SIEMENS

SINUMERIK 840D/810D

Description of Functions **01.2002 Edition**

SINUMERIK Tool Data Communication SinTDC

Manufacturer Documentation

SIEMENS

SINUMERIK 840D/810D

SINUMERIK Tool Data Communication SinTDC

Description of Functions

Valid for

Control	*Software version*
SINUMERIK 840D	5
SINUMERIK 810D	3

01.02 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 reprint with new Order No.
C .... Revised edition with new status.

Edition	Order No.	Remarks
01.02	6FC5 297-5AF30-0BP0	**A**

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

Edition	Order No.	Remarks
09.02	6FC5 298-6CA00-0BG3	**C**

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

This publication was produced with WinWord V8.0, Designer V7.0 and DokuTool AutWinDoc .

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 we cannot therefore 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 297-5AF30-0BP0
Printed in Germany

Siemens Aktiengesellschaft

Preface

Organization of the documentation

The SINUMERIK documentation is organized on 3 separate levels:

- General Documentation
- User Documentation
- Manufacturer/Service Documentation

Target group

The documentation is intended for manufacturers of machine tools using SINUMERIK 840D and 810D.

Objective

The Installation and Start-Up Guide explains the functionality of the *SinTDC* software and how a PLC program can make use of it.

Search aids

To improve orientation, we have provided the following aids in the Appendix, as well as lists of contents, screens and tables:

1. Abbreviations
2. References
3. Index

Standard scope

This manual is the Installation and Start-Up Guide for *SinTDC*.

Important

This Installation and Start-Up Guide is valid for:
SinTDC V1.0

((Further references))

References: //

Notes

The following special symbols and keywords have been used in this documentation:

Note

This symbol appears in this documentation whenever it is necessary to draw your attention to an important item of information.

Important

This symbol appears in this documentation whenever it is necessary to draw your attention to an important item of information.

Supplementary order data

In this document, you will find the symbol depicted with a reference to an ordering code. Please note that the function described can operate only if the specified option is installed in the control.

Warnings

The following warnings with varying degrees of severity appear in this document:

Danger

This warning notice means that loss of life, severe personal injury or substantial material damage **will** result if the appropriate precautions are not taken.

Warning

This warning notice means that loss of life, severe personal injury or substantial material damage **could** result if the appropriate precautions are not taken.

Caution

This warning notice (with warning triangle) means that a minor personal injury **may** result if the appropriate precautions are not taken.

Caution

This warning notice (without warning triangle) means that a material damage **may** result if the appropriate precautions are not taken.

Notice

This warning notice means that an undesired event or an undesired state **may** result if the appropriate notices are not observed.

Technical comments

Trademarks

IBM® is a registered trademark of the International Business Corporation. MS-DOS® and WINDOWS® are registered trademarks of the Microsoft Corporation.

Effectiveness of alterations

After altering data (e.g. machine data), attention must also be paid to the question as to when the alterations will become effective (e.g. after power ON or IMMEDIATE). For this reason, the point of time is always indicated.

■

Contents

1

1 Introduction

1.1 SinTDC

SinTDC is a software module for Sinumerik 810D/840D which enables you to easily load and unload tools with tool-identification systems either from the PLC or via the standard tool management user-interface. The tools can be loaded from the tool cabinet, the **code carrier** or from the master computer into the NC kernel. In addition, the tools can be unloaded from the NC kernel into the tool cabinet, onto the **code carrier** or into the master computer.

Services are also available for deleting/exporting tool data from, or importing it to the tool list. In addition, tools can be deleted in the NC kernel and the tool data updated on the code carrier.

Different code carrier readers (Moby, Bilz, Balluff) can be conneceted via the PROFIBUS. The data for these devices must be stored in a data block so that they can then be read in by *SinTDC*.

The following code carrier readers on RS-232 are also supported:

- Balluff BIS-C600
- Bar-code reader

There is a description of the tool data held on the code chip in the file wkonvert.txt (Section 3.5 "Structure of the description file").

The tool data not held on the code chip can be assigned with default values using the file deftooldat.txt.

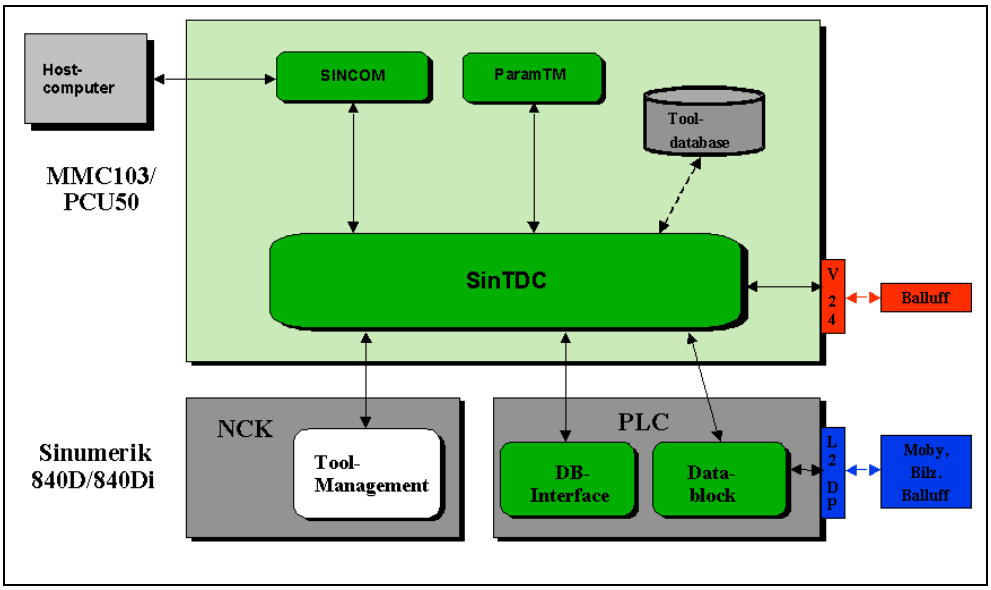

Fig. 1-1 SinTDC - overview

To load and unload the tools, *SinTDC* uses the *tool management server* (*TMS*) which is included in the scope of delivery, which establishes the connection to the host computer, to the tool management in the NC kernal, to the code chip reader and to the tool database (cabinet/catalog). See Fig. 1-1. The connections to the various readers are made via the *tool data server* (*COMTDS*). The communication between the *tool management server* and host computer takes place through *SINCOM* - *SINCOM* is not included in the scope of delivery of *SinTDC*

SinTDC is parameterized using a PLC connection, and the service to be executed is started. The PLC connection is made via the PLC interface in the data block DB19. The PLC interface structure is explained in Chapter 4.

The service to be executed is specified using the command code entered in the interface data block. The meaning of the various command codes is described in Chapter 5. The services belonging to the command codes are explained in Chapter 7. The necessary parameter values for the individual services must also be specified in the interface data block; these are described in Chapter 6.

Chapter 8 describes how to call up and acknowledge a service. Chapter 9 contains examples showing how the parameters for the different services can be assigned.

1.2 PLC interface

The *SinTDC* services are parameterized and started using this interface. In addition, the interface shows the service status and any error messages. The interface is decribed in detail in Chapter 4.

Programming knowledge in Step7 is required to be able to operate the PLC interface.

1.3 Connection to the standard tool management user interface

SinTDC offers a connection to the standard TM user interface (*ParamTM)* as from MMC 103 V5.3.18 (incl. Batch) and HMI Advanced V6.2.10. This allows data from a tool code chip to be read or written with softkey from the standard TM user interface to/from the code chip.

■

2

2 Installation

2.1 System requirements

SinTDC can be installed on the *SIEMENS SINUMERIK 840D* control systems *MMC 103* or *PCU50*. Each of these must fulfil different requirements.

In MMC 103, the interface to the Siemens TM user interface (*ParamTM*) **is only available** from Version 5.3.18 incl. batch .

In PCU50, the interface to the Siemens TM operator interface (*ParamTM*) **is only available** from HMI Advanced 6.02.10 and above.

The values specified for the required hard-disk storage also apply to the memory required for the installation. However, in addition, sufficient memory should remain available for the correct functioning of *Windows*®.

2.1.1 MMC 103

- MMC environment Version 05.03.18 or higher must be installed.
- Approx. 30 MB free hard-disk storage on drive C: must be available.

2.1.2 PCU50

- HMI environment Version 06.00.29 or higher must be installed.
- Approx. 20MB free hard-disk storage on drive E: must be available.
- Approx. 10MB free hard-disk storage on drive F: must be available.

2.1.3 Tool management

SINUMERIK 840D Tool Management must be installed and configured on the control before *SinTDC* is installed. Tool Management is available as an option with*SINUMERIK 840D*.

The interface is activated automatically during installation with the following entry in the file mmc.ini.
[ToolMgmt]
WToolIdSys=**SINTDC**

2.1.4 Master computer services

SinTDC offers services for the loading and unloading of tools using a host computer. For this, you also require the software *SINCOM* (V2 or higher) for the computer link. This is not included in the scope of delivery for *SinTDC* and must therefore be installed separately, in order to be able to use the master computer functionality. The installation of *SINCOM* must also have been completed before *SinTDC* is installed.

2.1.5 Expansion of DB19

If the PLC data block DB19 does not yet have the PLC interface for *SinTDC*, then it must be expanded prior to the first commissioning of *SinTDC*. An STL file is available for this on the installation CD in the directory DB19. The DB19 can be prepared for *SinTDC* using this file.

2.2 Executing the installation

2.2.1 Notes on installation

As most controls do not have a CD-ROM drive, it is recommended that you copy the contents of the installation CD using a *PG* into a separate directory on the hard-disk of the control and start from there. This directory should be deleted after the installation, so that additional memory capacity becomes available again.

If the control is integrated in a network, the installation can also be started from there.

MMC 103

For the installation, start up *Windows*® in Service mode using the Service menu (*Windows*® with MMC drivers), then run the SETUP.EXE on the installation CD.

PCU50

Windows® is started up in Service mode and the contents of the installation CD copied into the *PCU50* directory D:\Install. *Windows*® is then restarted, and during the boot procedure, the setup is automatically started.

The installation of *SinTDC* occurs automatically following the start of the setup. During the installation, it is only necessary to select and configure the code carrier reader being used, and to acknowledge the standard dialog boxes.

The installation path is determined and the version of the installed MMC or HMI environment checked automatically. If none of the software versions specified in Section 2.1 are installed, the setup is ended and a message to this effect issued.

2.2.2 Installation sequence

After selecting the setup language and acknowledging the welcome dialog box, the setup should be executed in the following sequence of steps:

1. Acceptance of licensing agreement
2. User registration
3. Verification of the installed MMC or HMI environment plus version check
4. Selection of the code carrier reader
5. According to reader: Selection of the interface which the reader is connected to, or configuration of the PLC interface (Subsection 2.2.3)
6. Display of installation sequence summary
7. Installation of DCOM95 (only for MMC 103), after which the setup for the shutdown and subsequent restart of *Windows*® is exited. The restart must be carried out via the Service menu (**not** via Start→ShutDown→Restart !!!). After the restart in Service mode, the setup is automatically resumed.
8. Installation of the *Microsoft*® database access components
9. Installation of the*SinTDC* software components and required system files
10. Update of REGIE.INI in the directory \Add_on
11. Update of the installed INI files
12. Exit setup

After the installation has been completed, the entries for the selected reader in the file COMTDS.INI should be checked before *Windows*® is shut down and the MMC or HMI environment started.

! **Important**

On some systems, a system reset may be carried out after the installation of the *Microsoft*® database access components. In this case, *Windows*® must be restarted in Service mode. The setup should then be started using the Parameter /continue; the setup resumes from the point where the installation was broken off.

Call example: C:\Tmp\CDSinTDC\setup.exe /continue

Note

If a setup dialog box is displayed after completion of the installation suggesting a restart of *Windows*®, the restart should be executed. If this dialog box is acknowledged, the setup automatically executes a restart for *Windows*®.

2.2.3 Selection of the code carrier reader

The dialog box for the selection of the code carrier reader to be used offers three different types for selection (see Fig. 2-1). The following types have been tested and approved:

- Moby (Connection type: PROFIBUS)
- Balluff (Connection type: PROFIBUS and RS-232)
- Bilz (Connection type: PROFIBUS)
- Bar-code reader (Connection type: RS-232)

The software components used by *SinTDC* are configured during the installation using the selected reader.

The interface to which the reader is connected is selected after the selection of the code carrier reader, if this is being run on a serial interface. Either one of the two serial interfaces COM1 and COM2 can be selected (see Fig. 2-2).

If the reader is connected to the PLC, for example, with *PROFIBUS*, a dialog box appears for the selection of the number of reader heads to be used (see Fig. 2-3). There is a separate dialog box for each reader head (Fig. 2-4), in which the PLC data block, the offset and the number of bytes can be specified. The number of bytes depends on the settings of the file Wkonvert.txt, which is also installed. When there are several reader heads, the setup suggests settings which ensure that the data areas for each reader head are arranged next to each other in the PLC.

After exiting setup, the parameters for the reader in the file COMTDS.INI should be checked and where necessary corrected. This file can be found in \Add_on\SinTDC\ and should be checked before the MMC or HMI environment is restarted.

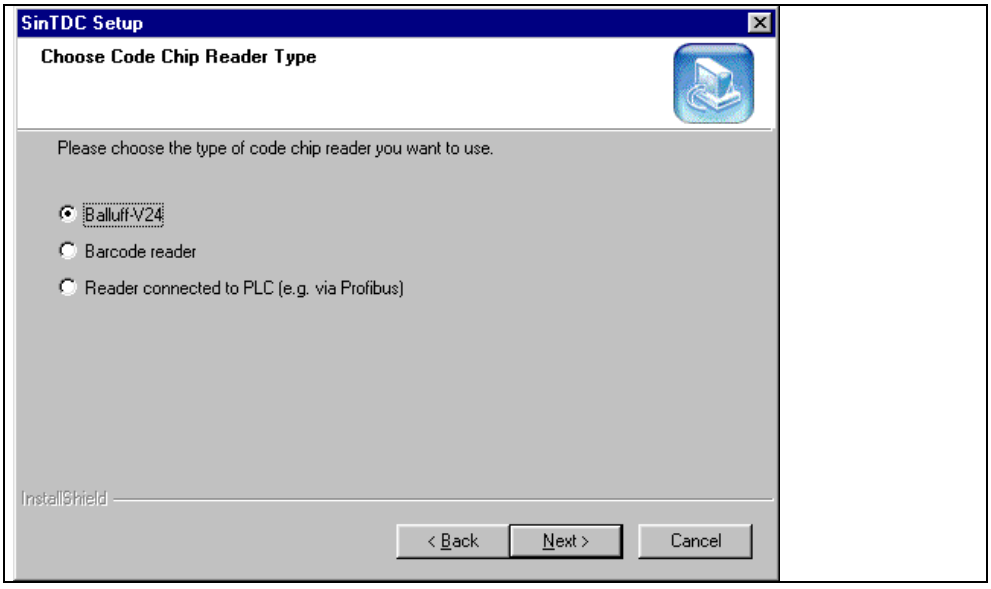

Fig. 2-1 Selection of code carrier reader

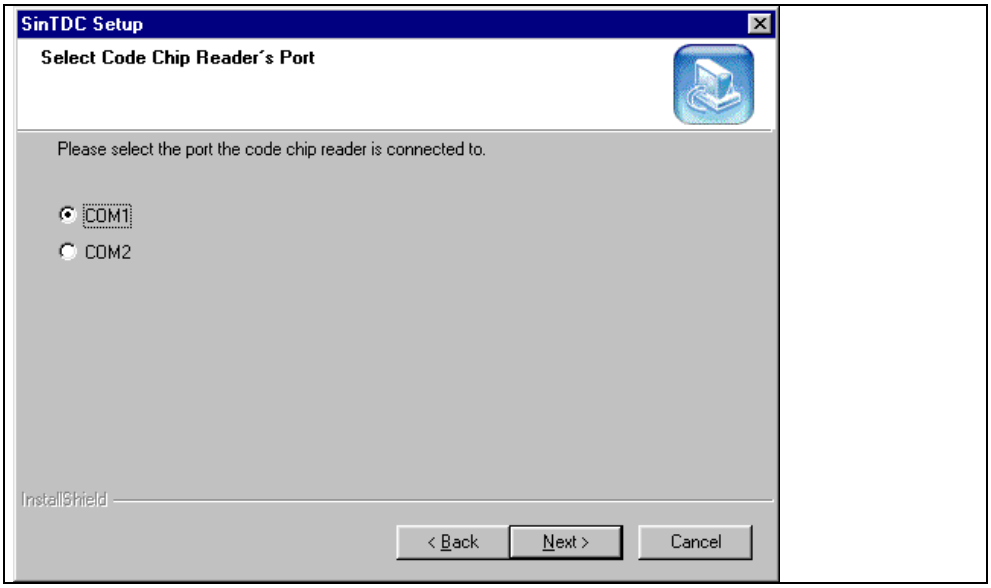

Fig. 2-2 Selection of reader interface

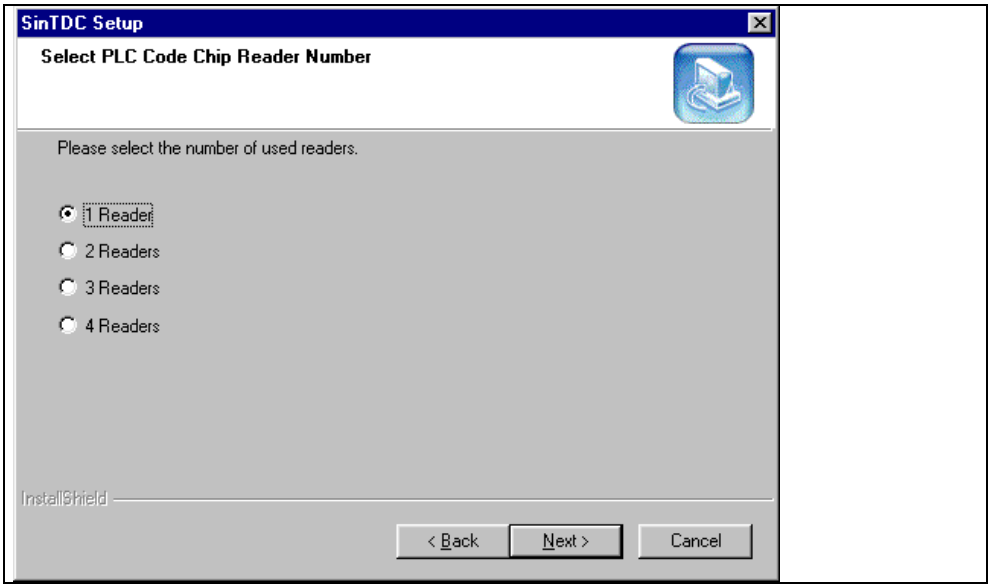

Fig. 2-3 Selection of number of reader (reader heads) used

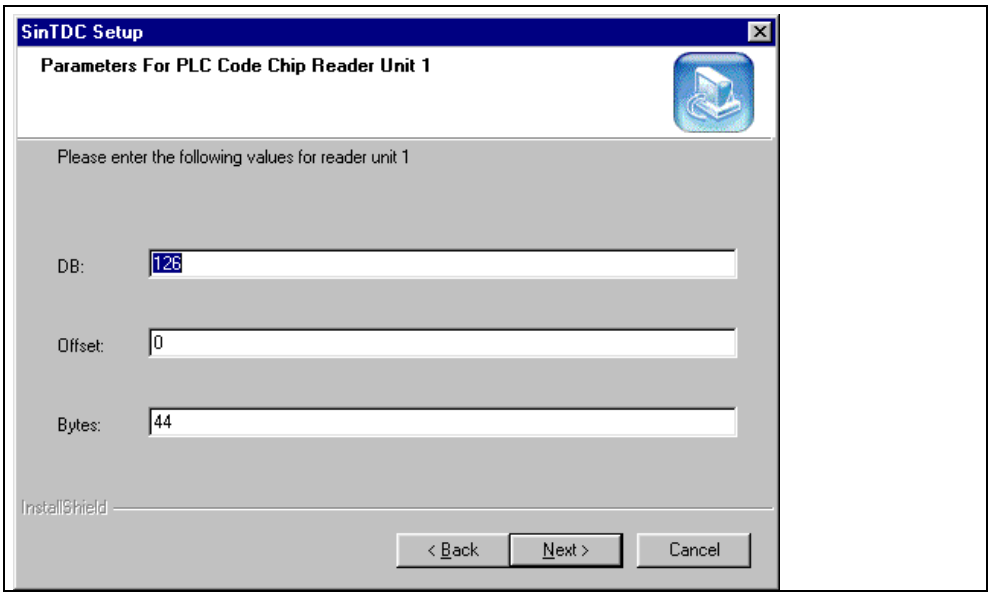

Fig. 2-4 Configuration of PLC reader

■

3

3 Configuration

3.1 SinTDC.INI

SinTDC is configured via the file SinTDC.INI. This file is situated in the directory \Add_on and can be changed with a text editor, for example, *NOTEPAD*. The file is divided into different sections, each one containing different entries. The sections and their entries are described in Table 3-1.

Table 3-1 Structure of the initialization file SinTDC.INI

Section	Item	Meaning
Global	ServiceTopic	Service Topic for the DDE connection to the NCDDE server, for example, ncdde
	NCUName	Name of default NCU, for example, NCU840D
	MMCHome	Root directory of the MMC or HMI software without a final backslash (\ ), for example, C: or F:
	COMTDSHome	Directory where the component COMTDS is stored
	TimeOut	Maximum time in seconds for the execution of a service
	DisablePLC Interface	Deactivation of the PLC interface, for example, when only the ParamTM interface is to be used
	MoveToolPLC	Traverse tool via program to the reader (ParamTM interface)
	MoveToolFirst	Traverse tool first (ParamTM interface)
FileName	Prefix	Prefix of the import or export files, which is added using the transfer parameter *FileIndex* and thus gives the whole file name, for example, Tools_
	Extension	Extension for the import/export files, for example, .txt
	ToolsHome	Directory for the import/export files without final backslash (\ ), for example, C:\ToolData
DataBlocks	Interface	PLC interface data block
	Parameter	Data block of the PLC interface parameters
	InterfaceOffset	Offset of the PLC interface in the data block
	Parameter Offset	Parameter area offset in the data block
	PLCInterface	Data block of the PLC interface of the PLC program for the ParamTM interface
	PLCParameters	Data block of the parameters for the PLC program (ParamTM interface)
	PLCInterface Offset	PLC interface offset of the PLC program for the ParamTM-Interface
	PLCParameter Offset	Offset of the parameters for the PLC program (ParamTM interface)
Logging	LogLevel	Logging mode (0...2) 0: Errors, warnings and messages 1: Trace level 1 2: Trace level 2
	MaxFileSize	Maximum size of log file in KB

If the execution of the services is aborted due to timeout, a longer time period can be specified via *TimeOut*. A timeout may occur, for example, if the upload of a tool is not acknowledged by the user. The default time limit for the execution of a service is 300 seconds.

The items *DisablePLCInterface*, *MoveToolPLC* and *MoveToolFirst* are relevant for the connection to *SINUMERIK* standard software *ParamTM*.
DisablePLCInterface=true can be used to specify that *SinTDC* may only be addressed via *ParamTM*; the PLC interface is then deactivated. The item *MoveToolPLC=true* is used if the tool has to be transported to the reader before reading or writing the data from or on the code chip – in this case, *SinTDC* communicates via the *ParamTM* PLC interface with the PLC program, which then transports the tool. *MoveToolFirst=true* is used to transport the tool to the code chip before reading or writing, otherwise this is carried out after reading.

The names of the import and export files for the import or export of tool data are composed of the entries *Prefix*, *Extension* and the transfer parameter *FileIndex*. After the export, these files can be found in the directory which was specified in *ToolsHome* and are also read from here for the import. This directory must already exist, as it is not created automatically. An export file could, for example, be called Tools_4711.txt, if the details above are used and specified as *FileIndex* "4711".

The area *DataBlocks* does not normally need to be available, as *SinTDC* uses by default the data blocks and offsets described in the following chapters. These are set up in the PLC with the STL file included in delivery (see Section 2.3).

The higher the *LogLevel* entry, the higher the number of messages stored in the log file SinTDC.LOG. This file is used in diagnostics, and contains, for example, notes on faulty transfer parameters. The error file SinTDC.ERR is created parallel to this log file, and contains only error messages.

3.2 TMS.INI

In the file TMS.INI, you can specify whether the tool data should be mixed with data from the tool list or from the NC kernel, if this data was not delivered with the tool. This data is found in the directory \Add_on\SinTDC.

Important

For *SinTDC* to function correctly, only the entries made in Table 3-2 should be changed.

Table 3-2 Structure of the initialization file TMS.INI

Section	Item	Meaning
Uninit Tool Data	MergeDBData	If this item is set at "1", non-initialized values of the tool from the tool list are used.
	MergeNCKData	If this item is set at "1", the non-initialized values of the tool are initialized with data from the NC kernel.

3.3 COMTDS.INI

The file COMTDS.INI contains the parameters for the component which communicates with the reader. It is divided into different sections, which are configured according to the reader used. However, not all sections are required for every reader. This file is also located in the directory \Add_on\SinTDC.

Important

This file does not usually have to be changed since all default settings are made automatically during installation.

3.3.1 Specification of the reader to be used

The reader type to be used is specified in the section GLOBAL. This entry determines the section where the reader parameters are stored. Table 3-3 shows the possible entries for thie section.

Table 3-3 Specification of the reader to be used

Section	Item	Meaning
Global	DEVICE	Specification of the reader to be used Balluff_V24: Balluff RS-232-C Datalogic DL910: Bar-code reader PLCDATA DDE: Reader on the PLC

3.3.2 Balluff RS-232-C Reader

The character sequence which represents the end of the transferred string is specified in the section Balluff_V24. Table 3-4 contains the default value of this item.

Table 3-4 Reader Balluff V24

Section	Item	Meaning
Balluff_RS232C	EOT	End of transferred character string Default: EOT = 0x2F2F

The COM interface to which the reader is connected must also be parameterized for the reader.

3.3.3 Bar-code reader

SinTDC processes pure ASCII code at the RS-232 interface. The conversion of the code types in ASCII must be carried out in the connected bar code device.
The section Datalogic DL910 contains the specifications for the identification of the end of the transferred string (see Table 3-5).

Table 3-5 Bar-code reader

Section	Item	Meaning
Datalogic DL910	EOT	End of transferred character string Default: EOT = 0x0A0D

The COM interface to which the reader is connected must also be parameterized for the reader.

3.3.4 Reader on the PLC

Data about the reader PLC interface is entered in the section PLCDATA-DDE. There is a separate area for the data of each individual reader head in this interface. The following table displays the required parameters.

Table 3-6 Reader on the PLC

Section	Item	Meaning
PLCDATA-DDE	Units	Number of reader heads used Default: Units = 1
	Linktopic1	LinkTopic for the DDE communication Default: Linktopic1 = ncdde|ncu840d
	DataBlock*n*	PLC data block containing the data of reader head *n* (*n* = 1...4)
	Offset*n*	Offset of reader head n data
	MAX_CC_ CAPACITY*n*	Maximum amount of data for the reader head *n* in bytes
	EOT*n*	Identification of transferred string end by reader head *n* Default: EOTn = 0x2F2F

3.3.5 Parameterization of the COM interface

The COM interface being used must be parameterized for the Balluff RS-232-C and the bar-code reader in the section COMPORT. The required settings are shown in Table 3-7, whereby the default values for both readers are specified.

Table 3-7 Parameterization of the COM interface

Section	Item	Meaning
COMPORT	PORT	Index of the COM interface used. 1: COM1 2: COM2 Default: Balluff RS-232-C: PORT = 1 Bar code: PORT = 1

Section	Item	Meaning
	Baud	Baud rate for the transmission of the data. 0: 300 1: 1200 2: 2400 3: 4800 4: 9600 5: 19200 6: 38400 7: 57600 8: 115200 Default: Balluff RS-232-C: Baud = 4 Bar code: Baud = 4
	DataBits	Number of data bits per transferred byte 0: 5 data bits 1: 6 data bits 2: 7 data bits 3: 8 data bits Default: Balluff RS-232-C: DataBits = 3 Bar code: DataBits = 3
	Parity	Specification of parity check 0: none 1: even parity 2: odd parity Default: Balluff RS-232-C: Parity = 1 Bar code: Parity = 0
	StopBits	Number of stop bits per transferred byte 0: 1 stop bit 1: 2 stop bits Default: Balluff RS-232-C: stop bits = 0 Bar code: StopBits = 0
	FlowControl	Data for the data flow check used 0: none 1: Xon/Xoff 2: RTS/CTS Default: Balluff RS-232-C: FlowControl = 0 Bar code: FlowControl = 2

3.4 Structure of the description file. wkonvert.txt

Description file

All data on the code carrier is stored in a specific sequence. This sequence is defined when the code carried system is commissioned. To enable the tool management to read and/or write this data stream, there is a conversion specification in the form of a description file. This description file consists of precisely specified tool and cutting edge dialog data. Only this dialog data can be processed by the tool management. No other data on the code carrier may be assigned to dialog variables, or it will not be processed. However, an OEM application can access this data.

The description file can be created as an ASCII file with a standard text editor. The path and name of the description file is in the file TMS.INI under the item "WkonvertFile".

Tool–dialog data

The tool–dialog data is defined as follows:

Dialog variable	Data type	Designation	Assignment $TC...
T1	String	Tool name, max. 32 places	$TC_TP2
T2	Integer	Duplo number	$TC_TP1
T3	Integer	Number of cutting edges	$P_TOOLND[tNo] tNo=tool number
T4	Integer	Tool size in left half-locations	$TC_TP3
T5	Integer	Tool size in right half-locations	$TC_TP4
T6	Integer	Tool size in upper half-locations	$TC_TP5
T7	Integer	Tool size in lower half-locations	$TC_TP6
T8	String	Magazine location type	$TC_TP7*
T9	Integer	Tool status	$TC_TP8
T10	Integer	Type of tool watchdog	$TC_TP9
T11	Integer	Type of tool search	$TC_TP11

* The character string stored here is an internal MMC location type which is assigned to the value in $TC_TP7. This text is specified using the commissioning TM and is stored in the database.

Cutting edge dialog data

Dialog variable	Data type	Designation	Assignment $TC...
C1	Integer	Sub-type	$TC_DP1
C4	Integer	Cutting edge position	$TC_DP2
		Geometry–Tool length compensation	
C5	Double	Length 1	$TC_DP3
C6	Double	Length 2	$TC_DP4
C7	Double	Length 3	$TC_DP5
		Geometry–Tool radius compensation	
C8	Double	Length 1	$TC_DP8
C9	Double	Length 2	$TC_DP9
C10	Double	Radius 1	$TC_DP6
C11	Double	Radius 2	$TC_DP7
C12	Double	Angle 1	$TC_DP10
C13	Double	Angle 2	$TC_DP11
		Wear–Tool length compensation	
C14	Double	Length 1	$TC_DP12
C15	Double	Length 2	$TC_DP13
C16	Double	Length 3	$TC_DP14
		Wear–Tool radius compensation	
C17	Double	Length 1	$TC_DP17
C18	Double	Length 2	$TC_DP18
C19	Double	Radius 1	$TC_DP15
C20	Double	Radius 2	$TC_DP16
C21	Double	Angle 1	$TC_DP19
C22	Double	Angle 2	$TC_DP20
		Base/adapter dimension– Tool length compensation	
C23	Double	Base length 1	$TC_DP21
C24	Double	Base length 2	$TC_DP22
C25	Double	Base length 3	$TC_DP23
C26	Double	Tool clearance angle	$TC_DP24
C27	Integer	Use of tool inverse	$TC_DP25
C28	Integer	Cutting edge number - for addressing of variables	-
C29*	Integer	Down-time in minutes	$TC_MOP2
C30*	Integer	Down-time prewarning limit in minutes	$TC_MOP1
C31	Integer	Number of pieces still to be produced	$TC_MOP4
C32	Integer	Prewarning limit for number of pieces still to be produced	$TC_MOP3
C33	Double	Target service life in minutes	$TC_MOP11
C34	Integer	Target number of pieces	$TC_MOP13
C35	Double	Wear prewarning limit	$TC_MOP5
C36	Double	Wear	$TC_MOP6
C37	Double	Target wear	$TC_MOP15
C38*	Double	Down-time in minutes	$TC_MOP2
C39*	Double	Down-time prewarning limit in minutes	$TC_MOP1

The dialog variables C2 and C3 are managed internally only.
* see note below

Note

C38 and C39 can only be used alternatively to C29 und C30.

Tool–user parameters and the new watchdog parameters are also now used for code carriers. The following new configuration variables are available for the file wkonvert.txt:

A1 - A10: User–tool data $TC_TPCx[t])
U1 - U10: User–cutting edge data (see $TC_DPCx[t,d])
S1 - S10: User-watchdog data (see $TC_MOPCx[t,d])

Data types

The dialog variable data types are defined as follows:
- Integer: Value range -32768 to32767
- Double: Double-precision floating point
- String: Character string from ASCII characters

Keywords

The code carrier data is assigned to the dialog data via the code carrier description file. As an ASCII file, the data can be created and/or modifed with standard editors.The code carrier description file is structured in line form, whereby each line is introduced with one of the following **keywords**:

Inverted comma

The **'** (inverted comma) marks the start of a comment. All subsequent characters to line ending are skipped.
Example:

```
' This is a comment
```

Note

An inverted comma is used to mark the beginning of a comment only in the code carrier description file. Otherwise, the beginning of a comment is introduced with a semi-colon.

Datalen

DATALEN=CONST | VARIABLE 0x<separator>
The following data has either a constant (**CONST**) or variable (**VARIABLE**) data length. Data with a variable length is terminated with 0x<separator>.

Example:
```
DATALEN=VARIABLE 0x0A  ' variable data length, separator LF
```

Item

Item<n>**=<**line**>**
<n> := current number of code carrier data, increasing consecutively from 1
<line> := <(max.) length in bytes> <code carrier data format> <dialog variable>
<code carrier data format> :
<dialog variable> : Code carrier assignment– to dialog data

If a user keyword has been defined immediately before item <n>, then the <dialog variable> has the value <keyword>

Conversion specification for code carrier date<n>

Example:
```
Item1   32   ASCII  T3 ' Relocate Tool identifier to/from
                       ' Tool dialog data 3
```

BItem

BItem<n>**=<**line**>**
<n> := current number of code carrier data within block <i>, increasing consecutively from 1
<line> := analog item<n>

Conversion specification for code carrier data <n> within a block. If tool–dialog data T<n> is assigned to the code carrier, then the first value of the code carrier data in the block is assigned to the dialog data.

Example:
```
BItem1   1      BCD    C1
```

Block

Block<n> <repeat instrucion>
<n> := current number of block, increasing consecutively from 1
<repeat instrucion> := * Item<n>

There follows a data block **BItem**<n> (to keyword **End_Block**<n>), which is stored / will be stored onto the code carrier according to the <repeat instrucion>.

Note

For **Block**<n> ***Item**<n>, Item<n> must be defined before Block<n>.

Example:
```
Block1 * Item6  ' Repeat Block1 according to the value of Item6

End_Block<n>
```

End_Block

End identification 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 • Least significant byte to most significant address (opposite to Intel format)
BCD	• Binary coded decimal number (if applicable, with sign and decimal point) • Non-relevant decades are pre-assigned a zero in the left position
BCD_SIN850	BCD with following specifications (SINUMERIK 850): • Each BCD number occupies 12 decades • Sign always in the 3rd decade • 0 = positive / 1 = negative • Non-relevant decades are pre-assigned a zero in the left-justified position • Value range max. 8 places with decimal point, max. 9 places without decimal point

Assignment between code carrier data and dialog data

The conversion specification for **Item<n>** or **BItem<n>** contains, as well as other data, the assignment to no/one/several dialog variable(s), where applicable with a conversion specification, which is described in more detail in this section.

The general conversion specification for **Item<n>** or **BItem<n>** is:
(B)Item<n>**=<**line**>**
<n> := current number of code carrier data, increasing consecutively
<line> := <(max.) length in bytes><code carrier data format><dialog variable>

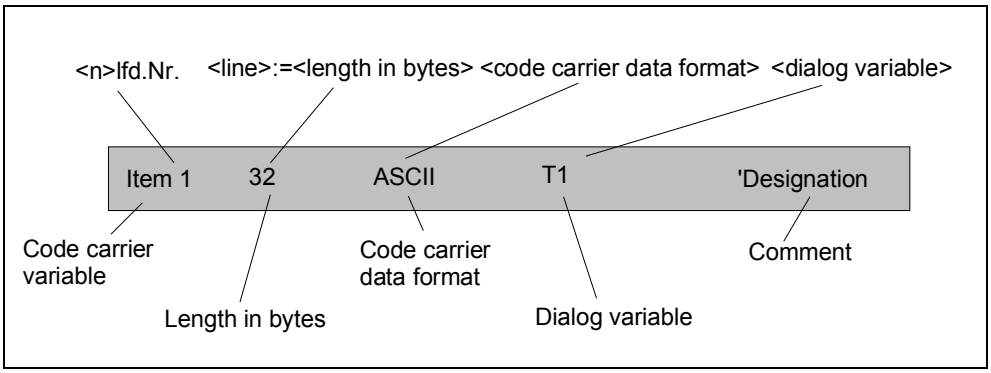

Fig. 3-1 Conversion specification

Dialog variable

<**dialog variable**> := <dvar1>[=(<uv>)] [, <dvar2>[=(<uv>)] [,<dvar3>
[&<dvar4>]=(<uv>)] [, <dvarN>[=(<uv>)]
<dvar> := T<index> | C<index> | – Fig. 3-1

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 valid for <dvar1> and <dvar2>
uv := <Tetn>
Tetn := nth tetrade in byte sequence
Byte1, = Tet1 and Tet2
Byte2, = Tet3 and Tet4

Division of the tetrades from code carrier variables (in BCD format) onto dialog variable.

Example:
```
T5=(Tet1), T6=(Tet2), T7=(Tet3), T8=(Tet4)
```

If the code carrier variable has, for example, the value 0x1234, then the dialog variable T5 is given the value 1 and the dialog variable T8 the value 4.

3.4.1 Examples for decription files

Name of the description or conversion file

The file name must be entered into the ...\user\mmc.ini at WToolIdSysKonv = **wkonvert.txt**.

Standard for SINUMERIK 840D with SinTDC

The name of the file after installation is **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	2	BCD	T9	' Status
Item6	1	BCD	T3	' Number of cutting edges
Item7	1	BCD	T10	' Type of tool watchdog
Item8	1	BCD	T11	' Type of tool search
'Cutting edge data Block1 * Item6				
Bitem1	2	BCD	C1	' Sub-type, type
Bitem2	1	BCD	C4	' Cutting edge position
' Tool length comp.				
Bitem3	4	BCD	C5	' Length 1
BItem4	4	BCD	C6	' Length 2
BItem5	4	BCD	C7	' Length 3
'Tool radius compensation				
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				
BItem12	4	BCD	C14	' Length 1
BItem13	4	BCD	C15	' Length 2
BItem14	4	BCD	C16	' Length 3
'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

'Tool 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	' Tool clearance angle
BItem25	1	BCD	C27	' Use of tool inverse
BItem26	2	BCD	C29	' Service life in minutes
BItem27	2	BCD	C30	' Service life prewarning limit
BItem28	2	BCD	C33	' Target service life in minutes
BItem29	2	BCD	C31	' Number of pieces
BItem30	2	BCD	C32	' Prewarning limit quantity
BItem31	2	BCD	C32	' Target quantity
End_Block1				

Example with tool and cutting edge OEM data

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	2	BCD	T9	' Status
Item6	1	BCD	T3	' Number of cutting edges
Item7	1	BCD	T10	' Type of tool–watchdog
Item8	1	BCD	T11	' Type of tool–search
Item9	**4**	**BCD**	**A1**	**'Tool OEM data 1**
'Cutting edge data Block1 * Item6				
Bitem1	2	BCD	C1	' Sub-type, type
Bitem2	1	BCD	C4	' Cutting edge position
' Tool length comp.				
Bitem3	4	BCD	C5	' Length 1
BItem4	4	BCD	C6	' Length 2
BItem5	4	BCD	C7	' Length 3
'Tool radius compensation				
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				
BItem12	4	BCD	C14	' Length 1
BItem13	4	BCD	C15	' Length 2
BItem14	4	BCD	C16	' Length 3

'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
'Tool 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	' Tool clearance angle
BItem25	1	BCD	C27	' Use of tool inverse
BItem26	2	BCD	C29	' Service life in minutes
BItem27	2	BCD	C30	' Service life prewarning limit
BItem28	2	BCD	C33	' Target service life in minutes
BItem29	2	BCD	C31	' Number of pieces
BItem30	2	BCD	C32	' Prewarning limit quantity
BItem31	2	BCD	C32	' Target quantity
BItem32	**4**	**BCD**	**U1**	**' Cutting edge OEM data 1**
End_Block1				

■

4

4 PLC Interface in DB19

4.1 Structure

The software *SinTDC* is addressed via a PLC interface, situated in the data block DB19 and supplied with data (see Fig. 4-1).

A further area contains the transfer and return parameters for the various services.

There is also a separate PLC interface for the *SinTDC ParamTM* interface, which is used to activate a PLC program. The parameters for this PLC program and the status bits are both also situated in this area.

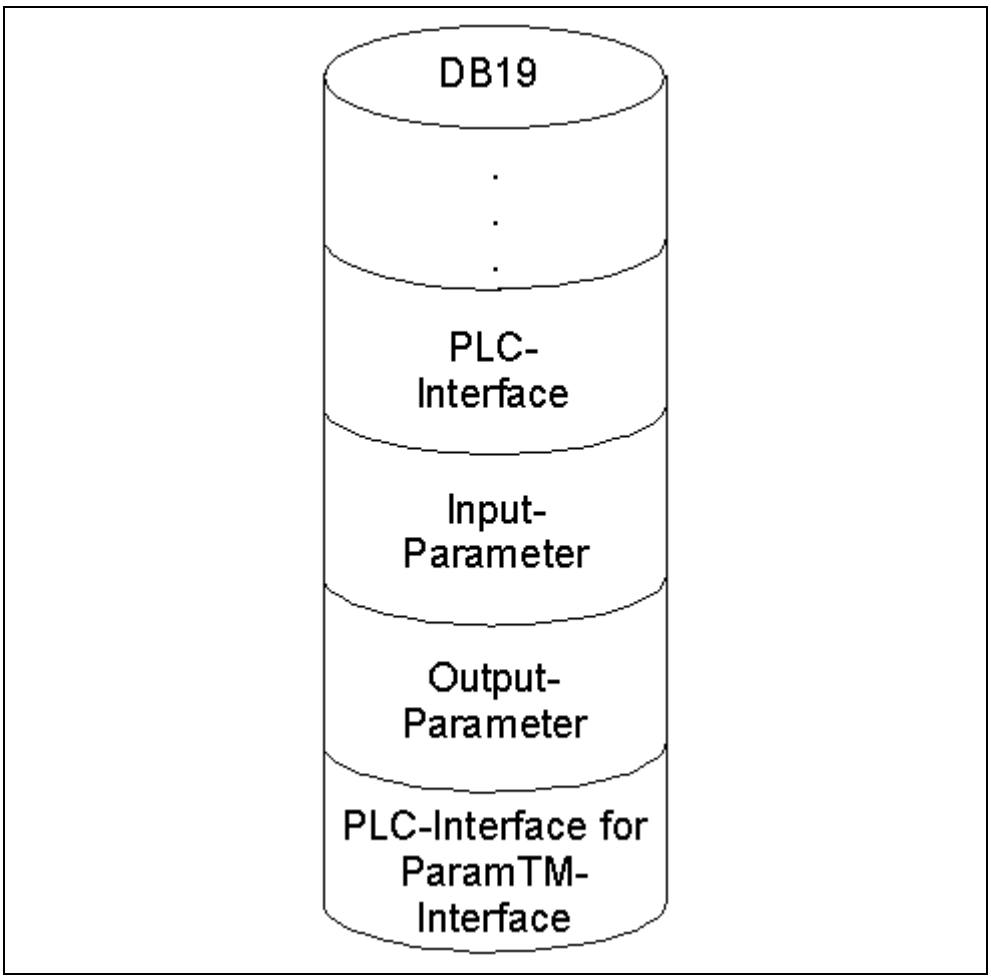

Fig. 4-1 Structure of the PLC interface

4.2 Assignment of the SinTDC PLC interface

Table 4-1 Assignment of the SinTDC PLC interface

PLC data		Identifier	Type	Value range	Meaning
DBX	250.0	Request	BOOL	TRUE, FALSE	Job request
DBB	251	C_Code	BYTE	0-127	Command code
DBX	252.0	Done	BOOL	TRUE, FALSE	Job successfully completed
DBX	252.1	Error	BOOL	TRUE, FALSE	Job completed with error
DBX	252.2	Active	BOOL	TRUE, FALSE	Job is being executed
DBX	252.3	Reserve	BOOL	TRUE, FALSE	
DBX	252.4	Spare	BOOL	TRUE, FALSE	
DBX	252.5	Spare	BOOL	TRUE, FALSE	
DBX	252.6	Spare	BOOL	TRUE, FALSE	
DBX	252.7	Spare	BOOL	TRUE, FALSE	
DBB	253	Spare	BYTE	-128...127	
DBW	254	Error_Code	INT	-32768...32767	Error code

While *SinTDC* is being ramped up, the request bit is set at "0", to ensure that no unrequested service is executed if, for example, the control has been switched off during the execution of a service.

If the request bit has already been set, it is reset and the error code 0xFFFF written in the *Error_Code* word in the PLC.

4.3 Transfer and return parameters of

Table4-2 Assignment of the PLC interface for the transfer parameters

PLC data		Identifier	Type	Value range	Meaning
Transfer parameters					
DBW	140	Subtype	INT	-1...32000	Tool subtype
DBW	142	Duplo	INT	-1...32000	Tool duplo number
DBB	144	Ident	String	32 characters	Tool identifier
DBW	178	TNumber	INT	-1...32000	T-number of the tool
DBW	180	MagazinePlace	INT	-1...32000	Magazine location of tool
DBW	182	MagazinePlaceType	INT	-1...32000	T-number of the tool
DBB	184	ToolSize_Upper	BYTE	1...127	Tool size in upper half-locations
DBB	185	ToolSize_Down	BYTE	1...127	Tool size in lower half-locations
DBB	186	ToolSize_Left	BYTE	1...127	Tool size in left half-locations
DBB	187	ToolSize_Right	BYTE	1...127	Tool size in right half-locations
DBB	188	Channel	BYTE	1...127	Tool channel number
DBW	190	Magazine	INT	0...3200	Tool magazine number
DBB	192	Unit	BYTE	1...127	Tool code carrier number
DBB	193	NCU_Index	BYTE	32 characters	Name of NCU
DBW	194	FileIndex	INT	0...3200	File index for import/export
DBX	196.0	Active Enable	BOOL	TRUE, FALSE	Concealment of the NC kernel status bit "Tool active"
DBX	196.1	Override	BOOL	TRUE, FALSE	Overwrite file if available?
DBX	196.2	LoadTool	BOOL	TRUE, FALSE	Load tool after creating
DBX	196.3	DeleteTool	BOOL	TRUE, FALSE	Delete tool after unloading

Table 4-3 Assignment of the PLC interface for the return parameters

PLC data	Identifier	Type	Value range	Meaning
Return parameters				
DBW 198	Subtype	INT	-1...32000	Tool subtype
DBW 200	Duplo	INT	-1...32000	Tool duplo number
DBB 202	Ident	String	32 characters	Tool identifier
DBW 236	TNumber	INT	-1...32000	T-number of the tool
DBW 238	ToolState	INT	-1...32000	Tool status
DBW 240	MagazinePlaceType	INT	-1...32000	Tool size
DBB 242	ToolSize_Upper	BYTE	1...127	Tool size in upper half-locations
DBB 243	ToolSize_Down	BYTE	1...127	Tool size in lower half-locations
DBB 244	ToolSize_Left	BYTE	1...127	Tool size in left half-locations
DBB 245	ToolSize_Right	BYTE	1...127	Tool size in right half-locations
DBB 246	Magazine	INT	0...3200	Tool magazine number
DBB 248	MagazinePlace	INT	0...3200	Magazine location of tool

The permissible parameter values for the individual services can vary and are therefore specified in Chapter 7 for each service separately.

4.4 PLC program interface (ParamTM-Interface)

Table 4-4 Assignment of interface for the PLC program

PLC data	Identifier	Type	Value range	Meaning
Interface				
DBX 256.0	Req	BOOL	TRUE, FALSE	Request to PLC program
DBB 257	Reserve	BYTE	-128...127	
DBX 258.0	Done	BOOL	TRUE, FALSE	Job successfully completed
DBX 258.1	Error	BOOL	TRUE, FALSE	Job completed with error
DBX 258.2	Active	BOOL	TRUE, FALSE	Job is being executed
DBX 258.3	Reserve	BOOL	TRUE, FALSE	
DBX 258.4	Reserve	BOOL	TRUE, FALSE	
DBX 258.5	Reserve	BOOL	TRUE, FALSE	
DBX 258.6	Reserve	BOOL	TRUE, FALSE	
DBX 258.7	Reserve	BOOL	TRUE, FALSE	
DBB 259	Reserve	BYTE	-128...127	
DBW 260	Error_Code	INT	-32768...32767	Error numbers
Transfer parameters				
DBW 262	Magazine	INT	0...3200	Tool magazine number
DBW 264	Magazine Place	INT	0...3200	Magazine location of tool
DBW 266	TNumber	INT	0...3200	T-number of the tool

■

5

5 Command Codes

5.1 Description

Table 5-1 Description of command codes (C code)

C code	Description
0	Loading from code carrier
1	Unloading to code carrier
2	Loading from tool cabinet
3	Unloading to tool cabinet
4	Loading from host computer
5	Unloading to host computer
6	Unloading all tools to tool cabinet
7	Loading tool
8	Unloading tool
9	Searching for empty location
10	Deleting tool in the NCK
11	Deleting tool in database
12	Exporting tool data
13	Importing tool data
14	Updating code carrier
15	Reading data from code carrier
16	Storing tool data in the NCK

■

6

6 Parameters

6.1 Description

Table 6-1 Description of parameters

Parameters	Description
ActiveEnable	Transfers tool status bit from the database into the NC kernel (TRUE) / conceals the status bit during transfer (FALSE)
Channel	Channel number
DeleteTool	Delete tool after unloading (TRUE) / do not delete (FALSE)
Duplo	Duplo number
FileIndex	Index of file from which or into which the import/export is to take place. The file name, which is extended by the index, can be preset in the INI-file.
Ident	Tool name
LoadTool	Load tool after reading (TRUE) / do not load (FALSE)
Magazine	Magazine number
MagazinePlace	Tool place
MagazinePlaceType	Tool place type
NCU_Index	Index of the NCU to be addressed for this job. The following values are permissible: 0 → NCU name from SinTDC.INI 1 → NCU1 2 → NCU2 >2 → MachineSwitch The names for NCU1 and NCU2 are determined from the file NETNAMES.INI.
Override	Overwrite export file (TRUE) / expand (FALSE)
Subtype	Subtype or type
TNumber	T-number
ToolSize_Down	Tool size in lower half-locations
ToolSize_Left	Tool size in left half-locations
ToolSize_Right	Tool size in right half-locations
ToolSize_Upper	Tool size in upper half-locations
ToolState	Tool status
Unit	Code carrier unit (reader head) which is to be read from. According to reader 1..4

■

7

7 Services

7.1 Loading from code carrier

Reads the tool data from the code carrier, whose unit is specified with *Unit*. Then the tool is created if it is not already available in the NC kernel. It is also loaded if *LoadTool* has been set. The index of the NCU, the channel number (*Channel*) and the magazine number (*Magazine*) must be specified.

If parameter *Magazine* is set to "0", the tool is loaded into the magazine in which an empty location is found.

If the value for *MagazinePlace* > 0, an attempt is made to load the tool into this magazine location.

Table 7-1 Parameters for "Loading from code carrier"

PLC data	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 180	MagazinePlace	INT	0...3200	Magazine location
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 192	Unit	BYTE	1...127	Number of reader head
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
DBX 196.2	LoadTool	BOOL	TRUE, FALSE	Load tool after creating?
Return parameters				
DBW 198	Subtype	INT	1...3200	Sub-type
DBW 200	Duplo	INT	0...3200	Duplo number
DBB 202	Ident	String	32 characters	Tool identifier
DBW 236	TNumber	INT	0...3200	T-number
DBW 238	ToolState	INT	0...3200	Tool status
DBW 240	MagazinePlace Type	INT	0...3200	Magazine location type
DBB 242	ToolSize_ Upper	BYTE	1...127	Tool size in upper half-locations
DBB 243	ToolSize_Down	BYTE	1...127	Tool size in lower half-locations
DBB 244	ToolSize_Left	BYTE	1...127	Tool size in left half-locations
DBB 245	ToolSize_Right	BYTE	1...127	Tool size in right half-locations
DBW 246	Magazine	INT	0...3200	Magazine number
DBW 248	MagazinePlace	INT	0...3200	Magazine location

7.2 Unloading into code carrier

Unloads the tool with the identifier specified in *Ident*. The magazine number, the channel number, the duplo number and the index of the NCU must be specified. If *DeleteTool* is set, the tool is deleted from the NC kernel.

If the parameter *Magazine* is set to "0", the tool is unloaded from an arbitrary magazine.

If the parameters *Magazine* and *MagazinePlace* > 0, an attempt will be made to unload the tool which is on the location. The parameters *Duplo* and *Ident* are then not taken into account.

Table 7-2 Parameters for "Unloading into code carrier"

PLC data	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 142	Duplo	INT	1...3200	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 180	MagazinePlace	INT	0...3200	Magazine location
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 192	Unit	BYTE	1...127	Number of the reader head
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
DBX 196.3	DeleteTool	BOOL	TRUE, FALSE	Delete tool after unloading?
Return parameters				
None				

7.3 Loading from tool cabinet

This service is mainly used in combination with the "Reading from code carrier" service (Section 7.16), if only partial information is contained on the code chip.

This is usually always the case if low-price read-only code chips are used.

Reads the tool data from the database and creates the tool in the NC kernel if it does not already exist. The parameters *Subtype*, *Duplo* and *Ident* must not necessarily be assigned. If *Subtype* and *Duplo* are not assigned, they must be specified with "-1". An empty string is specified with *Ident*, if this should not be assigned.

If *Subtype*, *Ident* or *Duplo* have not been uniquely assigned, several tools can be found in the database.

If the T-number is specified with "0", the T-number is assigned by the NC kernel. If a T-number greater than "0" is specified, an attempt is made to create and load the tool with this number. If this T-number cannot be used, it is assigned by the NC kernel.

If several selected tools are to be loaded by this service and the T-number is specified as greater than "0", then the desired tool numbers are incremented continuously internally. However no check is made here whether these incremented tool numbers are free in the NC kernel.

The parameter *TNumber* has no effect if the tool already exists in the tool list.

Via the *ActiveEnable* parameter, the handling of the tool status bit "Tool active" is specified. If this parameter is set (TRUE), this status bit is transferred to the NC kernel from the database unaltered. If however *ActiveEnable* is FALSE, this status bit is concealed when reading from the database and therefore entered in the NC kernel as not being set.

If the magazine number is specified as "0", the tool or tools are loaded into the magazine where an empty location is found, if *LoadTool* is set.

If the value for *MagazinePlace* > 0, an attempt is made to load the tool into this magazine location.

The parameters of this service are shown in Table 7-3. Depending on the parameter value, several tools can be loaded from the database to the NC kernel at the same time. Therefore an overview of possible parameter values is shown in Table 7-4.

Table 7-3 Parameters for "Loading from tool cabinet"

PLC data	**Identifier**	**Type**	**Value range**	**Meaning**
Transfer parameters				
DBW 140	Subtype	INT	-1, 1...32000	Sub-type
DBW 142	Duplo	INT	-1...32000	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 178	TNumber	INT	0...3200	T-number
DBW 180	MagazinePlace	INT	0...3200	Magazine location
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
DBX 196.0	ActiveEnable	BOOL	TRUE, FALSE	Concealment of NCK status bit "Tool active"?
DBX 196.2	LoadTool	BOOL	TRUE, FALSE	Loading tool after creating?
Return parameters				
DBW 198	Subtype	INT	1...3200	Sub-type
DBW 200	Duplo	INT	0...3200	Duplo number
DBB 202	Ident	String	32 characters	Tool identifier
DBW 236	TNumber	INT	0...3200	T-number
DBW 238	ToolState	INT	0...3200	Tool status

DBW	240	MagazinePlace Type	INT	0...3200	Magazine location type
DBB	242	ToolSize_ Upper	BYTE	1...127	Tool size in upper half-locations
DBB	243	ToolSize_Down	BYTE	1...127	Tool size in lower half-locations
DBB	244	ToolSize_Left	BYTE	1...127	Tool size in left half-locations
DBB	245	ToolSize_Right	BYTE	1...127	Tool size in right half-locations
DBW	246	Magazine	INT	0...3200	Magazine number
DBW	248	MagazinePlace	INT	0...3200	Magazine location

Table 7-4 Parameter combinations for "Loading from tool cabinet"

PLC date		**Identifier**	**Value**	**Meaning**
DBW	140	Subtype	-1	Several tools can be found in the database.
			>0	Tool(s) with this subtype is(are) loaded.
DBW	142	Duplo	-1	Several tools can be found in the database.
			>-1	Tool(s) with this duplo number is(are) loaded.
DBW	144	Ident	""	Several tools can be found in the database.
			Name	Tool(s) with this name is(are) loaded.
DBW	178	TNumber	0	The NCK assigns a free T-number.
			>0	An attempt is made to create and load the tool with this T-number.
DBW	190	Magazine	0	The tool(s) is(are) loaded into an arbitrary magazine.
			>0	The tool(s) is(are) loaded into the specified magazine.

7.4 Unloading into tool cabinet

This service is mainly used in combination with the "Reading from code carrier" service (Section 7.16), if only partial information is contained on the code chip.

This is usually always the case if low-price read-only code chips are used.

Unloads the tool identified with *Ident* from the NCK into the tool cabinet. The parameters *NCU*, *Channel*, *Ident*, *Duplo* and *Magazine* must be assigned.

If the magazine number is specified as "0", the tool is unloaded from an arbitrary magazine.

If the value for *MagazinePlace* > 0, an attempt is made to unload the tool from this magazine location.

After the successful unload, the tool is deleted in the NCK if *DeleteTool* has been set.

Table 7-5 Parameters for "Unloading into tool cabinet"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 142	Duplo	INT	1...3200	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 180	MagazinePlace	INT	0...3200	Magazine location
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
DBX 196.3	DeleteTool	BOOL	TRUE, FALSE	Delete tool after unloading?
Return parameters				
None				

7.5 Loading from host computer

Requests the data for the tool identified with *Ident* and creates it in the NC kernel, if it does not already exist. The parameters *NCU_Index*, *Channel*, *Ident*, *Duplo* and *Magazine* must be assigned.

If the magazine number is specified as "0", the tool is loaded into the magazine where an empty location is found, if *LoadTool* is set.

If the value for *MagazinePlace* > 0, an attempt is made to load the tool into this magazine location.

Table 7-6 Parameters for "Loading from host computer"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 142	Duplo	INT	1...3200	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 180	MagazinePlace	INT	0...3200	Magazine location
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
DBX 196.2	LoadTool	BOOL	TRUE, FALSE	Load tool after creating?
Return parameters				
DBW 198	Subtype	INT	1...3200	Sub-type
DBW 200	Duplo	INT	0...3200	Duplo number
DBB 202	Ident	String	32 characters	Tool identifier
DBW 236	TNumber	INT	0...3200	T-number
DBW 238	ToolState	INT	0...3200	Tool status
DBW 240	MagazinePlace Type	INT	0...3200	Magazine location type
DBB 242	ToolSize_ Upper	BYTE	1...127	Tool size in upper half-locations
DBB 243	ToolSize_Down	BYTE	1...127	Tool size in lower half-locations
DBB 244	ToolSize_Left	BYTE	1...127	Tool size in left half-locations
DBB 245	ToolSize_Right	BYTE	1...127	Tool size in right half-locations
DBW 246	Magazine	INT	0...3200	Magazine number
DBW 248	MagazinePlace	INT	0...3200	Magazine location

7.6 Unloading into host computer

Unloads the tool identified with *Ident* from the NC kernel and transfers the data to the host computer. The parameters *NCU_Index*, *Channel*, *Ident*, *Duplo* and *Magazine* must be assigned.

If "0" is specified for the magazine number, the tool is unloaded from an arbitrary magazine and then deleted, if *DeleteTool*is set.

If the value for *MagazinePlace* > 0, an attempt is made to unload the tool from this magazine location.

Table 7-7 Parameters for "Unloading into host computer"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 142	Duplo	INT	1...3200	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 180	MagazinePlace	INT	0...3200	Magazine location
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
DBX 196.3	DeleteTool	BOOL	TRUE, FALSE	Delete tool after unloading?
Return parameters				
None				

7.7 Unloading of all tools in tool cabinet

With this service, the user can empty his magazine simply and back up the tool data in the tool cabinet.

The data of all the tools loaded in the specified magazine are read from the NC kernel and written to the database. Then the tools are unloaded from the magazine and deleted, if *DeleteTool* is set. The parameters *NCU_Index*, *Channel* and *Magazine* must be assigned.

If "0" is set in magazine, the loaded tools of all magazines existing in the NC kernel are unloaded, whereby the spindle magazine is also taken into account. The unloading sequence corresponds to the sequence in which the tools were loaded.

Table 7-8 Parameters for "Unloading of all tools in tool cabinet"

PLC date	**Identifier**	**Type**	**Value range**	**Meaning**
Transfer parameters				
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
DBX 196.3	DeleteTool	BOOL	TRUE, FALSE	Delete tool after unloading?
Return parameters				
None				

7.8 Loading of a tool

A tool existing in the tool list is loaded into a tool location.

The tool specified with *Channel*, *Duplo* and *Ident* is loaded into the tool magazine specified in *Magazine*. The NCU is specified with *NCU_Index*.

If "0" is specified as the magazine number, the next free magazine location that can be used for the tool is searched for.

If the value for *MagazinePlace* > 0, an attempt is made to load the tool into this magazine location.

Table 7-9 Parameters for "Loading of a tool"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 142	Duplo	INT	1...3200	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 180	MagazinePlace	INT	0...3200	Magazine location
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
Return parameters				
None				

7.9 Unloading of a tool

The tool specified with *Channel*, *Duplo* and *Ident* is unloaded from the magazine and then deleted, if the parameter *DeleteTool* is set.

If the value for *MagazinePlace* > 0, an attempt is made to unload the tool from this magazine location.

Table 7-10 Parameters for "Unloading of a tool"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 142	Duplo	INT	1...3200	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 180	MagazinePlace	INT	0...3200	Magazine location
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
DBX 196.3	DeleteTool	BOOL	TRUE, FALSE	Delete tool after unloading?
Return parameters				
None				

7.10 Searching for empty location

Searches in the specified magazine for an empty tool location for the tool spcified with *TNumber*, *Ident* and *Duplo*.

If *TNumber* is specified, the required tool data is retrieved from the tool list and an empty tool location searched for.

If *Magazine* is assigned "0", a search is carried out for the next empty tool location in all the magazines.

Table 7-11 Parameters for "Searching for empty location"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 142	Duplo	INT	1...3200	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 178	TNumber	INT	0...3200	T-number
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
Return parameters				
DBW 198	Subtype	INT	1...3200	Sub-type
DBW 200	Duplo	INT	0...3200	Duplo number
DBB 202	Ident	String	32 characters	Tool identifier
DBW 236	TNumber	INT	0...3200	T-number
DBW 238	ToolState	INT	0...3200	Tool status
DBW 240	MagazinePlace Type	INT	0...3200	Magazine location type
DBB 242	ToolSize_ Upper	BYTE	1...127	Tool size in upper half-locations
DBB 243	ToolSize_Down	BYTE	1...127	Tool size in lower half-locations
DBB 244	ToolSize_Left	BYTE	1...127	Tool size in left half-locations
DBB 245	ToolSize_Right	BYTE	1...127	Tool size in right half-locations
DBW 246	Magazine	INT	0...3200	Magazine number
DBW 248	MagazinePlace	INT	0...3200	Magazine location

7.11 Deleting tool in NCK

Deletes tools in the NC kernel if these are not loaded.

Table 7-12 Parameters for "Deleting tool in the NCK"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 140	Subtype	INT	-1, 1...32000	Sub-type
DBW 142	Duplo	INT	1...3200	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 178	TNumber	INT	0...3200	T-number
DBB 188	Channel	BYTE	1...127	Channel number
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
Return parameters				
None				

7.12 Deleting tool in database

Deletes the data of tools in the database. When deleting, a check is made whether tools of the selected group have been loaded to the NCK. The data of loaded tools is not deleted in the database.

The parameters *Subtype*, *Ident* and *Duplo* must not necessarily be assigned. If *Subtype* and *Duplo* are not to be assigned, they must receive the value "-1". The empty string is used for *Ident*, if no specific tool is to be deleted.

If the subtype is not assigned, several tools can be deleted.

If *Duplo* is not assigned, all tools with the corresponding identifier and subtype are deleted in the database. If "0" is specified as the duplo number, master as well as operating data is deleted.

If *Duplo* and *Ident* are not specified, all tools of the corresponding subtype are deleted in the cabinet.

If the subtype < 10 and *Duplo* and *Ident* not assigned, the tool type is deleted. E.g. with value "1", all milling tools are deleted.

If with subtype "-1", an empty string is specified for the identifier and "-1" for the duplo number, the complete tool cabinet is deleted.

If the subtype is "-1", the identifier an empty string and the duplo number "0", the complete tool catalog and the tool cabinet are deleted.

Table 7-13 Parameters for "Deleting tool in database"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 140	Subtype	INT	-1, 1...32000	Sub-type
DBW 142	Duplo	INT	-1...32000	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
Return parameters				
None				

Table 7-14 Parameter combinations for "Deleting tool in database"

DBW 140	Subtype	>0	Deletion of the tools with the corresponding subtype and name
DBW 142	Duplo	-1	
DBB 144	Ident	Name	
DBW 140	Subtype	>0	Deletion of master and operating data
DBW 142	Duplo	0	
DBB 144	Ident	Name	
DBW 140	Subtype	>0	Deletion of all tools of the subtype
DBW 142	Duplo	-1	
DBB 144	Ident	""	
DBW 140	Subtype	<10	Deletion of the tool type
DBW 142	Duplo	-1	
DBB 144	Ident	""	
DBW 140	Subtype	-1	Deletion of the complete tool cabinet
DBW 142	Duplo	-1	
DBB 144	Ident	""	
DBW 140	Subtype	-1	Deletion of the complete tool cabinet and tool catalog
DBW 142	Duplo	0	
DBB 144	Ident	""	

7.13 Export of tool data

Export of tools from the database to a file in the NCK-INITIAL.INI file format.

The parameters *Subtype*, *Ident* and *Duplo* must not necessarily be assigned. If *Subtype* and *Duplo* are not to be assigned, they must receive the value "-1". The empty string is used for *Ident*, if no specific tool is to be exported.

If *Duplo* is not specified, all tools with the corresponding identifier and *Subtyp* are exported.

If *Duplo* and identifier are not specified, all tools of the corresponding subtype are exported.

If the subtype is less than "10" and *Duplo* and *Ident* are not assigned, the complete tool type is exported. E.g. with value "1" all milling tools are exported.

The complete database is exported if the subtype is "-1", the identifier is an empty string and *Duplo* is "-1".

All tools of the selected group are saved in the file whose index is specified in *FileIndex*. A check is made here, whether some of the tools whose data is to be exported are loaded in the NC kernel. The current data of loaded tools is read from the NC kernel and the database updated before the data is written to the file.

Table 7-15 Parameters for "Export tool data"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 140	Subtype	INT	-1, 1...32000	Sub-type
DBW 142	Duplo	INT	-1...32000	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 194	FileIndex	INT	0...3200	File index for export
DBX 196.1	Override	BOOL	TRUE, FALSE	Overwrite file if available?
Return parameters				
None				

Table 7-16 Parameter combinations for "Exporting tool data"

PLC date	Identifier	Value	Meaning
DBW 140	Subtype	>0	Exporting of the tools with the corresponding subtype and name
DBW 142	Duplo	-1	
DBB 144	Ident	Name	
DBW 140	Subtype	>0	Exporting of the tools with the corresponding subtype
DBW 142	Duplo	-1	
DBB 144	Ident	""	
DBW 140	Subtype	<10	Exporting of the complete tool type
DBW 142	Duplo	-1	
DBB 144	Ident	""	
DBW 140	Subtype	-1	Exporting of the complete database
DBW 142	Duplo	-1	
DBB 144	Ident	""	

7.14 Importing tool data

Imports tools from a file in the NCK-INITIAL file format into the database.

Only the index of the file in *FileIndex* from which the data is to be imported is specified as the parameter.

The import is always performed in the tool cabinet only. If a tool is not yet known in the database, it is also imported into the catalog.

If some of the tools to be imported have already been loaded into the NC kernel, the import of the data records of these tools is refused with an error message and an alarm triggered via the alarm server. The corresponding data records are skipped in the import file.

Table 7-17 Parameter for "Import tool data"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 194	FileIndex	INT	0...3200	File index for import
Return parameters				
None				

7.15 Updating code carrier

Updates the data on the code carrier.

The data of the tool specified with *Channel*, *Duplo*, *Ident*, *Magazine* and *NCU_Index* is read from the NC kernel and written to the code carrier of the tool. The code carrier is specified via *Unit*.

If "0" is transfered with magazine, the tool is searched for in all magazines

Table 7-18 Parameters for "Updating code carrier"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBW 142	Duplo	INT	1...3200	Duplo number
DBB 144	Ident	String	32 characters	Tool identifier
DBW 180	MagazinePlace	INT	0...3200	Magazine location
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 192	Unit	BYTE	1...127	Number of the reader head
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
Return parameters				
None				

7.16 Reading data from code carrier

Reading of the tool data from the code carrier and writing of the return parameters to the PLC. The code carrier is specified via *Unit*.

In this way it is possible, for example, to read the two data items duplo number and tool identifier from a code chip which only has this data stored on it. Then, for example, the services "Loading/unloading from the tool cabinet" (Sections 7.3 and 7.7) can be called.
In this way, low-price "read-only" code chips can be used.

Table 7-19 Parameters for "Reading data from code carrier"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBB 192	Unit	BYTE	1...127	Number of the reader head
Return parameters				
DBW 198	Subtype	INT	1...3200	Sub-type
DBW 200	Duplo	INT	0...3200	Duplo number
DBB 202	Ident	String	32 characters	Tool identifier
DBW 236	TNumber	INT	0...3200	T-number
DBW 238	ToolState	INT	0...3200	Tool status
DBW 240	MagazinePlace Type	INT	0...3200	Magazine location type
DBB 242	ToolSize_ Upper	BYTE	1...127	Tool size in upper half-locations
DBB 243	ToolSize_Down	BYTE	1...127	Tool size in lower half-locations
DBB 244	ToolSize_Left	BYTE	1...127	Tool size in left half-locations
DBB 245	ToolSize_Right	BYTE	1...127	Tool size in right half-locations
DBW 246	Magazine	INT	0...3200	Magazine number
DBW 248	MagazinePlace	INT	0...3200	Magazine location

7.17 Storing tool data in NCK

Creates the tool which has been read in from the code carrier in the NC kernel. If *LoadTool* is set, the tool is loaded immediately.

An attempt is made to create the tool with the specified T-number in the specified magazine of the NCU.

After the service has been carried out, the current tool data is returned in the return parameters.

Table 7-20 Parameters for "Storing tool data in NCK"

PLC date	Identifier	Type	Value range	Meaning
Transfer parameters				
DBB 188	Channel	BYTE	1...127	Channel number
DBW 190	Magazine	INT	0...3200	Magazine number
DBB 193	NCU_Index	BYTE	0...127	Index of the NCU
DBX 196.2	LoadTool	BOOL	TRUE, FALSE	Load tool after creating?
Return parameters				
DBW 198	Subtype	INT	1...3200	Sub-type
DBW 200	Duplo	INT	0...3200	Duplo number
DBB 202	Ident	String	32 characters	Tool identifier
DBW 236	TNumber	INT	0...3200	T-number
DBW 238	ToolState	INT	0...3200	Tool status
DBW 240	MagazinePlace Type	INT	0...3200	Magazine location type
DBB 242	ToolSize_ Upper	BYTE	1...127	Tool size in upper half-locations
DBB 243	ToolSize_Down	BYTE	1...127	Tool size in lower half-locations
DBB 244	ToolSize_Left	BYTE	1...127	Tool size in left half-locations
DBB 245	ToolSize_Right	BYTE	1...127	Tool size in right half-locations
DBW 246	Magazine	INT	0...3200	Magazine number
DBW 248	MagazinePlace	INT	0...3200	Magazine location

■

8

8 Start-Up Services

8.1 Execute services via the PLC interface

The services are executed via the PLC interface in the following sequence:

1. Specify command code
2. Set *Request* bit
3. Wait until *Done* bit has been set
4. Check error code
5. Reset *Request* bit

This sequence is shown in Fig. 8-1 as an impulse diagram, whereby the bits which are set and reset by *SinTDC* during the execution of the service are also displayed.

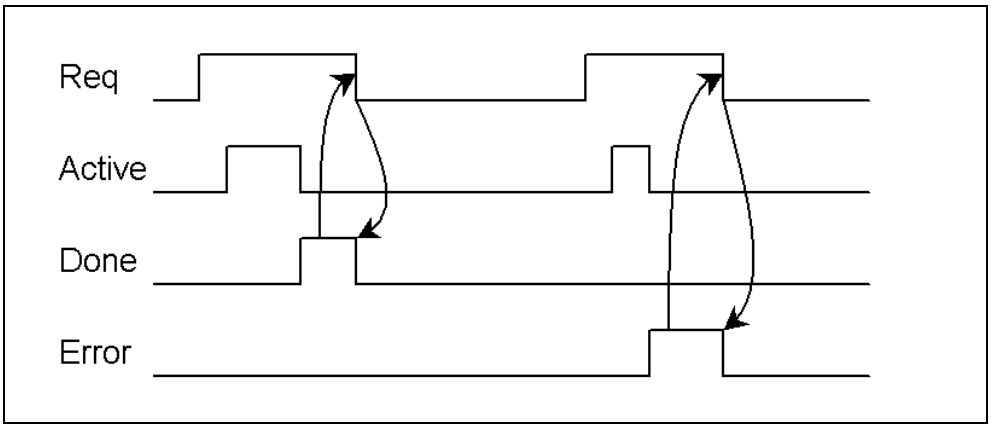

Fig. 8-1 Impulse diagram

The service which has been set with the command code is started by setting the *Request* bit (*Req*). Then, the *Active* bit is set in order to show that the job is being processed. If the job is completed with no errors, then the *Done* bit is set and the *Active* bit reset.

In the case of an error, the *Error* bit is set, and an error message written in the corresponding PLC-word; the *Active* bit is also reset in this case.

After the *Request* bit has been reset by the PLC, the *Done* or *Error* bit is reset again. In addition, the *Error* word is set to "0", so that the PLC must evaluate the error message before resetting the *Request* bit.

! Important

During the execution of the service, the *Request* bit must not be reset by the PLC. It should be reset only after the *Active* bit has been reset by *SinTDC* and the *Done* or *Error* bit set.

■

9

9 Examples

9.1 Loading from code carrier

In this example, the data from the imaginary tool "Tool_Test13" is read from a code carrier, the tool created in the NC kernel and loaded. The tool should be loaded in magazine "1" of the standard NCU in channel "1".

The parameters required for this service are shown in Table 9-1. Table 9-2 shows the return parameters supplied by *SinTDC*.

Before the service is executed, the value "0" is written in the command code byte (*K_Code*) of the PLC interface in the data block DB19. This service is then started by setting the *Request* bit. If the *Active* bit has been reset and the *Done* bit set, the error byte can be evaluated by the PLC. The *Request* bit must then be reset. The sequence is shown clearly in Table 9-3.

Table 9-1 Transfer parameters for "Loading from code carrier"

PLC date	**Identifier**	**Value**
DBW 180	MagazinePlace	0
DBB 188	Channel	1
DBW 190	Magazine	1
DBB 192	Unit	1
DBB 193	NCU_Index	0
DBX 196.2	LoadTool	TRUE

Table 9-2 Return parameters of "Loading from code carrier"

PLC date	**Identifier**	**Value**
DBW 198	Subtype	120
DBW 200	Duplo	1
DBB 202	Ident	Tool_Test13
DBW 236	TNumber	289
DBW 238	ToolState	14
DBW 240	MagazinePlaceType	1
DBB 242	ToolSize_Upper	1
DBB 243	ToolSize_Down	1
DBB 244	ToolSize_Left	1
DBB 245	ToolSize_Right	1
DBW 246	Magazine	1
DBW 248	MagazinePlace	1

Table 9-3 Execution of "Loading from code carrier"

SinTDC	**PLC**
...waits for request from PLC (Request bit from 0->1)	
	...writes the command code "0" in the PLC interface byte K_Code
	...sets the PLC interface Request bit (Request bit = "1")
	...waits until the Active bit has been set (Active bit from 0->1)
...receives the request from the PLC and starts the service, if a valid command code has been specified. Otherwise, the Error bit is set and an error message written in the Error_Code byte.	
...sets Active bit (Active bit = "1")	
	...waits until Active bit has been reset (Active bit from 1->0)
...reads the data from the code chip or from the PLC data block (according to reader)	
...loads the tool in the NC kernel	
...waits for the acknowledgement of the PI service via the PLC	
	...acknowledges the PI service
...writes the error code in the Error_Byte of the PLC interface (not equal to 0 in case of error)	
...sets the Done or Error bit (Done bit = "1" or Error bit = "1")	
...resets Active bit (Active bit = "0")	
...waits for request end (Request bit from 1->0)	
	...evaluates Done and Error bits and where applicable Error_Code
	...resets Request bit (Request bit = "0")
...receives request end (Request bit = "0")	
...resets Done and Error bits (Done bit = "0" and Error bit = "0")	
...deletes error code (Error_Code byte = "0")	
...waits for request via PLC (Request bit from 0->1)	
:	:

9.2 Unloading into code carrier

The loaded tool from the example above should be unloaded and the current data written to the code carrier. After unloading, the tool is deleted in the NC kernel.

The transfer parameters required for this are shown in Table 9-3. This service does not return any parameters, which means that there are no new values in the PLC interface return parameters.

Before the service is executed, the command code "1" must be written to the corresponding PLC byte (*K_Code*). The start of the service and the procedure is the same as the procedure described in 9.1. The sequence is shown clearly in Table 9-5.

Table 9-4 Transfer parameters for "Unloading into code carrier"

PLC date	**Identifier**	**Value**
DBW 142	Duplo	1
DBB 144	Ident	Tool_Test13
DBW 180	MagazinePlace	1
DBB 188	Channel	1
DBW 190	Magazine	1
DBB 192	Unit	1
DBB 193	NCU_Index	0
DBX 196.3	DeleteTool	TRUE

Table 9-5 Execution of "Unloading into code carrier"

SinTDC	**PLC**
...waits for request from PLC (Request bit from 0->1)	
	...writes the command code "0" in the PLC interface byte K_Code
	...sets the PLC interface Request bit (Request bit = "1")
	...waits until Active bit has been set (Active bit from 0->1)
...receives the request from the PLC and starts the service, if a valid command code has been specified. Otherwise, the Error bit is set and an error message written in the Error_Code byte.	
...sets Active bit (Active bit = "1")	
	...waits until Active bit has been reset (Active bit from 1->0)
...writes the data to the code chip or the PLC data block (according to reader)	
...unloads the tool from the NC kernel	
	...acknowledges the PI service
...writes the error code in the Error_Byte of the PLC interface (not equal to 0 in case of error)	
...sets the Done or Error bit (Done bit = "1" or Error bit = "1")	
...resets Active bit (Active bit = "0")	
...waits for the request end (Request bit from 1->0)	
	...evaluates Done and Error bits and where applicable Error_Code
	...resets Request bit (Request bit = "0")
...receives request end (Request bit = "0")	
...resets Done and Error bits (Done bit = "0" and Error bit = "0")	
...deletes error code (Error_Code byte = "0")	
...waits for request via PLC (Request bit from 0->1)	
:	:

■

A

A Appendix

A.1 Error codes

The error codes stored in the PLC byte *Error_Code* of the PLC interface are in hexadecimal format and indicate the point at which the job execution was aborted with an error.

If an error code is shown in decimal notation, it must be converted into hexadecimal format in order to calculate the cause of error from the tables A-1 and A-2.

The error codes are composed of a high byte and a low byte, each having a different meaning. The high byte (XX) identifies the command code in which the error occurred. The low byte specifies the cause of error.

Format

XX	YY

High byte

Table A-1 High byte of error code

XX	Service
00	No error
01	Loading from the code carrier
02	Unloading into code carrier
03	Loading from the tool cabinet
04	Unloading into the tool cabinet
05	Loading from the host computer
06	Unloading into the host computer
07	Unloading of all tools in tool cabinet
08	Loading of tool
09	Unloading of tool
0A	Searching for empty location
0B	Deletion of tool in NCK
0C	Deletion of tool in the database
0D	Export of tool data
0E	Import of tool data
0F	Updating code carrier

10	Reading data from code carrier
11	Storing tool data in NC kernel
...	Spare
F0	Invalid commando code
...	Reserve

Low byte

The low byte shows the cause of the error which resulted in the abort of the job.

Table A-2 Low byte of error code

YY	**Cause of error**
00	No error
01	Error while reading the parameters out of DB19
02	Error while calling up the internal service
03	Timeout while executing service
04	No tool data has been determined
05	Error while executing the internal service
...	Spare
	Error in transfer parameter
10	Subtype
11	Duplo
12	Ident
13	TNumber
14	MagazinePlace
15	MagazinePlaceType
16	ToolSize_Upper
17	ToolSize_Down
18	ToolSize_Left
19	ToolSize_Right
1A	Channel
1B	Magazine
1C	Unit
1D	NCU_Index
1E	FileIndex
1F	ActiveEnable
20	Override
21	LoadTool
22	DeleteTool
23	Length of Ident
...	Reserve

A.2 PLC data types

Table A-3 describes the PLC data types used and their interpretation by *SinTDC*.

Table A-3 Used PLC data types

Data type	Value range	Comment
BOOL	TRUE, FALSE	Used if single bits are to be set in the PLC. (TRUE: Bit = 1, FALSE: Bit = 0)
BYTE	-128...127	PLC byte
INT	-32768...32767	PLC word whose first byte contains the high byte and the second byte, the low byte. *Example:* PLC word 140 is composed of both PLC bytes 140 and 141 and should be assigned the value 120. Byte 140: 0 Byte 141: 120
String	xx characters	Strings occupy xx+2 bytes in the PLC, since the first two bytes contain the maximum length and the current length of the string. The actual string therefore only begins from the third byte. The individual characters are stored as ASCII code in the corresponding bytes. PLC strings do not have to be terminated with zero, as the current length is always supplied in the second byte of the string. *Example:* "Hello" should be saved in a string which begins at PLC byte 144 and which can have a maximum of 32 characters. The resulting byte assignment is as follows: Byte 144 = 32 Max. length of string Byte 145 = 5 Current length of string Byte 146 = 72 H Byte 147 = 97 e Byte 148 = 108 l Byte 149 = 108 l Byte 150 = 111 o

A.3 Abbreviations

ASCII	American Standard Code for Information Interchange
BA	Operator's Guide
CD	Compact Disc
COM	Communications
COM	Component Object Model (Programming model from *Microsoft*®)
DB	Data Block
DCOM	Distributed Component Object Model
HMI	Human Machine Interface
INT	Integer
MB	Megabyte
MMC	Man Machine Communication
NC	Numerical Control
NCK	Numerical Control Kernel
NCU	Numerical Control Unit
PCU	Personal Computer Unit
PG	Programming guide
PLC	Programmable Logic Control
ROM	Read-Only Memory
SINCOM	Sinumerik Communication
SinTDC	Sinumerik Tool Data Communication
VB	Visual Basic
VC	Visual C++

A.4 References

General Documentation

/BU/ SINUMERIK 840D/840Di/810D/802S, C, D
Ordering information
Catalog NC 60
Order No.: E86060-K4460-A101-A9-7600

/ST7/ **SIMATIC**
SIMATIC S7 Programmable Logic Controllers
Catalog ST 70
Order No.: E86060-K4670-A111-A3

/Z/ SINUMERIK, SIROTEC, SIMODRIVE
Accessories and Equipment for Special-Purpose Machines
Catalog NC Z
Order No.: E86060-K4490-A001-A8-7600

Electronic Documentation

/CD1/ The SINUMERIK system
DOC ON CD (09.02 Edition)
(with all SINUMERIK 840D/840Di/810D/802 and
SIMODRIVE publications)
Order No.: 6FC5 298-6CA00-0BG3

User Documentation

/AUK/	SINUMERIK 840D/810D **Short Guide AutoTurn Operation** Order No.: 6FC5 298-4AA30-0AP3	(11.01 Edition)
/AUP/	SINUMERIK 840D/810D **AutoTurn Graphic Programming System** Operator's Guide Programming / Setting up Order No.: 6FC5 298-4AA40-0BP3	(11.01 Edition)
/BA/	SINUMERIK 840D/810D **Operator's Guide: MMC** Order No.: 6FC5 298-6AA00-0BP0	(10.00 Edition)
/BAA/	SINUMERIK 840D/840Di/810D **Operator's Guide: @Event** Order No.: 6AU1900-0CL20-0BA0	(01.02 Edition)
/BAD/	SINUMERIK 840D/840Di/810D **Operator's Guide: HMI Advanced** Order No.: 6FC5 298-6AF00-0BP1	(11.01 Edition)
/BEM/	SINUMERIK 840D/810D **Operator's Guide HMI Embedded** Order No.: 6FC5 298-6AC00-0BP1	(11.01 Edition)
/BAH/	SINUMERIK 840D/840Di/810D **Operator's Guide HAT 6 (PHG new)** Order No.: 6FC5 298-0AD60-0BP2	(11.01 Edition)
/BAK/	SINUMERIK 840D/840Di/810D **Short Guide Operation** Order No.: 6FC5 298-6AA10-0BP0	(02.01 Edition)
/BAM/	SINUMERIK 810D/840D **Operator's Guide ManualTurn** Order No.: 6FC5 298-6AD00-0BP0	(08.00 Edition)
/BAS/	SINUMERIK 840D/810D **Operator's Guide ShopMill** Order No.: 6FC5 298-6AD10-0BP0	(12.01 Edition)
/BAT/	SINUMERIK 840D/810D **Operator's Guide ShopTurn** Order No.: 6FC5 298-6AD50-0AP1	(01.02 Edition)
/BNM/	SINUMERIK 840D840Di//810D **User Guide Measuring Cycles** Order No.: 6FC5 298-6AA70-0BP1	(09.01 Edition)

/DA/ SINUMERIK 840D/840Di/810D
Diagnostic Guide (09.01 Edition)
Order No.: 6FC5 298-6AA20-0BP1

/KAM/ SINUMERIK 840D/810D
Short Guide ManualTurn (04.01 Edition)
Order No.: 6FC5 298-5AD40-0BP0

/KAS/ SINUMERIK 840D/810D
Short Guide ShopMill (04.01 Edition)
Order No.: 6FC5 298-5AD30-0BP0

/KAT/ SINUMERIK 840D/810D
Short Guide ShopTurn (07.01 Edition)
Order No.: 6FC5 298-5AF20-0BP0

/PG/ SINUMERIK 840D/840Di/810D
Programing Guide, Fundamentals (09.01 Edition)
Order No.: 6FC5 298-6AB00-0BP1

/PGA/ SINUMERIK 840D/840Di/810D
Programming Guide Advanced (09.01 Edition)
Order No.: 6FC5 298-6AB10-0BP1

/PGK/ SINUMERIK 840D/840Di/810D
Short Guide Programming (02.01 Edition)
Order No.: 6FC5 298-6AB30-0BP1

/PGM/ SINUMERIK 840D/840Di/810D
Programming Guide ISO Milling (10.01 Edition)
Order No.: 6FC5 298-6AC20-0BP1

/PGT/ SINUMERIK 840D/840Di/810D
Programming Guide ISO Turning (10.01 Edition)
Order No.: 6FC5 298-6AC10-0BP1

/PGZ/ SINUMERIK 840D840Di//810D
Programming Guide Cycles (09.01 Edition)
Order No.: 6FC5 298-6AB40-0BP1

/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 (02.01 Edition)
Order No.: 6FC5 298-6AE40-0BP0

/SYS/ SINUMERIK 840D840Di//810D
Inbetriebnahme-Tool-SinCOM NC (02.02 Edition)
Systemüberblick
Order No.: (Part of the Online Help and available in pdf format)

Manufacturer/Service Documentation

a) Lists

/LIS/ SINUMERIK 840D/840Di/810D
SIMODRIVE 611D
Lists (09.01 Edition)
Order No.: 6FC5 297-6AB70-0BP1

b) Hardware

/BH/ SINUMERIK 840D840Di//810D
Operator Components Manual (HW) (09.01 Edition)
Order No.: 6FC5 297-6AA50-0BP1

/BHA/ SIMODRIVE **Sensor**
Absolute Encoder with Profibus-DP
User Guide (HW) (02.99 Edition)
Order No.: 6SN1 197-0AB10-0YP1

/EMV/ SINUMERIK, SIROTEC, SIMODRIVE
EMC Directive (06.99 Edition)
Planning Guide (HW)
Order No.: 6FC5 297-0AD30-0BP1

/PHC/ SINUMERIK 810D
Manual Configuring (HW) (12.01 Edition)
Order No.: 6FC5 297-6AD10-0BP1

/PHD/ SINUMERIK 840D
NCU 571.2-573.2 Configuring Manual (HW) (07.02 Edition)
Order No.: 6FC5 297-6AC10-0BP2

/PHF/ SINUMERIK FM-NC
NCU 570 Configuring Manual (HW) (04.96 Edition)
Order No.: 6FC5 297-3AC00-0BP0

/PMH/ SIMODRIVE **Sensor**
Measuring System for Main Spindle Drives
Configuring Installation Guide, SIMAG-H (HW) (05.99 Edition)
Order No.: 6SN1197-0AB30-0BP0

c) Software

/FB1/ SINUMERIK 840D/840Di/810D/FM-NC
Desription of Functions, Basic Machine (Part 1) (09.01 Edition)
(the various sections are listed below)
Order No.: 6FC5 297-6AC20-0BP1

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, Channels, Program Operation
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)
Desription of Functions, Extended Functions (Part 2) (09.01 Edition)
including FM-NC: Turning, Stepping Motor
(the various sections are listed below)
Order No.: 6FC5 297-6AC30-0BP1

A4	Digital and Analog NCK I/Os
B3	Several Operator Panels and NCUs
B4	Operation via PC/PG
F3	Remote Diagnostics
H1	Jog with/without Handwheel
K3	Compensations
K5	Mode Groups, Channels, Axis Exchange
L1	FM-NC Local Bus
M1	Kinematic Transformation
M5	Measurements
N3	Software Cams, Position Switching Signals
N4	Punching and Nibbling
P2	Positioning Axes
P5	Oscillation
R2	Rotary Axes
S3	Synchronous Spindles
S5	Synchronized Actions (SW 3 and earlier)
S6	Stepping Motors
S7	Memory Configuration
T1	Indexing Axes
W3	Tool Change
W4	Grinding

/FB3/ SINUMERIK 840D/840Di/810D(CCU2)
Description of Functions, Special Functions (Part 3) (09.01 Edition)
(the various sections are listed below)
Order No.: 6FC5 297-6AC80-0BP1

F2 3 to 5 Axis Transformation
G1 Gantry Axes
G3 Cycle Times
K6 Contour Tunnel Monitoring
M3 Coupled Axes and ESR
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 Handling Transformation Package
TE5 Setpoint Exchange
TE6 MCS Coupling
TE7 Retrace Support
TE8 Clock-Independent, Path-Synchronous Switching Signal Output

/FBA/ SIMODRIVE 611D/SINUMERIK 840D/810D
Description of Functions, Drive Functions (12.01 Edition)
(the various sections are listed below)
Order No.: 6SN1 197-0AA80-0BP8

DB1 Operational Messages/Alarm Reactions
DD1 Diagnostic Functions
DD2 Speed Control Loop
DE1 Extended Drive Functions
DF1 Enables
DG1 Encoder Parameterization
DM1 Calculation of Motor/Power Section Parameters and Controller Data
DS1 Current Control Loop
DÜ1 Monitors/Limitations

/FBAN/ SINUMERIK 840D/SIMODRIVE 611 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 Installation
DI2 Scanning with Tactile Sensors (scancad scan)
DI3 Scanning with Lasers (scancad laser)
DI4 Milling Program Generation (scancad mill)

/FBDN/ SINUMERIK 840D/810D
IT-Solutions
NC Data Management Server (DNC NT-2000)
Description of Functions (05.00 Edition)
Order No.: 6FC5 297-5AE50-0BP1

/FBDT/ SINUMERIK 840D/810D
IT-Solutions
NC Data Transmission (SinDNC)
Description of Functions (03.01 Edition)
Order No.: 6FC5 297-1AE70-0BP0

/FBFA/ SINUMERIK 840D/840Di/810D
ISO Dialects for SINUMERIK (09.01 Edition)
Description of Functions
Order No.: 6FC5 297-6AE10-0BP1

/FBFE/ SINUMERIK 840D/810D
Description of Functions **Remote Diagnosis** (11.01 Edition)
Order No.: 6FC5 297-0AF00-0BP1

FE1 Remote Diagnosis
FE2 Interrupt-Controlled Email Messaging: @Event

/FBH/ SINUMERIK 840D/840Di/810D
HMI Configuring Package (10.01 Edition)
Order No.: (Part of the SW package)

Part 1 User's Guide
Part 2 Description of Functions

/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** (08.00 Edition)
Order No.: 6FC5 297-6AD50-0BP0

/FBO/ SINUMERIK 840D/810D
Configuring the User Interface OP 030 (09.01 Edition)
Description of Functions
(the various sections are listed below)
Order No.: 6FC5 297-6AC40-0BP0

BA Operator's Guide
EU Development Environment (Configuring Package)
PSE Introduction to Configuring of Operator Interface
IK Screen Kit: Software Update and Configuration

PS Online only: Configuring syntax (Configuring package)

/FBP/ SINUMERIK 840D
Description of Functions **C-PLC- Programming** (03.96 Edition)
Order No.: 6FC5 297-3AB60-0BP0

/FBR/ SINUMERIK 840D/810D
IT-Solutions
Computer Link (SinCOM) (04.00 Edition)
Description of Functions
Order No.: 6FC5 297-5AD60-0BP0

NFL Interface to Host Computer
NPL Interface to PLC/NCK

/FBSI/ SINUMERIK 840D/SIMODRIVE
Description of Functions **SINUMERIK Safety Integrated** (03.01 Edition)
Order No.: 6FC5 297-6AB80-0BP0

/FBSP SINUMERIK 840D/810D
Description of Functions **ShopMill** (12.01 Edition)
Order No.: 6FC5 297-6AD80-0BP1

/FBST/ **SIMATIC**
FM STEPDRIVE/SIMOSTEP (01.01 Edition)
Description of Functions
Order No.: 6SN1 197-0AA70-0YP4

/FBSY/ SINUMERIK 840D/810D
Description of Functions **Synchronized Actions** (09.01 Edition)
for Wood, Glass, Ceramics and Presses
Order No.: 6FC5 297-6AD40-0BP1

/FBT/ SINUMERIK 840D/810D
Description of Functions **ShopTurn** (01.02 Edition)
Order No.: 6FC5 297-6AD70-0BP1

/FBTC/ SINUMERIK 840D/810D
IT-Solutions
SINUMERIK Tool Data Communication SinTDC (01.02 Edition)
Description of Functions
Order No.: 6FC5 297-5AF30-0BP0

/FBTD/ SINUMERIK 840D/810D
IT-Solutions
Tool Data Information System SINTDI (03.01 Edition)
with Online Help
Description of Functions
Order No.: 6FC5 297-6AE00-0BP0

/FBU/ **SIMODRIVE 611 universal**
Description of Functions (08.01 Edition)
Control Components for Speed Control
and Positioning
Order No.: 6SN1 197-0AB20-0BP4

/FBW/ SINUMERIK 840D/810D
Description of Functions **Tool Management** (10.01 Edition)
Order No.: 6FC5 297-6AC60-0BP1

/HBI/ SINUMERIK 840Di
Manual (09.01 Edition)
Order No.: 6FC5 297-6AE50-0BP0

/PFK/ **SIMODRIVE**
Configurig Guide for **1FT5/1FT6/1FK6 Motors** (12.01 Edition)
Transistor PWM Inverters for AC Feed Servo Drives
and AC Main Spindle Drives
Order No.: 6SN1 197-0AC20-0BP0

/PJE/ SINUMERIK 840D/810D
Projektierpaket HMI Embedded (08.01 Edition)
Description of Functions: Software Update,
Configuration, Installation
Order number: 6FC5 297-6EA10-0BP0
(the document CS Configuring Syntax is supplied with
the software and available as a pdf file)

/PJFE/ **SIMODRIVE**
Synchronous Integrated Motors 1FE1 (09.01 Edition)
Planning Guide
AC Motors for Main Spindle Drives
Order No.: 6SN1 197-0AC00-0BP1

/PJLM/ SIMODRIVE
Planning Guide **Linear motors 1FN1, 1FN3** (11.01 Edition)

ALL General Information about Linear Motors
1FN1 Three-Phase AC Linear Motor
1FN3 Three-Phase AC Linear Motor
CON Connections
Order No.: 6SN1 197-0AB70-0BP2

/PJM/ **SIMODRIVE**
Planning Guide **Motors**
Transistor PWM Inverters for AC Feed Drives (09.00 Edition)
and AC Main Spindle Drives
Order No.: 6SN1 197-0AA20-0BP4

/PJU/ **SIMODRIVE 611**
Planning Guide **Inverters** (05.01 Edition)
Order No.: 6SN1 197-0AA00-0BP5

/POS1/ **SIMODRIVE POSMO A**
User Guide (04.01 Edition)
Distributed Positioning Motor on PROFIBUS DP
Order No.: 6SN2 197-0AA00-0BP2

/POS2/ **SIMODRIVE POSMO A**
Assembly Instructions (supplied with every POSMO A) (12.98 Edition)
Order No.: 462 008 0815 00

/POS3/ **SIMODRIVE POSMO SI/CD/CA**
Distributed Servo Drive Systems, User Manual (08.01 Edition)
Order No.: 6SN2 197-0AA20-0BP1

/PPH/	**SIMODRIVE** Planning Guide **1PH2/1PH4/1PH7 Motors** AC Induction Motors for Main Spindle Drives Order No.: 6SN1 197-0AC60-0BP0	(12.01 Edition)
/PPM/	**SIMODRIVE** Planning Guide **Hollow Shaft Motors** Hollow Shaft Motors for Main Spindle Drives 1PM4 and 1PM6 Order No.: 6SN1 197-0AD03-0BP0	(10.01 Edition)
/S7H/	**SIMATIC S7-300** - Manual: Assembly, CPU Data (HW Description) - Reference Manual: Module Data Order No.: 6ES7 398-8AA03-8AA0	(10.98 Edition)
/S7HT/	**SIMATIC S7-300** Manual: STEP 7, Fundamentals, V. 3.1 Order No.: 6ES7 810-4CA02-8AA0	(03.97 Edition)
/S7HR/	**SIMATIC S7-300** Manual: STEP 7, Reference Manuals, V. 3.1 Order No.: 6ES7 810-4CA02-8AR0	(03.97 Edition)
/S7S/	**SIMATIC S7-300** **FM 353** Servo Drive Positioning Module for Stepper Drives Order in conjunction with Configuring Package	(04.97 Edition)
/S7L/	**SIMATIC S7-300** **FM 354** Multi-Axis Module for Servo Drives Order in conjunction with Configuring Package	(04.97 Edition)
/S7M/	**SIMATIC S7-300** Multi-Axis Module **FM 357.2** for Servo or Stepper Drive Order in conjunction with Configuring Package	(01.01 Edition)
/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 (10.00 Edition)
(including description of start-up software for SIMODRIVE 611D)
Order No.: 6SN 1197-0AA60-0BP6

/IAC/ SINUMERIK 810D
Installation and Start-Up Guide (10.01 Edition)
(including description of SIMODRIVE 611D start-up software)
Order No.: 6FC5 297-4AD20-0BP1

/IAD/ SINUMERIK 840D/SIMODRIVE 611D
Installation and Start-Up Guide (09.01 Edition)
(including description of SIMODRIVE 611D start-up software)
Order No.: 6FC5 297-6AB10-0BP1

/IAF/ SINUMERIK FM-NC
Installation and Start-Up Guide (07.00 Edition)
Order No.: 6FC5 297-3AB00-0BP1

/IAM/ SINUMERIK 840D/840Di/810D
Start-Up Guide HMI/MMC (11.01 Edition)
Order No.: 6FC5 297-6AE20-0BP1

AE1	Updates/Extensions
BE1	Supplement operator interface
HE1	Online Help
IM2	Installation and Start-Up HMI Embedded
IM4	Installation and Start-Up HMI Advanced
TX1	Creating Foreign Language Texts

■

I Index

L

M

N

O

P

R

S

T

U

W

■

SIEMENS AG
A&D MC BMS
P.O. Box 3180

D–91050 Erlangen, Germany

(Tel. 0180 / 5050 – 222 [Hotline]
Fax 09131 / 98 – 2176
email: motioncontrol.docu@erlf.siemens.de)

Suggestions

Corrections

For Publication/Manual:

SINUMERIK 840D/810D
SINUMERIK Tool Data Communication (SinTDC)

Manufacturer Documentation

From

Name

Company/Dept.

Address

Telephone: /

Telefax: /

Manual

Order No.: 6FC5297-5AF30-0BP0
Edition: 01.2002

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

Overview of SINUMERIK 840D/840Di/810D Documentation (07.2002)

General Documentation

- *SINUMERIK* 840D/810D — Brochure
- *SINUMERIK* 840D/840Di/ 810D/ — Catalog Order. Info NC 60 ***)**
- *SINUMERIK SIROTEC SIMODRIVE* Accessories — Catalog Accessories NC–Z

User Documentation

- *SINUMERIK* 840D/810D/ FM–NC — **AutoTurn**
  – Short Guide
  – Programming /Setup
- *SINUMERIK* 840D/840Di 810D — Operator's Guide
  – HT 6
  – @Event
- *SINUMERIK* 840D/840Di/ 810D — **Diagnostics Guide *)**
- *SINUMERIK* 840D/840Di/ 810D — **Operator's Guide *)**
  – Short Guide
  – HMI Embedded
  – HMI Advanced

User Documentation

- *SINUMERIK* 840D/840Di/ 810D — **Program. Guide**
  – Short Guide
  – Fundamentals ***)**
  – Advanced ***)**
  – Cycles
  – Measuring Cycles
  – ISO Turning/Milling
- *SINUMERIK* 840D/810D — Operator's Guide
  – **ManualTurn**
  – Short Guide ManualTurn
  – **ShopMill**
  – Short Guide ShopMill
  – **ShopTurn**
  – Short Guide ShopTurn
- *SINUMERIK* 840Di — **System Overview**

Manufacturer/Service Documentation

- *SINUMERIK* — **Configuring** (HW) ***)**
  – 810D
  – 840D
- *SINUMERIK* 840D/840Di/ 810D — **Operator Components** (HW) ***)**
- *SINUMERIK* 840D/810D — Descr. of Functions
  – **ManualTurn**
  – **ShopMill**
  – **ShopTurn**
- *SINUMERIK* 840D/840Di/ 810D — Descr. of Functions **Synchron. Actions**

Manufacturer/Service Documentation

- *SINUMERIK SIMODRIVE* 611D 840D/810D — Descr. of Functions **Drive Functions *)**
- *SINUMERIK* 840D/840Di/ 810D — **Descr. of Functions**
  – Basic Machine ***)**
  – Extended Functions
  – Special Functions
- *SINUMERIK* 840D/840Di/ 810D — Descr. of Functions **Tool Management**
- *SINUMERIK* 840D/810D — **Configuring Kit** HMI Embedded
- *SINUMERIK* 840D/810D — Descr. of Functions Operator Interface **OP 030**
- *SINUMERIK* 840D/810D — **IT Solutions**
  – Computer Link
  – Tool Dat. Inf. Syst.
  – NC Data Mgmt. & Distr.
  – NC Data Transm. over a Network
  – Tool Data Communica.

Manufacturer/Service Documentation

- *SINUMERIK SIMODRIVE* — Descr. of Functions SINUMERIK **Safety Integrated**
- *SINUMERIK* 840D — Descr. of Functions **Digitizing**
- *SINUMERIK SIMODRIVE* — **Inst.&Start-Up Guide *)**
  – 810D
  – 840D/611D
  – HMI
- *SINUMERIK SIMODRIVE* 840D/840Di 810D 611D — **Lists *)**
- *SINUMERIK SIMODRIVE* 840D 611D — Descr. of Functions **Linear Motor**
- *SINUMERIK SIMODRIVE* 840D 611D — Descr. of Functions
  – Hydraulics Module
  – Analog Module
- *SINUMERIK SIMODRIVE SIROTEC* — **EMC** Guidelines

Electronic Documentation

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

Manufacturer/Service Documentation

- *SINUMERIK* 840D/840Di/ 810D — Descr. of Functions **ISO–Dialects** for SINUMERIK
- *SINUMERIK* 840Di — **Manual** (HW + Inst.&Start-Up)
- *SINUMERIK* 840D/810D — Descr. of Functions **Remote Diagnosis**

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