CL400 / CL500

PBK5MIF Software Module
Software Module Description

Edition 102

BOSCH
Automation

CL400 / CL500

PBK5MIF Software Module
Software Module Description

1070 072 135-102 (99.11) GB

Contents

Page

1 Safety Instructions

Before you start working with the PBK5MIF Software Module, we recommend that you thoroughly familiarize yourself with the contents of this instruction manual. Keep this manual in a place where it is always accessible to all users.

1.1 Standard operation

This instruction manual presents a comprehensive set of instructions and information about the software module required for installation and operation of the Peripheral Bus Interface Module.

The products described hereunder

- were developed, manufactured, tested and documented in accordance with the relevant safety standards. In standard operation, and provided that the specifications and safety instructions relating to the project phase, installation and proper operation of the product are followed, there should arise no risk of danger to personnel or property.

The prerequisites for trouble-free service and safe operation of the product are proper transport, handling and storage, placement and installation, plus careful operation of the equipment.

1.2 Qualified personnel

The requirements pertaining to qualified personnel are based on the job specifications as outlined by the ZVEI (central association of the electrical industry) and VDMA (association of German machine and plant builders) professional associations in Germany. Please refer to the following German-language publication:
Weiterbildung in der Automatisierungstechnik
Hrsg.: ZVEI und VDMA
MaschinenbauVerlag
Postfach 71 08 64
60498 Frankfurt

This instruction manual is specifically designed for PLC specialists. They will require specific knowledge of the CL400/CL500 Programmable Logic Controllers and of the PBK5MIF Peripheral Bus Interface Module.

Interventions in the hardware and software of our products which are not described in this instruction manual may only be performed by specially trained Bosch personnel.

Unqualified interventions in the hardware or software or non-compliance with the warnings listed in this instruction manual or indicated on the product may result in serious personal injury or damage of property.

Installation and maintenance of the products described hereunder is the exclusive domain of trained electricians as per VDE 1000-10, who are familiar with the contents of this manual.

Trained electricians are persons of whom the following is true:

- They are capable, due to their professional training, skills and expertise, and based upon their knowledge of and familiarity with applicable technical standards, of assessing the work to be carried out, and of recognizing possible hazards.
- They possess, subsequent to several years' experience in a comparable field of endeavour, a level of knowledge and skills that may be deemed commensurate with that attainable in the course of a formal professional education in this area.

With regard to the foregoing, please read the information about our comprehensive training program. You'll find a listing of our seminars on the front inside cover of this instruction manual. The professional staff at our training centre will be pleased to provide detailed information. You may contact the centre by telephone at (+49) (0)6062 78-258.

1.3 Safety markings on components

Danger: High voltage!

Danger: Battery acid!

Electrostatically sensitive components!

Disconnect at mains before opening!

Pin for connecting PE conductor only!

Functional earthing/low noise earth

For screened conductor only!

1.4 Safety instructions in this manual

DANGEROUS ELECTRICAL VOLTAGE
This symbol is used to warn of the presence of a **dangerous electrical voltage**. Insufficient compliance with or failure to observe this warning may result in **personal injury**.

DANGER
This symbol is used wherever insufficient or lacking compliance with instructions may result in **personal injury**.

CAUTION
This symbol is used whenever insufficient or lacking compliance with instructions may result in **damage to equipment or data files**.

☞ This symbol is used to alert the user to an item of special interest.

1.5 Safety instructions concerning the described product

DANGER
Danger to Personnel and Equipment!
Test every new program before operating the system!

DANGER
Retrofits or modifications may adversely affect the safety of the products described!

The consequences may be severe personal injury, damage to equipment or environmental hazards. Therefore, any system retrofitting or modification utilizing third-party components will require express approval by Bosch.

1.6 Documentation, version and trademark

Documentation

The present instruction manual provides information about the PBK5MIF Software Module required for the installation and operation of the PBK Peripheral Bus Interface Module, and about the MOBY I/F Channel Module.

Overview of instruction manuals:

Instruction Manuals	Language	Part no.
PBK Peripheral Bus Interface Module, Module Description	German	1070 070 127
	English	1070 072 133
	Spanish	1070 072 330
MOBY I/F Channel Module, Module Description	German	1070 070 139
	English	1070 072 136
	Spanish	1070 072 331
PBK5MIF Software Module, Software Module Description	German	1070 072 035
	English	1070 072 135
	Spanish	1070 072 332

★ This asterisk symbol indicates that the instruction manual is describing an activity you shall be required to perform.

Amendments

Changes in this instruction manual from the previous manual version are denoted by black vertical bars in the right-hand margin.

☞ **The “LEERER MERKER Amendments” section lists the changes made since the preceding edition.**

Trademarks

All trademarks referring to software that is installed on Bosch products when shipped from the factory represent the property of the respective manufacturers.

When shipped from the factory, all installed software is protected by copyright. It may therefore be duplicated only with prior permission by Bosch or in accordance with the licensing agreements with the respective manufacturer or copyright owner.

MS-DOS® und Windows™ are registered trademarks of Microsoft Corporation.

MOBY® is a registered trademark of Siemens AG.

2 PBK5MIF and MOBY-I Mobile Data Storage System

The PBK5MIF Software Module controls the data traffic between the PLC program and the PBK Peripheral Bus Interface Module equipped with the MOBY I/F channel module.

The MOBY I/F Channel Module is a component of the MOBY-I inductive identification system manufactured by Siemens AG. The identification system encompasses

- the MOBY I/F Channel Module,
- the SLG Read/write unit, and
- the MDS Mobile Data Storage unit.

The MOBY I/F channel module is inserted in the PBK Peripheral Bus Interface Module.

In addition, for the purpose of startup and testing the MOBY I/F, the STG 4F Service and Testing unit is available.

With regard to its functionality, the Bosch MOBY I/F Channel Module corresponds to the Siemens CM 423 Channel Module.

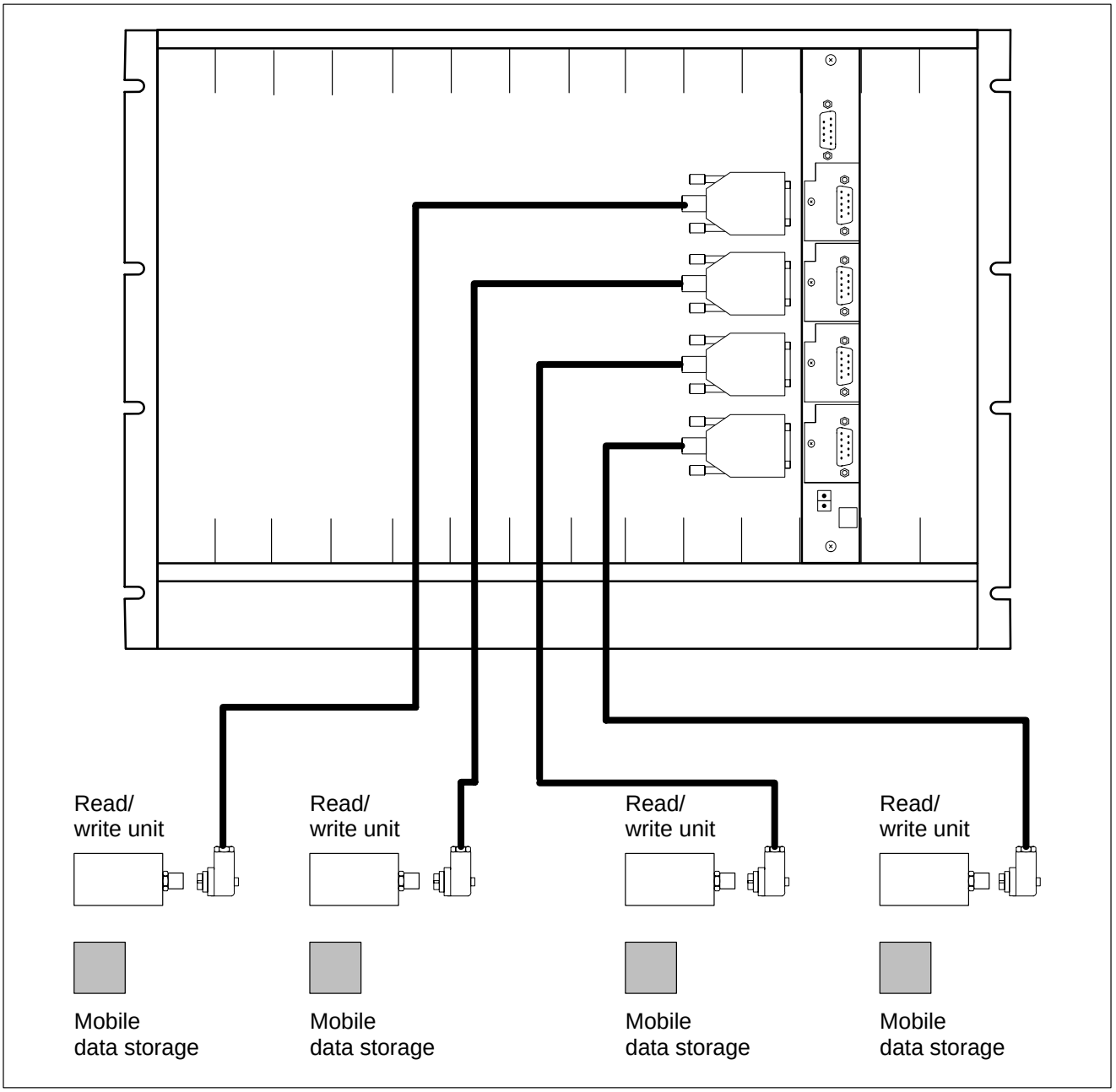
☞ **For detailed information about the MOBY-I Identification System, refer to the documentation provided by Siemens AG. The following catalogue may provide introductory reading:**
MOBY
Identsystem MOBY-I
E86060-K6910-A101-A2

For a detailed description of the PBK Peripheral Bus Interface Module, refer to the following instruction manual:
PBK Peripheral Bus Interface Module
Module Description
Order no. 1070 072 133

For a detailed description of the MOBY I/F Channel Module, refer to the following instruction manual:
MOBY® I/F Channel Module
Module Description
Order no. 1070 072 136

The intelligent SLG Read/write unit is connected to the peripheral bus of the control unit via the MOBY I/F Channel Module and the PBK Peripheral Bus Interface Module.

The data transmission between the SLG and the MDS is handled inductively via frequency modulation.



The PBK5MIF Software Module and the MOBY I/F support the Filehandler file management system.

To handle the data transfer, the use of the PBK5MIF Software Module is required. The PBK5MIF must be integrated into the PLC program.

☞ **The PLC program may utilize only the PBK5MIF module to access data on the PBK! All other types of read or write accesses to the addresses on the PBK Peripheral Bus Interface Module will cause loss of synchronization.**

Major PBK5MIF functions:

- Adapting the data structure of the PLC application program to that of the PBK and MOBY I/F.
- Exchanging commands and data with the PBK.
- Processing error messages for the PLC program.

The data transfer sequence between the PBK5MIF and MDS may be divided into 3 sections:

- Transfer of commands, parameter values and/or data to the SLG.
- Data exchange between SLG and MDS.
- Transfer of parameter values or data between CL500 and SLG.

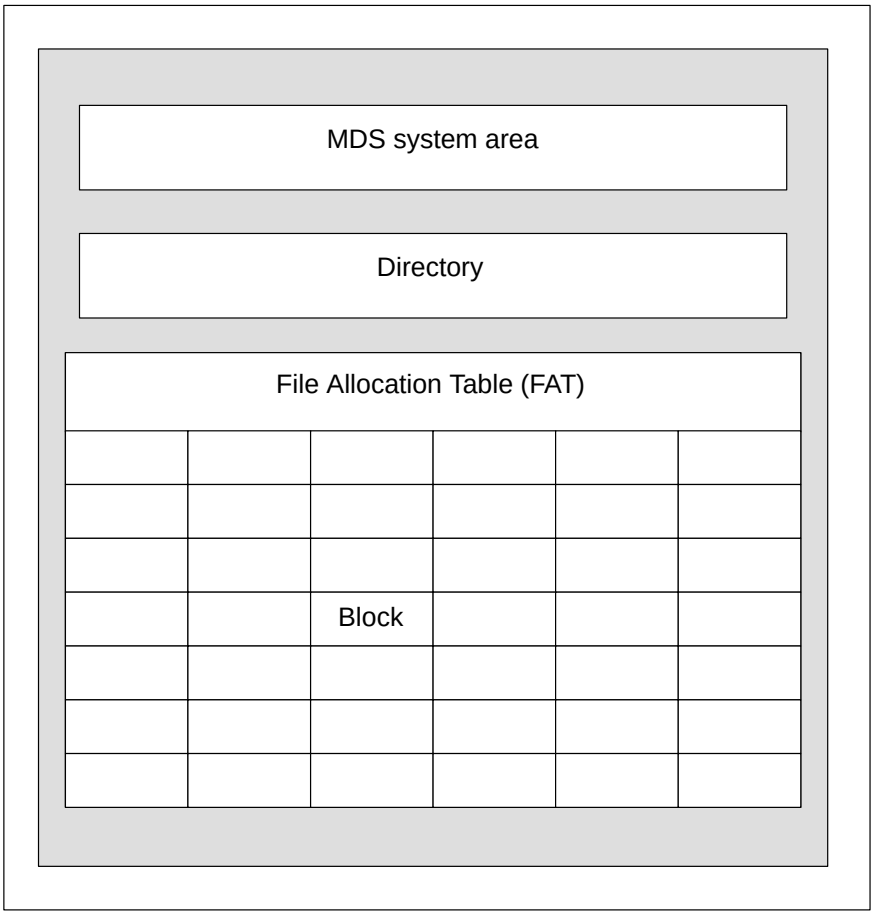
2.1 Filehandler

The Filehandler file management system facilitates a simple means of managing files on the MDS. The system has the characteristics of a DOS operating system:

- The user can use filenames to address his data.
- Files of different sizes can be managed.
- Access rights can be assigned to files.

Filehandler	DOS operating system
MDS: The data is stored in a RAM or EEPROM module onboard the MDS.	**Floppy disk:** The data is stored on a floppy disk.
SLG: The SLG modulates the data inbound from the MOBY I/F and demodulates the data inbound from the MDS. The MDS must be positioned within the transmission range of the SLG. Depending on the type of SLG employed, the transmission range may be several centimetres.	**Floppy disk drive:** The floppy disk drive modulates the data from the PC into a write current and demodulates the read current. The positioning of the floppy disk above the read/write head must be very precise because the magnetic fields are very small.
PLC: The MOBY I/F and PBK comprise the interface to the PLC.	**PC:** The PC controls the floppy disk drive via an interface.

2.2 Internal Structure of Mobile Data Storage Unit



MDS system area contains all MDS-specific data (e.g., MDS name, MDS type, last SLG number, etc.).

The directory contains the names of all files being stored, together with the corresponding parameter values. In each case, the number of available directory entries depends on the respective MDS type.

The FAT File Allocation Table handles the allocation of all data blocks belonging to a particular file.

The payload data area is divided into blocks. In each case, both the number of blocks and the block size depend on the MDS type being used.

From the data held in both directory and FAT, the MOBY I/F Channel Module calculates a checksum. This checksum is stored onboard the MOBY I/F.

The “Move” and “Load” commands und Load can be used to write the information describing both the directory and the FAT into the data field of the of the PLC. In this way, the data will be available immediatly after a power failure, without having to be newly calculated.

When a new MDS arrives at an SLG, the MOBY I/F first compares the checksum.

- If the checksum is identical to the checksum stored onboard the MOBY I/F, the system will initiate the execution of the command.
- If the checksum stored onboard the MOBY I/F is not identical with that of the MDS, the MDS checksum will be newly calculated.
  - If the calculated checksum matches that of the MDS, this indicates that the data structure onboard the MDS has been modified. The corresponding directory and FAT information will then be stored onboard the MOBY I/F. This procedure can be extremely time-consuming, causing the production cycle to be halted.

    To prevent this problem from occurring, the manufacturing process should utilize, onboard all MDS units, a standardized, fixed file structure; refer to Section 2.4 Manufacturing Sequence.

  - If the checksum calculation produces a value that differs from the MDS checksum, the MDS will be defective.

2.3 Communications between PBK5MIF and PBK

On the hardware side, communications between the CL500 and the PBK/ MOBY I/F are implemented via a Dual-Port-RAM module.

The software-based interface is implemented on the basis of command and acknowledgement messages.

- When the PLC program initiates a command, it is transferred to the MOBY I/F via data channel 1.
- The filehandler reads the data, starts command-specific functions and then returns a corresponding command acknowledgement to the CL500.
- The PBK5MIF responds by updating the status bits, and by setting the Ready bit.

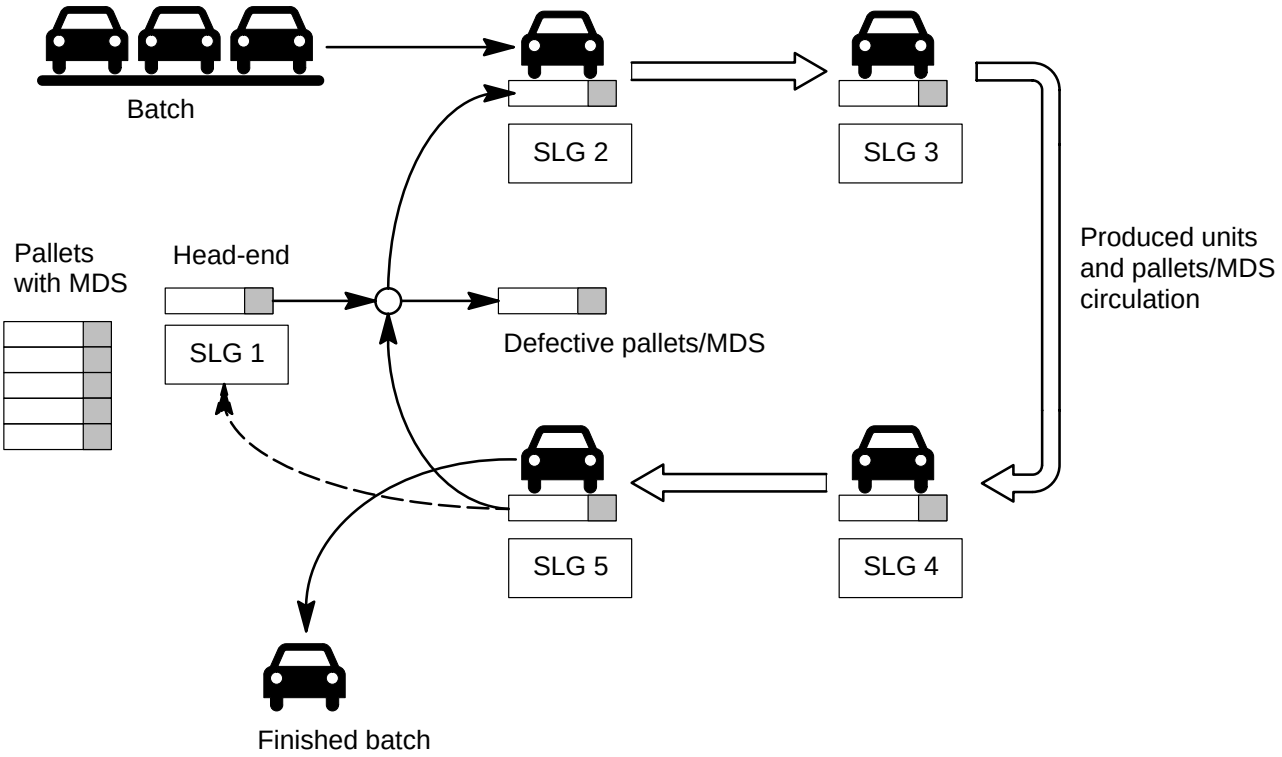
The maximum size of the user data per message (and thus also per cycle) is

- Reading: 240 bytes
- Writing: 226 bytes

2.4 Manufacturing Sequence

To ensure properly timed sequences within the manufacturing process, a uniform and permanent file structure is required for all MDS units. During the actual manufacturing process, data access by SLG Read/write units on the various MDS is read/write only.

Time-intensive commands, such as “Format”, are executed onboard a special SLG (head-end) outside of the actual manufacturing process. This ensures that all MDS units within the manufacturing process always utilize the same file structure. The checksum is therefore always identical, dispensing with the requirement of time-intensive reloading of MDS directory information and File Allocation Table.



As dictated by requirements, an MDS at the end of the production process (SLG 5) can be reprocessed at SLG 1 (head-end).

The head-end performs the following functions:

- Formatting MDS units.
- Creating files with the use of the “Create” command.
- Sorting out defective MDS units.
- Monitoring production data.

3 PBK5MIF Module Structure

This chapter describes the internal structure of the PBK5MIF Software Module. The first section discusses the module call in conjunction with the PBK5MIF parameters. The second section explains the PBK5MIF data structure, and the division into the data modules.

3.1 Differences between PROFI and WinSPS-based programming

The present documentation discusses the representation of constants and program module calls in the notation generated by the PROFI programming device software. With the use of WinSPS however, constants and program module calls will appear in different notation due to the adaptation to the IEC1131-3 standard.

The referred differences are shown in the following side-by-side comparison:

Differences in programming and notation of word constants

Data type		PLC utility programs	
Explanation	**Notation**	**PROFI**	**WinSPS**
UINT unsigned integer	Binary/Dual	K00000000 00000000B K11111111 11111111B	2#0000000000000000 2#1111111111111111
	Decimal, word	K00000D – K63535D	00000 – 65535
	Decimal, byte/byte	K000/000 – K255/255	in IEC1131 Part 3 not defined
	Hexadecimal	K0000H – KFFFFH	16#0000 – 16#FFFF
INT integer	Decimal, word	K–32768 – K+32767 K–32768D – K+32767D	–32768 – +32767
Text, STRING(2)	ASCII	K'AB'	'AB'
Time value, TVALUE	Time value (+time base r) r: 0 = 10 ms, 1 = 100 ms 2 = 1 s, 3 = 10 s	K0.r – K1023.r	T#10ms – T#10230s T#0.r – T#1023.r

Differences in programming and notation of module calls

	PLC utility programs	
	PROFI	**WinSPS**
Program module/function call (IEC1131-3)	CM PM	CM FC

Differences in programming and notation of jump instructions

	PLC utility program	
	PROFI	**WinSPS**
Jump instruction	JPx –label	JPx label
Jump destination	–label	label:

3.2 Module Calls and Parameters

Module call

The module call for the PBK5MIF must be issued cyclically, including the initial program cycle.

The PBK5MIF utilizes 4 parameters:

- 3 input parameters
  - KanalNr (channel number)
  - WorkDBNr (working data module no.)
  - PBK Adr (PBK address)
- 1 output parameter
  - Status

```
CM       -PBK5MIF,4          ;Module call

P0    W  K1D                 ;Channel number
P1    W  K204D               ;Data module number
P2    W  AZ8                 ;Start address
P3    BY -StatusK1
```

P0 KanalNr

Channel number of MOBY I/F channel module onboard the PBK, word.

This parameter value transfers the channel number of the MOBY I/F to the PBK5MIF. The channel number must be between 1 and 4, e.g., K1D for channel 1.

P1 WorkDBNr

Data module number for the work area of the PBK5MIF, word.

For each MOBY I/F channel module installed onboard the PBK, one WorkDB data module (min. 450 Byte) must be generated. This parameter value transfers the data module number as a numerical value without operand identifier. The permissible value is between 0 and 511 (effective with function block v1.7), e.g., K16D for data module DB16.

P2 PBKAdr

PBK start address, word

The PBK occupies 4 input bytes and 4 output bytes on the CL500 control unit.

The PBK start address can only be a multiple of 4. The PBK start address must be entered with the operand identifier. The output range must be specified. The input range will then be automatically reserved in parallel with the input range. The following address ranges are permitted:

- IO0 through IO60
- EO0 through EO60

The PBK start address is selected by setting the S1 DIP switch on the PBK.

The address range is set by positioning jumpers JP1 and JP2.

☞ **Refer to technical documentation:**
CL500
PBK Peripheral Bus Interface Module
Module Description
Order no. 1070 070 133

The following assignments apply to address ranges:

PBK address range	Jumpers JP1/JP2	Address range in PLC program	Example of P2
I/O field	2 through 3	II/IO	IO4
EI/EO field	1 through 2	EI/EO	EO4

P3 Status

PBK5MIF status, byte

The value of parameter P3 transfers the status of the PBK5MIF to the PLC program.

Bit	Name	Explanation
0	Active	1: PBK5MIF is enabled for command execution 0: PBK5MIF is not active
1	WorkDB	1: The data module specified with parameter P1 does not exist, or has not been programmed with the required 450 byte minimum size. 0: No error
2 through 6		not used
7	Error	1: An error has occurred during command processing. The precise cause of the error is stored in the WorkDB data module, in D7 Type and D8 ErrCode; refer to Chapter 4, WorkDB Working Area, and Chapter 9, Error Messages. ☞ **The bit remains set (HIGH) for the duration of only one PLC cycle.** For this reason, the data in D7 and D8 of the WorkDB data module must be interpreted or stored immediately. 0: No error detected.

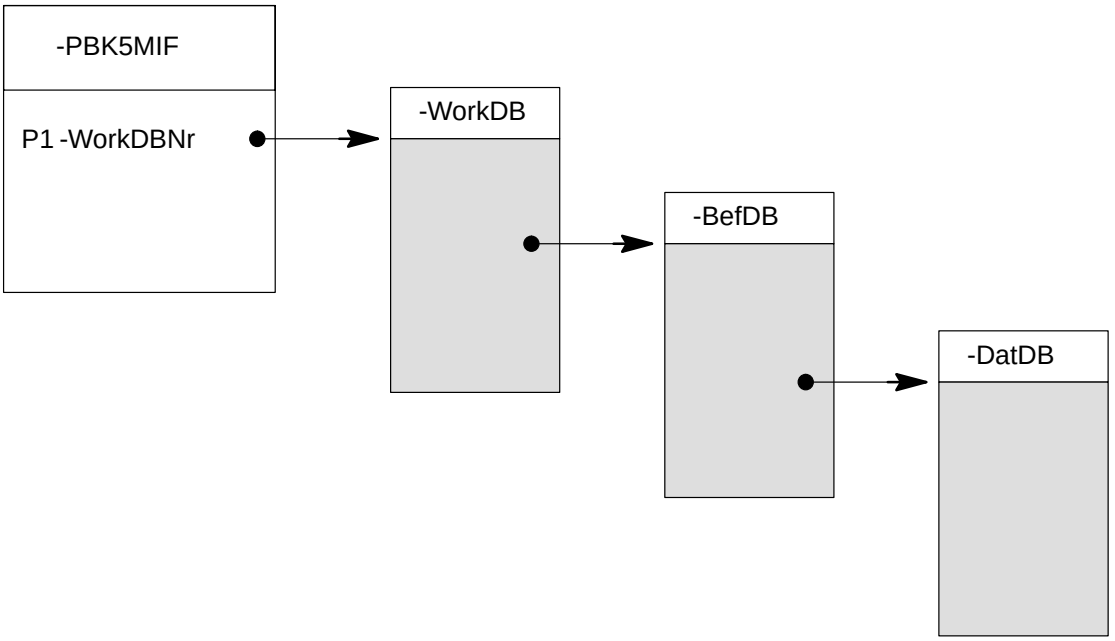
3.3 PBK5MIF Data Structure

The data structure of the PBK5MIF encompasses 3 areas:

- Working area : WorkDB, refer to Chapter 4
- Command area : BefDB, refer to Chapter 5
- Data area : DatDB, refer to Chapter 6

The areas are linked by means of pointers.

Example of PBK5MIF data structure distribution to data modules



Example of data module layout table:

DM No.	Name	Comment	R/E	Length
DM 0			R	0
DM 1	WorkDB1	Work area for channel module 1	R	450
DM 2	BefDB	DM contains command parameters	R	98
DM 3	QuellDB	Source data for “Write” command	R	512
DM 4	ZielDB	Destination area for “Read” command	R	512
DM 5	DirDaten	Destination area for “Dir” command	R	512

☞ **The WorkDB data module must always be present and may not be used for other data!**

A WorkDB must be created for each MOBY I/F channel module, e.g., WorkDB1, WorkDB2, and so forth.

☞ **Appropriate programming can be used to combine the BefDB and DatDB data modules.**

To effect the division, or segregation, of commands and data, two structures are possible:

- All commands in one data module, and the data in one or more separate data modules (refer to illustration).
- Each command can be stored in a separate data module containing the parameters and data for that command.

In the example, all commands are entered in one data module. The data is handled by three data modules.

- One QuellDB source data module, containing the data for the “Write” command.
- One ZielDB destination data module, into which the data for the “Read” command is written.
- One DirDB directory data module, into which the “Dir” command writes the MDS directory.

4 WorkDB – Working Area Data Module WorkDB

☞ **A WorkDB data module must be created for each MOBY I/F channel module.**

The size of the data module must be a minimum of 450 bytes. The WorkDB data module encompasses 3 data areas:

- PBK parameters, D0 through D14
- MOBY I/F parameters, D18 through D28
- Internal area, from D30 onwards

☞ **The WorkDB data module must always be present, and may not be used for other data!**

4.1 PBK Parameters

☞ **Some data in the WorkDB data module are processed and written in byte-wise fashion!**

DB 1 Name: WorkDB1 Comment: FB working area for Channel1							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field / Comment**	**Explanation**	**F**
D 0	2	PBKCom	binary	N	0	PBK commands, all-purpose	B
D 2	2	BefDBNr	decimal	N	2	Pointer at parameterized command	D
D 4	2	BefOff	decimal	N	0	Pointer at parameterized command	D
D 6	1	Chk	ASCII	N	0	Data transfer with/without checksum	
D 7	1	Typ	ASCII	N		Refer to Chapter 9, Error Messages	
D 8	4	ErrCode	ASCII	N		Refer to Chapter 9, Error Messages	
D 12	1	Typ/Code	ASCII	N		Active command type	
D 14	1	Module	decimal	N	00	Channel module type onboard the PBK	H
D 15	1			N	not assigned		

D0 PBKCom

Bit	Description
0	Start signal
1 through 15	not assigned

Start signal for the PBK5MIF to initiate execution of the command parameterized in the BefDB data module. The status, and not the Der Zustand, and not the edge, of Bit 0 is interpreted. Bit 0 must be set by the PLC program. Subsequent to the start of command execution, the PBK5MIF resets the bit.

D2 BefDBNr

Pointer to the BefDB data module. This data module stores the command that is to be applied on the MDS.

Example of D2:
DM2 data module: 2 D

D4 BefOff

Pointer to the data word in the BefDB data module containing the command.

Example of D4:
Offset for data word number 0 : 0 D

D6 Chk

Enabling/disabling the control byte which safeguards data transmission between the ZS and the PBK on the system bus of the CL500.

Data type: ASCII

- 1 : Operation with control byte
- 0 : Operation without control byte

☞ **The control byte extends the cycle time for reading files by the factor 1.3, and for writing to files by the factor 1.5.**

D7 Typ

The value in D7 will be valid only as long as Bit 7 in status parameter P3 of the PBK5MIF is set.

The error type is described with ASCII characters 0 or 1. For a detailed description, refer to Chapter 9, Error Messages.

D8 ErrCode

The value in D8 will be valid only as long as Bit 7 in status parameter P3 Status of the PBK5MIF is set.

The error code consists of 4 ASCII characters. For a detailed description, refer to Chapter 9, Error Messages.

D12 Typ/Code

D12 contains the current or the last-processed command type and command code.

D14 Modul

D14 contains the code for the module type:

- MOBY I/F 00

4.2 MOBY I/F Parameters

☞ **Some data in the WorkDB data module are processed and written in byte-wise fashion!**

DB 1 Name: WorkDB1 Comment: FB working area for Channel1						RAM / EPROM: R	
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field / Comment**	**Explanation**	**F**
D 16	2	Reserve					D
D 18	2	SLGNr	decimal	N	65535; Test number	Read/write unit number	D
D 20	1	EAKO	ASCII	N	4	Control of approach/departure monitoring	
D 21	1	ECC	ASCII	N	0	with/without error correcting code	D
D 22	1	ResetTyp	ASCII	N	C	“Reset” FH command cancellation type	
D 23	1	ABTA	hex	N		Scanning time when addressing an MDS	B
D 24	2	FHCom	binary	N	0	Short commands without parameters in BefDB	B
D 26	2	FHStatus	binary	N	0	Status messages from file handler module	B
D 28	2	DatSize	decimal	N	0	Size of received payload data in bytes	D

D18 SLGNr2

SLG Read/write unit number; this number is written to the MDS prior to every processing task. It enables the Filehandler to distinguish the MDS from one another.

Possible SLG numbers range from 1 through 65534.

CAUTION
The number 65535 has been reserved for testing purposes. It has the effect that the next MDS subsequent to a “Next” command can be the same MDS.

☞ **Changes to data word D18 will take effect only after a Reset has been performed in D24 of FHCom.**

D20 EAKO

Control of approach/departure monitoring, refer to Chapter 8, EAKO – Approach/Departure Monitoring.

Data type: ASCII

- 0, Presence fluctuation are reported to the PLC program. The “Next” command must be issued by the PLC program.
- 1, Prior to the initiation of the command, an MDS must be within transmission range of the SLG. The “Next” command is not mandatory.
- 4, The detection or recognition of an MDS is handled by the PLC program, e.g., via position switch.

☞ **Changes to data word D20 will take effect only after a Reset has been performed in D24 of FHCom.**

D21 ECC

Error Correcting Code, ECC driver for increased data security on the MDS.

☞ **The ECC driver extends the access time to the data onboard the MDS.**

Data type: ASCII

- 0, ECC driver disabled
- 1, ECC driver enabled

☞ **Changes to data word D21 will take effect only after a Reset has been performed in D24 of FHCom.**

D22 ResetTyp

The ResetTyp (Reset type) indicates the type and mode of the abort (or reset).

Data type: ASCII

There are 2 kinds of ResetType:

- B, Hard Reset, the Reset occurs immediately and without concern for the data structure.
- C, Soft Reset, e.g., within the framing of a “Write” command, causes the currently valid internal MDS system areas (DIR, FAT) to be updated.

CAUTION
The responsibility for data rests with the user!

☞ **Changes made in data word D22 will take effect only after a reset has been performed in D24 of FHCom.**

D23 ABTA

Scanning interval in the presence of an MDS. The parameterization described below is applicable for all MDS types of the MOBY-I product range.

Bit	Description
0 through 5	Time value
6 and 7	Time factor

Time value

- 0 through 3Fh (0 = continuous field scan)

Time factor:

- 00 = 0,01 s
- 01 = 0,1 s
- 10 = 1,0 s
- 11 = 10 s

D24 FHCom

Bit	Description
0	Reset
1	Next
2 through 15	internal use

Explanation:

- Reset

  The Reset is initiated by the PLC program or PBK5MIF, and accepted at any point in time. Subsequent to the start of the command, the Reset bit is reset by the PBK5MIF.

  The Reset command resynchronizes data exchange and also contains the “Next” command.

- Next

  The “Next” command is set by the PLC program. The processing of an MDS is concluded. The next command is executed with the next MDS that follows.

  In plants requiring critical timing accuracy, this procedure can be used to transfer the next command for the succeeding MDS already at a point where the old MDS is still within the transmission range of the SLG Read/write unit, refer to Chapter 8, EAKO – Approach/Departure Monitoring.

D26 FHStatus

Bit	Explanation
0	ECC
1	Bat1
2	Bat2
3	War1
4	War2
5	Anw
6	STG enabled
7	Sys
8 through 15	not assigned

Explanation:

- ECC
  The MDS data error correction was carried out.
- Bat1
  The MDS battery must be replaced.
- Bat2
  The auxiliary battery of the MDS must be replaced immediately! (Applies to MDS507 only, otherwise always 1.)
- War1
  Warning. During command processing, the MDS was located outside of the SLG transmission range. Command execution was completed. However, there exists a possibility that the data read and/or write access onboard the MDS was incomplete.
- War2
  Warning. The correct execution of the “Read” command is not possible because the end of the file was reached.
- Anw
  Presence of an MDS within the SLG transmission range.
  - 0 no MDS within SLG transmission range
  - 1 MDS within SLG transmission range
- STG enabled
  Goes HIGH in the following conditions:
  - An STG 4F Service and Testing Unit is communicating with the PBK. and
  - A command awaiting execution is present.

☞ **This may delay execution of the command.**

- Sys
  The checksum of the MDS positioned within the SLG transmission range fails to match the checksum stored onboard the MOBY I/F, refer to Section 2.2, Internal Structure of Mobile Data Storage Unit.

☞ **This extends the execution time for the current command!**

D28 “transmission range” DatSize

D28 contains information about the received payload data.

- Subsequent to invoking the “Read” command with the addition of the “Read Entire File” switch, the address D28 DatSize will contain the file size in bytes.
- Subsequent to invoking the “Dir” command, the address D28 DatSize will contain the number of the files stored onboard the MDS.

5 BefDB – Command Area Data Module

The PBK5MIF differentiates between two types of commands:

- Commands without parameterization:
  - “Reset” and
  - “Next”

  These are invoked by setting the associated bit in D24 of FHCom in the WorkDB data module.

- Parameterized commands,

  these are invoked by setting Bit 0, Start signal, in D0 of PBKCom in the WorkDB data module.

This chapter describes the parameterized commands and related programming procedures.

The parameterized commands must be written into a data module. The payload data for the commands must also be written into a data module, refer to Chapter 6, Datenbereich DatDB.

D2, BefDBNr, of the WorkDB data module contains the BefDB data module in which the parameterized commands are stored.

D4, BefOff, of the WorkDB data module contains the data word in the BefDB that contains the selected command. The subsequent data words contain the parameter values for the command.

☞ **Each command block in the BefDB data module must begin with an even-numbered byte address!**

Example of the command area, BefDB data module

DM 2 Name: BefDB Comment: DM contains command parameter values							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/Comment**	**Explanation**	**F**
D 0		Read	ASCII	N	1R		
D 2			ASCII	N	File.001		
D 10			Word	N	0		D
D 12			Word	Y	–1		D
D 14			Word	N	4		D
D 16			Word	N	0		D
D 18		Write	ASCII	N	1W		
D 20			ASCII	N	File.001		
D 28			Word	Y	0		D
D 30			Word	N	512		D
D 32			Word	N	3		D
D 34			Word	N	0		D
D 36		Format	ASCII	N	1I		
D 38			ASCII	N	[BOSCH]		

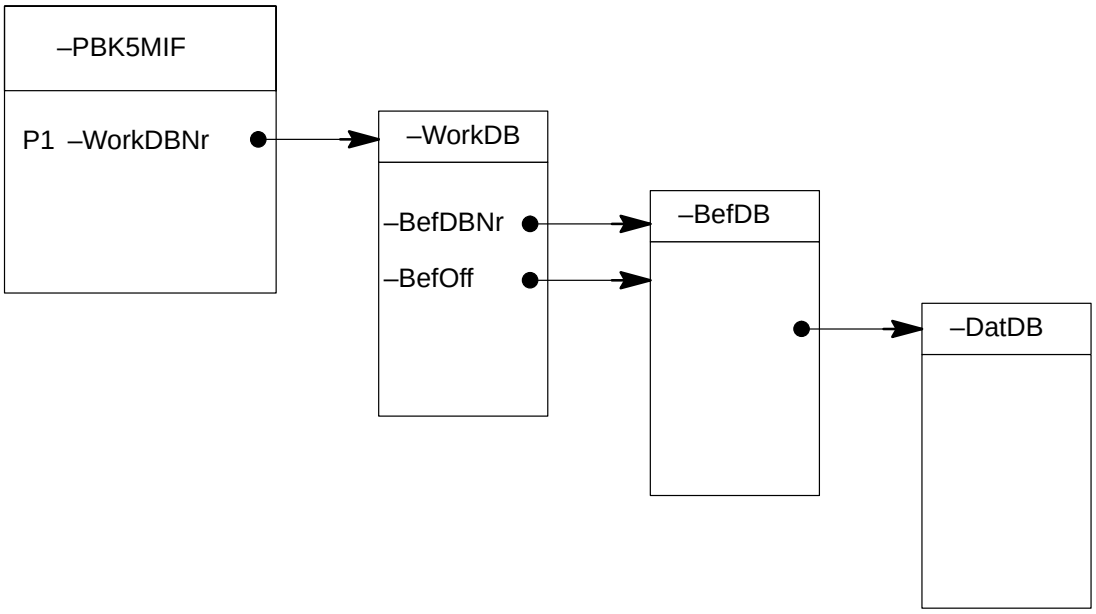
Parameterizable commands

Command	Explanation	Command type/Code
Format	Format MDS	1I
Create	Create new file onboard formatted MDS	1B
Cover	Lock up MDS file structure	1G
Write	Write data to file	1W
Read	Read data from file	1R
Delete	Delete file onboard MDS	1D
Attribute	Assign attribute to file	1Y
Dir	Read MDS directory	1G
Load	MDS system area (PLC → MOBY I/F)	1L
Move	MDS system area (MOBY I/F → PLC)	1M

Initiating a parameterized command

☞ **Refer also to Chapter 7, Sample Program, Automatic “Read” command initiation.**

Several data modules, connected by pointers, are required to initiate and execute a command.



- Transfer the WorkDBNr data module number for the work area of the PBK5MIF in parameter P1.
- Enter data module for BefDB commands in D2, BefDBNr, of the WorkDB data module.
- Enter pointer to the data word in the BefDB data module containing the command in D4, BefOff.
- Set start signal, Bit 0 in D0 of PBCom of WorkDB data module, to HIGH (1).

5.1 Format

The “Format” command is used to format an MDS. The MDS is given a name, the type, and an attribute.

CAUTION
Entire MDS data will be deleted!

DM 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Explanation**	**F**
D	36	Format	ASCII	N	1I		
D	38		ASCII	N	[BOSCH]		
D	46		Word	N	0006		H
D	48		Word	N	0001		D

Command structure

No.	No. of bytes	Designation	Description
D36	2	Command type/Code	1I
D38	8	MDS name	max. 8 ASCII characters
D46	2	MDS type	refer to “MDS Type” table
D48	1	Attribute	refer to “Attribute” table

MDS type

☞ **If an MDS is formatted for ECC operation, D21, ECC, must also be set, refer to Section 4.2, MOBY I/F-Parameters.**

MDS type (hex)	**MDS capacity** (brutto), * operating with ECC	**No. of files**	**Payload data** (Bytes)
01 02 81	62 62 42	1 2 1	27 12 7
03 83	128 112 *	3 3	60 45
04 84	2,045 1,778 *	16 16	1,680 1,440
05 85	8,189 7,154 *	32 32	7,456 6,464
06 86	32,765 28,658 *	64 64	31,488 27,520
07 87	65,533 57,330 *	128 128	62,976 54,784
08 88	131,069 114,674 *	256 256	126,720 110,336
09 89	262,141 229,362 *	256 256	257,024 224,256
0A 8A	524,285 458,738 *	512 512	514,048 448,768
0B 8B	1,048,573 917,490 *	512 512	1,035,264 904,960

Attribute

Bit	**Explanation**
0	0 Always reads entire system area 1 Enables checksum mechanism
1	0 Deletes entire old directory 1 Sets up the old MDS directory on the MDS. Retains the existing file payload data size. If LOW (0), the payload data area will be deleted.
2	0 Deletes entire MDS 1 Deletes DIR and FAT, retains payload data area.
3	0 normal 1 In each case, counters for write cycles and/or EEC corrections are retained. The counters will be retained even if the system area is faulty.
4 through 7	not used

5.2 Create

The “Create” command generates a new file onboard a formatted MDS. The file is entered in the directory and is assigned the file size 0.

☞ **It is not permitted to create a file with a file name that is already in use.**

DM 2 Name: BefDB Comment: DM contains command parameter values							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Explanation**	**F**
D	50	Create	ASCII	N	1B		
D	52		ASCII	N	File.001		
D	60		Word	N	0		D

Command structure

No.	No. of bytes	Designation	Explanation
D50	2	Command type/Code	1B
D52	8	File name	max. 8 ASCII characters
D60	1	Attribute	0 No attribute 1 Read-only 2 Single write access 4 Write access may not increase the file size. 5 As in 4h, but the file may only be read-accessed.

5.3 Cover

The “Cover” command can be used to lock the file structure of an MDS. This means that any commands causing checksum modifications and/or additional DIR and/or FAT entries, can be transmitted only by specially autorized users.

Unauthorized access has occurred, if:

- Files are deleted and/or created.
- File sizes are reduced and/or increased.
- Data storage units (data media) are formatted in the event that these are formatted already.
- Attributes are deleted and/or newly assigned.

DB 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Explanation**	**F**
D		COVER					
D							
D							

Command structure

No.	No. of bytes	Designation	Explanation
D	2	Command type/Code	1C
D	8	File name	max. 8 ASCII characters
D	1	Attribute	0 Modification of MDS file structure again permitted 1 MDS is to be locked

5.4 Write

The “Write” command can be used to write payload data from a data module to a file which was created onboard a formatted MDS using the Create command.

DM 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Explanation**	**F**
D	18	Write	ASCII	N	1W		
D	20		ASCII	N	File.001		
D	28		Word	Y	0		D
D	30		Word	N	512		D
D	32		Word	N	3		D
D	34		Word	N	0		D

Command structure

No.	No. of bytes	Designation	Explanation
D18	2	Command type/Code	1W
D20	8	File name	max. 8 ASCII characters
D28	2	Offset	Start position for data write-access in the file.
D30	2	Size	Size of payload data, in bytes, that is to be written.
D32	2	Data module number	DM containing the data to be written.
D34	2	DBOffset	Position in the DM marking the start point for the data write-access.

D28 Offset

The offset can be used to define the position within the file that will be the starting point for data write-access. If the data is to be appended to the existing end of file, the offset value must be defined as –1.

5.5 Read

The “Read” command is used to read data from the MDS and write them into a data module.

DM 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Explanation**	**F**
D	0	Read	ASCII	N	1R		
D	2		ASCII	N	File.001		
D	10		Word	N	0		D
D	12		Word	Y	–1		D
D	14		Word	N	4		D
D	16		Word	N	0		D

Command structure

No.	No. of bytes	Designation	Explanation
D0	2	Command type/Code	1R
D2	8	File name	max. 8 ASCII characters
D10	2	Offset	Start position for readaccess in the file.
D12	2	Size	Size of payload data, in bytes, that is to be read.
D14	2	Data module number	DM to which the data from the MDS is to be written.
D16	2	DBOffset	Start position for data write-access in the DM.

D12 Size

If the size is defined as “–1”, the entire file will be read. The size of the payload data, expressed in bytes, can be taken from D28, size, in the WorkDB data module; refer to Section 4.2 MOBY I/F Parameters.

5.6 Delete

The “Delete” command removes a file from the MDS.

CAUTION
All data in the file will be lost!

DM 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Explanation**	**F**
D	68	Delete	ASCII	N	1D		
D	70		ASCII	N	File.001		

Command structure

No.	No. of bytes	Designation	Explanation
D68	2	Command type/Code	1D
D70	8	File name	max. 8 ASCII characters

5.7 Attribute

The “Attribute” command can be used to assign read or write attributes to a file.

DM 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Explanation**	**F**
D	78	Attribute	ASCII	N	1Y		
D	80		ASCII	N	File.001		
D	88		Word	N	0		D

Command structure

No.	No. of bytes	Designation	Explanation
D78	2	Command type/Code	1Y
D80	8	File name	max. 8 ASCII characters
D88	1	Attribute	0 No attribute 1 Read-only, file data cannot be modified 2 One-time write-access following “Create” 4 File size must not be changed by commands requiring write-access. 5 As in 4h, but the file may only be read-accessed.

5.8 Dir

The “Dir” command writes the directory of the MDS into a selected data module.

DM 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Description**	**F**
D	62	Dir	ASCII	N	1G		
D	64		Word	N	5		D

Command structure

No.	No. of bytes	Designation	Explanation
D62	2	Command type/Code	1G
D64	2	Data module number	DM to which the MDS directory is written.
D66	2	DBOffset	Start position for directory write-access in the data module.

The directory can be written into a special DirDaten data module.

Directory structure in the DirDaten data module.

DM 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Explanation**	**F**
D	0	MDS Name	ASCII	N	.....		
D	8	ChkSum	Word	N	0000		H
D	10	freeSpce	Word	N			H
D	12		Word	N	0000		H
D	14	Name1	ASCII	N	.....		
D	22		Word	N			H
D	24		Word	N			H
D	26	Name2	ASCII	N	.....		
D	34		Word	N			H
D	36		Word	N			H

Command structure

No.	No. of bytes	Designation	Explanation
D0	8	MDS Name	max. 8 ASCII characters
D8	2	ChkSum	Internal checksum
D10	4	freeSpce	Remaining available memory for payload data onboard MDS.
D14 through D24		Repeated for every file in the directory.	
D14	8	Namex	File name comprising max. 8 ASCII characters
D22	3		File size in bytes
D24	1		0 no attribute 1 read only 2 write once

Data word D28, DatSize, in the WorkDB data module contains the number of files stored onboard the MDS.

In the event that the DirDaten data module is configured with insufficient size, the data will be written to the end of the data module, with the remainder being lost. In the WorkDB data module, an error code is stored at address D8, ErrCode; refer also to Chapter 9, Error Messages.

5.9 Load

The “Load” command is used subsequent to the system startup to load the MDS system area stored in the data field of the PLC into the MOBY I/F channel module.

DM 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Explanation**	**F**
D	94	Load	ASCII	N	1L		
D	96		Word	N	0		D

Command structure

No.	No. of bytes	Designation	Explanation
D94	2	Command type/Code	1L
D96	2	Offset	Offset in data field to the MDS system area Value range: K0D through K24532D

5.10 Move

The “Move” command is used to store the system area of the current MDS in the data field of the PLC control unit. The command can be carried out, for example, if Bit 7, Sys, in data word D26, FHStatus, of the WorkDB data module indicates that the system area of the MDS has undergone a change.

DM 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/comment**	**Explanation**	**F**
D	90	Move	ASCII	N	1M		
D	92		Word	N	0		D

Command structure

No.	No. of bytes	Designation	Explanation
D90	2	Command type/ Code	1M
D92	2	Offset	Offset in data field under which the MDS system range is to be stored. Value range: K0D through K24532D

☞ **The administration of the data field is the responsibility of the user.**

Area overlaps are not detected by the system. Only the data field limit DF24575 is monitored.

The “MDS System Area Size” table below indicates the number of bytes that must be reserved in the data field to accommodate the system areas of the various MDS types.

MDS System Area Size

MDS type * operating with ECC	Size of MDS system area (bytes)
01H 02H 81H	25 41 25
03H 83H	63 61
04H 84H	443 413
05H 85H	923 861
06H 86H	1,397 1,335
07H 87H	2,785 2,657
08H 88H	4,583 4,455
09H 89H	5,601 5,345
0AH 8AH	11,193 10,687
0BH 8BH	15,265 14,247

6 DatDB – Data Area Data Module

Data exchange for the commands

- Write
- Read
- Dir

is handled via the DatDB data module.

☞ **The data modules BefDB and DatDB can be combined by adjusting the program accordingly.**

In the command frame for the respective command in the BefDB data module, the data module handling data traffic must always be entered with its data module number and the DBOffset value.

The following tables provide an example of the “Write” command.

DM 2 Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**S**	**Data field/Comment**	**Explanation**	**F**
D	18	Write	ASCII	N	1W	D	
D	20		ASCII	N	File.001	D	
D	28		Word	Y	0	D	D
D	30		Word	N	512	D	D
D	32		Word	N	3	D	D
D	34		Word	N	0	D	D

No.	No. of bytes	Designation	Explanation
D32	2	Data module number	DM which contains the data to be written.
D34	2	DBOffset	Position in the DB marking the start of data writeaccess.

The data for the “Write”, “Read” and “Dir” commands may also be clearly divided over 3 data modules.

Data module layout table

DM No.	Name	Comment	R/E	Size	
DM 3	QuellDB	Source data for “Write” command	R	512	
DM 4	ZielDB	Destination area for “Read” command	R	512	
DM 5	DirDaten	Destination area for “Dir” command	R	512	

- A QuellDB data module, containing the data for the “Write” command.
- A ZielDB data module, into which the data of the “Read” command is written.
- A DirDB data module, into which the “Dir” command writes the directory of the MDS.

Notes:

7 Sample Program

To describe the PBK5MIF software module, short extracts from a sample program have been used throughout Chapters 2 through 5. In the present chapter, this sample program is shown in full length.

This sample program is supplied as a component of the standard PBK5MIF software package; it is located on the diskette labelled: Function modules for PBK Peripheral Bus Interface Module, Order no.: 1070 070 042.

The sample program comprises the following files:

- Module file OB1.P5O
- Module file Anlauf.P5O
- Module file PBK5MIF.P5O
- Symbol file Beispiel.S5S

Program requirements

The sample program will be executable without modifications only provided the following conditions are met:

- PBK start address: AZ20
- A MOBY I/F v1.3 channel module is installed in module slot M1 (Channel 1)
- MDS is properly formatted
- A file named File.001 is available.

The sample program was tested in conjunction with the following components:

- MOBY I/F v1.3
- SLG 1
- MDS 505, 32k

7.1 OB1.P5O

The module call for the PBK5MIF must be issued cyclically, including the initial cycle (Cycle 1). It is therefore useful to program the module call instruction in OM1.

PBK5MIF parameter definitions

```
DEF     K1D,-Kanal1         ;Module slot M1 of MOBY I/F channel module
DEF     K1D,-WorkDB1        ;Data module number for WorkDB1 for module slot M1
DEF     AZ20,-PBKAdr        ;Start address of PBK, extended I/O, byte address 20
DEF     M0,-StatusK1        ;PBK5MIF status message for module slot M1
```

WorkDB definitions

```
DEF     D0,-PBKCom          ;PBK command, all-purpose
DEF     D7,-Typ             ;Error type
DEF     D8,-CodeHigh        ;Error code Bytes 1 and 2
DEF     D10,-CodeLow        ;Error code, Bytes 3 and 4
DEF     D24,-FHCom          ;Reset and Next commands
DEF     D26,-FHStatus       ;Status message of PBK5MIF
```

Mask definition

```
DEF     K0001H,-Start       ;Mask for bit position 0
```

PBK5MIF module call instruction

```
CM      -PBK5MIF,4
P0      -Kanal1             ;Module slot M1 of MOBY I/F channel module
P1      -WorkDB1            ;Data module number for WorkDB1 for module slot M1
P2      -PBKAdr             ;Start address der PBK, extended I/O, byte address 20
P3   BY -StatusK1           ;PBK5MIF status message for module slot M1
```

Display of start parameter

```
L    BY -StatusK1,C         ;Load -StatusK1 in register C
```

Is the PBK5MIF active?

```
A    B  M0.0                ;PBK5MIF enabled bit
JPC     -Warten             ;Is the PBK5MIF working? -> yes
```

Has an error occurred?

```
AN   B  M0.7                ;P3 -StatusK1 Bit 7 Error
JPC     -NoError            ;Has an error occurred? -> no
```

Read error type and error code from WorkDB

☞ **Bit 7, Error, is set for the duration of one PLC cycle only.**

The information in D7, type, and D8, ErrCode, of the WorkDB must be interpreted immediately after the PBK5MIF has been called.

```
L        -WorkDB1,A
CM       [A]                ;Open WorkDB1
L    BY  -Typ,A             ;Error type in D7 of WorkDB
L        -CodeHigh,B        ;Bytes 1 and 2 of error code
L        -CodeLow,C         ;Bytes 3 and 4 of error code
```

At this point, there follows the interpretation of the error message, refer also to Chapter 9, Error Messages. In addition, an application-specific response can be programmed.

```
EP

         -NoError
```

Automatic “Read” command initiation

☞ **D2 file name in DB2 of BefDB data module for the “Read” command must be adapted, refer also to table in Section 5.5 and BefDB Table in Section 7.3.**

```
L        -WorkDB1,A
CM       [A]                ;Open WorkDB1
L    W   -Read,A            ;Address of command in BefDM data module
                            ;via register A
T    W   A,-BefOff          ;transfer to D4, BefOff, of WorkDB1

L    W   -PBKCom,A          ;Load D0 of PBKCom
O    W   -Start,A           ;Set Bit 0 HIGH (1)
T    W   A,-PBKCom

         -Warten

EP
```

7.2 Anlauf.P5O

The PBK5MIF module must also be called in the initial cycle. It is therefore useful to program the module call instruction in Anlauf.P50 as well. The trigger pulse, which is necessary for initializing the internal data areas of the PBK5MIF, occurs only in the first cycle.

PBK5MIF parameter definitions

```
DEF      K1D,-Kanal1          ;Module slot M1 for MOBY I/F channel module
DEF      K1D,-WorkDB1         ;Data module number for WorkDB1 for module slot M1
DEF      AZ20,-PBKAdr         ;Start address of PBK, extended I/O, byte address 20
DEF      M0,-StatusK1         ;PBK5MIF status message for module slot M1
```

PBK5MIF module call instruction

```
CM       -PBK5MIF,4
P0       -Kanal1              ;Module slot M1 of MOBY I/F channel module
P1       -WorkDB1             ;Data module number for WorkDB1 for module slot M1
DEF      AZ20,-PBKAdr         ;Start address of PBK, extended I/O, byte address 20
P3    BY -StatusK1            ;PBK5MIF status message for module slot M1

EP
```

7.3 Beispiel.S5S

Program modules

Type	Module name ;Comment	R/E
OM 1	OB1	R
OM 5	Anlauf (startup)	R

Organization modules

Type	Module name ;Comment	R/E
PM 0	PBK5MIF	R

Special markers

Address	Symbol	Comment	Type
SM 22.0	IstZykl	Actual cycle time in ms	
SM 24.0	MaxZykl	Maximum cycle time in ms	
SM 26.0	MinZykl	Minimum cycle time in ms	

Overview of data modules

DM No.	Name	Comment	R/E	Size
DM 1	WorkDB1	Work area for module slot M1	R	450
DM 2	BefDB	Command parameters	R	98
DM 3	QuellDB	Source data for “Write” command	R	512
DM 4	ZielDB	Destination area for “Read” command	R	512
DM 5	DirDaten	Destination area for “Dir” command	R	512

WorkDB1

DB 1 Name: WorkDB1 Comment: Work area FB for Channel1							RAM/EPROM: R
No.	Byte	Symbol	Type	Sg	Data field/Comment	Explanation	F
D	0	PBKCom	Word	N	0		B
D	2	BefDBNr	Word	N	2		D
D	4	BefOff	Word	N	0		D
D	6	Chk/Typ	ASCII	N	1		
D	8	ErrCode	ASCII	N			
D	12	Typ/Code	ASCII	N			
D	14		ASCII	N			
D	18	SLGNr	Word	N	65535; Test number		D
D	20	EAKO/ ECC	ASCII	N	40		
D	22	ResetTyp	ASCII	N	C		
D	24	FHCom	Word	N	0		B
D	26	FHStatus	Word	N	0		B
D	28	DatSize	Word	N	0		D

BefDB

DM 2Name: BefDB Comment: DM contains command parameters							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/Comment**	**Explanation**	**F**
D	0	Read	ASCII	N	1R		
D	2		ASCII	N	File.001		
D	10		Word	N	0		D
D	12		Word	Y	–1		D
D	14		Word	N	4		D
D	16		Word	N	0		D
D	18	Write	ASCII	N	1W		
D	20		ASCII	N	File.001		
D	28		Word	Y	0		D
D	30		Word	N	512		D
D	32		Word	N	3		D
D	34		Word	N	0		D
D	36	Format	ASCII	N	1I		
D	38		ASCII	N	[BOSCH]		
D	46		Word	N	0006		H
D	48		Word	N	0001		H
D	50	Create	ASCII	N	1B		
D	52		ASCII	N	File.001		
D	60		Word	N	0		D
D	62	Dir	ASCII	N	1G		
D	64		Word	N	5		D
D	66		Word	N	0		D
D	68	Delete	ASCII	N	1D		
D	70		ASCII	N	File.001		
D	78	Attribut	ASCII	N	1Y		
D	80		ASCII	N	File.001		
D	88		Word	N	0		D
D	90	Move	ASCII	N	1M		
D	92		Word	N	0		D
D	94	Load	ASCII	N	1L		
D	96		Word	N	0		D

QuellDB

DM 3 Name: QuellDB Comment: Source data for “Write” command						RAM/EPROM: R	
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/Comment**	**Explanation**	**F**
D	0		ASCII	N	Descriptive Text		
D	44		ASCII	N	Descriptive Text		
D	88		ASCII	N	Descriptive Text		
D	132		ASCII	N	Descriptive Text		
D	176		ASCII	N	Descriptive Text		
D	220		ASCII	N	Descriptive Text		
D	264		ASCII	N	Descriptive Text		
D	388		ASCII	N	Descriptive Text		
D	352		ASCII	N	Descriptive Text		
D	396		ASCII	N	Descriptive Text		
D	440		ASCII	N	Descriptive Text		
D	484		ASCII	N	Descriptive Text		

ZielDB

DB 4 Name: ZielDB Comment: Destination area for “Read” command						RAM/EPROM: R	
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/Comment**	**Explanation**	**F**
D	0		ASCII	N	.....		
D	44		ASCII	N	.....		
D	88		ASCII	N	.....		
D	132		ASCII	N	.....		
D	176		ASCII	N	.....		
D	220		ASCII	N	.....		
D	264		ASCII	N	.....		
D	308		ASCII	N	.....		
D	352		ASCII	N	.....		
D	396		ASCII	N	.....		
D	440		ASCII	N	.....		
D	484		ASCII	N	.....		

DirDaten

DB 5 Name: DirDaten Comment: Destination area for “Dir” command							RAM/EPROM: R
No.	**Byte**	**Symbol**	**Type**	**Sg**	**Data field/Comment**	**Explanation**	**F**
D	0	MDS Name	ASCII	N	.....		
D	8	ChkSum	Word	N	0000		H
D	10	freeSpce	Word	N			H
D	12		Word	N	0000		H
D	14	Name1	ASCII	N	.....		
D	22		Word	N			H
D	24		Word	N			H
D	26	Name2	ASCII	N	.....		
D	34		Word	N			H
D	36		Word	N			H
D	38	Name3	ASCII	N	.....		
D			.	.	.		.
D			.	.	.		.
D	482	Name40	ASCII	N	.....		
D	490		Word	N			H
D	492		Word	N			H
D	494	Name41	ASCII	N	.....		
D	502		Word	N			H
D	504		Word	N			H
D	506	Name42	ASCII	N	.....		

8 EAKO – Approach/Departure Monitoring

The term “EAKO approach/departure monitoring” describes a procedure that makes it possible to determine whether an MDS is positioned within the transmission range of an SLG. The user is provided with a choice of three different methods.

EAKO	Explanation
0	• MDS recognition via range scanning. • Command is stored by the Filehandler until an authorized MDS enters the SLG transmission range or leaves that range during command execution. • The “Next” command is mandatory. • Contiguous MDS flow control via the Filehandler.
1	• MDS recognition occurs in the PLC program, e.g., via position switch. • Command execution is immediate. • The “Next” command is optional but not mandatory. • No contiguous MDS flow control.
4	• MDS recognition occurs in the PLC program, e.g., via position switch. • Command is stored by the Filehandler until an authorized MDS enters the SLG transmission range or leaves that range during command execution. • The “Next” command is optional but not mandatory. • No contiguous MDS flow control.

The procedure is selected in data word D20, EAKO, of the WorkDB data module. The data type is ASCII.

8.1 Definition of Terms

Before embarking on the following description of the 3 procedures, a few terms related to approach/departure monitoring shall be defined

☞ **Prerequisite: All SLG (read/write unit) stations feature different SLG numbers.**

SLG number

Each SLG features a uniuqe SLG number. If work is being performed by an MDS on a given SLG, the SLG number of the SLG will be written into the MDS system area. The SLG number is written automatically before the first command is executed with the MDS. Therefore the SLG number reveals whether a new or old MDS is currently positioned within the SLG transmission range.

new MDS

The SLG number of the SLG and the SLG number on the MDS do not match.

old MDS

The SLG number of the SLG and the SLG number on the MDS are identical.

current MDS

All commands currently being executed refer to this MDS.

Example of an operational sequence on an SLG

- The old MDS remains current – with all commands referring to this MDS – until it is concluded by a “Next” command.
- The PLC program sets the “Next” command, i.e.,Bit 1 in D24, FHCom, of the WorkDB data module. All subsequent commands will refer to the next new MDS.
- If the next MDS to enter the SLG transmission range is again the old MDS, this will be recognized by the unchanged SLG number. No command will be executed.
- If a new MDS enters the SLG transmission range and its SLG number is different, the stations’s local SLG number will be written into the system area of the MDS. This turns the new MDS into the current MDS.

8.2 EAKO 0

Subsequent to receiving a command, the PBK5MIF continuously attempts to establish communications with the MDS via the connected SLG. As soon as an MDS appears within SLG transmission range, command processing commences. With this procedure, the PLC program can initiate a command at any ppoint in time. The command will be stored until it can be processed with, or by, an MDS.

In this operating mode, the “Next” command has a decisive function. Starting the “Next” command signifies the following:

- Processing of the old/current MDS has concluded.
- A new command can be initiated immediately following the “Next” command. However, this command will be processed only once a new MDS has arrived.

If the old MDS reenters the SLG transmission range, the new command will not be executed.

Initiation of the “Next” command is always permitted.

This operating mode provides a contiguous MDS flow control.

In the event that a new MDS enters the SLG transmission range before the old/current MDS was concluded with the “Next” command, the procedural sequence will be as follows:

- If a command other than “Next” is initiated, an error message will be returned.
- If the “Next” command is initiated, the old MDS will be concluded, and the new MDS will become the current MDS. All subsequent commands will be executed with this current MDS until processing is terminated with another “Next” command.
- If the current MDS again exits the SLG transmission range without having processed a command, an error condition will result. The error message is returned only once a command has been initiated.

8.3 EAKO 1

In this operating mode, the PLC program a part of the MDS flow control.

When a command is initiated, the PLC program must ensure that an MDS is positioned within the SLG transmission range. The command is executed immediately. If there is no MDS in the SLG transmission range, an error message will be returned.

This operating mode can be utilized in all situations where the PLC program is able to detect the precence of an MDS, e.g., by means of position switches.

The “Next” command can be used but is not mandatory.

8.4 EAKO 4

This operating mode does not provide any means of manipulating the MDS flow control with the PBK5MIF software module. The entire MDS flow control must be handled by the PLC program, e.g., with the use of position switches.

A command will be stored until an MDS enters the SLG transmission range.

The “Next” command can be used but is not mandatory.

9 Error Messages

The PBK5MIF enters error messages in D7, Typ, and D8, ErrCode, of the WorkDB data module. The messages will be valid only if Bit 7 in parameter P3, Status, of the PBK5MIF is HIGH.

A distinction is made between two types of errors:

- Typ 0 Error messages originating in PBK5MIF or PBK.
- Typ 1 Error messages generated by the MOBY I/F.

Some Type 1 error messages are also indicated by the red Error LED on the front panel of the MOBY I/F channel module. Here, the error type is determined by the number of flashing pulses separated by an intervening pause.

☞ **In the standard operating mode, a flashing error indicator may be safely ignored, provided that the system continues to operate correctly.**

Flashing pulses/ErrCode

Flashing pulses	Type	ErrCode
1	1	D001
2	1	C006
3	1	B001
4	1	C002
5	1	C007
6	1	C008
7	1	C009
8	1	C010
9	1	C011
10	1	C012
11	1	C013
12	1	C014
13	1	C015
14	1	C016
15	1	C017

9.1 Error type 0

Error messages originating in PBK5MIF or PBK.

9.1.1 P PBK5MIF parameter error

P001

The PBK cannot be contacted at the address defined in parameter P2 PBKAdr.

Check the settings of the S1 DIP and positions of jumpers JP1/JP2 on the PBK, refer to Section 3.2, Module Calls and Parameters.

P002

The BefDB data module specified in the WorkDB data module via data words D2, BefDBNr, BefDBNr and D4, BefOff, has not been programmed, or has been programmed with excessive size.

P003

Unknown command type in BefDB data module.

P004

Unknown command code in BefDB data module.

P005

Either the data module number specified in the BefDB data module for a “Write”, “Read”, or “Dir” command, or the DBOffset has not been programmed, or has been programmed with excessive size.

P007

A value other than 1 or 0 was found in WorkDB data module, data word D6, Chk.

9.1.2 D Request-specific errors

D001

MOBY I/F transmits a message with an unknown error code.

D002

The received part block does not correspond to the requested part block.

D003

The full data complement received with the “Read” command does not fit into the specified destination data module. The data is being written up to the end of the data module; the remaining data will be lost.

Increase the size of the data module.

D004

Due to a preceding error occurrence, Type 1 MOBY I/F are no longer processed. The only command being processed is the “Reset” command in Bit 0 of data word D24 FHCom in the WorkDB data module.

Initiate “Reset” command.

D005

The MDS system area requested with the “Move” command does not fit into the data field area intended for the subject system area.

Check the Offset for the “Move” command in the BefDB data module.

9.1.3 > System Errors reported by the PBK hardware module

>001

Checkbyte D6 Chk in WorkDB data module is enabled, and an error was detected.

The error may have been caused by unauthorized instances of read/write access to the PBK start address.

>002

MOBY I/F is not connected or is defective.

>003

internal error

>004

internal error

>005

internal error

>006

internal error

>007

Parameter error in BefDB data module.

9.1.4 > System Error reproted by PBK5MIF FB

Fxxxx

A fatal system fault has occurred.

Example: Application program has overwritten internal FB data areas.

Both PBK and FB are completely reinitialized. Active requests that may still be present will be lost.

9.2 Error Type 1

MOBY I/F error messages

9.2.1 A Protocol Errors

A006

Unknown error

A011

The message control parameters are not being received in the correct order.

A015

Checkbyte D6, Chk, in WorkDB data module is HIGH. The check byte generated by the PBK5MIF does not match the command message.

A016

MOBY I/F currently occupied by the STG 4F Service and Testing unit. Command processing will be delayed until the STG 4F has completed its task.

If necessary, reinitiate the command.

9.2.2 B SLG Errors

B001

Flashing red Error LED on MOBY I/F: 3 flashing pulses.

Errors in the connection to the SLG.

- Check the cable between MOBY I/F and SLG.
- Missing 24 V power supply.
- Defective MOBY I/F or SLG.

B002

- EAKO : 1

  A command was initiated but there is no MDS within the SLG transmission range.

- EAKO : 0
  - The old/current MDS has exited the transmission range, and the next/new MDS has entered the SLG transmission range. The old/current MDS was not concluded with the “Next” command.
  - A new MDS has entered the SLG transmission range and is exiting again without a command having been processed with the MDS.

9.2.3 C MDS Errors

C002

Flashing red Error LED on MOBY I/F: 4 flashing pulses.

The MDS reports a memory error. The MDS was never write-accessed or has lost its memory contents due to a battery failure; not applicable to EEPROM-equipped MDS.

- Replace MDS if Bit 1, battery monitor, Bat in D26 of FHStatus in the WorkDB data module, has gone HIGH.
- Initialize MDS with the STG 4F
- Format MDS with “Format” command.

C006

Flashing red Error LED on MOBY I/F: 2 flashing pulses.

The MDS has exited the SLG transmission range during an ongoing command execution.

- Reinitiate the command.
- MDS is positioned in the border area of the SLG transmission range.

C007

Flashing red Error LED on MOBY I/F: 5 flashing pulses.

- The “Format” command was started with an incorrect parameter. The memory capacity onboard the MDS falls short of the specification given in the command.
- The “Read” or “Write” commands point to an address that does not exist on the MDS.

C008

Flashing red Error LED on MOBY I/F: 6 flashing pulses.

SLG receives interference pulses from its environment.

- External noise field, the noise field can be detected by means of the inductive field indicator of the STG 4F.
- The distance between two SLGs is too small and fails to conform to the project guidelines.
- The connecting cable between MOBY I/F and SLG is faulty, too long or fails to conform to the project guidelines.

☞ **For configuration guidelines, refer to the catalog published by Siemens AG:**
MOBY
Identsystem MOBY-I (MOBY-I Identification System)
E86060-K6910-A101-A2

C009

Flashing red Error LED on MOBY I/F: 7 flashing pulses.

Too many transmission errors have occurred. After several attempts, MDS was unable to receive the complete command or data from the MOBY I/F.

- MDS is positioned in the border area of the SLG transmission range.
- Transmission to the MDS is affected by the external noise field.

C010

Flashing red Error LED on MOBY I/F: 8 flashing pulses.

The file handler has detected an error during data transmission.

- External noise field, the noise field can be detected by means of the inductive field indicator of the STG 4F.
- The distance between two SLGs is too small and fails to conform to the project guidelines.
- The connecting cable between MOBY I/F and SLG is defective, too long or fails to conform to the project guidelines.

☞ **For configuration guidelines, refer to the catalog published by Siemens AG:**
MOBY
Identsystem MOBY-I (MOBY-I Identification System)
E86060-K6910-A101-A2

C011

Flashing red Error LED on MOBY I/F: 9 flashing pulses.

refer to C008

C012

Flashing red Error LED on MOBY I/F: 10 flashing pulses.

MDS is unable to execute the “Format” command. MDS is defective.

C013

Flashing red Error LED on MOBY I/F: 11 flashing pulses.

Error with “Format” command.

- MDS is positioned in the border area of the SLG transmission range.
- MDS is using too much electricity. MDS is defective.

C014

Flashing red Error LED on MOBY I/F: 12 flashing pulses.

MDS cannot be write-accessed.

- The memory capacity onboard the MDS falls short of the specification given in the “Format” command. MDS of appropriate type must be newly formatted.
- MDS is defective.
- EEPROM of the MDS has reached the end of its service life.

C015

Flashing red Error LED on MOBY I/F: 13 flashing pulses.

MDS address range is being exceeded.

Wrong MDS type.

C016

Flashing red Error LED on MOBY I/F: 14 flashing pulses.

An ECC error has occurred. Data on the MDS cannot be read-accessed.

- MDS data is lost, the MDS is defective.
- MDS was not formatted with the use of the ECC driver. MDS must be re-formatted.
- EEPROM onboard MDS has reached the end of its useful service life. Replace MDS.

C017

Flashing red Error LED on MOBY I/F: 15 flashing pulses.

Filehandler is not functioning correctly.

- Check command structure or command sequence.
- Defective MOBY I/F hardware or firmware.

9.2.4 D Request-specific errors

D001

Flashing red Error LED on MOBY I/F: 1 flashing pulse.

Filehandler has not yet been initialized by means of the “Reset” command.

This condition can be terminated only through the use of the “Reset” command.

D005

The “Format”, “Create”, or “Write” were transmitted with inappropriate parameter values.

- “Format” command with invalid MDS name or MDS type.
- “Create” command with invalid file name.
- “Write” command with file size “0” (zero) 0.

D007

The MDS system area stored in the data field is faulty.

- A valid system area has not yet been written into the datab field (“Move” command).
- The segregation of the data field is faulty, e.g., area overlap.
- The data field is being overwritten due to a PLC program error, e.g., indirect addressing.

D009

Invalid parameters (or parameter values) in “Reset” command.

Error originates in PLC program. Check PBK5MIF parameters.

D014

- “Write” command, MDS memory is full. Therefore, only truncated data is written to the MDS.
- “Create” command, file cannot be created onboard the MDS due to insufficient number of free blocks.

D015

The filehandler was unable to identify the MDS.

Reformat the MDS.

D018

The selected address is outside the file.

Reformat MDS.

D022

The data memory was closed with the use of the “Cover” command. A command requiring write-access (e.g., “Create” and “Format”) may not be permitted to destroy the data field, and is therefore rejected.

D023

"Cover" command: The MDS name specified in the command fails to match the actual MDS name.

9.2.5 E Directory-related errors

E001

- The MDS type which is present within the SLG transmission range does not match the selected ECC driver.
  Reformat the MDS using the ECC driver.
- MDS is not an MDS filehandler.
  Reformat MDS.

E002

Free directory entry no longer available onboard this MDS. The file specified in the "Create" command cannot be created.

E003

The file name specified in the "Create" command is already in use.

E005

An error has occurred during "Read" or "Write" command execution.

Reformat MDS.

9.2.6 F File-specific errors

F001

- The accessed file does not exist.
  Generate the file with the use of the "Create" command.
- The file name in the in the BefDB data module may possibly not be written in ASCII notation.

F005

Attempted write-access by means of "Write" command to a file which is write-protected.

F007

QUEUE-READ: Specified file size smaller than actual file size.

F008

QUEUE-READ: The skip calculated by the Filehandler is greater than 0FFF Hex.

Notes:

10 Characteristic Data

Characteristic Data	PBK5MIF
Module name	PBK5MIF
Module length	5345 words
Parameters	4
Processing time	154 µs, no-load
Assigned markers	M230 to M255
Call	cyclic, incl. 1st cycle

Cycle loading through PBK5MIF

n = The number of user data per cycle to be processed. A maximum of 240 bytes can be read or 226 bytes written in one cycle. Larger files are distributed over several cycles.

Idle run	154 µs
Read file	(370 + n · 5.7) µs
Write file	(370 + n · 7.0) µs

☞ **The times specified for reading/writing files are valid only if the program is run without a checksum. If the program is run with a checksum, the cycle time for reading files is extended by the factor 1.3, and for writing files by the factor 1.5.**

Notes:

A Appendix

A.1 Abbreviations

Abbreviation	Meaning
BefDB	Command Area data module
DatDB	Data Area data module
DM	Data module
DOS	Disk Operating System
EI	Extended input
EO	Extended output
FAT	File Allocation Table
I	Input
II	Interface input
IO	Interface output
MDS	Mobile Data Storage unit
O	Output
PBK	Peripheral Bus Interface Module
PBK5MIF	CL500 function module for the PBK module and the MOBY I/F channel module
SLG	Read/write unit
STG 4F	Service and Testing unit
ZS	Central processing unit of controller

A.2 Amendments

In this revised edition 102, amendments have been made to the former edition 101 on the following pages.

Chapter	Page	Change
3	3–1	Section 3.1
	3–2	P1 WorkDBNr
4	4–1	Table was augmented.
	4–4	Table was augmented.
	4–3	Description of ’SatzNr’ variable was added.
	4–6	Description of ’ABTA’ variable was added.
	4–7	D26 FHStatus
5	5–2	Table was augmented.
	5–5	Attribute
	5–6	Command structure
	5–7	Description of “Cover” command was added.
	5–11	Command structure
9	9–4	Sections 9.1.3 and 9.1.4
	9–9	Description of request-specific error messages were augmented by adding “D007” and “D022”.
	9–10	Description of request-specific error messages was augmented by adding “D023”.
	9–10	Description of file-specific error messages was augmented by adding “F007” and “F008”.

A.3 Index

Bosch Automation Technology

Australia
Robert Bosch (Australia) Pty. Ltd.
Head Office
Cnr. Centre - McNaughton Roads
P.O. Box 66
AUS-3168 Clayton, Victoria
Fax (03) 95 41 77 03

Great Britain
Robert Bosch Limited
Automation Technology Division
Meridian South Meridian Business Park
GB-LE3 2WY Braunstone
Leicestershire
Fax (01 16) 28-9 28 78

Canada
Robert Bosch Corporation
Automation Technology Division
6811 Century Avenue
CAN-Mississauga, Ontario L5N 1R1
Fax (905) 5 42-42 81

USA
Robert Bosch Corporation
Automation Technology Division
Fluid Power Products
7505 Durand Avenue
USA-Racine, Wisconsin 53406
Fax (414) 5 54-81 03

Robert Bosch Corporation
Automation Technology Division
Factory Automation Products
816 East Third Street
USA-Buchanan, MI 49107
Fax (616) 6 95-53 63

Robert Bosch Corporation
Automation Technology Division
Industrial Electronic Products
40 Darling Drive
USA-Avon, CT 0 60 01-42 17
Fax (860) 4 09-70 80

Your concessionary

BOSCH

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Antriebs- und Steuerungstechnik
Postfach 11 62
D-64701 Erbach
Fax +49 (0) 60 62 78-4 28

1070 072 135-102 (99.11) GB · HB SP· AT/PLS · Printed in Germany