. input limit: 1	Min. input limit: 6
Change valid after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 1 and higher	
Meaning:	The following applies to MM_NUM_SUMCORR > 0: This data does not reserve memory, but is used for monitoring purposes only. The following applies to MM_NUM_SUMCORR > –1: This data reserves non-volatile memory.	
Related to...	MD 18100: MM_NUM_CUTTING_EDGE_IN_TOA MD 18108: MM_NUM_SUMCORR	
References:	Description of Functions: Memory Configuration (S7)	

18112 MD number	**MM_KIND_OF_SUMCORR** Properties of sum offsets in NCK	
Default setting: 0	Min. input limit: 0	Min. input limit: 0x1F
Change valid after POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 5 and higher	
Meaning:	Bit 0=0 Sum offsets are saved when the tool data are backed up. Bit 0=1 Sum offsets are not saved when the tool data are backed up. Bit 1=0 Setup offsets are saved when the tool data are backed up. Bit 1=1 Setup offsets are not saved when the tool data are backed up. Bit 2=0 If the “TM” function is in use, the existing sum offsets are not affected when the tool status is set to “active”. Bit 2=1 When the tool status is set to “active”, the existing sum offsets are set to the value 0. Bit 3=0 If the functions “TM” + “Adapter” are in use: Sum offsets are transformed. Bit 3=1 Sum offsets are not transformed. Bit 4=0 No setup offset data records. Bit 4=1 Setup offset data records are set up additionally. The sum offset is thus the product of the setup offset + “sum offset fine”. Changing the states of bits 0, 1, 2 and 3 does not alter the memory configuration. Changing the status of bit 4 causes the non-volatile memory to be re-configured with the next Power ON operation.	
Related to...	MD 18100: MM_NUM_CUTTING_EDGE_IN_TOA MD 18108: MM_NUM_SUMCORR MD 18110: MM_MAX_SUMCORR_PER_CUTTEDGE MD 18080: MM_TOOL_MANAGEMENT_MASK MD 20310: MC_TOOL_MANAGEMENT_MASK MD 18086: MM_NUM_MAGAZINE_LOCATION MD 18104: MM_NUM_TOOL_ADAPTER	
References:	Description of Functions: Tool Offsets (W1)	

8.1.4 Channel-specific machine data

20310 MD number	**TOOL_MANAGEMENT_MASK** Channel-specific activation of the tool management	
Default setting: 0x0, ...	Min. input limit: 0	Min. input limit: 0xFFFFF
Change valid after: POWER ON	Protection level: 2/7	Unit: HEX
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	MD = 0: TM inoperative Bit 0=1: TM operative The tool monitoring functions are enabled for the current channel. Bit 1=1: TM monitoring function active The functions for monitoring tools (tool life and workpiece count) are enabled. Bit 2=1: OEM functions operative The memory for user data can be utilized (see also MD 18090 to 18098) Bit 3=1: “Consider adjacent location” operative **Bits 0 to 3** must be set identically to MD 18080: MM_TOOL_MANAGEMENT_MASK. Bit 4=1: The PLC has the option of requesting another tool change preparation operation with modified parameters. Part program stops at T selection or M06 until it has been acknowledged by the PLC program.	
	Bit 5 Main run / PLC synchronization in connection with “Tool Change Load” command for the main spindle takes place with the transport acknowledgement to the NCK. Bit 19 = 0 Synchronization in relation to tool change. The command is not deemed to be valid until the specified acknowledgement from the PLC has arrived in the NCK. Bit 19 = 1 Synchronization in relation to IPO block, i.e. the main run block remains active until the specified acknowledgement from the PLC has arrived PLC has arrived in the NCK.	
	Bit 6 Main run / PLC synchronization in connection with “Tool Change Load” command for a secondary spindle takes place with the transport acknowledgement. Bit 19 = 0 Synchronization in relation to tool change. The command is not deemed to be valid until the specified acknowledgement from the PLC has arrived in the NCK. Bit 19 = 1 Synchronization in relation to IPO block, i.e. the main run block remains active until the specified acknowledgement from the PLC has arrived in the NCK.	
	Bit 7 Main run / PLC synchronization in connection with “Tool Change Load” command for the main spindle does not take place until the PLC acknowledges that the “Tool Change Load” command has been executed. Bit 19 = 0 Synchronization in relation to tool change. The command is not deemed to be valid until the specified acknowledgement from the PLC has arrived in the NCK. Bit 19 = 1 Synchronization in relation to IPO block, i.e. the main run block remains active until the specified acknowledgement from the PLC has arrived PLC has arrived in the NCK.	

20310 MD number	**TOOL_MANAGEMENT_MASK** Channel-specific activation of the tool management	
Meaning:	Bit 8 Main run / PLC synchronization in connection with “Tool Change Load” command for a secondary spindle does not take place until the PLC acknowledges that the “Tool Change Load” command has been executed.	
	Bit 19 = 0	Synchronization in relation to tool change. The command is not deemed to be valid until the specified acknowledgement from the PLC has arrived in the NCK.
	Bit 19 = 1	Synchronization in relation to IPO block, i.e. the main run block remains active until the specified acknowledgement from the PLC has arrived PLC has arrived in the NCK.
	Bit 9:	Reserved for test purposes may also be used by machine manufacturers in the test phase, until the PLC program can handle tool change
Meaning:	Bit 10=1:	M06 is delayed until “Prepare change” has been executed by the PLC via FC8 (status 1 ...). The change signal (e.g. M06) is not output until the tool selection via FC8 FC8 (DBX [n+0].2 = 0) has been acknowledged. The part program is stopped at M06 until the T selection has been acknowledged.
	Bit 10=0	The “Tool Change Load” command is not output from NCK–> PLC until the PLC preparation acknowledgement has been received. This is relevant for PLC command 3 (i.e. programming of M06 in a block which does not contain a T).
Meaning:	Bit 11=1:	The prepare command is output again, even if it has already been issued once for the same tool. The purpose of this repeat output is to position the chain with the first ‘Tx’ call and check whether the tool is in the correct change location (e.g. in front of a tool change station) with the 2nd call.
	Bit 11=0	The prepare command can be output only once for a tool.
Meaning:	Bit 12=1:	The prepare command is executed even if the tool is already loaded in the spindle, i.e. the T selection signal (DB72.DBXn.2) is set even if it has already been set once for the same tool. (Tx...Tx).
	Bit 12=0:	The prepare command is not executed if the tool is already loaded in the spindle.
	Bit 13=1:	The commands from the diagnostics buffer are stored in the passive file system (NCATR xx.MPF under part program) on Reset. This file is required by the hotline. The tool sequences are only recorded in the diagnostic buffer on systems with sufficient memory (NCU572, NCU573).
	Bit 14=1:	Tool offset selection on reset Start according to the settings of machine data MD20112 START_MODE_MASK MD20120 TOOL_RESET_NAME MD20110 RESET_MODE_MASK MD20124 TOOL_MANAGEMENT_TOOLHOLDER If the tool defined in TOOL_RESET_NAME must be loaded (acc. to setting in RESET_MODE_MASK), a selection and tool change command is output to the user interface on RESET and START. (DB 72) If RESET_MODE_MASK is set to retain the active tool and the active tool is disabled in the spindle (by user), a tool change command for a replacement tool is output to the user interface. MD22562 is displayed in the event of a tool change error.
	Bit 14=0:	No automatic tool change takes place on RESET and Start.

20310 MD number	**TOOL_MANAGEMENT_MASK** Channel-specific activation of the tool management
Meaning:	Bit 15=1: No return transport of tool if several prepare commands have been issued (Tx–>Tx). Bit 15=0: Tool is returned from any defined buffers. Bit 16=1: T=Location number is active Bit 16=0: T=“T name” Bit 17=1: Tool life decrementation can be started/stopped via the PLC in channel DB 2.1...DBx 1.3. Bit 18=1: Activation of monitoring mode “Last tool in tool group” Bit 18=0: No monitoring for “Last tool in tool group” Bit 19=1: The synchronizations defined by bits 5 to 8 refer to the main run block Bit 19=0: The synchronizations defined by bits 5 to 8 refer to the TM command output
Meaning:	Bit 20=0: With PLC signal “Program testing active”, the generated commands are not output to the PLC. NCK acknowledges the commands itself. Magazine and tool data are not altered. Bit 20=1: With PLC signal “Program testing active”, the generated commands are output to the PLC; depending on acknowledgement mode, this may alter tool/magazine data in the NCK. If the acknowledgement parameters for the “target magazine” are set to the same values as the “source magazine”, the tool is not returned and therefore no data altered in the NCK. In the default setting, i.e. bits 6–9=0, synchronization takes place (in relation to a tool change for the main spindle) in the block in which a cutting edge of the new tool is selected for the first time. Synchronization can be implemented either via the transport acknowledgement or the end acknowledgement. Bit 0 must be set to make bit 1 and higher operative. Bit 0 cannot be set if the TM option is not installed.
	Bit 21=0: Default setting: Ignore tool status “W” on tool selection Bit 21=1: Tools with status “W” cannot be selected by another tool change, tool preparation command.
Related to...	
References:	

Activation of tool management (only if TM option is set).
The complete tool change process generally consists of 2 commands to the PLC:
– Tool change preparation and
– Tool change load
You must be familiar with these terms if you wish to utilize the following setting options.
Bits 5, 6, 7 and 8 slow down the block processing sequence. In this case, bits 7 and 8 slow the process more than bits 5 and 6.
Bit 18 increases the length of a search operation for a suitable tool, especially if the search area includes a large number of disabled replacement tools.
Bit 19 in combination with set bits 5, 6, 7 and 8 slow down block processing.

20110 MD number	**RESET_MODE_MASK** Definition of initial setting of control system after power-up and Reset	
Default setting: 0x0	Min. input limit: 0	Min. input limit: 0x7FFF
Change valid after: RESET	Protection level: 2/7	Unit: HEX
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	Definition of initial setting of control system after power-up and Reset/end of part program in relation to G codes (in particular, current plane and settable zero offset), tool length compensation and transformation on basis of following bit settings (only the bits marked in bold type are relevant for the tool management): **Bit 0 Reset mode** Bit 1 Suppress auxiliary function output on tool selection **Bit 2 Select Reset response after Power On; e.g. tool offset** **Bit 3 Select Reset response at end of test mode in relation to active tool offsets. This bit is relevant only if bits 0 and 6 are set.** **It identifies the program to which “current setting for active tool length compensation” refers;** **– the program that was active at the end of test mode or** **– the program that was active prior to activation of test mode.** Bit 4 Reserved! Setting is now made in $MC_GCODE_RESET_MODE[..] Bit 5 Reserved! Setting is now made in $MC_GCODE_RESET_MODE[..] **Bit 6 Reset response “Active tool length compensation”** Bit 7 Reset response “Active kinematic transformation” Bit 8 Reset response “Coupled-motion axes” Bit 9 Reset response “Tangential follow-up” Bit 10 Reset response “Synchronous spindle” Bit 11 Reset response “Revolutional feedrate” Bit 12 Reset response “Geo-axis replacement” Bit 13 Reset response “Master value coupling” Bit 14 Reset response “Basic frame” Bits 4 to 11 are evaluated only if bit 0=1.	
Related to...		
References:	Description of Functions: Coordinate Systems (K2)	

20112 MD number	**START_MODE_MASK** Definition of initial setting of control system after part program start	
Default setting: 0x400	Min. input limit: 0	Min. input limit: 0x7FFF
Change valid after: RESET	Protection level: 2/7	Unit: HEX
Data type: DWORD	Applies to SW 3 and higher	
Meaning:	Definition of initial setting of control system on part program start in relation to G codes (in particular, current plane and settable zero offset), tool length compensation, transformation and axis coupling on basis of following bit settings (only the bits marked in bold type are relevant for the tool management): **Bit 0 Not assigned: $MC_START_MODE_MASK is evaluated every time a part program is started** Bit 1 Suppress auxiliary function output on tool selection Bit 4 Start response G code “Current plane” Bit 5 Start response G code “Settable zero offset” **Bit 6 Start response “Active tool length compensation”** Bit 7 Start response “Active kinematic transformation” Bit 8 Start response “Coupled-motion axes” Bit 9 Start response “Tangential follow-up” Bit 10 Start response “Synchronous spindle” Bit 11 Reserved Bit 12 Start response “Geo-axis replacement” Bit 13 Start response “Master value coupling” Bit 14 Start response “Basic frame”	
Related to...		
References:	Description of Functions: Coordinate Systems (K2)	

20120 MD number	**TOOL_RESET_VALUE (without TM only)** Tool whose length compensation is selected during power-up	
Default setting: 0	Min. input limit: 0	Min. input limit: 32000
Change valid after: RESET	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	This data is valid only when the TM function is not active. Definition of tool of which length compensation is selected during power-up and on Reset/ end of part program as a function of MD 20110 and on start of part program as a function of MD 20112.	
Related to...	MD 20110 and 20112	
References:	Description of Functions: Coordinate Systems (K2)	

20121 MD number	**TOOL_PRESEL_RESET_VALUE** Preselect tool on Reset		
Default setting: 0	Min. input limit: 0	Min. input limit: 32000	
Change valid after: RESET	Protection level: 2/7	Unit: –	
Data type: DWORD	Applies to SW 4.1 and higher		
Meaning:	This data is valid only when the TM function is not active. Definition of preselected tool when MD 20310=1. A tool is preselected after power-up and on Reset or end of part program as a function of MD 20110 and on start of part program as a function of MD 20112.		
Related to...	MD 20110: RESET_MODE_MASK and 20112: START_MODE_MASK		
References:	Description of Functions: Coordinate Systems (K2)		

20122 MD number	**TOOL_RESET_NAME** Active tool on RESET and tool management		
Default setting: –	Min. input limit: –	Min. input limit: –	
Change valid after: RESET	Protection level: 2/7	Unit: –	
Data type: STRING	Applies to SW 3.2 and higher		
Meaning:	This data is valid only if the TM function is active. Definition of the tool with which tool length compensation is selected during power-up and on Reset or end of part program as a function of MD 20110 RESET_MODE_MASK and on start of part program as a function of MD 20112 START_MODE_MASK.		
Related to...	MD 20110: RESET_MODE_MASK, MD 20112: START_MODE_MASK MD 20124: TOOL_MANAGEMENT_TOOLHOLDER MD 20130: CUTTING_EDGE_RESET_VALUE		
References:			

<table>
<tr><td><b>20124</b>
MD number</td><td colspan="5"><b>TOOL_MANAGEMENT_TOOLHOLDER</b>
Toolholder number</td></tr>
<tr><td colspan="2">Default setting: DBB n + 0,0,0</td><td colspan="2">Min. input limit: 0</td><td colspan="2">Min. input limit: 16</td></tr>
<tr><td colspan="3">Change valid after: POWER ON</td><td colspan="2">Protection level: 2/7</td><td>Unit: –</td></tr>
<tr><td colspan="4">Data type: DWORD</td><td colspan="2">Applies to SW 3.2 and higher.</td></tr>
<tr><td>Meaning:</td><td colspan="5">This data is relevant only when the tool management function is active.
Defines whether the toolholder no. or the spindle no. is specified in order to establish the insertion location of a tool to be loaded. The tool management must be informed of the toolholder in which a tool is to be loaded.
If the MD is greater than 0, the spindle numbers $TC_MPP5 are interpreted as toolholder numbers.
The automatic address extension of T and M06 is then the value of this MD and no longer the value of MD 20090 SPIND_DEF_MASTER_SPIND.
On machines with several toolholders without a declared master spindle, the MD is used as a default value in order to determine the toolholder into which the tool is to be placed on the tool change. SETMTH(n) declares toolholder n as the master toolholder.
Tools which are to be loaded in a buffer location of the spindle type and which have the value $TC_MPP5=n correct the tool path. Tools with a value not equal to n have no effect on the correction.
The command SETMTH is used to declare the toolholder defined in the MD as master toolholder again.
When defining the magazine locations of internal magazines, spindle locations –$TC_MPP1=2=spindle location– can be assigned a location type index ($TC_MPP5). This assigns a specific toolholder to the location.</td></tr>
<tr><td>Related to...</td><td colspan="5">MD 20090 SPIND_DEF_MASTER_SPIND
MD 20110: RESET_MODE_MASK, MD 20112: START_MODE_MASK
MD 20122: TOOL_RESET_NAME
MD 20130: CUTTING_EDGE_RESET_VALUE</td></tr>
<tr><td>References:</td><td colspan="5"></td></tr>
</table>

<table>
<tr><td><b>20126</b>
MD number</td><td colspan="5"><b>TOOL_CARRIER_RESET_VALUE</b>
Operative toolholder on Reset</td></tr>
<tr><td colspan="2">Default setting: 0</td><td colspan="2">Min. input limit: 0.0</td><td colspan="2">Min. input limit: plus</td></tr>
<tr><td colspan="3">Change valid after: Reset</td><td colspan="2">Protection level: 2/7</td><td>Unit: –</td></tr>
<tr><td colspan="4">Data type: DWORD</td><td colspan="2">Applies to SW 4.1 and higher</td></tr>
<tr><td>Meaning:</td><td colspan="5">Definition of tool with which tool length compensation is selected during power-up and on Reset or end of part program as a function of machine data $MC_RESET_MODE_MASK and on start of part program as a function of machine data $MC_START_MODE_MASK.</td></tr>
<tr><td>Related to...</td><td colspan="5">MD 20110; MD 20112</td></tr>
<tr><td>References:</td><td colspan="5">Description of Functions: Tool Offsets (W1)</td></tr>
</table>

© Siemens AG 2000. All Rights Reserved

20128 MD number	**COLLECT_TOOL_CHANGE** Collect tool change in search run	
Default setting: 1	Min. input limit: –	Min. input limit: –
Change valid after: Immediately	Protection level: 2/7	Unit: –
Data type: BOOLEAN	Applies to SW 4.3 and higher	
Meaning:	The MD is only relevant when tool management is active. It determines whether the tool change M code defined in MD 22560: TOOL_CHANGE_M_CODE must be collected during block searches with calculation. TRUE: Tool change M code is collected FALSE: Tool change M code is not collected The tool determined in the search run is displayed and treated as the current tool. The T number output is not affected by this. The tool offset data determined in the NCK become effective. No change takes place in the magazine data, etc. Without tool management, the tool change M code is not collected if it is not assigned to an auxiliary function group.	
Related to...	MD 22560: TOOL_CHANGE_M_CODE	
References:		

20130 MD number	**CUTTING_EDGE_RESET_VALUE** Tool cutting edge whose length compensation is selected during power-up	
Default setting: 0	Min. input limit: 0	Min. input limit: 32000
Change valid after: RESET	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	Definition of tool cutting edge with which tool length compensation is selected during power-up and on Reset or end of part program as a function of machine data $MC_RESET_MODE_MASK and on start of part program as a function of machine data $MC_START_MODE_MASK. When tool management is active and bits 0 and 6 set in $MC_RESET_MODE_MASK, the last offset of the tool which was active on power-off – generally the tool in the spindle – is operative after power-on.	
Related to...	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK	
References:	Description of Functions: Coordinate Systems (K2)	

20132 MD number	**SUMCORR_RESET_VALUE** Effective sum offset on Reset	
Default setting: 0	Min. input limit: 0	Min. input limit: 6
Change valid after: RESET	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 5 and higher	
Meaning:	Definition of sum offset with which tool length compensation is selected during power-up and on Reset or end of part program as a function of machine data $MC_RESET_MODE_MASK and on start of part program as a function of machine data $MC_START_MODE_MASK. Machine data $MN_MAX_SUMCORR_PERCUTTING_EDGE determines the maximum meaningful value which can be entered.	
Related to...	MD 20110: MD 20112	
References:	Description of Functions: Tool Offsets (W1)	

20270 MD number	**CUTTING_EDGE_DEFAULT** Initial setting of tool edge after tool change without programming	
Default setting: 1	Min. input limit: –2	Min. input limit: 32000
Change valid after: POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 5.2 and higher	
Meaning:	If no cutting edge is programmed after a tool change, then the edge number preset in CUTTING_EDGE_DEFAULT is applied. Value = 0: No cutting edge is initially active after a tool change. Edge is selected only when D number is programmed. Value = 1: MD_SLMAXCUTTINGEDGENUMBER, no. of cutting edge (equals 9 in SW 4 and earlier) Value = –1: Edge number of old tool also applies to new tool Value = –2: Edge (offset) of old tool remains active until a D number is programmed.	
Related to...		
References:	Description of Functions: Tool Offsets (W1)	

20272 MD number	**SUMCORR_DEFAULT** Initial setting of sum offset without programming	
Default setting: 0	Min. input limit: –1	Min. input limit: 6
Change valid after: POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 5 and higher	
Meaning:	The MD specifies the number of the sum offset of the tool edge which becomes active if a new cutting edge offset is activated when a programmed DL value is not available. Machine data $MN_MAX_SUMCORR_PERCUTTING_EDGE determines the maximum meaningful value which can be entered. Value Meaning > 0 Number of sum offset = 0 No sum offset active with programmed D = –1 The sum offset number for previously programmed D is applied.	
Related to...	MD 20270: CUTTING_EDGE_DEFAULT	
References:	Description of Functions: Tool Offsets (W1)	

20320 MD number	**TOOL_TIME_MONITOR_MASK** Time monitoring for tool in spindle	
Default setting: 0x0	Min. input limit: –	Min. input limit: –
Change valid after: POWER ON	Protection level: 2/7	Unit: HEX
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	Activation of tool time monitoring function for spindle 1....x. As soon as the path axes are moved (not with G00), the time monitoring data for the tool loaded in the appropriate spindle are updated. Bit 0...x–1: Monitoring of active tool in spindle 1...x	
Related to...		
References:	Description of Functions: Memory Configuration (S7)	

22550 MD number	**TOOL_CHANGE_MODE** New tool offset for M function	
Default setting: 0	Min. input limit: 0	Min. input limit: 1
Change valid after: POWER ON	Protection level: 2/7	Unit: –
Data type: BYTE	Applies to SW 1.1 and higher	
Meaning:	This machine data determines the mode of tool change MD: TOOL_CHANGE_MODE = 0 The new tool data become effective directly when T or D is programmed. This setting is predominantly used for turning machines with a tool turret. MD: TOOL_CHANGE_MODE = 1 The new tool is prepared for the tool change with the T function. This setting is used mainly on milling machines with tool magazine in order to move the new tool to the tool change position in parallel with the machining time (i.e. the machining operation is not interrupted). With the M function set in MD: TOOL_CHANGE_MODE, the old tool is removed from the spindle and the new tool loaded into it. This tool change must be programmed with M function M06 according to DIN 66025.	
Related to...	MD 22560: TOOL_CHANGE_M_MODE	
References:	Description of Functions: Tool Offsets (W1)	

22560 MD number	**TOOL_CHANGE_M_CODE** M function for tool change	
Default setting: 6	Min. input limit: 0	Min. input limit: 99999999
Change valid after: POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 1 and higher	
Meaning:	This machine data is effective only when TOOL_CHANGE_MODE = 1. If the T function is used only to prepare a new tool for tool change (for milling machines that use a tool magazine, this setting is mainly used to synchronize the new tool with the tool change position in parallel with the machining time), another M function must be taken for the tool change. The M function entered in TOOL_CHANGE_M_CODE triggers the tool change (old tool removed from spindle and new tool inserted into spindle). This tool change must be programmed with M function M06 according to DIN 66025.	
Related to...	MD 22550: TOOL_CHANGE_MODE	
References:	Functional description for tool offset (W1)	

22562 MD number	**TOOL_CHANGE_ERROR_MODE** Response to tool changing errors	
Default setting: 0x0	Min. input limit: 0	Min. input limit: 0x1F
Change valid after: POWER ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies to SW 5.1 and higher	
Meaning:	Bit 0=0 Standard response: Stop at NC block containing error Bit 0=1 If an error is detected in the block containing the tool change preparation command, the alarm is delayed until the associated tool change command (M06) is interpreted in the program run. Only then is the alarm output. The operator can make corrections in this block. This machine data is relevant only if the setting MC_TOOL_CHANGE_MODE = 1 is used. Bit 1 is relevant only when TM function is active. Bit 1=0 Standard response: When the tool change is being prepared, only tools are detected whose data are assigned to a magazine. Bit 1=1 A tool will be loaded if its data are registered in the NCK, but not assigned to a magazine location. The data are then automatically assigned to the programmed toolholder. The user is requested to place tools in the toolholder or remove tools from it. Bit 2 Qualification of offset programming Bit 2=0 T0 and Dx (x > 0) produce offset 0. D0 and DL = x (x > 0) produce offset 0 and thus also sum offset 0. Bit 2=1 T0 and Dx (x > 0) result in an alarm. D0 and DL = x (x > 0) result in an alarm. Bit 3=0 Default: If the tool on the spindle is blocked: create a tool change command to request a replacement tool. If there is no replacement, an alarm is produced. Bit 3=1 The blocked state of the spindle tool is ignored. The tool is active. The following part program should then be formulated so that no parts are machined with the blocked HARMONIZE tool. Bit 4=0 Default: An attempt has been made to activate the spindle tool or the replacement tool Bit 4=1 If the tool on the spindle is blocked, Initsat T0 is programmed in the start.	
Related to...		
References:	Description of Functions: Tool Offsets (W1)	

20090 MD number	**SPIND_DEF_MASTER_SPIND** Initial setting of master spindle in channel	
Default setting: 1, 1, 1, 1,...	Min. input limit: 1	Min. input limit: 15
Change valid after: POWER ON	Protection level: 2/7	Unit: –
Data type: BYTE	Applies to SW 1 and higher	
Meaning:	Definition of master spindle in channel. The number of the spindle is set. Example: 1 equals spindle S1. When S is programmed, the current master spindle is automatically addressed. (The SETMS(n) command can be programmed to declare the spindle number as the master spindle. SETMS declares the spindle defined in the MD to be the master spindle again.	

20090 MD number	**SPIND_DEF_MASTER_SPIND** Initial setting of master spindle in channel
Related to...	
References:	Description of Functions: Spindles (S1)

28085 MD number	**LINK_TOA_UNIT** Assignment of a TO unit to a channel	
Default setting: 1, 2, 3, 4, 5, ...	Min. input limit: 1	Min. input limit: 10
Change valid after: POWER ON	Protection level: 2/7	Unit:
Data type: DWORD	Applies to SW 2 and higher	
Meaning:	The T0 area includes all the tool, magazine, ... data blocks which the NCK can identify. The maximum number of units in the TO area match the maximum number of channels. If LINK_TOA_UNIT = default, then each individual channel is assigned a TO unit. When LINK_TOA_UNIT = i, the channel is assigned TO unit i. In this way it is possible to assign one TO unit to several channels. *Important* The upper limit value does not imply that the value is always meaningful or without conflict. If one channel (the first) is active and the other not on a system with a total of 2 channels, the MD on channel 1 can be formally set to a value of 2. However, the NCK cannot work with this setting because it would mean that channel 1 possesses no data blocks for tool offsets since a channel with Id=2 does not exist. The NCK detects this conflict during power-on or a warm restart and reacts by independently changing the (incorrect) setting to the default setting for the MD.	
Related to...		
References:	Description of Functions: Memory Configuration (S7)	

8.1.5 Machine data for function replacement

10715 MD number	**M_NO_FCT_CYCLE** M function number for cycle call	
Default setting: –1	Min. input limit: –1	Min. input limit: –
Change valid after POWER ON	Protection level: 2/4	Unit: –
Data type: DWORD	Applies to SW 5.2 and higher	
Meaning:	M number with which a subroutine is called. The name of the subroutine is stored in $MN_M_NO_FCT_CYCLE_NAME. If the M function defined by $MN_M_NO_FCT_CYCLE is programmed in a part program, the subroutine defined in M_NO_FCT_CYCLE_NAME is started by the end of the block. If the M function is programmed again in the subroutine, it is not replaced a second time by a subroutine call. $MN_M_NO_FCT_CYCLE is effective both in Siemens mode G290 and in external language mode G291. A subroutine call may not be superimposed on M functions with a fixed meaning. Alarm 4150 is generated in case of a conflict: – M0 to M5, – M17, M30, – M40 to M45, – M function for switching between spindle and axis modes as set in $MC_SPIND_RIGID_TAPPING_M_NR (default M70) – M functions for nibbling/punching as configured in $MC_NIBBLE_PUNCH_CODE provided they have been activated via $MC_PUNCHNIB_ACTIVATION. – With applied external language ($MN_MM_EXTERN_LANGUAGE) M19, M96–M99. Exception: The M functions defined via $MC_TOOL_CHANGE_M_CODE for tool changes. The subroutines configured via $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME must not be activated simultaneously in one block (part program line), in other words, only one M/T function replacement can be programmed in each block. Neither an M98 nor a modal subroutine call may be programmed in the same block as the M function replacement . It is also illegal to program subroutine return jumps and end of part program. Alarm 14016 is generated if these conventions are not observed.	
Related to...		
References:	ISO dialects for Sinumerik (FBFA)	

10716 MD number	**M_NO_FCT_CYCLE_NAME** Name for tool change cycle with M functions from MD $MN_NO_FCT_CYCLE	
Default setting: –	Min. input limit: –	Min. input limit: –
Change valid after POWER ON	Protection level: 2/4	Unit: –
Data type: STRING	Applies to SW 5.2 and higher	

10716 MD number	**M_NO_FCT_CYCLE_NAME** Name for tool change cycle with M functions from MD $MN_NO_FCT_CYCLE
Meaning:	The cycle name is stored in the machine data. This cycle is called if the M function has been programmed from machine data $MN_M_NO_FCT_CYCLE. If the M function is programmed in a motion block, then the cycle is executed after the motion. $MN_M_NO_FCT_CYCLE is effective both in Siemens mode G290 and in external language mode G291. If a T number is programmed in the call block, the programmed T number can be scanned in the cycle under variable $P_TOOL. $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be activated simultaneously in the same block, in other words, only one M/T function replacement can be programmed in each block. Neither an M98 nor a modal subroutine call may be programmed in the same block as the M function replacement. It is also illegal to program a subroutine return jump or end of part program. Alarm 14016 is generated if these conventions are not observed.
Related to...	
References:	ISO dialects for Sinumerik (FBFA)

<table>
<tr><td colspan="2"><b>10717</b>
MD number</td><td colspan="4"><b>T_NO_FCT_CYCLE_NAME</b>
Name for tool change cycle with T number</td></tr>
<tr><td colspan="2">Default setting: –</td><td colspan="2">Min. input limit: –</td><td colspan="2">Min. input limit: –</td></tr>
<tr><td colspan="3">Change valid after POWER ON</td><td colspan="2">Protection level: 2/4</td><td>Unit: –</td></tr>
<tr><td colspan="3">Data type: STRING</td><td colspan="3">Applies to SW 5.2 and higher</td></tr>
<tr><td colspan="2">Meaning:</td><td colspan="4">This machine data defines the name of a tool cycle that is called when T is programmed. If the T number is programmed in a motion block, the cycle is called after the motion.

The programmed T number can be scanned in the cycle with $C_T.
$MN_T_NO_FCT_CYCLE_NAME is effective both in Siemens mode G290 and in external language mode G291.

$MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be activated simultaneously in the same block, in other words, only one M/T function replacement can be programmed in each block.
Neither an M98 nor a modal subroutine call may be programmed in the block with the T function replacement. It is also illegal to program a subroutine return jump or end of part program.
Alarm 14016 is generated if these conventions are not observed.</td></tr>
<tr><td colspan="2">Related to...</td><td colspan="4"></td></tr>
<tr><td colspan="2">References:</td><td colspan="4">ISO dialects for Sinumerik (FBFA)</td></tr>
</table>

8.1.6 Machine data of Siemens user data

The numbers of the Siemens machine data are listed below. These data are defined by Siemens and must not be used by customers. No detailed description of them is given for this reason.

18200

18201

18202

18203

18204

18205

18206

18207

18208

18209

Note

A detailed description of machine data 18091, 18093, 18095, 18097 and 18099 has been provided, but these MD may be used only if they are set to their respective defaults.

Signal Descriptions PLC Interfaces 9

Overview of data blocks

The table below shows an overview of the data blocks used for data management. DB 71 to DB 73 are the tool management interfaces.

DB 71	for load/unload points
DB 72	for spindle as change position
DB 73	for turret as change position
DB 74	internal data block of basic program for tool management

1. The interfaces for load magazines are organized in such a way in DB 71 that a separate interface area is defined for every configured load magazine. The interface area for load point 1 is generally intended for load operations to spindles. It also receives commands for relocating and positioning tools in any location.
2. DB 72 includes an independent interface area for every spindle defined in the tool management system.
3. DB 73 includes an independent interface area for every turret in the magazine configuration. The turret numbers are counted contiguously from the lowest to the highest magazine number.

All interfaces are designed to receive TM commands (load, tool change, etc.). Basic program blocks FC 7 and FC 8 are used to communicate the current positions of tools.

One of the interfaces is updated by the NCK via the basic program after a command (e.g. operator selection of “Load” function or part program “Tool change” function).

Note

If magazine, buffer or load/unload position data are changed in the installation branch, then:

1. Press softkey Create PLC data (MMC103) or change the assignments of DB 4 in the PLC program and
2. delete data blocks DB 71 to DB 74 and restart (cold) the PLC.

9.1 Interface for loading/unloading magazine

DB71 Data block	Load/unload point signals NCK–>PLC interface							
Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	Interfaces							
DBB 0	IFC 8	IFC 7	IFC 6	IFC 5	IFC 4	IFC 3	IFC 2	IFC 1
DBB 1	IFC 16	IFC 15	IFC 14	IFC 13	IFC 12	IFC 11	IFC 10	IFC 9
DBB 2, 3								
DBB n + 0				NC program positions magazine	Position to load point	Relocate	Unload	Load
DBB n + 1	Unassigned							
DBB n + 2	Assigned channel (8 bit int)							
DBB n + 3	Tool management number (8 bit int)							
DBD n + 4	Reserved							
DBD n + 8	Reserved							
DBD n + 12	Reserved							
DBW n + 16	Identifier for load/unload point (int), (fixed value 9999)							
DBW n + 18	Location no. of load/unload point (Int)							
DBW n + 20	Magazine no. (source) for unload/relocate/position (Int)							
DBW n + 22	Location no. (source) for unload/relocate/position (Int)							
DBW n + 24	Magazine no. (target) for load/relocate/position (Int)							
DBW n + 26	Location no. (target) for load/relocate/position (Int)							
DBW n + 28	Spare							

Initial addresses for load/unload points:

Load/unload point 1: n = 4
2: n = 34
3: n = 64
4: n = 94

Example of calculation of address DBW n+24 (magazine no. target)

n = (m–1) * len + 4 m = location no. of load station/point
len = 30 (length of a load point)

m =2 ; len = 30 n = (2–1) * 30 + 4 ==> n = 34

DBW (34 + 24) = DBW 58

Address for magazine no. (target) of 2nd load point is DBW 58.

Load point 1 is provided for load/unload operations to/from all spindles. This must be taken into account when load interface assignments are made (applies only to MMC 100.2; implemented automatically on MMC 103). Load point 1 is also used to relocate/position tools in any location (e.g. buffer location).

DB 71 **DBX 0.0 – 0.15**	**Active status of interfaces 1 – 16**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signals valid from version: **2**
Signal state 1	The active interface has a valid data record. A task bit has been set in DBB (n+0). There are 16 interfaces. Address “n” must always be calculated for the active interface.	
Signal state 0	Operation for this interface has ended. Reset by FC 8.	
References		

DB 71 **DBX(n+0).0**	**Command: Load**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state 1	Load operation for a tool is initiated. The magazine location into which the tool is to be loaded is defined in DBW (n+24) and DBW (n+26). The load point in question is the location number of the load point. It also appears in DBW (n+18).	
Related to...	DB71 DBW(n+16) and (n+18) or (n+24) and (n+26)	
References		

DB 71 **DBX(n+0).1**	**Command: Unload**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state 1	Unload operation for a tool is triggered. The magazine location from which the tool is to be unloaded is defined in DBW (n+20) and DBW (n+22). The number of the unload point is defined in DBW (n+18).	
Related to...	DB71 DBW(n+16) and (n+18) or (n+20) and (n+22)	
References		

DB 71 DBX(n+0).2	**Command: Relocate**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state 1	Relocate operation for a tool is triggered. From magazine/location (n+20, n+22= source) to magazine/location (n+24, n+26= target)	
Related to...		
References		

Note

The bits in DBB (n+0) (load, unload,....) are not updated by the basic program until a new task exists for this interface. They are only up to date if the corresponding interface bit in DBB0 is “1”. The user can reset the bits as required.

DB 71 DBX(n+0).3	**Command: Position at load point**	
Edge evaluation	Signal(s) updated:	Signal(s) valid from version: **3.2**
Signal state 1	A magazine location (magazine no. 9999) is to be positioned at the load point. The magazine location to be moved to the load point is defined in DB71.DBW n+20 and n+22. The load point is stored in DB71.DBW n+18.	
Related to...		

DB71. DBB(n+2)	**Assigned channel**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Number of channel to which active interface applies	
Related to...		

DB71. DBB(n+3)	**Tool management no.**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Associated tool management number; corresponds to the number of the TO unit within a TO area	
Related to...		

DB71. DBW(n+16)	**Code for load/unload point (fixed value 9999)**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	The code for the load/unload point is permanently fixed to 9999.	
Related to...		
References		

DB71. DBW(n+18)	**Location no. of load/unload point**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	The location no. of the load/unload point is displayed.	
Related to...		
References		

DB71. DBW(n+20)	**Magazine no. (source) for unload/relocate/position**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Unload: Magazine from which tool is to be unloaded Relocate: Magazine from which the tool is taken Position: Magazine to be positioned	
Related to...	DBW(n+22)	

DB71. DBW(n+22)	**Location no. (source) for unload/relocate/position**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Unload: Location from which tool is to be unloaded Relocate: Location from which the tool is taken Position: Location to be positioned at load point DBW(n+18)	
Related to...	DBW(n+20)	

DB71. DBW(n+24) MD number	**Magazine no. (target) for load/relocate/position**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Load: Magazine into which tool is to be loaded Relocate: Magazine into which the tool is to be placed Position: Magazine at which tool must be positioned Tool remains at original location Significant only for interface 1. If values other than 0 are entered here, the data define the magazine or location for positioning (language command POSM).	
Related to...	DBW(n +26)	

DB71. DBW(n+26) MD number	**Location no. (target) for load/relocate/position**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Load: Location into which tool is to be loaded Relocate: Location into which tool is to be placed Position: Location at which tool must be positioned Tool remains at original location Significant only for interface 1. If values other than 0 are entered here, the data define the magazine or location for positioning (language command POSM).	
Related to...	DBW(n+24)	

9.2 Interface for spindle as change position

DB72 Data block	Spindle as change position NCK–>PLC interface							
Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DBB 0	IFC 8	IFC 7	IFC 6	IFC 5	IFC 4	IFC 3	IFC 2	IFC 1
DBB 1	IFC 16	IFC 15	IFC 14	IFC 13	IFC 12	IFC 11	IFC 10	IFC 9
DBB 2, 3								
DBB n + 0	Reserved	Unload manual tool	Load manual tool	OldTL in BUF no. (n+42)	T0	Prepare change	Execute change (initiated by: M06)	Obligatory change
DBB n + 1	Unassigned							
DBB n + 2	Assigned channel (8 bit int)							
DBB n + 3	Tool management number (8 bit int)							
DBD n + 4	$P_VDITCP[0] User parameter 0 (DWord)							
DBD n + 8	$P_VDITCP[1] User parameter 1 (DWord)							
DBD n +12	$P_VDITCP[2] User parameter 2 (DWord)							
DBW n + 16	Buffer identifier (int), fixed value 9998) corresponds to “target position for new tool”							
DBW n +18	Relative location (target) in buffer magazine (Int)							
DBW n +20	Magazine no. (source) for new tool (Int)							
DBW n +.22	Location no. (source) for new tool (Int)							
DBW n + 24	Magazine no. (target) for old tool (Int)							
DBW n + 26	Location no. (target) for old tool (Int)							
DBW n + 28	New tool: Location type (Int)							
DBW n + 30	New tool: Size left (Int)							
DBW n + 32	New tool: Size right (Int)							
DBW n + 34	New tool: Size above (Int)							
DBW n + 36	New tool: Size below (Int)							
DBW n + 38	Tool status for new tool							
	Tool was in use	WZ fixed-location-coded		Prewarning limit reached	Tool measured		Tool enabled	Active tool
DBW n + 40	New tool: Internal T no. of NCK (Int)							
DBW n + 42	If DBX (n+0.4) = 1, then buffer location of old tool must be entered here							
DBW n + 44	Spare							

DB72 Data block	Spindle as change position NCK–>PLC interface
DBW n + 46	Spare

Initial addresses of spindle:	Spindle 1: n = 4 Spindle 2: n = 52 Spindle 3: n = 100

n = (m–1)* len + 4	m = location no. of change position
	len = 48

Note

If M06 is programmed on its own, then only user parameters (SW 6 and later), channel, tool management number and the bit for “Execute change” are updated.

DB72. **DBX 0.0 – 0.15**	**Active status of interfaces 1–16**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**	Assigned interface has a valid data record, tool change or tool preparation task has been triggered.	
Signal state **0**	Operation for this interface has ended. Reset by FC 8.	
Related to...		

DB72. **DBX(n+0).0**	**Command code: Obligatory change**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**	The new tool is fixed-location coded	
Signal state **0**		
Related to...	Position of tools involved	
References		

DB72. DBX(n+0).1	**Command code: Execute change with M06**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**	M06 command for tool change has been programmed, tool change can now be performed.	
Signal state **0**		

DB72. DBX(n+0).2	**Command code: Prepare change**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**	Prepare new tool for change operation. If necessary, move location for old tool to spindle.	
Signal state **0**		
Related to...		
References		

DB72. DBX(n+0).3	**Command code: T0**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**	Indicates that T0 has been programmed (no-load spindle traversal)	
Signal state **0**		
Related to...		
References		

DB72. DBX(n+0).4	**Command code: Old tool in BUF no.**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**	The buffer number of the tool to be unloaded is stored in DB72.DBW (n+42)	
Signal state **0**		
Related to...		
References		

DB72. DBX(n+0).5	Command code: Load manual tool	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**	One manual tool must be loaded. The tool to be loaded is displayed on the MMC.	
Signal state **0**		
Related to...		
References		

DB72. DBX(n+0).6	Command code: Unload manual tool	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**	The tool must be unloaded manually.	
Signal state **0**		
Related to...		
References		

Note

The bit in DBB (n+0).2 (prepare change) is **not** reset by the system when a tool change command is output. The bits in DBB(n+0)... are up to date only if the corresponding interface bit in DBB0 is set to “1”. The user can reset the bits as required.
Simultaneous activation of DBX(n+0).1 and DBX(n+0).2 means that T and M06 were programmed in the same block.

DB72. DBB(n+2)	Assigned channel	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Number of channel to which active interface applies	
Related to...		
References		

DB72. DBB(n+3)	**Tool management no.**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Associated tool management number (TO area)	
Related to...		
References		

DB72. DBD(n+4)	**User parameter 0 (D int)**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	A value can be transferred to the PLC via the part program by programming $P_VDITCP[0]=(value);. Values are transferred with a T call.	
Related to...		
References		

DB72. DBD(n+8)	**User parameter 1 (D int)**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	A value can be transferred to the PLC via the part program by programming $P_VDITCP[1]=(value);.	
Related to...		
References		

DB72. DBD(n+12)	**User parameter 2 (D int)**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	A value can be transferred to the PLC via the part program by programming $P_VDITCP[2]=(value);.	
Related to...		
References		

DB72. **DBW(n+16)**	**Buffer magazine no. (fixed value 9998) target position for new tool**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Magazine no. 9998 (buffer magazine); Target magazine for new tool	
Related to...		
References		

DB72. **DBW(n+18)**	**Location in buffer magazine (spindle)**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Location no. of buffer magazine in which new tool must be placed. This is usually the spindle. The location number defined for this buffer during start-up is output.	
Related to...		
References		

DB72. **DBW(n+20)**	**Magazine no. (source) for new tool to be loaded**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Magazine no. from which new spindle tool is fetched	
Related to...	DBW(n+22)	
References		

DB72. **DBW(n+22)**	**Location no. (source) for new tool**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Location no. of magazine from which new spindle tool is fetched	
Related to...	DBW(n+20)	
References		

DB72. DBW(n+24)	**Magazine no. (target) for old tool to be replaced**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Number of magazine in which old tool must be placed.	
Related to	DBW(n+26)	
References		

DB72. DBW(n+26)	**Location no. (target) for old tool**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Magazine location for old tool.	
Related to...		
References		

DB72. DBW(n+28)	**New tool: Location type**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	The location type of the new spindle tool is entered here.	
Related to...	Tool size: left, right, top, bottom	
References		

DB72. DBW(n+30)	**New tool: Size left**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Specification of tool size **on left** in half locations for new spindle tool.	
Related to...		
References		

DB72. DBW(n+32)	**New tool: Size right**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Specification of tool size **on right** in half locations for new spindle tool.	

DB72. DBW(n+32)	**New tool: Size right**
Related to...	
References	

DB72. DBW(n+34)	**New tool: Size top**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Specification of tool size **at top** in half locations for new spindle tool.	
Related to...		
References		

DB72. DBW(n+36)	**New tool: Size bottom**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Specification of tool size **at bottom** in half locations for new spindle tool.	
Related to...		
References		

DB72. DBW(n+38)	**Tool status for new tool**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	**Bit 0**: Active tool **Bit 1**: Tool enabled **Bit 2**: Disabled **Bit 3**: Tool measured **Bit 4**: Prewarning limit reached **Bit 6**: Tool is fixed-location coded **Bit 7**: Tool was in use	
Related to...		
References		

DB72. DBW(n+40)	**New tool: Internal T no. of NCK**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Display of internal NCK T no. for the new spindle tool.	
Related to...		
References		

DB72. DBW(n+42)	**Buffer location of old tool**	
Edge evaluation	Signal(s) updated:	Signal(s) valid from version:
Meaning	If DB72.(n+0.4) = 1, the buffer location of the old tool is entered here. This can be any buffer (including gripper).	
Related to...		
References		

DB72. DBW(n+44)	**Spare**	
Edge evaluation	Signal(s) updated:	Signal(s) valid from version:
References		

DB72. DBW(n+46)	**Spare**	
Edge evaluation	Signal(s) updated:	Signal(s) valid from version:
References		

9.3 Interface for turret as change position

DB73 Data block	Turret as change position NCK–>PLC interface							
Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DBB 0	IFC 8	IFC 7	IFC 6	IFC 5	IFC 4	IFC 3	IFC 2	IFC 1
DBB 1	IFC 16	IFC 15	IFC 14	IFC 13	IFC 12	IFC 11	IFC 10	IFC 9
DBB 2, 3								
DBB n + 0					T0		Execute change triggered by: T no.	Obligatory change
DBB n + 1	Unassigned							
DBB n + 2	Assigned channel (8 bit int)							
DBB n + 3	Tool management number (8 bit int)							
DBD n + 4	$P_VDITCP[0] User parameter 0 (DWord)							
DBD n + 8	$P_VDITCP[1] User parameter 1 (DWord)							
DBD n + 12	$P_VDITCP[2] User parameter 2 (DWord)							
DBW n + 16	Reserved							
DBW n + 18	Reserved							
DBW n + 20	Magazine no. of turret (Int)							
DBW n + 22	Location no. of new tool (Int)							
DBW n + 24	Reserved							
DBW n + 26	Location no. of old tool (Int)							
DBW n + 28	New tool: Location type (Int)							
DBW n + 30	New tool: Size left (Int)							
DBW n + 32	New tool: Size right (Int)							
DBW n + 34	New tool: Size above (Int)							
DBW n + 36	New tool: Size below (Int)							

DB73 Data block	Turret as change position NCK–>PLC interface							
DBW n + 38	Tool status for tool							
	Tool was in use	WZ fixed-loc.-coded		Prewarning limit reached	Tool measured		Tool enabled	Active tool
DBW n + 40	New tool: Internal T no. of NCK (Int)							
DBW n + 42	Reserved							

Initial addresses of turret:
Turret 1: n = 4
Turret 2: n = 48
Turret 3: n = 92

n = (m–1)* len + 4
m = location no. of change position
len = 44

Example for change position 3: n = (3–1)*n 44+ 4= 2*44 + 4= 88 + 4= 92

DB73 – DBX 0.0 – 0.15	**Active status of interfaces 1–16**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**	Assigned interface has a valid data record	
Signal state **0**	Operation for this interface has ended. Reset by FC 7.	
References		

DB73. DBX(n+0).0	**Command code: Obligatory change**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**		
Signal state **0**		
Related to	Position of tools involved	
References		

DB73. DBX(n+0).1	**Command code: Change tool**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Signal state **1**	Perform tool change	
Signal state **0**		
References		

DB73. DBB(n+0).3	**T0**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Indicates that T0 has been programmed	
References		

DB73. DBB(n+2)	**Assigned channel**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Number of channel from which T word was programmed.	
References		

DB73. DBB(n+3)	**Tool management no.**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Associated tool management number (TO area) of channel	
Related to...		
References		

Note

The bits in DBB (n+0) (obligatory change, execute change,...) are **not** reset by the system. They are only up to date if the corresponding interface bit in DBB0 is “1”. The user can reset the bits as required.

DB73. DBD(n+4)	**User parameter 0 (D int)**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	A value can be transferred to the PLC via the part program by programming $P_VDITCP[0]=(value);. Parameters 0–2 are transferred with a T command.	
Related to...		
References		

DB73. DBD(n+8)	**User parameter 1 (D int)**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	A value can be transferred to the PLC via the part program by programming $P_VDITCP[1]=(value);.	
Related to...		
References		

DB73. DBD(n+12)	**User parameter 2 (D int)**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	A value can be transferred to the PLC via the part program by programming $P_VDITCP[2]=(value);.	
Related to...		
References		

DB73. DBW(n+16)	**Reserved**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning		

DB73. DBW(n+18)	**Reserved**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning		

DB73. DBW(n+20)	**Magazine no. of new tool**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Magazine no. of new tool to be used in machining operation.	
Related to...	DBW(n+22)	
References		

DB73. DBW(n+22)	**Location no. of new tool to be loaded**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Location no. of new tool to be used in machining operation.	
Related to...	DBW(n+20)	
References		

DB73. DBW(n+24)	**Reserved**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning		
Related to		
References		

DB73. DBW(n+26)	**Location no. of old tool to be replaced**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Location no. of old tool (used in machining operation until now)	
Related to...		
References		

DB73. DBW(n+28)	**New tool: Location type**
Meaning	The location type of the new tool is entered here.
Related to...	Tool size: left, right, top, bottom
References	

DB73. DBW(n+30)	**New tool: Size left**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Specification of tool size **on left** in half locations for new tool.	
Related to		
References		

DB73. DBW(n+32)	**New tool: Size right**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Specification of tool size **on right** in half locations for new tool.	
Related to...		
References		

DB73. DBW(n+34)	**New tool: Size top**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Specification of tool size **at top** in half locations for new tool.	
Related to...		
References		

DB73. DBW(n+36)	**New tool: Size bottom**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Specification of tool size **at bottom** in half locations for new tool.	
Related to...		
References		

DB73. DBW(n+38)	**Tool status for new tool**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	**Bit 0**: Active tool **Bit 1**: Tool enabled **Bit 2**: **Bit 3**: Tool measured **Bit 4**: Prewarning limit reached **Bit 6**: Tool is fixed-location coded **Bit 7**: Tool was in use	
Related to...		
References		

DB73. DBW(n+40)	**New tool: Internal T no. of NCK**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
Meaning	Display of internal NCK T no. for the new tool. This T no. allows tool management variables to be read or written via FB 2/FB 3.	
Related to...		
References		

DB73. DBW(n+42)	**Reserved**	
Edge evaluation	Signal(s) updated: **Conditionally**	Signal(s) valid from version: **2**
References		

9.4 NC channels interface

The following data blocks are required by the tool management function.

DB21–30 Data block	Signals to/from NC CHANNEL PLC–>NCK interface							
Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DBB 1	Activate program test	PLC action ended	CLC over-ride	CLC stop	**Time monitoring active (TM)**	Synchronized action Off	Enable protection zones	Activate referencing
DBB 29	**Do not disable tool**	**Deactivate wear monitoring**	**Deactivate workpiece counter**	Activate PTP travel	Activate fixed feedrate 4	Activate fixed feedrate 3	Activate fixed feedrate 2	Activate fixed feedrate 1
Cyclical signals from NC channel								
DBB 317	**Tool missing**	PTP travel active						External language mode active
Modification signals for TM functions								
DBB 344					**Last replacement tool in tool group**	**Transition to new replacement tool**	**Tool limit value reached**	**Tool prewarning limit reached**
Transferred TM functions								
DBD 348	**T number for TM prewarning limit (DInt)**							
DBD 352	**T number for tool limit value (DInt)**							
DBD 356	**T number of new replacement tool (DInt)**							
DBD 360	**T number of last replacement tool (DInt)**							

The data relevant to the TM function are printed in bold.

DB21. DBX 1.3	The user can start and stop the tool life monitoring function using PLC signal “Time monitoring active”. The effectiveness of this control is set via MD 20310 bit 17.

DB21. DBX 29.5	Activates/deactivates the workpiece count function.

DB21. DBX 29.6	Activates/deactivates the wear monitoring function.

DB21. DBX 29.7	With **VDI signal “Tool disable inoperative”** (bit value = 1), the NCK does not process tool status “disabled” during tool searches. With **VDI signal “Tool disable operative”** (bit value = 0), the NCK does process tool status “disabled” during tool searches.

DB21. DBX 317.7	Display in PLC indicating that the programmed tool is missing.

DB21.DBX 344.0–344.3	**Modification signals of the tool management functions**	
Edge evaluation	Signal(s) updated: job-controlled by NCK	Signal(s) valid from version: **5.1**
Meaning	One T number for tool prewarning limit, limiting value, new replacement tool, last replacement tool has been output to the interface with a value in conjunction with the associated modification signal at the start of an OB1 cycle. The modification signal indicates that the relevant value is valid.	
Related to...		
References		

Alarms 10

Alarm No.	Brief Description
6402	Tool change not possible, magaze number does not exist
6403	Tool change not possible, specified magazine location does not exist
6404	Tool change not possible, tool not available or missing
6405	Commands has invalid PLC acknowledgement parameter
6406	PLC acknowledge missing
6407	Tool cannot be placed in a magazine location that does not meet the requirements for loading
6410	One cutting edge of monitored tool has reached prewarning limit
6411	One cutting edge of monitored tool has reached prewarning limit
6412	One cutting edge of monitored tool has reached monitoring limit
6413	One cutting edge of monitored tool has reached monitoring limit
6421	No location for tool free in magazine
6422	Tool not moved. Magazine not defined
6423	Tool not moved. Magazine location not available
6424	Tool not moved. Tool not available or missing
6425	Tool not moved. Tool cannot be placed on location in magazine
6430	Workpiece counter: overflow in table of monitored cutting edges
6431	Function not allowed. Tool management/TM monitoring is not active
6432	Function cannot be executed. NO tool assigned to spindle
16924	Tool data are altered during program testing
17001	No memory left for tool magazine data
17160	Tool not selected
17180	Illegal D number
17181	Unknown D no.
17182	Illegal total offset number
17188	The specified D number in the TO unit of the channel is not unique
17189	D number is not unique

Alarm No.	Brief Description
17191	Unknown tool identifier
17192	No further replacement tools possible
17194	No suitable tool found
17202	Cannot delete magazine data
17212	Manual tool must be changed
17214	Remove manual tool from spindle/toolholder
17216	Manual tools must be changed
17220	Tool not available
17230	Duplo no. already disposed
17240	Invalid definition of tool
17250	Invalid definition of magazine
17260	Invalid definition of magazine location
17262	Illegal tool adapter assignment
20150	PLC terminates the interrupted command
20160	PLC can terminate only incorrectly aborted commands
22065	Tool move not possible, since tool not in magazine
22066	Tool change not possible, since tool not in magazine
22067	Tool not changed, since tool group does not contain a tool which is ready for use
22068	Tool group does not include a tool which is ready for use
22069	Tool group does not include a tool which is ready for use
22070	Change tool into magazine. Repeat data backup
22071	Tool has “active” status in an “inactive” wear grouping
400604	Set change with M06 in machine data
410151	Magazine data for tool management missing in PLC

MD 11410 SUPPRESS_ALARM_MASK can be set to suppress particular alarms, as shown in the table below.

Bit	Alarm number
2	16924
4	17189
5	22071
7	22070
8	6411, 6413
9	6410, 6412

10.1 Description of alarms

Alarm No.	
6402	Channel %1 tool not changed. Magazine no. %2 not available
Explanation	%1 = Channel ID, %2 = Magazine number The desired tool change is not possible. The magazine with the specified number is not available.
Reaction	Display of alarm Interface signals are set NC start disable NC stop on alarm
Remedy	– Check whether the magazine data are correctly defined. – Check whether the magazine is connected to the desired spindle via a distance relation
Continue program run	Reset alarm with RESET key, restart the part program.

Alarm No.	
6403	Channel %1 tool not changed. Magazine location %2 on magazine %3 not available
Explanation	%1 = Channel ID, %2 = Magazine number, %3 = Magazine location number The desired tool change is not possible. The specified magazine location is not contained in the specified magazine.
Reaction	Alarm display Interface signals are set NC Start disable NC Stop on alarm
Remedy	– Check whether the magazine data are correctly defined.
Continue program run	Reset alarm with RESET key, restart the part program.

Alarm No.	
6404	Channel %1 tool not changed. Tool %2 not available or missing
Explanation	%1 = Channel ID, %2 = String (identifier) The desired tool change is not possible. The specified tool does not exist or cannot be changed.
Reaction	Alarm display Interface signals are set NC Start disable NC Stop on alarm
Remedy	– Check whether the part program is written correctly – Check whether the tool data are correctly defined – Check whether there is a replacement tool which can be used for the specified tool
Continue program run	Reset alarm with RESET key, restart the part program.

Alarm No.	
6405	Channel %1 command %2 has invalid PLC acknowledgement parameter %3 identifier %4
Explanation	%1 = Channel ID, %2 = Command no., %3 = PLC acknowledgement parameter, %4 = Error identifier The specified command has been answered by a PLC with an invalid acknowledgement in the current combination. The following assignments are defined for “command no.”: 1 Move tool,load or unload magazine 2 Prepare tool change 3 Execute tool change 4 Prepare tool change and execute with T command 5 Prepare tool change and execute with M command 7 End aborted tool command 8 Check “Move tool” with reservation 9 Check “Move tool” 0 Transport acknowledgement The tool change requested by the command cannot be executed. The magazine location in the errored parameter does not exist in the magazine. The error code (%4) explains the alarm in more detail: 0 Not defined 1 Status not legal now or undefined status received from PLC 2 Source and/or target magazine no./location no. unknown 3 Not defined 4 Target magazine no. and/or target location no. in Move Tool command is not final destination 5 Not defined 6 Source and/or target magazine no./location no. for tool change is not known 7 PLC command with inconsistent data: Either magazine addresses in VDI are inconsistent, or NCK command does not match PLC acknowledgement, or both 8 Not defined 9 Not defined 10 Reservation of a buffer location for asynchronous tool motion is not defined.
Reaction	Alarm display Interface signals are set NC Start disable NC Stop on alarm
Remedy	Please inform the authorized personnel / service department Erroneous PLC communication: correct PLC program
Continue program run	Reset alarm with RESET key, restart the part program.

Alarm No.	
6406	Channel %1 PLC acknowledge for command %2 missing
Explanation	%1 = Channel ID, %2 = Command no. There is still no acknowledgement from the PLC for the tool change. The NCK cannot continue processing until it receives this acknowledgement for the specified command number. Possible values are described for alarm 6405
Reaction	Alarm display Interface signals are set NC Start disable
Remedy	Please inform the authorized personnel / service department – Erroneous PLC communication: Correct the PLC program. – IT is possible to release NCK from the wait condition with the PLC command 7. This aborts the waiting command.
Continue program run	Reset alarm with RESET key, restart the part program.

Alarm No.	
6407	Channel %1 Tool %2 cannot be placed in the magazine %3 on location %4. Invalid definition of magazine!
Explanation	%1 = Channel ID, %2 = String (identifier), %3 = Magazine number, %4 = Magazine location number A tool change request or a verification request was issued to put the tool in a location which does not satisfy the prerequisites for filling. The following causes for the error are possible: – Location is blocked or not free – Tool type does not match the location type – Tool possibly too large, adjacent locations are not free
Reaction	Alarm display Interface signals are set NC Start disable NC Stop on alarm
Remedy	– Check whether the magazine data are correctly defined (especially the location type) – Check whether the tool data are correctly defined (especially the location type)
Continue program run	Reset alarm with RESET key, restart the part program.

Alarm No.	
6410	TO unit %1 tool %2 with duplo no. %3 has reached tool prewarning limit
Explanation	%1 = TO unit, %2 = Tool identifier (name), %3 = duplo number This alarm indicates that at least one cutting edge of the tool being monitored for time or workpiece count has reached its prewarning limit. It is activated via the OPI interface (MMC, PLC). Since the channel context is not defined, the TO unit is specified instead.
Reaction	Display of alarm Interface signals are set
Remedy	For information purposes only. User is responsible for further action.
Continue program run	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6411	Channel %1 tool %2 with duplo no. %3 has reached tool prewarning limit
Explanation	%1 = Channel number %2 = Tool identifier (name), %3 = duplo number This alarm indicates that at least one cutting edge of the tool being monitored for time or workpiece count has reached its prewarning limit. Limit is monitored in a channel-related context. The alarm is generated while the NC program is running. The channel context is defined.
Reaction	Display of alarm Interface signals are set
Remedy	For information purposes only. User is responsible for further action.
Continue program run	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6412	TO unit %1 tool %2 with duplo no. %3 has reached tool monitoring limit
Explanation	%1 = TO unit, %2 = Tool identifier (name), %3 = duplo number This alarm indicates that at least one cutting edge of the tool being monitored for time or workpiece count has reached its monitoring limit. It is activated via the OPI interface (MMC, PLC). The channel context is not defined, so the TO unit is specified
Reaction	Display of alarm Interface signals are set
Remedy	For information purposes only. User is responsible for further action.
Continue program run	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6413	Channel %1 tool %2 with duplo no. %3 has reached tool monitoring limit
Explanation	%1 = Channel number, %2 = Tool identifier (name), %3 = duplo number This message informs that at least one cutting edge of the time or quantity monitored tool has reached its monitoring limit. Limit is monitored in a channel-related context. The alarm is generated while the NC program is running. The channel context is defined.
Reaction	Display of alarm Interface signals are set
Remedy	For information purposes only. User is responsible for further action.
Continue program run	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6421	Channel %1 tool not moved. Empty location for tool %2 duplo no. %3 in magazine %4 not available
Explanation	%1 = Channel ID, %2 = String (identifier), %3 = duplo number, %4 = Magazine number The desired tool motion command – triggered from the MMC or PLC – is not possible. The specified magazine location is not contained in the specified magazine. There is no location available for this tool.
Reaction	Display of alarm Interface signals are set NC Start disable
Remedy	– Check whether the magazine data are correctly defined (e.g. the magazine must not be blocked). – Check whether the tool data are correctly defined (e.g. the location type of the tool must match the allowable location types). – Check whether there is still room in the magazine to add another tool; there may not be due to operating procedures. – Check whether a location type hierarchy is defined and whether it, for example, does not allow insertion of a type ‘A’ tool in a free location with type ‘B’.
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6422	Channel %1 tool not moved. Magazine no. %2 not available
Explanation	%1 = Channel ID, %2 = Magazine number The desired tool motion command – triggered from the MMC or PLC – is not possible. The magazine with the specified number is not available.
Reaction	Display of alarm Interface signals are set NC Start disable
Remedy	– Check whether the magazine data are correctly defined. – If the PLC issued the command for motion: check whether the PLC program is correct. – If the MMC issued the command for motion: check whether the MMC command was assigned correct parameters.
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6423	Channel %1 tool not moved. Location %2 on magazine %3 not available!
Explanation	%1 = Channel ID, %2 = Magazine location number, %3 = Magazine number The desired tool motion command – triggered from the MMC or PLC – is not possible. The specified magazine location is not contained in the specified magazine.
Reaction	Display of alarm Interface signals are set NC Start disable
Remedy	– Check whether the magazine data are correctly defined.
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6424	Channel %1 tool not moved. Tool %2 not available or missing
Explanation	%1 = Channel ID, %2 = String (identifier) The desired tool motion command – triggered from the MMC or PLC – is not possible. The specified tool is not defined.
Reaction	Display of alarm Interface signals are set NC Start disable
Remedy	– Check whether the tool data are correctly defined. – Check whether the motion command was correctly parameterized.
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6425	Channel %1 Tool %2 cannot be placed in the magazine %3 on location %4. Invalid definition of magazine
Explanation	%1 = Channel ID, %2 = String (identifier), %3 = Magazine number, %4 = Magazine location number The desired tool motion command – triggered from the MMC or PLC – is not possible. A movement request was issued to put the tool in a location which does not satisfy the prerequisites for filling. The following causes for the error are possible: – Location is blocked or not free – Tool type does not match location type. – Tool possibly too large, adjacent locations are not free.
Reaction	Display of alarm Interface signals are set NC Start disable
Remedy	– Check whether the magazine data are correctly defined. – Check whether there is still room in the magazine to add another tool there may not be due to operating procedures. – Check whether a location type hierarchy is defined and whether it, for example, does not allow insertion of a type ‘A’ tool in a free location with type ‘B’.
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6430	Workpiece counter: overflow in table of monitored cutting edges
Explanation	No more cutting edges can be entered in the piece counter table. As many cutting edges can be noted for the workpiece counter as are possible in total in the NCK. This means that if for each tool each cutting edge in each TO unit is used precisely once for a workpiece then the limit is reached. If several workpieces are made on several spindles simultaneously, it is possible to note cutting 18100 MM_NUM_CUTTING_EDGES_IN_TOA for the totaling counter for all ofthe workpieces. If this alarm occurs, it means that cutting edges used subsequently are no longer quantity monitored until the table has been emptied again, e.g. by means of the NC language command SETPIECE or by the relevant job from MMC, PLC (PI service).
Reaction	Display of alarm Interface signals are set NC Start disable
Remedy	– Was decrementing of the piece counter forgotten? Then program SETPIECE in the part program, or add the correct command in the PLC program. – If the part program/PLC program is correct, then more memory should be set for tool cutting edges via the machine data $MM_NUM_CUTTING_EDGES_IN_TOA (can only be performed with the necessary access rights).
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6431	Function not allowed.. Tool management is not active
Explanation	Occurs when a data management function is called which is not available because ToolMan is deactivated. For example, the language commands GETT, SETPIECE, GETSELT, NEWT, DELT.
Reaction	Display of alarm Interface signals are set Interpreter stop NC Start disable
Remedy	– Please inform the authorized personnel/service department. – Check how the numerical control is to be configured. Is ToolMan or tool monitoring needed but not activated? – Is a part program used that is meant for a numerical control with ToolMan/tool monitoring? either run the part program on a suitable NC or modify the part program. – Activate tool management/monitoring by setting the appropriate machine data. See $MM_TOOL_MANAGEMENT_MASK, $MC_TOOL_MANAGEMENT_MASK. – Check whether the required option is set accordingly.
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
6432	Function not allowed. No tool assigned to spindle
Explanation	An attempt was made to perform an operation that requires a tool to be located on the spindle. This can be the quantity monitoring function, for example.
Reaction	Display of alarm Interface signals are set
Remedy	– Select another function, another spindle, position tool on spindle.
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
16924	Channel %1 Caution: Program test alters tool management data
Explanation	%1 = Channel number Tool data are altered during program testing. You cannot automatically correct the tool data again on termination of program test mode. This alarm prompts you to create a backup of the tool data which must be copied back in when you have finished testing the program.
Reaction	Display of alarm
Remedy	Please inform the authorized personnel/service department. – Save tool data on MMC and reimport data after “ProgtestOff”.
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
17001	Channel % 1 block % 2 No more memory available for tool/magazine data
Explanation	%1 = Channel number, %2= Block number, Label The available memory for defining adapter data has been used up. If the alarm occurs when you are writing one of the $TC_ADPT parameters, you have tried to define more adapter data records than permitted by the setting in MD MM_NUM_TOOL_ADAPTER.
Reaction	Display of alarm Interface signals are set Interpreter stop NC Start disable
Remedy	Please contact the authorized personnel / servicing department. – Modify machine data – Modify the NC program, i.e. reduce the number of variables which caused the error condition
Program continuation	Clear alarm with the RESET key. Restart part program.

Alarm No.	
17160	Channel % 1 Block % 2 no tool selected
Explanation	%1 = Channel number, %2 = Block number, Label An attempt has been made to access the current tool offset data via the system variables: $P_AD[n] Contents of the parameter (n: 1 – 25) $P_TOOL Active D number (tool edge number) $P_TOOLL[n] Active tool length (n: 1 – 3) $P_TOOLR Active tool radius although no tool had been selected previously.

Alarm No.	
Reaction	Display of alarm Interface signals are set Interpreter stop NC Start disable
Remedy	Program or activate a tool offset in the NC program before using the system variables. Example: N100 G.. ... T5 D1 ...LF With the channel-specific machine data: 22550: **TOOL_CHANGE_MODE** New tool offset for M function 22560: **TOOL_CHANGE_M_MODE** M function with tool change It is established whether a tool offset is activated in the block with the T word or whether the new offset values are allowed for only when the M word for tool change occurs.
Program continuation	Clear alarm with the RESET key. Restart part program.

Alarm No.	
17180	Channel %1 Block %2 illegal D number
Explanation	%1 = Channel number %2 = Block number, Label In the displayed block, access is made to a D number (tool edge number) that is not initialized and therefore is not available.
Reaction	Display of alarm Interface signals are set Interpreter stop NC Start disable
Remedy	Check tool call in the NC part program: – Correct tool edge number programmed? If no tool edge number is specified, then D1 is automatically activated. – Tool parameters P1 – P25 defined? The dimensions of the tool edge must have been entered previously either through the operator panel or through the V.24 interface.
Program continuation	Clear alarm with the RESET key. Restart part program.

Alarm No.	
17181	Channel %1 block %2 T no. = % 3, D no. = % 4 does not exist
Explanation	%1 = Channel number, %2 = Block number, Label, %3 = T number, %4 = D number You have programmed a D number which the NCK does not recognize. With a standard application, the D number refers to the specified T number. If the “flat D number” function is activated, T = 1 is output.
Reaction	Display of alarm Interface signals are set Correction block with reorganization
Remedy	In case of a programming error, eliminate the error with a correction block and continue the program run. If the data block is missing, load a block for the relevant T/D values to the NCK (via MMC using Overwrote) and continue the program run.
Program continuation	Cancel alarm with NC Start and continue machining.

Alarm No.	
17182	Channel %1 Block %2 illegal total offset number
Explanation	%1 = Channel number, %2 = Block number, Label An attempt was made to access a non-defined total offset of the current tool edge.
Reaction	Display of alarm Interface signals are set Correction block with reorganization
Remedy	Access the total offset memory with $TC_SCUP*, $TC_CEP*, check the total offset selection DIx or tool selection Ti or offset selection Dz. Cancel alarm with NC Start and continue machining.
Program continuation	Cancel alarm with NC START and continue machining.

Alarm No.	
17188	Channel %1 D number %2 defined for tool T number %3 and %4
Explanation	%1 = Channel number, %2 = Offset number D D, %3 = T number of first tool, %4 = T number of second tool The specified D number %2 in the TO unit of channel %1 is not unique. The specified T numbers %3 and %4 each have an offset with number %2. If tool management is active: The specified T numbers belong to tool groups with different identifiers.
Reaction	Display of alarm Interface signals are set
Remedy	– Ensure that the D numbers within the TO unit are unique – If unique numbering is not necessary for subsequent operations, do not use the command.
Program continuation	Alarm display showing cause of alarm disappears. No further operator action necessary.

Alarm No.	
17189	Channel %1 D number %2 of tools in magazine / location %3 and %4 defined
Explanation	%1 = Channel ID, %2 = D number, %3 = Magazine/location number, '/' as separator, %4 = Magazine/location, '/' as separator Alarm is generated only when TM function is active The specified D number %2 in the TO unit of channel %1 is not unique. The tools in the specified magazine locations %3 and %3 each have an offset with the number %2. If tool management is active: The specified T numbers belong to tool groups with different identifiers.
Reaction	Display of alarm Set interface signals
Remedy	– Ensure that the D numbers within the TO unit are unique, e.g. by renaming the D numbers, – If unique numbering is not necessary for subsequent operations do not use the CHKDM command. The alarm is only a prompt. It can be suppressed by setting bit 4 of MD 11410 SUPPRESS_ALARM_MASK.
Program continuation	Alarm display showing cause of alarm disappears. No further operator action necessary.

Alarm No.	
17191	Channel %1 block %2 T = %3 does not exist, program %4
Explanation	You have programmed a tool identifier that the NCK does not recognize.
Reaction	Display of alarm Interface signals are set Correction block with reorganization
Remedy	%1 = Channel number, %2 = Block number, Label, %3 = T number or T identifier, %4 = Program name If the program pointer is at an NC block which contains the specified T identifier: If the program is incorrect, remedy the error with a correction block and continue the program. If the data block is missing, create a data block, i.e. load the tool data block containing all defined D numbers to the NCK (via MMC using Overwrote) and continue the program run. If the program pointer is indicating an NC block which does not contain the specified T identifier: The error occurred at an earlier point in the program where the T command appeared, but the alarm was not output until the change command was detected. If the program contains an error, e.g. T5 programmed instead of T55, the current block can be corrected with a correction block, i.e. if it contains only M06, then it can be corrected to T55 M06. The errored “T5 line” will remain in the program until this is aborted with RESET or end of program. In complex program structures with indirect programming, it may not be possible to correct the program. In this case, you can only intervene locally with an overwrote block – with T55 in the example. If the data block is missing, create a data block, i.e. load the tool data block containing all defined D numbers to the NCK (via MMC using Overwrote), program T using the Overwrote function and continue the program run.
Program continuation	Cancel alarm with NC START and continue machining.

Alarm No.	
17192	TO unit %1 invalid tool name for %2, duplo no. %3. No further replacement tools possible in %4
Explanation	%1 = TO unit number, %2 = Tool identifier, %3 = duplo number of tool to be renamed, %4 = Group identifier, only possible if TOOLMAN active The tool with the specified tool identifier, duplo number cannot accept the group identifier. Reason: The maximum permissible number of replacement tools has already been defined. The name entered for the tool has assigned or changed the assignment of the tool to a tool group which already includes the maximum permissible number of replacement tools for this particular machine.

Alarm No.	
Reaction	Display of alarm Interface signals are set
Remedy	Use fewer replacement tools. Unload replacement tools that are no longer required and delete their data in the NCK. Request other settings for the maximum number from the machine manufacturer.
Program continuation	Alarm display showing cause of alarm disappears. No further operator action necessary.

Alarm No.	
17194	Channel %1 Block %2 no suitable tool found
Explanation	%1 = Channel number, %2 = Block number, Label – An attempt was made to access a tool which has not been defined – The specified tool does not permit access – A tool with the desired properties is not available
Reaction	Display of alarm Interface signals are set Correction block with reorganization
Remedy	– Check access to tool: – Are the parameters of the command correctly programmed? – Does the status of the tool prevent access?
Program continuation	Cancel alarm with NC START and continue machining.

Alarm No.	
17202	Channel %1 block %2 Magazine data cannot be deleted
Explanation	%1 = Channel number, %2 = Block number, Label You have attempted to delete magazine data at a time when they cannot be deleted. The data for a magazine which currently has the status “Tool is moving” cannot be deleted. A tool adapter currently assigned to a magazine location cannot be deleted. A tool adapter cannot be deleted if machine data $MN_MM_NUM_TOOL_ADAPTER is set to –1.
Reaction	Display of alarm Interface signals are set Correction block with reorganization
Remedy	If your attempt to delete a magazine is rejected: Make sure that the relevant magazine does not have the “Tool is moving” status when you enter the Delete command. If your attempt to delete a tool adapter is rejected: Make sure that its assignment to the magazine location(s) is canceled in the appropriate MD beforehand.
Program continuation	Cancel alarm with NC START and continue machining.

Alarm No.	
17212	Channel %1 Tool management: Load manual tool %3, duplo no. %2 in spindle/ toolholder %4
Explanation	%1 = Channel number, %2 = duplo number, %3 = Tool identifier, %4 = Toolholder (spindle)number Indication that the specified manual tool must be brought to the specified toolholder or spindle before the program is continued. A manual tool is a tool whose data are registered in the NCK, but which is not assigned to a magazine location. As a result, it is not fully accessible for the purpose of automatic tool changes by the NCK or other operations on the machine.
Reaction	Display of alarm
Remedy	– Make sure that the specified manual tool is placed in the named toolholder. The alarm is automatically reset once the PLC has acknowledged the Tool Change Load command.
Program continuation	Alarm display showing cause of alarm disappears. No further operator action necessary.

Alarm No.	
17214	Channel %1 Tool management: Remove manual tool %3 from spindle/toolholder %2.
Explanation	%1 = Channel number, %2 = Toolholder (spindle) number %3 = Tool identifier, This alarm prompts you to remove the specified manual tool from the named toolholder or spindle before you continue the program. A manual tool is a tool whose data are registered in the NCK, but which is not assigned to a magazine location. As a result, it is not fully accessible for the purpose of automatic tool changes by the NCK or other operations on the machine.
Reaction	Display of alarm
Remedy	– Ensure that the specified manual tool is taken from the toolholder. The alarm is automatically reset once the PLC has acknowledged the Tool Change Load command. Manual tools can be used efficiently only if their application is appropriately supported by the PLC program.
Program continuation	Alarm display showing cause of alarm disappears. No further operator action necessary.

Alarm No.	
17216	Channel %1 Tool management: Remove manual tool from spindlel toolholder %4 and load manual tool %3, duplo no. %2.
Explanation	%1 = Channel number, %2 = duplo number, %3 = Tool identifier, %4 = Toolholder (spindle) number Indicates that the specified manual tool must be brought to the specified toolholder or spindle and the manual tool located there removed before the program is continued. A manual tool is a tool whose data are registered in the NCK, but which is not assigned to a magazine location. As a result, it is not fully accessible for the purpose of automatic tool changes by the NCK or other operations on the machine.
Reaction	Display of alarm
Remedy	– Make sure that the specified manual tools are exchanged. The alarm is automatically reset once the PLC has acknowledged the Tool Change Load command. Manual tools can be used efficiently only if their application is appropriately supported by the PLC program.
Program continuation	Alarm display showing cause of alarm disappears. No further operator action necessary.

Alarm No.	
17220	Channel %1 Block %2 tool not available
Explanation	%1 = Channel number, %2 = Block number, Label An attempt was made to access a tool via a T no. that has not (yet) been defined. For example, when tools are to be put into magazine locations by programming $TC_MPP6 = ‘toolNo’. This is possible only when both the magazine location and the tool given by ‘toolNo’ have been defined.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct the NC program
Program continuation	Clear alarm with the RESET key. Restart part program.

Alarm No.	
17230	Channel %1 Block %2 duplo no. already allocated
Explanation	%1 = Channel number, %2 = Block number, Label An attempt was made to write a tool duplo number to the name for which another tool (another T number) already exists with the same duplo number.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct the NC program
Program continuation	Clear alarm with the RESET key. Restart part program.

Alarm No.	
17240	Channel %1 Block %2 invalid definition of tool
Explanation	%1 = Channel number, %2 = Block number, Label An attempt was made to modify a tool data that would subsequently damage the data consistency or lead to a conflicting definition.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct the NC program
Program continuation	Clear alarm with the RESET key. Restart part program.

Alarm No.	
17250	Channel %1 Block %2 invalid definition of magazine
Explanation	%1 = Channel number, %2 = Block number, Label An attempt was made to modify a magazine data that would subsequently damage the data consistency or lead to a conflicting definition.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct the NC program
Program continuation	Clear alarm with the RESET key. Restart part program.

Alarm No.	
17260	Channel %1 Block %2 invalid definition of magazine location
Explanation	%1 = Channel number, %2 = Block number, Label An attempt was made to modify a magazine data that would subsequently damage the data consistency or lead to a conflicting definition.
Reaction	Alarm display Interface signals are set Interpreter stop NC Start disable
Remedy	– Correct the NC program
Program continuation	Clear alarm with the RESET key. Restart part program.

Alarm No.	
17262	Channel %1 block %2 Illegal tool adapter operation
Explanation	%1 = Channel number, %2 = Block number, Label This alarm is generated if you attempt to define or cancel the assignment between a tool adapter and a magazine location and the selected location already has another tool adapter and/or is already holding a tool or, if you are canceling the assignment, there is still another tool in the location. If machine data $MC_MM_NUM_SUMCORR is set to –1, then it is not possible to create adapters by a write operation to an adapter which is not yet defined. When the machine data is set in this way, then it is only possible to write adapter data that are already assigned (automatically) to magazine locations.
Reaction	Display of alarm Interface signals are set Correction block with reorganization
Remedy	– Do not assign more than one adapter to each magazine location – The magazine location must be empty – Machine data $MC_MM_NUM_SUMCORR with setting –1: If the alarm is generated when you are typing one of the system parameters $TC_ADAPTx (x=1,2,3,T), then you must change the write operation to ensure that it includes only adapter data which are already assigned to magazine locations.
Program continuation	Cancel alarm with NC Start. Restart part program.

Alarm No.	
20150	Channel %1 Tool management: PLC terminates interrupted command
Explanation	%1 = Channel number Indication that the PLC has terminated an interrupted command (with alarm output) from the tool management – tool change.
Reaction	Alarm display Interface signals are set
Remedy	For information only
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
20160	Channel %1 Tool management: PLC can only terminate commands interrupted due to an error
Explanation	%1 = Channel number Indication that the PLC wanted to interrupt an active command from the tool management (tool change); or that there is no command active for abort. NCK refuses because the channel status is either ‘active’ (abort is then not allowed), or ‘reset’ (then there is nothing to abort).
Reaction	Alarm display Interface signals are set
Remedy	For information only
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
22066	Channel %1 tool management: tool move not possible since there is no tool %2 with duplo no. %3 in magazine %4
Explanation	%1 = Channel number, %2 = String (identifier), %3 = duplo number, %4 = Magazine number The desired tool change is not possible. The specified tool is not contained in the specified magazine. (NCK can contain tools that are not assigned to a magazine. Such tools cannot be used to perform operations (movements, change)).
Reaction	NC Start disable Alarm display Interface signals are set NC Stop on alarm

Alarm No.	
Remedy	– Please contact the authorized personnel / servicing department. – Check that the specified tool is contained in the desired magazine, or program another tool to be changed – Check whether machine data $MC_RESET_MODE_MASK, $MC_START_MODE_MASK, $MC_TOOL_RESET_NAME match the current definition data
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
22067	Channel %1 tool management: Tool change not possible, since no ready in tool group %2
Explanation	%1 = Channel number, %2 = String (identifier) The desired tool change is not possible. The specified tool group does not contain a “ready to use” replacement tool which could be loaded. It may be that all of the tools in question have been set to the ‘Disabled’ state by the tool monitoring function.
Reaction	NC Start disable Alarm display Interface signals are set NC Stop on alarm
Remedy	– Ensure that the named tool group contains a tool that is ready for use when tool change is requested: – This can be achieved, for example, by replacing disabled tools, or – by releasing a disabled tool manually – Check whether the tool data are correctly defined. Have all intended tools in the group with the specified identifier been loaded?
Program continuation	Clear alarm with the RESET key. Restart part program.

Alarm No.	
22068	Channel %1 Tool management: Tool group %3 does not contain a ready tool
Explanation	%1 = Channel number, %2 = Block number, Label, %3 = String (identifier) The specified tool group does not contain a “ready to use” replacement tool which could be loaded. The tool monitoring function may have set all potentially suitable tools to the “disabled” status. The alarm can occur in conjunction with alarm 14710. In this specific situation, NCK attempts to replace the disabled tool located on the spindle with an available replacement tool (which does not exist in this error condition) (which does not exist in this error condition). The user must resolve this conflict, for example, by removing the tool locatedon the spindle from the spindle by issuing a movement command (e.g. through MMC operation).
Reaction	NC Start disable Alarm display Interface signals are set
Remedy	– Make sure that the named tool group contains a ready tool at the moment the tool change is requested. – This can be achieved, for example, by replacing disabled tools, or – by releasing a disabled tool manually. – Check whether the tool data are correctly defined. Have all the available tools in the group with the specified identifier already been loaded?
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
22069	Channel %1 block %1 Tool management: Tool group %3, program %4 does not contain a ready tool
Explanation	%1 = Channel number, %2 = Block number, Label, %3 = String (identifier), %4 = Program name The specified tool group does not contain a tool which is ready for use and could be used for tool change.The tool monitoring function may have set all potentially suitable tools to the “disabled” status. Parameter %4 = program name helps to identify the program that contains the command causing the problem (tool selection). This can be a subroutine, cycle, etc., which cannot be identified from the display.
Reaction	Display of alarm Interface signals are set Correction block with reorganization

Alarm No.	
Remedy	– Make sure that the named tool group contains a ready tool at the moment the tool change is requested, for example, by taking one of the following measures: – Replace disabled tools, – Manually enable a disabled tool. – Check whether the tool data are correctly defined. Have all the available tools in the group with the specified identifier already been loaded?
Program continuation	Cancel alarm with NC START and continue machining

Alarm No.	
22070	TO unit %1 Please load tool T=%2 into magazine. Repeat data back-up.
Explanation	%1 = TO unit, %2 = T number of tool Alarm is generated only when TM function is active You have started to make a backup of the tool/machine data. The system has detected that the buffer magazine still contains one or more tools. When a data backup is made, these tools will lose their assignment information, i.e. the magazine and location assignment data. For this reason, it is advisable to move all tools from the buffer to the magazine before commencing with the data back-up. If the above scenario does not apply, you have re-imported data with magazine locations set to the “reserved” status. You may have to reset this status manually. In the case of tools with a fixed-location coding, the loss of information about their location in the magazine is equivalent to a general empty location search on any subsequent change back to the magazine.
Reaction	Interface signals are set Display of alarm
Remedy	Make sure that there are no tools stored in the buffer magazine before you start to back up data. Repeat data backup operation after you have removed the tools from the buffer magazine.
Program continuation	Clear the alarm with Delete key. No further operator action necessary.

Alarm No.	
22071	TO unit %1 tool %2 duplo no. %3 is active, but not in the current wear grouping
Explanation	%1 = TO unit, %2 = T number of tool, %3 = duplo number The “Wear grouping” function is active. The setting “Set tool to active status” which applies when a new wear grouping is activated is also selected. This setting can also be programmed with language command SETTA or started via Analog Functions on the OPI. It has been detected that more than one tool from the tool group has the “active” status. The tool which has the “active” status in an “inactive” wear grouping is named in the alarm. The alarm is only a prompt. It can be suppressed by setting bit 5 of MD 11410 SUPPRESS_ALARM_MASK.
Reaction	Display of alarm Set interface signals
Remedy	Before you start the machining operation, make sure that the “active” status is not set for any of the tools in the magazine. You can do this by programming command SETTIA.
Program continuation	Clear the alarm with Delete key. No further operator action necessary.
Alarm No.	

Alarm No.	
400604	Set change with M06 in machine data
Explanation	Change is possible only with M06 for the magazine type used (box, chain). Check for invalid settings when using turret magazines.
Reaction	Display of alarm PLC Stop
Remedy	The value in 1 in channel-specific machine data 22550 TOOL_CHANGE_MODE

Alarm No.	
410151	No magazine data for tool management in PLC
Explanation	No magazine data available in the PLC. Start-up not complete although TOOLMAN option is active.
Reaction	Display of alarm
Remedy	Press the “Create PLC data” softkey via MMC 103 during start-up of the tool management. Set the data in data block DB4 starting at DBB64 for MMC 100.

PLC Sample Programs 11

This section gives practical examples which illustrate how function blocks can be adapted to suit a variety of realistic configurations.

These sample programs are stored in file wzv_bsp.exe in catalog Bsp_prog in the SINUMERIK 810D/840D Toolbox.

11.1 FB 90: QUIT_WZV Acknowledgements to TM

Description of functions

FB QUIT_WZV supports the user in acknowledging the tool management tasks, communicating position changes of the tools to the tool management and updating the tool change position of the PLC when using the integrated tool management.

32 user interfaces are available for transfer tasks to the tool management in the instance DB FB QUIT_WZV. Data are transferred to the NCK in FB QUIT_WZV using call FC TM_TRANS (FC 8). The parameters of FC TM_TRANS are defined as a variable in FB QUIT_WZV and must be assigned a value for each user interface. The symbolic names of the variables have the same names as the formal parameters of FC TM_TRANS. See the Description of Function basic PLC program in Section 4 of the Block description FC 8 for more information about the parameters.

The following variables must be assigned values in the branch target list of each user interface:

– TaskIdent
– TaskIdentNo
– NewToolMag
– NewToolLoc
– OldToolMag
– OldToolLoc
– Status

If the tool is transferred from the magazine to the spindle via a buffer (e.g. gripper), then the following variables

– NewToolMag_Change_S1
– NewToolLoc_Change_S1
– OldToolMag_Change_S1
– OldToolLoc_Change_S1

must also be assigned values on transfer to spindle 1.

For spindle 2, these variables have the same name with the ending '_S2'. If the tool change operation is reset or aborted, these variables must be used to assign the FC TM_TRANS parameters.

With a 1 signal at a user interface (DIB 0 – DIB 3), FC TM_TRANS is called with the parameter values programmed in the branch target list.

If the task is completed successfully (FC TM_TRANS Ready = 1), the user interface bit is reset by FB QUIT_WZV. If the task or transfer of FC TM_TRANS produces an error, error bit DIX 4.0 in the instance DB is set to 1 signal and the output parameter error of FC TM_TRANS is available as error number in DIW 6.

The user interface is reset in the event of an error. Further tasks are only processed after the error bit has been reset (by the user). For the meaning of the error numbers, please refer to Description of Functions, Basic PLC Program, Section 4, Description of Block FC 8 under 'Error'.

If several tasks are present simultaneously, the user interfaces are processed according to the following priority:

1. UI 25 →	UI 32
2. UI 17 →	UI 24
3. UI 9 →	UI 16
4. UI 1 →	UI 8

You must enter the actual magazine position of the tool change point in accordance with the selected FB–QUIT in the instance DB starting at DIW 10.

Declaration

FUNCTION_BLOCK FB 90
// no parameters
// user interface in the instance DB

Block call

CALL FB 90, DB xxx; // xxx no. instance DB

User interface

The user interface is stored in the instance DB from DIB 0 to DIB 46.
Bytes 47 to 64 are internal variables of FB QUIT_WZV, which can be output for support during installation. The variables ASS_alt (UI_old), ASS_Aenderung (UI_change) and ASS_aktiv (UI_active) have the same assignment as ASS_neu (UI_new) (DBB 0 to DBB 3).

DB instance	**User interface**							
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIB 0	UI 8	UI 7	UI 6	UI 5	UI 4	UI 3	UI 2	UI 1
DIB 1	UI 16	UI 15	UI 14	UI 13	UI 12	UI 11	UI 10	UI 9

DB instance	User interface							
DIB 2	UI 24	UI 23	UI 22	UI 21	UI 20	UI 19	UI 18	UI 17
DIB 3	UI 32	UI 31	UI 30	UI 29	UI 28	UI 27	UI 26	UI 25
DIB 4								Error
DIB 5	–							
DIB 6	Error number							
DIB 8	–							
DIB 10	ActPosChangePosMag1							
DIB 12	ActPosChangePosMag2							
DIB 14	ActPosChangePosGr1							
DIB 16	ActPosChangePosGr1							
DIB 18	–							
DIB 20	TaskIdent							
DIB 21	TaskIdentNo							

DB instance	User interface
DIW 22	NewToolMag
DWI 24	NewToolLoc
DWI 26	OldToolLoc
DWI 28	Status
DWI 30	NewToolMag_Change_S1
DWI 32	NewToolLoc_Change_S1
DWI 34	OldToolMag_Change_S1
DWI 36	OldToolMag_Change_S1
DWI 38	OldToolLoc_Change_S1
DWI 40	NewToolMag_Change_S2
DWI 42	NewToolLoc_Change_S2
DWI 44	OldToolMag_Change_S2
DWI 46	OldToolLoc_Change_S2
DID 48	UI_old
DID 52	UI_change
DID 56	UI_active

DB instance	User interface							
DIB 60							Ready	Start
DIB 61	–							
DIW 62	Error FC 8							
DIB 64	StepNo							

Abort/reset

If a task in progress such as “Load tool”, “Unload tool”, “Prepare change” or “Execute change” is aborted by the NC Reset or Emergency Stop signal, the PLC must acknowledge the task with FC TM_TRANS, status 3, if the task has not been completed. A task acknowledgement with status 3 is acknowledged negatively by the tool management with error no. 6405. This behavior is taken into account in FB QUIT_WZV in the error evaluation of FC TM_TRANS. No error is output here.

Configuration / start-up

When supplying parameters for the FC TM_TRANS, it is important to ensure that the correct magazine locations are assigned for the parameters NewToolMag/Loc and OldToolMag/Loc on each status change or end-of-job acknowledgement. The same applies to TaskIdent and TaskIdentNo. The tool management checks each parameter against FC TM_TRANS on acknowledgement. If a wrong value is detected by the tool management, the NC assumes the STOP state and NC error 6405 “Channel %1 command %2 has invalid PLC acknowledgement parameter %3” appears. If such a fault condition occurs, the variables in the parameters of FC TM_TRANS can be output and check in PLC status.

The status of the variables ASS_aktiv (DIB 44 DIB 47) shows which was the last task to be processed. The assignment of ASS_aktiv is identical to the ASS interface (DIB 0 – DIB 3).

Power Off / Restart

If the NCK power is switched off or NCK reset is performed while a task is being executed, the user must reset the user interface bits.
In addition, the following variables in the DB instance must be deleted in OB 100:

```
AUF DB xxx;   // Open DB instance FB QUIT_WZV
L 0;
T DBD 48;     // UI_old
T DBD 52;     // UI_change
T DBD 56;     // UI_active
```

T DBB 60; // Start and ready FC 8

As of version 2.0 of FB 90, the variables in FB 90 are deleted on restart. The instruction section can be omitted in OB100.

11.1.1 Sample programs

Sample programs

As an example of the use of FB QUIT_WZV, five different magazine configurations are programmed in FB 90. The setting for the user interface bits in FB 90 is programmed in FC 90. The blocks are contained in files QUIT_1.awl – QUIT_2.awl.

The following magazine types have been implemented as program examples:

- Chain magazine with one spindle as a pick-up magazine
- Chain magazine with one dual gripper and one spindle
- Chain magazine with two grippers and one spindle
- Two chain magazines with one spindle
- Chain magazine with two spindles

11.1.2 Chain magazine with one spindle as a pick-up magazine

Description

FB QUIT_WZV is programmed as FB 90 in QUIT_1.awl for the following magazine configuration:

Magazine no.	Location no.	Meaning
9999	1	Load point spindle
9999	2	Load point magazine
9998	1	Spindle
1	1	Magazine location 1
1	2	Magazine location 2
1	.	Magazine location .

Magazine no.	Location no.	Meaning
1	n–1	Magazine location n–1
1	n	Magazine location n

The tool is moved directly from the magazine to the spindle (pick-up magazine). If a tool is already located in the spindle it is returned to the magazine before the new tool is placed in the magazine. Loading is performed either via the loading point magazine or the loading point spindle.
For this configuration, 17 transfer job requests from the PLC to the tool management are programmed in FB QUIT_WZV. These requests are triggered by the user via the user interfaces UI 1–UI 20.
With an asynchronous job request, tool position changes outside a programmed sequence, e.g. for movements in JOG, can be sent to the tool management after a tool change has been aborted.

The following job requests are implemented in FB 90 and triggered in FC 90 in the example in QUIT_1.awl:

UI	Function
1	Acknowledgement load tool completed, load point magazine
2	Abort / Reset load tool, load point magazine
3	Acknowledgement unload tool completed, load point magazine
4	Abort / Reset unload tool, load point magazine
5	Acknowledgement load tool completed, load point spindle
6	Abort / Reset load tool, load point spindle
7	Acknowledgement unload tool completed, load point spindle
8	Abort / Reset unload tool, load point spindle
9	Acknowledgement prepare change completed
10	Abort / Reset prepare change
11	Status change spindle → magazine tool change
12	Status change magazine → spindle tool change
13	Abort / Reset change
14	–
15	Acknowledgement relocate (from MMC)
16	Asynchronous relocation spindle → magazine
17	Asynchronous relocation magazine → spindle
18	–
19	–
20	Actual position change magazine location change position
21	–

UI	Function
22	Acknowledgement positioning at loading point
23	Abort / Reset positioning at loading point

... ...

32	–

The actual position for job requests 16, 17 and 20 is taken from DB instance DBW 10 in FB 90. The actual position address can be changed by the user.

Caution:

With asynchronous relocation the magazine location state “Z” (reserved for tool in buffer) is not taken into account. This means that with asynchronous relocation from magazine to spindle the identifier “Z” is not set and with asynchronous relocation from spindle to magazine the identifier “Z” is not reset in the old location.

In this case, “Z” must be set and cleared with FB 3 (write NC variable).
With NC SW 3.2 and later, magazine location status “Z” is transferred with Task–Ident 5 for asynchronous relocation.

11.1.3 Chain magazine with one dual gripper and one spindle

Description

FB QUIT_WZV is programmed as FB 90 in QUIT_2.awl for the following magazine configuration:

Magazine no.	Location no.	Meaning
9999	1	Load point spindle
9999	2	Load point magazine
9998	1	Spindle
9998	1	Dual gripper, gripper 1
9998	1	Dual gripper, gripper 2
1	1	Magazine location 1
1	2	Magazine location 2

Magazine no.	Location no.	Meaning
1	.	Magazine location .
1	n	Magazine location n

The tool is moved via the dual gripper from the tool change position in the magazine to the spindle. The tools are moved to and from the magazine and spindle simultaneously. Before the tool is changed the gripper on the magazine side is gripper 2 and the gripper on the spindle side is gripper 1.
With this definition only two relocation commands are necessary.

Loading is performed either via the loading point magazine or the loading point spindle.
For this configuration, 19 transfer job requests from the PLC to the tool management are programmed in FB QUIT_WZV. These requests are triggered by the user via the user interfaces UI 1 – UI 20.
With an asynchronous job request, tool position changes outside a programmed sequence, e.g. for movements in JOG, can be sent to the tool management after a tool change has been aborted.

The following job requests are implemented in FB 90 and triggered in FC 90 in the example in QUIT_2.awl:

UI	Function
1	Acknowledgement load tool completed, load point magazine
2	Abort / Reset load tool, load point magazine
3	Acknowledgement unload tool completed, load point magazine
4	Abort / Reset unload tool, load point magazine
5	Acknowledgement load tool completed, load point spindle
6	Abort / Reset load tool, load point spindle
7	Acknowledgement unload tool completed, load point spindle
8	Abort / Reset unload tool, load point spindle
9	Acknowledgement prepare change completed
10	Abort / Reset prepare change
11	Status change spindle → gripper 1 and magazine → gripper 2 tool change
12	Status change magazine → magazine and gripper 2 → spindle tool change
13	Abort / Reset change
14	–
15	Acknowledgement relocate (from MMC)
16	Asynchronous relocation gripper 1 → spindle
17	Asynchronous relocation gripper 1 → magazine
18	Asynchronous relocation gripper 2 → spindle

UI	Function
19	Asynchronous relocation gripper 2 → magazine
20	Actual position change magazine location change position
21	–
22	Acknowledgement positioning at loading point
23	Abort / Reset positioning at loading point

... ...

32	–

The actual position for job requests 17, 19 and 20 is taken from DB instance. DIW 10 in FB 90. The actual position address can be changed by the user.

Caution:
With asynchronous relocation the magazine location state “Z” (reserved for tool in buffer) is not taken into account. This means that with asynchronous relocation from magazine to spindle the identifier “Z” is not set and with asynchronous relocation from spindle to magazine the identifier “Z” is not reset in the old location. In this case, “Z” must be set and cleared with FB 3 (write NC variable). With NC SW 3.2 and later, magazine location status “Z” is transferred with Task-Ident 5 for asynchronous relocation.

11.1.4 Chain magazine with two grippers and one spindle

Description

FB QUIT_WZV is programmed as FB 90 in QUIT_3.awl for the following magazine configuration:

Magazine no.	Location no.	Meaning
9999	1	Load point spindle
9999	2	Load point magazine
9998	1	Spindle
9998	2	Gripper 1
9998	3	Gripper 2
1	1	Magazine location 1
1	2	Magazine location 2
1	.	Magazine location .

Magazine no.	Location no.	Meaning
1	n	Magazine location n–1
1	n	Magazine location n

The tool is relocated from the tool change position in the magazine into the spindle via gripper 1 or gripper 2 and from the spindle into the magazine via gripper 2. Tools can only be loaded via the loading point of the magazine.

For this configuration, 20 transfer job requests from the PLC to the tool management are programmed in FB QUIT_WZV.
These tasks must be initiated by the user via the user interfaces UI 1 – UI 20. With an asynchronous job request, tool position changes outside a programmed sequence, e.g. for movements in JOG, can be sent to the tool management after a tool change has been aborted.

The following job requests are implemented in FB 90 and triggered in FC 90 in the example in QUIT_3.awl:

UI	Function
1	Acknowledgement load tool completed, load point magazine
2	Abort / Reset load tool, load point magazine
3	Acknowledgement unload tool completed, load point magazine
4	Abort / Reset unload tool, load point magazine
5	Acknowledgement prepare change completed
6	Abort / Reset prepare change
7	Status change magazine → gripper 1 tool change
8	Status change magazine → gripper 2 tool change
9	Status change spindle → gripper 2 tool change
10	Status change gripper 1 → spindle tool change
11	Status change gripper 2 → magazine tool change
12	Abort / Reset change
13	Acknowledgement relocate (from MMC)
14	Asynchronous relocation gripper 1 → magazine
15	Asynchronous relocation gripper 2 → magazine
16	Asynchronous relocation gripper 1 → spindle
17	Asynchronous relocation gripper 2 → spindle
18	Asynchronous relocation spindle → gripper 1
19	Asynchronous relocation spindle → gripper 2
20	Actual position change magazine location change position

UI	Function
21	–
22	Acknowledgement positioning at loading point
23	Abort / Reset positioning at loading point

... ...

32	–

The actual position is read from the DB instance DIW 10 for job 20 in FB 90.
The actual magazine position for gripper 1 is read from the DB instance DIW14 (UI 14) and the actual magazine position for gripper 2 from DB instance DIW16 (UI 15). The addresses of the actual positions can be changed by the user.
Loading and unloading of spindles is not programmed in FB QUIT_WZV. The user can program this function himself using a free user interface. Jump target lists IFC 1 – IFC 3 can be used as an example.

Caution:
With asynchronous relocation the magazine location state “Z” (reserved for tool in buffer) is not taken into account. This means that with asynchronous relocation from magazine to spindle the identifier “Z” is not set and with asynchronous relocation from spindle to magazine the identifier “Z” is not reset in the old location. In this case, “Z” must be set and cleared with FB 3 (write NC variable).
With NC SW 3.2 and later, magazine location status “Z” is transferred with Task-Ident 5 for asynchronous relocation.

11.1.5 Two chain magazines with one spindle as a pick-up magazine

Description

FB QUIT_WZV is programmed as FB 90 in QUIT_4.awl for the following magazine configuration:

Magazine no.	Location no.	Meaning
9999	1	Load point spindle
9999	2	Load point magazine
9998	1	Spindle
1	1	Magazine location 1

Magazine no.	Location no.	Meaning
1	2	Magazine location 2
1	.	Magazine location .
1	n–1	Magazine location n–1
1	n	Magazine location n
1	1	Magazine location 1
1	2	Magazine location 2
1	.	Magazine location .
1	n	Magazine location n–1
1	n	Magazine location n

The tool is moved directly from magazine 1 or magazine 2 to the spindle (pick-up magazine). If a tool is already located in the spindle it is returned to magazine 1 or magazine 2 before the new tool is placed in the magazine.
Loading is performed either via the loading point magazine or the loading point spindle. For this configuration, 22 transfer job requests from the PLC to the tool management are programmed in FB QUIT_WZV. These requests are triggered by the user via the user interfaces UI 1 – UI 22.
With an asynchronous job request, tool position changes outside a programmed sequence, e.g. for movements in JOG, can be sent to the tool management after a tool change has been aborted.

The following job requests are implemented in FB 90 and triggered in FC 90 in the example in QUIT_4.awl:

UI	Function
1	Acknowledgement load tool completed, load point magazine
2	Abort / Reset load tool, load point magazine
3	Acknowledgement unload tool completed, load point magazine
4	Abort / Reset unload tool, load point magazine
5	Acknowledgement load tool completed, load point spindle
6	Abort / Reset load tool, load point spindle
7	Acknowledgement unload tool completed, load point spindle
8	Abort / Reset unload tool, load point spindle
9	Acknowledgement prepare change completed
10	Abort / Reset prepare change
11	Status change spindle → magazine tool change
12	Status change magazine → spindle tool change
13	Abort / Reset change

UI	Function
14	–
15	Acknowledgement relocate (from MMC)
16	Asynchronous relocation spindle → magazine 1
17	Asynchronous relocation magazine 1 → spindle
18	Asynchronous relocation spindle → magazine 2
19	Asynchronous relocation magazine 2 → spindle
20	Actual position change magazine location change position magazine 1
21	Actual position change magazine location change position magazine 2
22	Acknowledgement positioning at loading point
23	Abort / Reset positioning at loading point

... ...

32	–

The actual position is read from the DB instance DIW10 or DIW12 for job 20 in FB 90. The addresses of the actual positions can be changed by the user.

Caution:
With asynchronous relocation the magazine location state “Z” (reserved for tool in buffer) is not taken into account. This means that with asynchronous relocation from magazine to spindle the identifier “Z” is not set and with asynchronous relocation from spindle to magazine the identifier “Z” is not reset in the old location. In this case, “Z” must be set and cleared with FB 3 (write NC variable).
With NC SW 3.2 and later, magazine location status “Z” is transferred with Task–Ident 5 for asynchronous relocation.

11.1.6 Chain magazine with two spindles

Description

FB QUIT_WZV is programmed as FB 90 in QUIT_5.awl for the following magazine configuration:

Magazine no.	Location no.	Meaning
9999	1	Load point for spindle1
9999	2	Load point for spindle 2

Magazine no.	Location no.	Meaning
9999	2	Load point magazine
9998	1	Spindle 1
9998	2	Spindle 2
1	1	Magazine location 1
1	2	Magazine location 2
1	.	Magazine location .
1	n	Magazine location n–1
1	n	Magazine location n

The tool is moved directly from the magazine to spindle 1 or spindle 2 (pick-up magazine). If a tool is already located in the spindle it is returned to the magazine before the new tool is placed in the magazine.

Spindle 1 is assigned to channel 1 and spindle 2 to channel 2. Therefore a tool function or tool change programmed in channel 1 is output in DB 72 in UI 1 and the new tool is placed on spindle 1.
Therefore, a tool call or tool change programmed in channel 2 is output in DB 72 in UI 2 and the new tool is placed on spindle 2. Tools can only be loaded via the loading point of the magazine.

For this configuration, 20 transfer job requests from the PLC to the tool management are programmed in FB QUIT_WZV. These requests are triggered by the user via the user interfaces UI 1 – UI 20.
With an asynchronous job request, tool position changes outside a programmed sequence, e.g. for movements in JOG can be sent to the tool management after a tool change has been aborted.

The following job requests are implemented in FB 90 and triggered in FC 90 in the example in QUIT_5.awl:

UI	Function
1	Acknowledgement load tool completed, load point magazine
2	Abort / Reset load tool, load point magazine
3	Acknowledgement unload tool completed, load point magazine
4	Abort / Reset unload tool, load point magazine
5	Acknowledgement prepare change completed, spindle 1
6	Abort / Reset prepare change, spindle 1
7	Acknowledgement prepare change completed, spindle 2
8	Abort / Reset prepare change spindle 2
9	Status change spindle 1 → magazine
10	Status change magazine → spindle 1

UI	Function
11	Abort / Reset change spindle 1
12	Status change spindle 2 → magazine
13	Status change magazine → spindle 2
14	Abort / Reset change spindle 2
15	Acknowledgement relocate (from MMC)
16	Asynchronous relocation spindle 1 → magazine
17	Asynchronous relocation magazine → spindle 1
18	Asynchronous relocation spindle 1 → magazine
19	Asynchronous relocation magazine → spindle 2
20	Change in actual magazine location position
21	–
22	Acknowledgement positioning at loading point
23	Abort / Reset positioning at loading point

... .

32	–

The actual position for job request 20 in FB 90 is taken from DB instance DIW 10. The actual position address can be changed by the user.
Loading and unloading of spindles is not programmed in FB QUIT_WZV. The user can program this function himself using a free user interface. Jump target lists IFC 1 – IFC 3 can be used as an example.

Caution:
With asynchronous relocation the magazine location state “Z” (reserved for tool in buffer) is not taken into account. This means that with asynchronous relocation from magazine to spindle the identifier “Z” is not set and with asynchronous relocation from spindle to magazine the identifier “Z” is not reset in the old location. In this case, “Z” must be set and cleared with FB 3 (write NC variable).
With NC SW 3.2 and later, magazine location status “Z” is transferred with Task-Ident 5 for asynchronous relocation.

11.2 FB 91: LE_SUCH Search for empty location for tool in buffer

Description of functions

A search for an empty location in the magazine for a tool in the buffer can be made with FB LE_SUCH.
A separate instance DB from the user area must be assigned to each FB 91 call. When FB 91 is called an empty location is searched in the magazine for a tool in the buffer on a positive edge change at control input Start.

The location in the buffer is sent to the FB in input parameter MagNo_ZW and LocNo_ZW and the magazine number in which the empty location search is being made is sent in parameter Mag no.

Successful execution is indicated by status parameter Done with logical “1”. The empty location is output via output parameter MagNo_Empty and LocNo_Empty. Any errors that have occurred are output via Error and State.

The empty location search extends over several PLC cycles. The block can only be called cyclically. FB 2 is called twice and FB 4 once in FB 91. These blocks are called with the multi-instance DB in FB 91.

Important!

FB 91 can perform the empty location search only if basic program parameter NCKomm has been set to “1” (in OB100: FB 1, DB 7).

Declaration

```
FUNCTION_BLOCK FB 91

VAR_INPUT

Start :       BOOL;
MagNo_ZW :    INT;
LocNo_ZW :    INT;
MagNo :       INT;

END_VAR

VAR_OUTPUT

Active:       BOOL;
Done :        BOOL;
Error :       BOOL;
```

```
State :          WORD;
MagNo_Empty: INT;
LocNo_Empty:  INT;

END_VAR
```

Explanations of formal parameters

All the formal parameters of block LE-SUCH are listed in the table below.

Signal	Type	Type	Value range	Remarks
Start	I	Bool		Start empty location search.
MagNo_ZW	I	int	–1:	Magazine number of the buffer
LocNo_ZWv	I	int	–1:	Location number of the buffer
MagNo	I	int	–1:	Magazine number of magazine in which empty location search is to be performed.
Active	O	Bool		Empty location search running
Done	O	Bool		Empty location found. Signal pending for one PLC cycle.
Error	O	Bool		Empty location search was acknowledged negatively or could not be executed. Signal pending for one PLC cycle. Error number is stored in State.
State	O	Word		See error identifiers
MagNo_Empty	O	int		Magazine number empty location
LocNo_Empty	O	int		Location number empty location

You can still use the following signals in the instance DB of FB91 to control the search for empty locations:

TNr_write = 1: The T number of the tool for the empty location search is stored in TNr_FB2. MagNo_ZW/LocNo_ZW are not evaluated.

MMCSEM =1: No semaphores set for PI service TMFDPL

Error detection

If it is not possible to execute a request in the empty location search, this is indicated in status parameter Error with ‘logical 1’. The cause of the error is coded at block output State:

State	Meaning	Note
1	Error while reading T number (FB 2) from MagNo_ZW and LocNo_ZW.	The error ID of FB 2 is stored as the variable StateFB2_WZGesp in the instance DB.
2	The logical T number of the magazine location is zero.	Check whether a tool is located in the magazine location of the buffer.
3	Error on PI service empty location search (FB 4).	The error ID of FB 4 is stored in the variable StateFB4Gesp.
4	Error on acknowledgement parameter of PI service read TMFDPL with FB 2.	The error ID of FB 2 is stored in the variable State FB 2_ParGesp in the in the instance DB.
5	Search for empty location terminated with error	No empty location available in the magazine
6	Invalid step number	Internal error in FB.
7	Error while reading variable numMagPlaceParams with FB 2.	Restart required.
8	Error FB 4 PI service MMCSEM	Semaphore for PI service TMFDPL on 1st event. Another job may be active (MMC)

Timing diagram

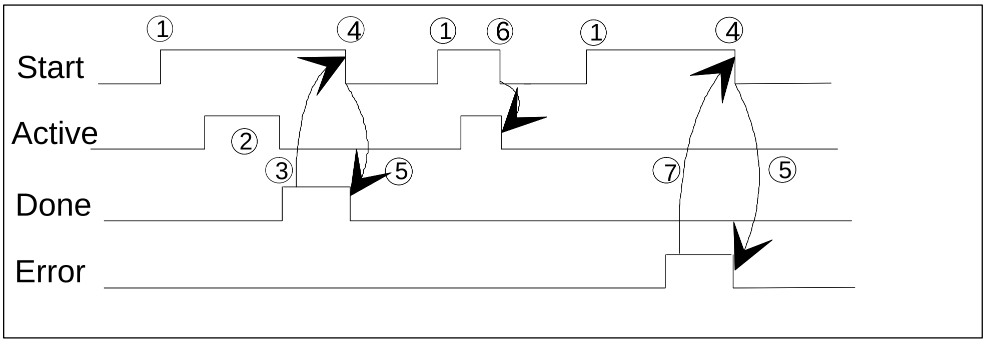


Fig. 11-1 Timing diagram for FB 91

1. Activation of function
2. Empty location search active
3. Positive acknowledgement: Empty location found
4. Reset of function activation signal after receipt of acknowledgement by user, signal change by FC
5. If function activation signal is reset before receipt of acknowledgement, the output signals are not updated; not relevant once the function is running
6. Negative acknowledgement: Error occurred. Error code in output parameter state

Example of call

```
U        DB21.DBX 204.0;     // M80 signal
S        M 150.0;            // Start empty location search
CAll FB91,DB 91(
Start : M 150.0,             // Start empty location search
MagNo_ZW : 9998,             // Magazine no.=buffer
LocNo_ZW :2,                 // Magazine loc. 2 = gripper

MagNo :          1           // Magazine no. for empty location = 1
Active:          M 150.1,    // Empty location search active
Done :           M 150.2,    // Empty location found
Error:           M 150.3,    // Error in empty loc. search
State :          MW 152,     // Error number
MagNo_Empty: MW 154,         // Magazine number, empty loc.
LocNo_Empty: MW156);         // Location number, empty loc.

U   M 150.2;                 // Empty location found
O   M 150.3;                 // Error in empty loc. search
State:  M 150.0;             // Start empty location search
U   M 150.3;
S   M 160.0;                 // Error in empty location search
```

Blocks to be loaded

FB 91, FB 2, FB 4, DB 91, DB 119

11.3 FB 92: GET_LOC Read magazine location and tool data

Description of functions

The magazine location data of a magazine location and the tool data of a tool can be read with FB GET_LOC.
A separate instance DB from the user area must be assigned to each FB 92 call.
Depending on the signal at input GetWkz, calling FB 92 reads the data on a positive edge change at control input Req. If input GETWKZ carries a 1-signal the magazine location data and tool data are read. If GETWKZ = 0 only the magazine location data area read.
The magazine location is transferred to the FB via input parameters MagNo and LocNr. Successful execution of the function is indicated at status parameter NDR with logical “1”. Any errors that have occurred are output via Error and State.

The following data are read:

- Magazine location data (TP):
  – Location status
- General tool data (TD):
  – Size to the left in half locations
  – Size to the right in half locations
  – Size to the top in half locations
  – Size to the bottom in half locations
  – Magazine location type
  – Tool status

The data are stored in the instance DB. A detailed description of the data is to be found in the Description Lists in Chapter 4, Variables, and in the Description of Functions Tool Management in the Section Programming.
Execution of the read functions extends over several PLC cycles. The block can only be called cyclically.

Note

FB 2 is called twice in FB 92. These blocks are called with multi.instance DB in FB 92.

Declaration

```
FUNCTION_BLOCK FB 92

VAR_INPUT
Req :          BOOL;
GetWkz:        BOOL;
MagNo :        INT;
LocNo :        INT;

END_VAR

VAR_OUTPUT
NDR:           BOOL;
Error:         BOOL;
State :        WORD;

END_VAR
```

Explanations of formal parameters

All the formal parameters of block GET_LOC are listed in the table below.

Signal	Type	Type	Value range	Remarks
NDR	I	Bool		Start Read state
GetWkz	I	Bool		0 signal: Read magazine location data 1 signal: Read magazine location and tool data
MagNo	I	int	–1:	Magazine number
LocNo	I	int	–1:	Location number
Done	O	Bool		Operation successfully executed.
Error	O	Bool		Task was acknowledged negatively or could not be executed. Error number is stored in State.
State	O	Word		See error messages

You can still use the following signals in the instance DB of the FB 92 to control the read job:

TNr_write = 1: The T number of the tool for Read Tool Data is stored in TNo (DIW28). MagNo/LocNo are not interpreted.
Only tool data are read.

Error detection

If it is not possible to execute a request, this is indicated in status parameter Error with “logical 1”. The cause of the error is coded at block output State:

State	Meaning	Note
1	Error on reading magazine location data (FB 2).	The error ID of FB 2 is stored as the variable StateFB2_TNrGesp. in the instance DB.
2	The logical T number of the magazine location is zero.	Check whether a tool is located in the specified magazine location.
3	Error on reading tool data (FB 2).	The error ID of FB 2 is stored as the variable StateFB2_WZGesp in the instance DB.
6	Invalid step number	Internal error in FB.
7	Error while reading variable num-MagPlaceParams with FB 2.	Restart required.

Data interface

DB instance	
Byte	Description of the data read
DIW 28	Logical T number
DIW 30	Location status
DIW 32	Size to the left in half locations
DIW 34	Size to the right in half locations
DIW 36	Size to the top in half locations
DIW 38	Size to the bottom in half locations
DIW 40	Magazine location type
DIW 42	Tool state

Timing diagram

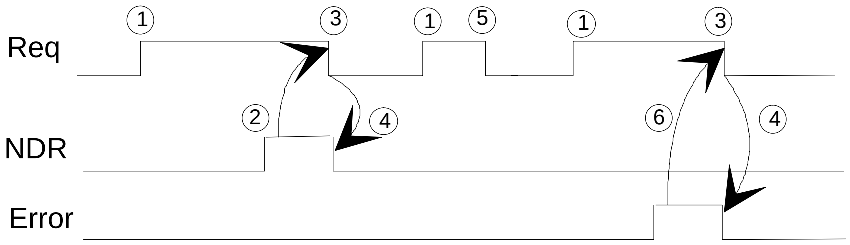


Fig. 11-2 Timing diagram for FB 92

1. Activation of function
2. Positive acknowledgement: Receive new data
3. Reset of function activation signal after receipt of acknowledgement
4. Signal change by FB
5. If function activation signal is reset before receipt of acknowledgement, the output signals are not updated; not relevant once the function is running
6. Negative acknowledgement: Error occurred. Error code in output parameter state

Example of call

```
U          DB21.DBX 204.1;     // M81 signal
S          M 160.0;            // Start Read states
CAll FB92,DB 92(
Req :      M 160.0,            // Start Read states
GetWkz:      true,             // Read magazine location and tool data
MagNo: 9998,                   Magazine no. = buffer
LocNo:2,                       // Magazine loc. 2 = gripper
NDR :      M 160.1,            // Task executed
Error:     M 160.2,            // Reading error
State :    MW 162);            // Error number

U          M 160.1;            // Data read
O          M 160.2;            // Reading error
State :    M 160.0;            // Start empty location search
U          M 160.2;
S          M 160.7;            // Tool data reading error
```

Blocks to be loaded

FB 92, FB 2, DB 92, DB 119

11.4 FB 93: PUT_LOC Write magazine location and tool data

Description of functions

The magazine location status of a magazine location and the tool status of a tool can be written with FB PUT_LOC.
A separate instance DB from the user area must be assigned to each FB 93 call.
Depending on the signal at input PutWkz, calling FB 93 writes the data on a positive edge change at control input Req. If input PutWkz carries a 1 signal the tool status is written, if PutWkz = 0, the magazine location status is written.
The magazine location is transferred to the FB via input parameters MagNo and LocNr. Successful execution is indicated by status parameter Done with logical “1”.
Any errors that have occurred are output via Error and State.
The status data are entered in the instance DB. A detailed description of the data is to be found in the Description Lists in Chapter 4, Variables, and in the Description of Functions Tool Management in the Section Programming.
Execution of the write functions extends over several PLC cycles. The block can only be called cyclically.
FB 2 is called once and FB 3 twice in FB 93. These blocks are called with multi instance DB in FB 92.

Note
FB 93 can execute the read operations only if basic program parameter NCKomm has been set to “1” (in OB100: FB 1, DB 7).

Declaration

```
FUNCTION_BLOCK FB 93

VAR_INPUT

Req :           BOOL;
PutWkz:         BOOL;
MagNo :         INT;
LocNo :         INT;
END_VAR

VAR_OUTPUT

Done            BOOL;
Error:          BOOL;
State :         WORD;

END_VAR
```

Explanations of formal parameters

All the formal parameters of block PUT_LOC are listed in the table below

Signal	Type	Type	Value range	Remarks
Done	I	Bool		Start write status
PutWkz	I	Bool		0 signal: Write magazine location state 1 signal: Write tool status
MagNo	I	int	1 ..	Magazine number
LocNo	I	int	1 ..	Location number
Done	O	Bool		Operation successfully executed.
Error	O	Bool		Task was acknowledged negatively or could not be executed. Error number is stored in State.
State	O	Word		See error messages

The write job can also be influenced with the following signals in the instance DB of FB93:

TNr_write = 1: The T number of the tool for Write Tool Data is stored in T No (DIW32). MagNo/LocNo are not evaluated

Error detection

If it is not possible to execute a request, this is indicated in status parameter Error with “logical 1”. The cause of the error is coded at block output State:

State	Meaning	Note
1	Error on reading magazine location data (FB 2).	The error ID of FB 2 is stored as the variable StateFB2_TNrGesp. in the instance DB.
2	The logical T number of the magazine location is zero.	Check whether a tool is located in the specified magazine location.
3	Error while writing magazine location data (FB 3).	The error ID of FB 3 is stored as the variable StateFB3_LocGesp in the instance DB.

State	Meaning	Note
4	Tool data writing error (FB 3).	The error ID of FB 3 is stored as the variable StateFB3_WZGesp in the instance DB.
6	Invalid step number	Internal error in FB.
7	Error while reading variable numMagPlaceParamswith FB 2.	Restart required.

Data interface

DB instance	User interface
Byte	Description of the data
DIW 32	Logical T number (read by FB)
DIW 34	Location state (read by FB)
DIW 36	Location state (data to tool management, block TP, parameter P5)
DIW 38	Tool status (data to tool management, block TD, toolState)

Timing diagram

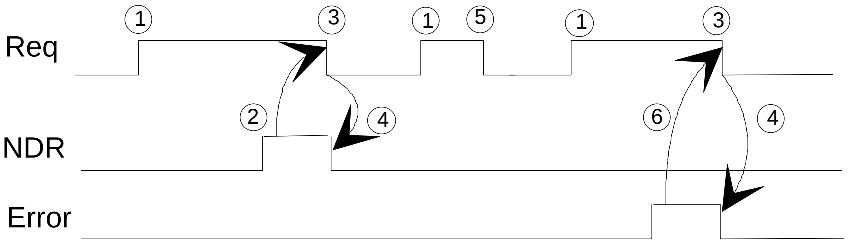


Fig. 11-3 Timing diagram for FB 93

1. Activation of function
2. Positive acknowledgement: Variables have been written
3. Reset of function activation signal after receipt of acknowledgement
4. Signal change by FB
5. If function activation signal is reset before receipt of acknowledgement, the output signals are not updated; not relevant once the function is running
6. Negative acknowledgement: Error occurred. Error code in output parameter state

Example of call

```
U          DB21.DBX 204.2;      // M82 signal
S          M 164.0;             // Start Read states
CAll FB93,DB 93(
Req :      M 164.0,             // Start Read states
GetWkz:      true,              // Read magazine location and tool data
MagNo: 1                        // Magazine no. = magazine 1
LocNo:10,                       // Magazine location 10
Done:      M 164.1,             // Task executed
Error:     M 164.2,             // Reading error
State :    MW 166);             // Error number

U          M 164.1;             // Data read
O          M 164.2;             // Reading error
State :    M 164.0;             // Start empty location search
U          M 164.2;
S          M 164.7;             // Tool data reading error
```

Blocks to be loaded

FB 93, FB 2, DB 93, DB 119

Notes

Abbreviations and Terms A

A.1 Abbreviations

ASUB	Asynchronous Subroutine
C1 .. C4	Channel 1 to channel 4
CUTOM	Cutter Radius Compensation: Tool radius compensation
DB	Data Block in the PLC
DBB	Data Block Byte in the PLC
DBW	Data Block Word in the PLC
DBX	Data Block Bit in the PLC
DDE	Dynamic Data Exchange
DW	Data Word
ENC	Encoder
EPROM	Erasable Programmable Read Only Memory
FB	Function Block
FC	Function Call: Function block in the PLC
FM-NC	Function Module – Numerical Control
GUD	Global User Data
HEX	Abbreviation for hexadecimal number
IBN	Installation and Start-up
INC	Increment
INI	Initializing Data
IPO	Interpolator
ISO Code	Special punchtape code, number of punched holes per character always even
K Bus	Communications Bus
MCS	Machine Coordinate System
MD	Machine Data
MDA	Manual Data Automatic

MMC	Man Machine Communication: User interface on numerical control systems for operator control, programming and simulation.
MSD	Main Spindle Drive
NC	Numerical Control
NCK	Numerical Control Kernel: Numerical kernel with block preparation, traversing range, etc.
OA	Open Architecture
OB	Organization Block in the PLC
OEM	Original Equipment Manufacturer: Manufacturer whose products are marketed under a different name.
OP	Operator Panel
OPI	Operator Panel Interface
PI	Program Invocation: Programming instance
PLC	Programmable Logic Controller
RAM	Random Access Memory in which data can be read and written
TLC	Tool Length Compensation
TM	Tool Management
TO	Tool Offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
TRC	Tool Radius Compensation
VDI	Virtual Device Interface
V.24	Serial interface (definition of interchange circuit between DTE and DCE)
WCS	Workpiece Coordinate System

A.2 Terms

Important terms are listed in alphabetical order. The symbol "–>" precedes terms which are explained under a separate entry in this list.

A

Access rights

Programs and other data are protected internally by a system of access rights based on 7 levels:

- Three password levels for system manufacturer, machine manufacturer and user and

Four keyswitch settings which can be evaluated via the PLC (depending on the keyswitch hardware).

Alarms

All messages and alarms are displayed on the operator panel in plaintext with date and time as well as the appropriate symbol for the reset criterion. Alarms and messages are displayed separately.

1. Alarms and messages in the part program
   Alarms and messages can be displayed directly from the part program in plaintext.
2. Alarms and messages from PLC
   Alarms and messages relating to the machine can be displayed from the PLC program in plaintext. No additional function block packages are required for this purpose.

Approach fixed machine point

Approach motion towards one of the predefined –> fixed machine points.

Archiving

Reading out of files and/or directories to an external storage medium.

Asynchronous subroutine

A part program that can be started out of synchronism with (or independently of) the current program status by means of an interrupt signal (e.g. "High-speed NC input" signal) (SW package 4 and later).

Auxiliary functions

Auxiliary functions can be used to pass –> parameters to the –> PLC in –> part programs, triggering reactions there which are defined by the machine manufacturer.

Axes

- CNC axes are classified according to their functional scope as:
- Axes: Interpolating path axes

Auxiliary axes: Non-interpolating infeed and positioning axes with axis-specific feed. Auxiliary axes are not involved in the actual machining operation, e.g. tool feeder, tool magazine.

Axis identifier

In compliance with DIN 66217, axes are identified as X, Y, Z
for a right-angled, rectangular –> coordinate system.
–> Rotary axes rotating around X, Y, Z are assigned the identifiers A, B, C. Additional axes, which are parallel to those specified, can be identified with other letters.

Axis/spindle exchange

An axis/spindle is permanently assigned to a particular channel via a machine data setting. Using program commands it is possible to release an axis/spindle and assign it to another channel.

B

Back up

Dump of contents of storage medium (hard disk) to an external storage device for the purpose of backing up and/or archiving data.

Basic coordinate system

Cartesian coordinate system, is mapped onto machine coordinate system by means of transformation.
In the –> part program, the programmer uses the axis names of the basic coordinate system. The basic coordinate system exists in parallel to the –> machine coordinate system when no –> transformation is active. The difference between the systems relates only to the axis identifiers.

Baud rate

Rate at which data transmission takes place (bit/s).

Block

A section of a –> part program terminated with a line feed. A distinction is made between –> main blocks and –> subblocks.

Block search

The block search function allows selection of any point in the part program at which machining must start or be continued. The function is provided for the purpose of testing part programs or continuing machining after an interruption.

Boot

Loading the system program after Power On.

C

Channel

A channel can execute a –> part program independently of other channels. A channel has exclusive control over the axes and spindles assigned to it. Part program sequences on different channels can be coordinated by –> synchronization.

Channel structure

The channel structure makes it possible to process the –> programs of the individual channels simultaneously and asynchronously.

Contour monitoring

The following error is monitored within a definable tolerance band as a means of measuring the contour accuracy. Overloading of the drive, for example, may result in an unacceptably large following error. In such cases, an alarm is output and the axes stopped.

Cycle

Subroutine for executing a frequently repeated machining operation on the workpiece.

Cycles support

The available cycles are listed in menu “Cycles support” in the “Program” operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plaintext.

D

D number

Number for the tool offset memory

Data block

1. Data unit on the –> PLC which can be accessed by –> HIGHSTEP programs.
2. Data unit on the –> NC: Data blocks contain data definitions for global user data. These data can be initialized directly when they are defined.

Data word

A data unit, two bytes in size, within a –> PLC data block.

Dimensional specification, metric and inches

Position and lead values can be programmed in inches in the machining program. The control is set to a basic system regardless of the programmable dimensional specification (G70/G71).

E

Editor

The editor makes it possible to create, modify, extend, join and import programs/texts/program blocks.

F

File type

Possible types of files, e.g. part programs, zero offsets, R parameters, etc.

Fixed machine point

A point defined uniquely by the machine tool, such as a machine reference point.

Fixed point approach

Machine tools can approach defined fixed points such as a tool change point, load point, pallet change point, etc. The coordinates of these points are stored on the control. Where possible, the control traverses these axes in –> rapid traverse.

Frame

A frame is a calculation rule that translates one Cartesian coordinate system into another Cartesian coordinate system. A frame contains the components Zero Offset, Rotation, Scaling and Mirroring.

I

Identifier

Words in compliance with DIN 66025 are supplemented by identifiers (names) for variables (arithmetic variables, system variables, user variables), for subroutines, for vocabulary words and for words with several address letters. These supplements have the same meaning as the words with respect to block format. Identifiers must be unique. It is not permissible to use the same identifier for different objects.

Increment

Traversing path length specification based on number of increments. The number of increments can be stored as a –> setting data or selected with keys labeled with 10, 100, 1000, 10000.

K

Keyswitch

The keyswitch is the mode selector switch on the CPU. The keyswitch is operated by means of a removable key.
The keyswitch on the –> machine control panel has 4 positions which are assigned functions by the operating system of the control. There are also three keys of different colors belonging to the keyswitch that can be removed in the specified positions.

L

Languages

The operator-prompt display texts, system messages and system alarms are available in five system languages:
German, English, French, Italian and Spanish.
The user can select two of the listed languages at a time in the control (Start-Up operating area).

M

Machine axes

Axes which exist physically on the machine tool.

Machine control panel

An operator panel on a machine tool with operating elements such as keys, rotary switches, etc. and simple indicators such as LEDs. It is used for direct control of the machine tool via –>PLC.

Machine coordinate system

A coordinate system which is related to the axes of the machine tool.

Machine zero

A fixed point on the machine tool which can be referenced by all (derived) measurement systems.

Machining channel

A channel structure makes it possible to reduce downtimes by allowing sequences of motions to be executed in parallel, e.g. to traverse a loading gantry while a machining operation is in progress. In this case, a CNC channel must be regarded as a separate CNC control with decoding, block preparation and interpolation.

Macros

Individual instructions in the programming language can be linked to create one instruction. This condensed instruction sequence is called by a user-defined name in the CNC program and the macro command executed in accordance with the individual instructions.

Magazine

The following categories of magazine are utilized in the TM system:

- Real magazine
  Actual magazine for storing tools, the NCK is capable of managing several real magazines.
- Internal magazine
  All other positions in which a tool may be located are handled logically in the NCK as a magazine (or magazine location).
  There are exactly 2 internal magazines: The load magazine and the buffer magazine.
- Virtual magazine
  This term is applied on MMCs to refer to all the real and internal magazines of one TO unit.
- Active magazine
  Magazine which is linked to a spindle and from which a tool change can be executed.
- Background magazine
  A magazine which is linked to a previous magazine via system parameter $TC_MAP5. Generally speaking, tool changes involve the relocation of tools.

Main block

A block prefixed by ":" containing all the parameters required to start execution of a –> part program.

Main program

–> Part program identified by a number or name in which further main programs, subroutines or –> cycles may be called.

Main run

The part program blocks which have been decoded and edited in the "preprocessing" run are executed in the "main run".

MDA

Control system operating mode: Manual Data Automatic. In the MDA mode, individual program blocks or block sequences with no reference to a main program or subroutine can be input and executed immediately afterwards through actuation of the NC start key.

Messages

All messages programmed in the part program and –> alarms detected by the system are displayed on the operator panel in plaintext. Alarms and messages are displayed separately.

Metric measurement system

Standardized system of units for lengths in millimeters (mm), meters (m), etc.

Mirroring

Mirroring exchanges the leading signs of the coordinate values of a contour in relation to an axis. Mirroring can be performed simultaneously in relation to several axes.

Module

"Module" is the term given to any file required for creating and processing programs.

N

NC

Numerical Control: It incorporates all the components of the machine tool control system: –> NCK, –> PLC, –> MMC, –> COM.
Note: CNC (computerized numerical control) would be a more appropriate description for the SINUMERIK 840D or FM-NC controls.

NCK

Numeric Control Kernel: Component of the NC control which executes –> part programs and essentially coordinates the movements on the machine tool.

NRK

Numeric Robotic Kernel (operating system of the –> NCK)

O

OEM

The scope for implementing individual solutions (OEM applications) for the SINUMERIK 840D has been provided for machine manufacturers who wish to create their own operator interface or integrate process-oriented functions in the control.

Offset memory

Data area in the control in which tool offset data are stored.

Operating mode

An operating concept on a SINUMERIK control. The modes
–> Jog, –> MDA, –> Automatic are defined.

Operator interface

The operator interface (OI) is the display medium of a CNC system. It takes the form of a screen and has eight horizontal and eight vertical softkeys.

Oriented spindle stop

Stops the workpiece spindle at a specified orientation angle, e.g. to perform an additional machining operation at a specific .position

Oriented tool retraction

RETTOOL: In the case of interruptions in the machining process (e.g. when a tool breaks), a program command can be used to retract the tool in a user-specified orientation by a defined distance.

Override

Manual or programmable control feature which enables the user to override programmed feedrates or speeds in order to adapt them to a specific workpiece or material.

P

Part program

A sequence of instructions to the NC control which combine to produce a specific –> workpiece by performing certain machining operations on a given –> blank.

PLC

Programmable Logic Controller: Component of the –> NC: Programmable controller for processing the control logic on the machine tool.

PLC program memory

- SINUMERIK 840D: The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory. The PLC user memory can be expanded up to 128 KB.
- SINUMERIK 810D: The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory of the CPU 314. The user memory in the basic configuration of the S7-CPU314 is 64 KB in size and can be optionally expanded to 128 KB.

R

Reference point

Point on the machine tool with which the measuring system of the –> machine axes is referenced.

Reference point approach

If the position measuring system used is not an absolute-value encoder, then a reference point approach operation is required to ensure that the actual values supplied by the measuring system are in accord with the machine coordinate values.

Replacement tool

A tool group generally contains several tools. For tool change purposes, only the identifier is specified in the part program. The tool with the “active” status is generally selected as the new tool. But if this is disabled, then one of the other twin tools, i.e. the replacement tool, is selected instead. –> Twin tool

REPOS

1. Reapproach contour by means of operator input
   REPOS allows the point of interruption to be reapproached by means of the direction keys.
2. Reapproach contour by means of program
   A selection of approach strategies are available in the form of program commands: : Approach point of interruption, approach start of block, approach end of block, approach a point on the path between start of block and interruption.

R parameter

Arithmetic parameter. The programmer of the –> part program can assign or request the values of the R parameter as required.

S

Safety functions

The control includes continuously active monitoring functions which detect faults in the –> CNC, the programmable controller (–> PLC) and the machine so early that damage to the workpiece, tool or machine rarely occurs. In the event of a fault, the machining operation is interrupted and the drives shut down. The fault cause is then stored and displayed as an alarm. At the same time, the PLC is informed that a CNC alarm is pending.

Setting data

Data which provide the control with information about properties of the machine tool in a way defined by the system software.

Softkey

A key whose name appears on an area of the screen. The choice of softkeys displayed is adapted dynamically to the operating situation. The freely assignable function keys are linked to defined functions in the software.

Spindles

- Spindle = toolholder
  Toolholder is generally the location for the machining tool. However, the term “spindle” is frequently used in this general context.
- Main spindle = master spindle
  This is the spindle with the number defined by machine data MD $MC_SPIND_DEF_MASTER_SPIND. Language command SETMS(n) can be programmed to declare the spindle with number n as the master spindle. A channel has exactly one master spindle.
- Secondary spindle
  This term refers to all spindles that are not the master spindle.

Standard cycles

Standard cycles are used to program machining operations which repeat frequently:

- For drilling/milling operations
- For turning operations

The available cycles are listed in menu “Cycle support” in the “Program” operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plaintext.

Subblock

Block prefixed by “N” containing information for an operation such as a position parameter.

Subroutine

A sequence of instructions of a –> part program which can be called repeatedly with various parameters. –> Cycles are a type of subroutine.

Synchronization

Instructions in –> part programs for coordination of the operations in different –> channels at specific machining points.

Synchronized actions

1. Auxiliary function output
   While a workpiece is being machined, technological functions (–> auxiliary functions) can be output from the CNC program to the PLC. These auxiliary functions control, for example, ancillary equipment on the machine tool such as the sleeve, gripper, chuck, etc.
2. High-speed auxiliary function output
   The acknowledgement times for the –> auxiliary functions can be minimized and unnecessary halts in the machining process avoided for time-critical switching functions.

Synchronous axes

Synchronized axes require the same amount of time to traverse their path as –> geometry axes for their path.

System variables

A variable which exists although it has not been programmed by the –> part program programmer. It is defined by the data type and the variable name, which is prefixed with **$**.
See also –> User-defined variable.

T

Tool nose radius compensation

A contour is programmed on the assumption that a pointed tool will be used. Since this is not always possible, the control makes allowance for the curvature radius of the tool being used. The curvature centre point displaced by the curvature radius is guided equidistantly to the contour.

Tool offset

A tool is selected by the programmed T function (5 decades, integer) in the block. Up to 12 cutting edges (D addresses) can be assigned to each T number. The number of tools to be managed in the control is set at the configuration stage.

Tool radius compensation

In order to program a desired –> workpiece contour directly, the control must traverse a path equidistant to the programmed contour, taking into account the radius of the tool used (G41/G42).

Transformation

Programming in a Cartesian coordinate system, execution in a non-Cartesian coordinate system (e.g. with machine axes as rotary axes).

Twin tool, tool group

Twin tools all have the same identifier, but a different duplo number. Twin tools with the same identifier are also referred to as a “tool group”.

U

User-defined variable

Users can define variables in the –> part program or data block (global user data) for their own use. A definition contains a data type specification and the variable name. See also –> system variable.

User program –> part program

User memory

All programs and data such as part programs, subroutines, comments, tool offsets, zero offsets/frames and channel and program user data can be stored in the joint CNC user memory.

V

Variable definition

A variable is defined through the specification of a data type and a variable name. The variable name can be used to address the value of the variable.

Vocabulary words

Words with a specific notation which have a defined meaning in the programming language for –> part programs.

W

Working memory

The working storage is a Random Access Memory in the –> CPU containing the user program which is accessed by the processor during program processing.

Workpiece

1. Part to be created/machined by the machine tool or
2. a workpiece is a directory in which programs and other data are stored. Workpieces are stored in another directory.

Workpiece coordinate system

The starting position of the workpiece coordinate system is the –> workpiece zero. In machining operations programmed in the workpiece coordinate system, the dimensions and directions refer to this system.

Workpiece zero

The workpiece zero is the starting point for the –> workpiece coordinate system. It is defined by its distance to the machine zero.

Notes

References

B

General Documentation

/BU/
SINUMERIK 840D/810D/FM-NC
Ordering Information
Catalog NC 60.1
Order No.: E86060-K4460-A101-A6-7600

/ST7/
SIMATIC
SIMATIC S7 Programmable Logic Controllers
Catalog ST 70
Order No.: E86 060-K4670-A111-A3

/VS/
SINUMERIK 840D/810D/FM-NC
Technical Catalog
Catalog NC 60.2
Order No.: E86060-K4460-A201-A4-7600

/W/
SINUMERIK 840D/810D/FM-NC
Brochure

/Z/
SINUMERIK, SIROTEC, SIMODRIVE
Accessories and Equipment for Special-Purpose Machines
Catalog NC Z
Order No.: E86060-K4490-A001-A6-7600

Electronic Documentation

/CD6/
The SINUMERIK System (04.00 Edition)
DOC ON CD
(includes all SINUMERIK 840D/810D/FM-NC and SIMODRIVE 611D publications)
Order No.: 6FC5 298-5CA00-0BG2

User Documentation

/AUE/
SINUMERIK 840D/810D/FM-NC
AutoTurn Graphic Programming System (07.99 Edition)
Part 2: Setup
Order No.: 6FC5 298-4AA50-0BP2

/AUK/
SINUMERIK 840D/810D/FM-NC
Short Guide AutoTurn Operation (07.99 Edition)
Order No.: 6FC5 298-4AA30-0BP2

/AUP/
SINUMERIK 840D/810D/FM-NC
AutoTurn Graphic Programming System (07.99 Edition)
Part 1: Programming
Order No.: 6FC5 298-4AA40-0BP2

/BA/
SINUMERIK 840D/810D/FM-NC
Operator's Guide (04.00 Edition)
Order No.: 6FC5 298-5AA00-0BP2

– Operator's Guide

– Operator's Guide Interactive Programming (MMC 102/103)

/BAE/
SINUMERIK 840D/810D/ FM-NC
Operator's Guide Unit Operator Panel (04.96 Edition)
Order No.: 6FC5 298-3AA60-0BP1

/BAH/
SINUMERIK 840D/810D
Operator's Guide HT6 (HPU new) (06.00 Edition)
Order No.: 6FC5 298-0AD60-0BP0

/BAK/
SINUMERIK 840D/810D/FM-NC
Short Operation Guide (12.98 Edition)
Order No.: 6FC5 298-5AA10-0BP0

/BAM/
SINUMERIK 840D/810D
Operator's Guide ManualTurn (02.00 Edition)
Order No.: 6FC5 298-5AD00-0BP0

/KAM/
SINUMERIK 840D/810D
Short Guide ManualTurn (11.98 Edition)
Order No.: 6FC5 298-2AD40-0BP0

/BAS/
SINUMERIK 840D/810D
Operator's Guide ShopMill (06.00 Edition)
Order No.: 6FC5 298-5AD10-0BP1

/KAS/ SINUMERIK 840D/810D
Short Guide ShopMill (01.98 Edition)
Order No.: 6FC5 298-2AD30-0BP0

/BAP/ SINUMERIK 840D/840Di/810D
Operator's Guide Handheld Programming Unit (04.00 Edition)
Order No.: 6FC5 298-5AD20-0BP1

/BNM/ SINUMERIK 840D/840Di/810D/FM-NC
User's Guide Measuring Cycles (04.00 Edition)
Order No.: 6FC5 298-5AA70-0BP2

/DA/ SINUMERIK 840D/840Di/810D/FM-NC
Diagnostics Guide (04.00 Edition)
Order No.: 6FC5 298-5AA20-0BP2

/PG/ SINUMERIK 840D/840Di/810D/FM-NC
Programming Guide Fundamentals (04.00 Edition)
Order No.: 6FC5 298-5AB00-0BP2

/PGA/ SINUMERIK 840D/840Di/810D/FM-NC
Programming Guide Advanced (04.00 Edition)
Order No.: 6FC5 298-5AB10-0BP2

/PGK/ SINUMERIK 840D/810D/FM-NC
Short Guide Programming (12.98 Edition)
Order No.: 6FC5 298-5AB30-0BP0

/PGZ/ SINUMERIK 840D/840Di/810D/FM-NC
Programming Guide Cycles (04.00 Edition)
Order No.: 6FC5 298-5AB40-0BP2

/PI/ **PCIN 4.4**
Software for Data Transfer to/from MMC Module
Order No.: 6FX2 060 4AA00-4XB0 (German, English, French)
Order from: WK Fürth

/SYI/ SINUMERIK 840Di
System Overview (01.00 Edition)
Order No.: 6FC5 298-5AE40-0BP0

Manufacturer/Service Documentation

a) Lists

/LIS/ SINUMERIK 840D/840Di/810D/FM-NC
SIMODRIVE 611D
Lists (04.00 Edition)
Order No.: 6FC5 297-5AB70-0BP2

b) Hardware

/BH/
SINUMERIK 840D/840Di/810D/FM-NC
Operator Components Manual (HW) (04.00 Edition)
Order No.: 6FC5 297-5AA50-0BP2

/BHA/
SIMODRIVE **Sensor**
Absolute Encoder with Profibus DP
User Guide (HW) (02.99 Edition)
Order No.: 6SN1197-0AB10-0YP1

/EMV/
SINUMERIK, SIROTEC, SIMODRIVE
EMC Installation Guide
Planning Guide (HW) (06.99 Edition)
Order No.: 6FC5 297-0AD30-0BP1

/PHC/
SINUMERIK 810D
Manual Configuring (HW) (04.00 Edition)
Order No.: 6FC5 297-3AD10-0BP2

/PHD/
SINUMERIK 840D
NCU 561.2–573.2 Configuring Manual (HW) (04.00 Edition)
Order No.: 6FC5 297-5AC10-0BP2

/PHF/
SINUMERIK FM-NC
NCU 570 Configuring Manual (HW) (04.96 Edition)
Order No.: 6FC5 297-3AC00-0BP0

/PMH/
SIMODRIVE **Sensor** (05.99 Edition)
Measuring System for Main Spindle Drives
Configuring/Installation Guide, SIMAG-H (HW)
Order No.: 6SN1197-0AB30-0BP0

c) Software

/FB1/
SINUMERIK 840D/840Di/810D/FM-NC
Description of Functions, Basic Machine (Part 1) (04.00 Edition)
(the various sections are listed below)
Order No.: 6FC5 297-5AC20-0BP2

A2	Various Interface Signals
A3	Axis Monitoring, Protection Zones
B1	Continuous Path Mode, Exact Stop and Look Ahead
B2	Acceleration
D1	Diagnostic Tools
D2	Interactive Programming
F1	Travel to Fixed Stop
G2	Velocities, Setpoint/Actual-Value Systems, Closed-Loop Control
H2	Output of Auxiliary Functions to PLC
K1	Mode Group, Channel, Program Operation Mode
K2	Axes, Coordinate Systems, Frames, Actual-Value System for Workpiece, External Zero Offset

K4	Communication
N2	EMERGENCY STOP
P1	Transverse Axes
P3	Basic PLC Program
R1	Reference Point Approach
S1	Spindles
V1	Feeds
W1	Tool Compensation

/FB2/

SINUMERIK 840D/840Di/810D(CCU2)/FM-NC
Description of Functions, Extended Functions (Part 2) (04.00 Edition)
including FM-NC: Turning, Stepper Motor
(the various sections are listed below)
Order No.: 6FC5 297-5AC30-0BP2

A4	Digital and Analog NCK I/Os
B3	Several Operator Panels and NCUs
B4	Operation via PG/PC
F3	Remote Diagnostics
H1	Jog with/without Handwheel
K3	Compensations
K5	Mode Groups, Channels, Axis Replacement
L1	FM-NC Local Bus
M1	Kinematic Transformation
M5	Measurement
N3	Software Cams, Position Switching Signals
N4	Punching and Nibbling
P2	Positioning Axes
P5	Oscillation
R2	Rotary Axes
S3	Synchronous Spindles
S5	Synchronized Actions (up to and including SW 3)
S6	Stepper Motor Control
S7	Memory Configuration
T1	Indexing Axes
W3	Tool Change
W4	Grinding

/FB3/

SINUMERIK 840D/840Di/810D(CCU2)/FM-NC
Description of Functions, Special Functions (Part 3) (04.00 Edition)
(the various sections are listed below)
Order No.: 6FC5 297-5AC80-0BP2

F2	3-Axis to 5-Axis Transformation
G1	Gantry Axes
G3	Cycle Times
K6	Contour Tunnel Monitoring
M3	Coupled Motion and Leading Value Coupling
S8	Constant Workpiece Speed for Centerless Grinding
T3	Tangential Control
V2	Preprocessing
W5	3D Tool Radius Compensation
TE1	Clearance Control

TE2 Analog Axis
TE3 Master-Slave for Drives
TE4 Transformation Package Handling
TE5 Setpoint Exchange
TE6 MCS Coupling

/FBA/ SIMODRIVE 611D/SINUMERIK 840D/810D
Description of Functions, Drive Functions (08.99 Edition)
(the various sections are listed below)
Order No.: 6SN1 197-0AA80-0BP5

DB1 Operational Messages/Alarm Reactions
DD1 Diagnostic Functions
DD2 Speed Control Loop
DE1 Extended Drive Functions
DF1 Enable Commands
DG1 Encoder Parameterization
DM1 Calculation of Motor/Power Section Parameters and Controller Data
DS1 Current Control Loop
DÜ1 Monitors/Limitations

/FBAN/ SINUMERIK 840D/SIMODRIVE 611D Digital
Description of Functions
ANA-Module (02.00 Edition)
Order No.: 6SN1 197-0AB80-0BP0

/FBD/ SINUMERIK 840D
Description of Functions **Digitizing** (07.99 Edition)
Order No.: 6FC5 297-4AC50-0BP0

DI1 Start-up
DI2 Scanning with Tactile Sensors (scancad scan)
DI3 Scanning with Lasers (scancad laser)
DI4 Milling Program Generation (scancad mill)

/FBDN/ CAM Integration DNC NT-2000
Description of Functions
System for NC Data Management and Data Distribution (05.00 Edition)
Order No.: 6FC5 297-5AE50-0BP1

/FBFA/ SINUMERIK 840D/810D
Description of Functions
ISO Dialects for SINUMERIK (04.00 Edition)
Order No.: 6FC5 297-5AE10-0BP1

/FBHLA/ SINUMERIK 840D/SIMODRIVE 611 digital
Description of Functions
HLA Module (04.00 Edition)
Order No.: 6SN1 197-0AB60-0BP2

/FBMA/
SINUMERIK 840D/810D
Description of Functions **ManualTurn** (02.00 Edition)
Order No.: 6FC5 297-5AD50-0BP0

/FBO/
SINUMERIK 840D/810D/FM-NC
Description of Functions (03.96 Edition)
Configuring of Operator Interface OP 030
(the various sections are listed below)
Order No.: 6FC5 297-3AC40-0BP0

BA Operator's Guide
EU Development Environment (Configuring Package)
PS Online only: Configuring Syntax (Configuring Package)
PSE Introduction to Configuring of Operator Interface
IK Screen Kit: Software Update and Configuration

/FBP/
SINUMERIK 840D
Description of Functions **C-PLC Programming** (03.96 Edition)
Order No.: 6FC5 297-3AB60-0BP0

/FBR/
SINUMERIK 840D/810D
Description of Functions
SINCOM Computer Link (02.00 Edition)
Order No.: 6FC5 297-5AD60-0BP0

NFL Host Computer Interface
NPL PLC/NCK Interface

/FBSI/
SINUMERIK 840D/SIMODRIVE
Description of Functions **SINUMERIK Safety Integrated** (05.00 Edition)
Order No.: 6FC5 297-5AB80-0BP1

/FBSP/
SINUMERIK 840D/810D
Description of Functions **ShopMill** (06.00 Edition)
Order No.: 6FC5 297-5AD80-0BP1

/FBST/
SIMATIC
FM STEPDRIVE/SIMOSTEP (11.98 Edition)
Description of Functions
Order No.: 6SN1 197-0AA70-0BP3

/FBSY/
SINUMERIK 840D/810D
Description of Functions **Synchronized Actions** (04.00 Edition)
for Wood, Glass, Ceramics, Presses
Order No.: 6FC5 297-5AD40-0BP2

/FBTD/
SINUMERIK 840D/810D
Description of Functions
Tool Information SINTDI with Online Help (04.99 Edition)
Order No.: 6FC5 297-5AE00-0BP0

/FBU/
SIMODRIVE 611 universal
Description of Functions (05.00 Edition)
Closed-Loop Control Component for Speed Control and Positioning
Order No.: 6SN1 197-0AB20-0BP3

/FBW/
SINUMERIK 840D/810D
Description of Functions **Tool Management** (07.00 Edition)
Order No.: 6FC5 297-5AC60-0BP2

/HBI/
SINUMERIK 840Di
Manual (06.00 Edition)
Order No.: 6FC5 297-5AE50-0BP0

/IK/
SINUMERIK 840D/810D/FM-NC
Screen Kit MMC 100/Unit Operator Panel (06.96 Edition)
Description of Functions: Software Update and Configuration
Order No.: 6FC5 297-3EA10-0BP1

/KBU/
SIMODRIVE 611 universal
Short Description (05.00 Edition)
Closed-Loop Control Component for Speed Control
Order No.: 6SN1 197-0AB40-0BP3

/PJLM/
SIMODRIVE
Planning Guide **Linear Motors** (05.00 Edition)
(on request)
ALL General Information about Linear Motors
1FN1 1FN1 Three-Phase AC Linear Motor
1FN3 1FN3 Three-Phase AC Linear Motor
CON Connections
Order No.: 6SN1 197-0AB70-0BP1

/PJM/
SIMODRIVE
Planning Guide **Motors** (09.00 Edition)
Three-Phase AC Motors for Feed and
Main Spindle Drives
Order No.: 6SN1 197-0AA20-0BP4

/PJMS/
SIMODRIVE
Planning Guide **Synchronous Integrated Motor 1FE1** (03.00 Edition)
AC Motors for Main Spindle Drives
Order No.: (on request)

/PJU/
SIMODRIVE 611-A/611-D
Planning Guide **Inverters** (08.98 Edition)
Transistor PWM Inverters for
AC Feed Drives and AC Main Spindle Drives
Order No.: 6SN1 197-0AA00-0BP4

/POS1/
SIMODRIVE POSMO A
User Manual (02.00 Edition)
Distributed Positioning Motor on PROFIBUS DP
Order No.: 6SN2 197-0AA00-0BP1

/POS2/
SIMODRIVE POSMO A
Installation Instructions (enclosed with POSMO A) (12.98 Edition)
Order No.: 462 008 0815 00

/S7H/
SIMATIC S7-300 (10.98 Edition)
– Manual: Assembly, CPU Data (HW)
– Reference Manual: Module Data
Order No.: 6ES7 398-8AA03-8AA0

/S7HT/
SIMATIC S7-300 (03.97 Edition)
Manual: STEP 7, Basic Information, V. 3.1
Order No.: 6ES7 810-4CA02-8AA0

/S7HR/
SIMATIC S7-300 (03.97 Edition)
Manual: STEP 7, Reference Manuals, V. 3.1
Order No.: 6ES7 810-4CA02-8AR0

/S7S/
SIMATIC S7-300 (04.97 Edition)
FM 353 Step Drive Positioning Module
Order in conjunction with Configuring Package

/S7L/
SIMATIC S7-300 (04.97 Edition)
FM 354 Servo Drive Positioning Module
Order in conjunction with Configuring Package

/S7M/
SIMATIC S7-300 (10.99 Edition)
FM 357 Multi-Axis Module for Servo and Stepper Drives
Order in conjunction with Configuring Package

/SHM/
SIMODRIVE 611 (01.98 Edition)
Manual Single-Axis Positioning for **MCU 172A**
Order No.: 6SN 1197-4MA00-0BP0

/SP/
SIMODRIVE 611-A/611-D, SimoPro 3.1
Program for Configuring Machine-Tool Drives
Order No.: 6SC6 111-6PC00-0AA□
Order from: WK Fürth

d) Installation and start-up

/IAA/ **SIMODRIVE 611A**
Installation and Start-Up Guide (04.00 Edition)
Order No.: 6SN 1197-0AA60-0BP5

/IAC/ SINUMERIK 810D
Installation and Start-Up Guide (04.00 Edition)
(incl. description of SIMODRIVE 611D start-up software)
Order No.: 6FC5 297-3AD20-0BP2

/IAD/ SINUMERIK 840D/SIMODRIVE 611D
Installation and Start-Up Guide (04.00 Edition)
(incl. description of SIMODRIVE 611D start-up software)
Order No.: 6FC5 297-5AB10-0BP2

/IAF/ SINUMERIK FM-NC
Installation and Start-Up Guide (04.96 Edition)
Order No.: 6FC5 297-3AB00-0BP0

/IAM/ SINUMERIK 840D/810D
MMC Installation and Start-Up Guide (04.00 Edition)
Order No.: 6FC5 297-5AE20-0BP2

IM1	Start-up functions for the MMC 100.2
IM3	Start-up functions for the MMC 103
IM4	Start-up functions for HMI Advanced (PCU 50)
HE1	Editor help
BE1	Supplement operator interface

Index

Notes

SIEMENS AG
A&D MC BMS
P.O. Box 3180

D–91050 Erlangen

Federal Republic of Germany

(Tel. 0180 / 525 – 8008 / 5009 [Hotline]
Fax +49 (0)9131 / 98 – 1145
email: motioncontrol.docu@erlf.siemens.de)

From:

Name

Company/Dept.

Address:

Telephone: /

Telefax: /

Suggestions

Corrections

For Publication/Manual:

SINUMERIK 840D/840Di/810D

Tool Management

Manufacturer/Service Documentation

Description of Functions

Order No.: 6FC5297–5AC60–0BP2
Edition: 07.00

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