SIEMENS

MOBY®

FC 46
Function for Filehandler

Technical Description

Table of Contents

6GT2097-3AC40-0DA2

Release 05.2002

Safety Guidelines

This manual contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger.

Danger

indicates that death, severe personal injury or substantial property damage will result if proper precautions are not taken.

Warning

indicates that death, severe personal injury or substantial property damage can result if proper precautions are not taken.

Caution

indicates that minor personal injury or property damage can result if proper precautions are not taken.

Note

draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

Qualified Personnel

The device/system may only be set up and operated in conjunction with this manual.
Only **qualified personnel** should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Correct Usage

Note the following:

Warning

This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

Trademarks

MOBY® and SIMATIC® are registered trademarks of SIEMENS AG.

Some of the other designations used in these documents are also registered trademarks; the owner's rights may be violated if they are used be third parties for their own purposes.

Siemens AG
Automation and Drives
Systems Engineering
PO Box 2355, D-90713 Fuerth

Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

Technical data subject to change.

Siemens Aktiengesellschaft

Order No. 6GT2097-3AC40-0DA2

Table of Contents

Brief Description

1

The function block described in this document controls the data transmission between a STEP 7 program and the ASM 451 and ASM 452 interface modules (starting with release 4). Throughout this documentation these interfaces will now only be referred to as ASM.

The FC 46 function uses the filehandler on the ASM/SLG. This gives the user a variety of advantages.

- The user addresses data via logical names (file names) consisting of up to 8 alphanumeric characters. Absolute addressing of data is therefore unnecessary.
- Sets of associated data are called files.
- Management of files of varying lengths
- Allocation of access rights to files

To make the advantages of the “file handler” accessible to the SIMATIC user, an appropriate S7 function (FC 46) has been created. This controls data transmission between SIMATIC and the ASM.

Principal functions of FC 46

- Conversion of data from the form specified by the user into a form that can be interpreted by the ASM
- Communication with the ASM through the exchange of commands and data
- Preprocessing of errors for the user

The data transmission sequence between FC and MDS can be divided into three parts.

- Providing the interface with the appropriate commands, parameters and/or data
- Transmitting the data between interface and data carrier
- Providing the S7 with the appropriate parameters or data from the interface

The FC 46 can be used with the following S7s.

- S7-300 with CPU 315-2 DP
- S7-400 with CPU 414-2 DP

Supported MOBY families:

- MOBY I
- MOBY U (starting with version V3.0 of FC 46 and release 4 of the ASM 452)

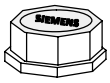
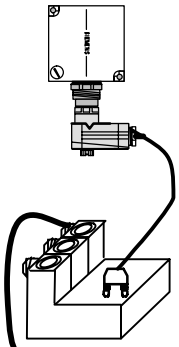
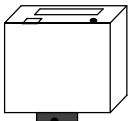
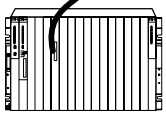
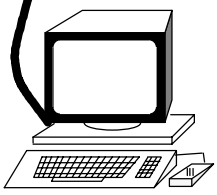
Use the SIEM804D.GSD file (ASM 451) or SIEM80B6.GSD (ASM 452) for PROFIBUS operation. The file is included with the product “Software MOBY” (6GT2 080-2AA10). It can also be downloaded from the Internet via **www.ad.siemens.de/support/** .

Notes on compatibility

The ASM 452 (starting with release 4) is compatible with the ASM 451. Starting 08/2002 the ASM 451 will be discontinued and replaced by the ASM 452. To ensure compatibility with existing systems, switch 8 (see chapter 2.1) must be set to OFF. This permits the existing system to continue working with the ASM 452 and the GSD file of the ASM 451.

1.1 Basic Information about the “File Handler”

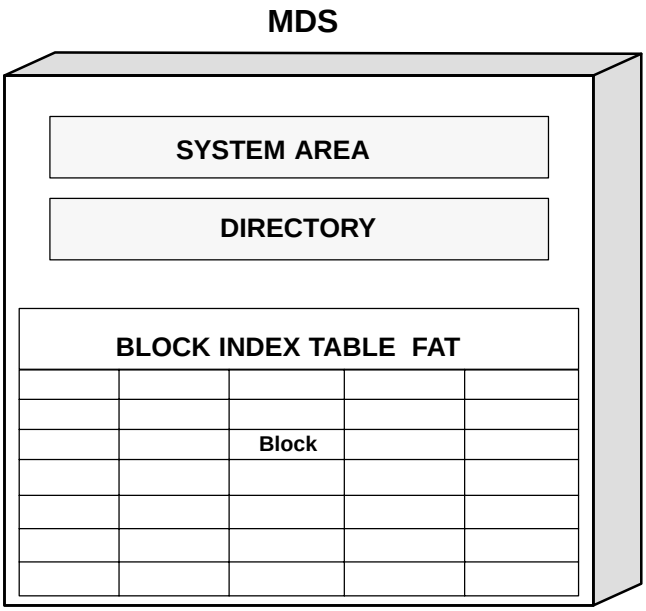
To give yourself a better idea of how the MOBY file handler works, think of it as a floppy disk system.

MOBY File Handler	Floppy Disk System
 **MDS:** Data is stored electronically in the MDS on a memory chip (RAM or EEPROM).	 **Floppy Disk:** Data is written magnetically on a floppy disk.
 **SLG:** The SLG modulates the data stream arriving from the ASM and demodulates the data stream arriving from the MDS. The MDS must lie within the transmission window of the SLG. The size of the transmission window depends on the MDS/SLG configuration.	 **Floppy Drive:** The floppy drive converts the data impulses arriving from the PC into a write string for the read/write head (or vice versa**)** The floppy disk must be positioned very exactly over the read/write head as this generates only very small magnetic fields.
 The MOBY interface module (ASM) provides the user interface. A USART module on the ASM produces the serial data stream for the SLG.	 A PC controls the floppy drive and exchanges data with it via an interface. The coding and decoding of the data takes place in a floppy controller in the PC.

As with a floppy disk system, the file handler also accesses data using logical file names, not physical memory addresses. This in turn requires that the files are created on a formatted (completely erased) data carrier. The data relating to a file can then be processed by the appropriate command, e.g. READ, WRITE, DELETE etc.

1.2 MDS Memory Structure

1.2.1 Logical Structure of MDS Memory



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

The directory contains the names of all existing files with their corresponding parameters. The number of directory entries possible depends on the type of MDS.

The FAT (File Allocation Table) points to all the blocks belonging to a file.

The user data area is divided into blocks. The number of blocks and the block length depends on the type of MDS.

Figure 1-1 MDS memory

1.2.2 Configurable memory sizes

MDS units are available with memory chips (RAM/EEPROM) of various sizes. When initialising (see section 4.7) the “file handler” must be told what type of MDS is involved so that the correct data management system can be created.

The following table shows how the file handler structures the various types of MDS.

Table 1-1 Memory sizes which can be formatted

Typ (Hex)	MDS-capacity in bytes (gross)	Max. no. of files	User data bytes (nett)	Number of blocks	Block size in bytes
Enter this parameter when initialising	Total MDS memory capacity	Max. number of files that can be created	Max. no. of user bytes that can be written to the MDS	The product of no. of blocks and block size gives the user data length (net) in bytes	
01H 81H	62 42 [1]	1 1	27 7	1 1	27 7
02H	62	2	12	2	6
03H 83H	128 112 [1]	3 3	60 45	6 5	10 9
04H 84H	2.045 1.778 [1]	16 16	1.680 1.440	105 90	16 16
05H 85H	8.189 7.154 [1]	32 32	7.456 6.464	233 202	32 32
06H 86H	32.765 28.658 [1]	64 64	31.488 27.520	246 215	128 128

1 ...with ECC operation

1.2.3 Check sum

To minimise the time it takes the SLG to process an MDS, a check sum (2 bytes long) is maintained in the system area of the MDS. Use of the check sum mechanism is **essential in dynamic read and write operations** (see section 6.1).

Each time the contents of the directory or the FAT are modified (execution of write commands e.g. CREATE, WRITE, ATTRIB etc.), the file handler calculates a check sum on the two areas and places this in the MDS system area. If two consecutive MDSs have the same check sum, their data structures (file names, file lengths etc.) will be totally identical.
The file handler does not therefore need to read the directory and FAT of the second MDS (additional time requirement), but can continue to operate with the current (internal to file handler) parameters.

Note:

The check sum is created from the directory and FAT only, and not from the file contents.

1.3 Communication Between FC 46 and ASM

Communication between FC 46 and ASM is implemented with cyclic and acyclic services of PROFIBUS-DP/V1.

The current status of the ASM is stored in a status byte which is cyclically polled by the S7 master and stored in the I/O area of the S7. The address of this status byte is specified in the formal operand ADR of FC 46 and must correspond to the PROFIBUS configuration.

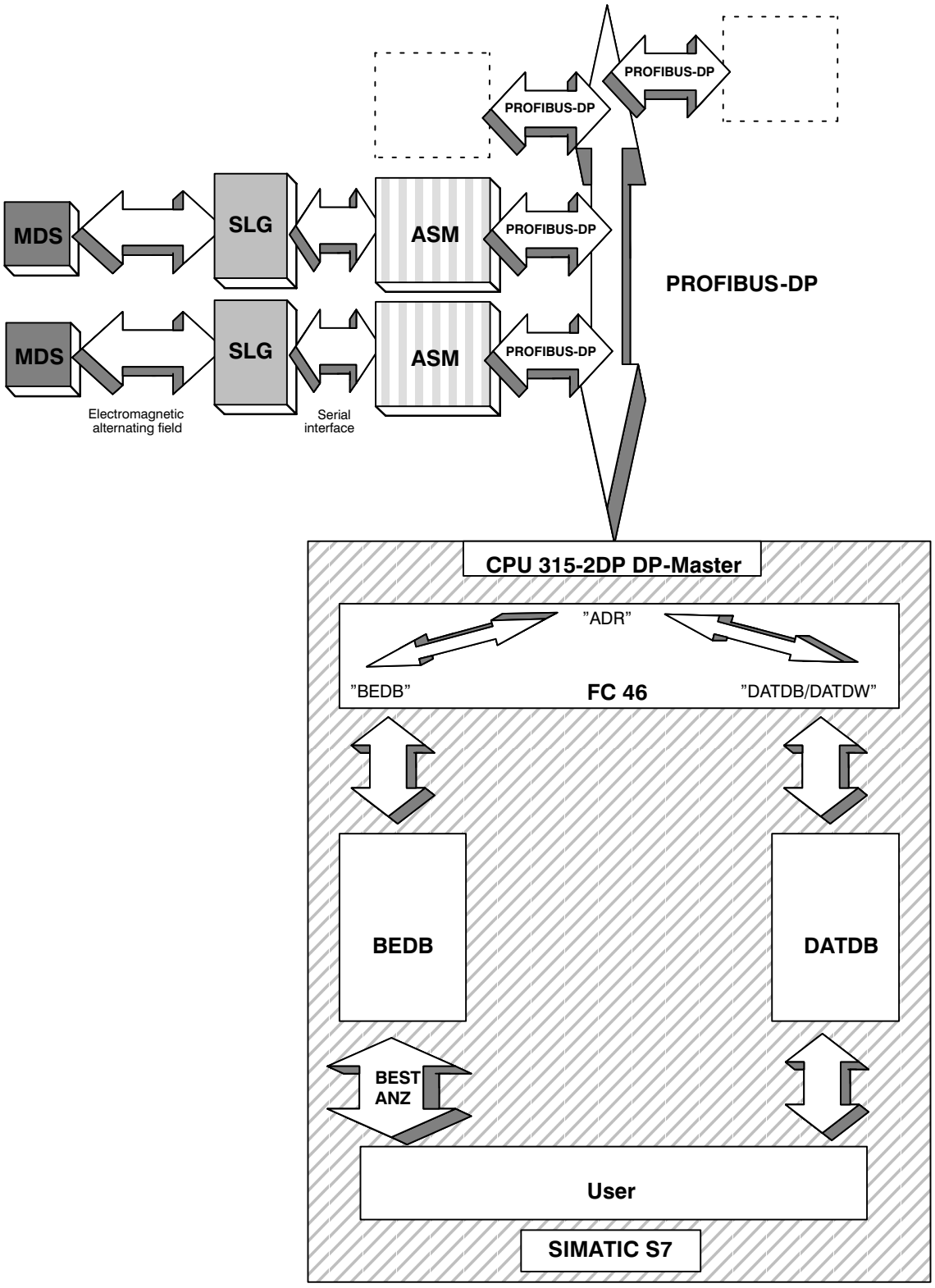
File handler telegram communication is implemented by command and acknowledgment telegrams which are transferred by acyclic DP/V1 services. System functions SFC 58 (write service) and SFC 59 (read service) are available for this purpose on the S7.

The file handler command telegram to be transferred is conditioned in the sending buffer of BEDB after the command start of FC 46 and is transferred asynchronously with SFC 58 to the ASM. The ASM starts the command-related procedures (e.g., read from MDS), and then prepares a file handler acknowledgment. The acknowledgment is read to the receiving buffer of BEDB by FC 46 via SFC 59.

FC 46 updates the inidcation bits (e.g., error, presence of the MDS, and so on), copies user data to the area provided by DATDB/DATDW (depending on the type of command), and starts the next partial or complete command telegram (for multi-block commands) to the ASM.

The maximum block length for command and acknowlegment telegrams is defined by PROFIBUS-DP/V1 and is 240 bytes for the SIMATIC S7. Since each command and acknowledgment telegram always contains a header (i.e., telegram header) plus user data, the maximum length of the user data is 229 bytes per telegram.

1.4 Program Flow Chart



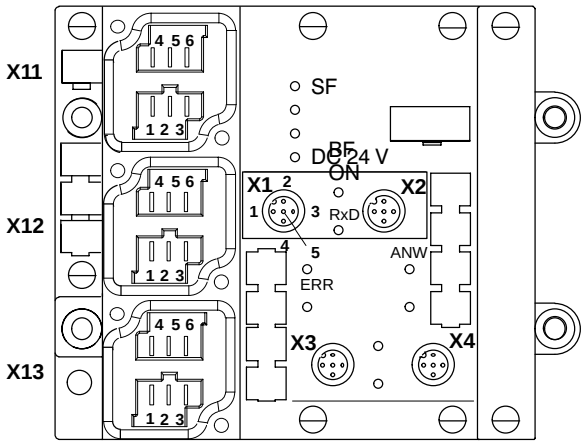
Hardware Description of ASM 451 and ASM 452

2

Pin assignment

The following figure shows the pin assignment of the ASM.

ASM 451



LEDs for PROFIBUS-DP
(See also section 9.1)

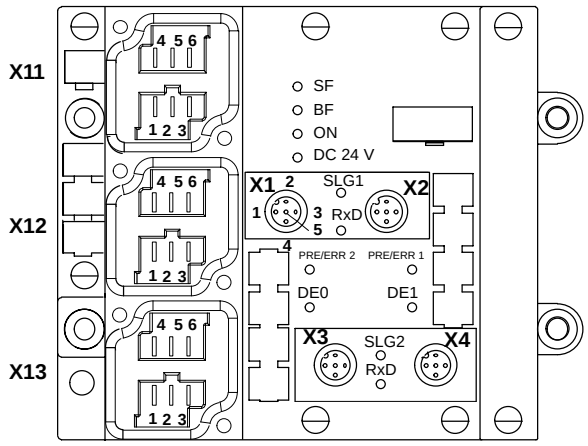
SF:	System fault (hardware error on the ASM)
BF:	Bus fault (error on PROFIBUS-DP)
ON:	On when the logic voltage is present on the ASM (is generated from the 24 V supply voltage)
DC 24 V:	On when the 24 V supply voltage is connected to the ASM

LEDs for MOBY and ASM 451

RxD:	SLG active
ANW:	MDS present
ERR:	Error indication

All other LEDs are not assigned.

ASM 452



LEDs for MOBY and ASM 452

RxD:	SLG with command active
PRE/ERR 1:	Channel 1: MDS present or error indication MDS present: LED is permanently ON. If there is more than 1 MDS in the field, the number of MDSs is indicated by short light pulses. There is no error output. Error indication: The LED is permanently OFF. The last error number is indicated with short light pulses.
PRE/ERR 2:	Channel 2 is OFF.
SLG 1:	SLG 1 is ON. Only SLG 1 can be selected on the filehandler.
SLG 2:	SLG 2 is OFF.

Socket	Pin Assignment	
X11 and X12 (PROFIBUS-DP)	1	Signal B
	2	PE
	3*	PE
	4	Signal A
	5*	L+
	6*	M
X13 (supply voltage)	1	PE
	2	L+
	3	M
	4	PE
	5	L+
	6	M

* Not circuited

Socket	Pin Assignment (SLG)	
X1	1	+RxD
	2	+TxD
	3	-TxD
	4	-RxD
	5	PE
X2	1	+24 V
	2	Res.
	3	0 V
	4	Res.
	5	PE

X3 and X4 not used with the filehandler

Figure 2-1 Pin assignment and LEDs of the ASM

Dimensional drawing of ASM with mounting holes

The following figure shows a dimensional drawing of the ASM with bus plug connectors. The length of the PG screw connections and the radius of the cable must be added to the total width and depth specified.

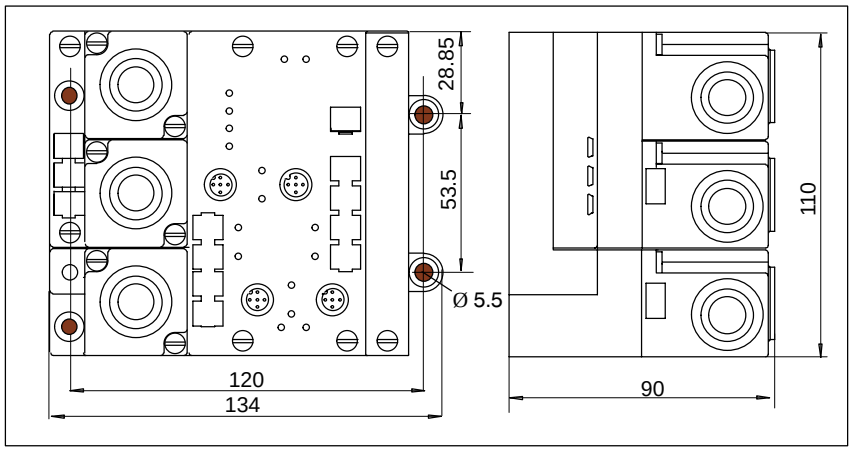


Figure 2-2 Dimensional drawing of the ASM

2.1 PROFIBUS Address and Terminal Resistance

PROFIBUS address

The PROFIBUS address is used to specify the address under which the ASM interface module will be addressed by the DP master of PROFIBUS-DP.

The 7 DIP switches on the basic module are used to set a PROFIBUS address from 1 to 125.

Purpose of the terminal resistance

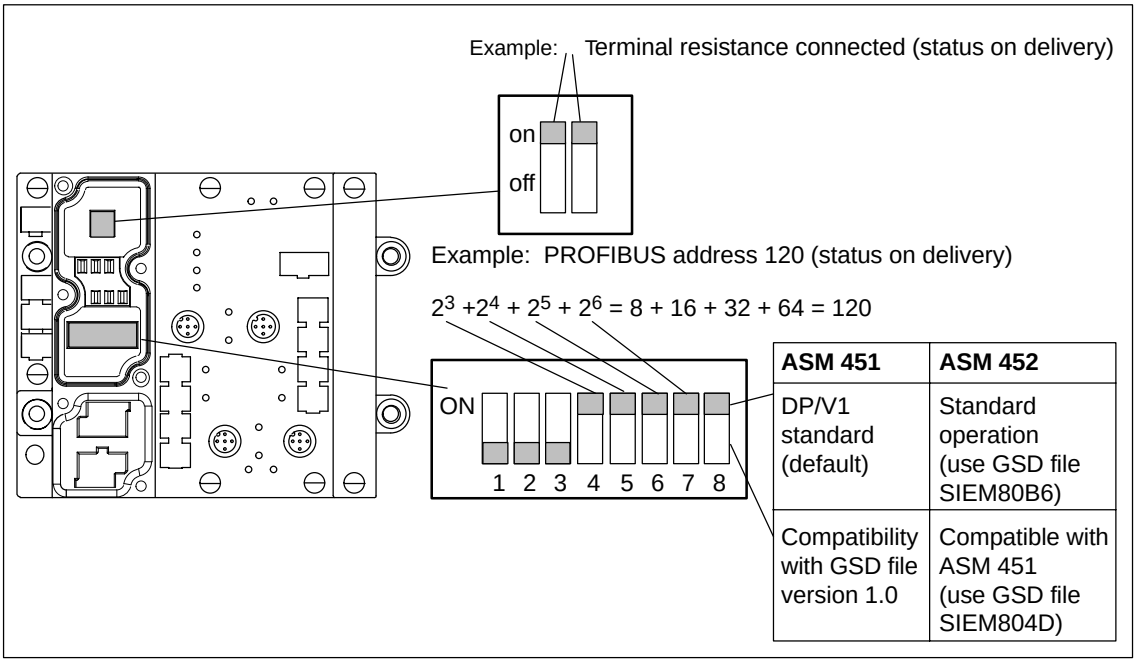
Both ends of a bus cable must be terminated with its impedance. The terminal resistance is connected in the first and last station of the network.

Location of the DIP switches

The DIP switches for setting the PROFIBUS address and connecting the terminal resistance are located inside the ASM under the plug connector plate for the connection plugs for PROFIBUS-DP and the supply voltage.

How to proceed

The plug connector plate of the ASM must be removed before the PROFIBUS address can be set and the terminal resistance connected. The plug connector plate covers the DIP switches. The following figure shows the location of the DIP switches on the ASM and gives an example of the setting of each.



ASM 451	ASM 452
DP/V1 standard (default)	Standard operation (use GSD file SIEM80B6)
Compatibility with GSD file version 1.0	Compatible with ASM 451 (use GSD file SIEM804D)

Figure 2-3 Setting the PROFIBUS address and connecting the terminal resistance

Note

- The PROFIBUS address on the ASM must always correspond to the PROFIBUS address specified by the configuration software for this ASM.
- To ensure that the terminal resistance functions correctly, always set **both** DIP switches of the terminal resistance to “on” or “off.”

2.2 Wiring the Plug Connectors for the Voltage Supply and PROFIBUS-DP

Possible connections

The voltage supply must be connected to each ASM separately (on X13). The plug connectors for PROFIBUS and the voltage supply are not included. You can order them from Siemens under order number 6ES7 194-1AA00-0XA0.

PROFIBUS-DP plug connector assignment

Connect the plug connectors as shown in the figure below. The pins for the PROFIBUS-DP connection are shown in bold type.

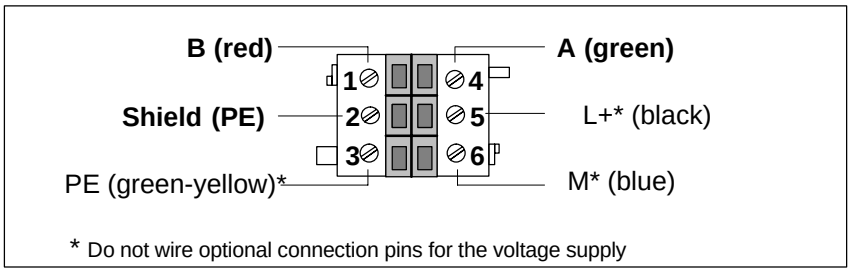


Figure 2-4 Pin assignment of the plug connector for wiring PROFIBUS-DP

Looping through PROFIBUS-DP

If you want to loop through PROFIBUS-DP to the next ASM, wire the second plug connector. This plug connector is wired the same as the plug connector for the connection to the first ASM. See figure 2-4.

Wiring the voltage supply

You will need the following materials for connecting the voltage supply (24 V DC).

- One plug connector
- 3-core, flexible copper cable

Connect the plug connector to pins 1, 2 and 3 as shown in the next figure. Terminals 1 and 4, 2 and 5, and 3 and 6 are jumpered internally.

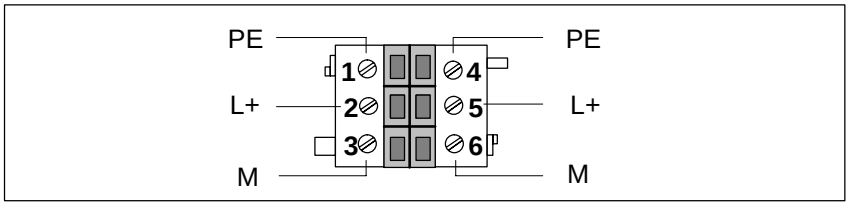


Figure 2-5 Pin assignment of the plug connector for connection of the voltage supply

Caution

Wiring the plug connector incorrectly can destroy all or part of the device.

Note

To ensure protection rating IP65, IP66 or IP67, all 3 plug connectors must always be connected to the ASM regardless of whether they are wired or not.

Caution
(Only applicable to ASM 451)

When the ASM is used with a baud rate of 3 Mbaud, 6 Mbaud or 12 Mbaud, a shielded cable must be used for the supply voltage or a type 742 7101 ferrite (from Würth Electronic) must be used to hinge the PROFIBUS cable so that the limit values for CE are adhered to.

2.3 SLG Connection

With prefabricated cable

All SLGs are connected with a 2-m prefabricated cable. Other cable lengths are available on request.

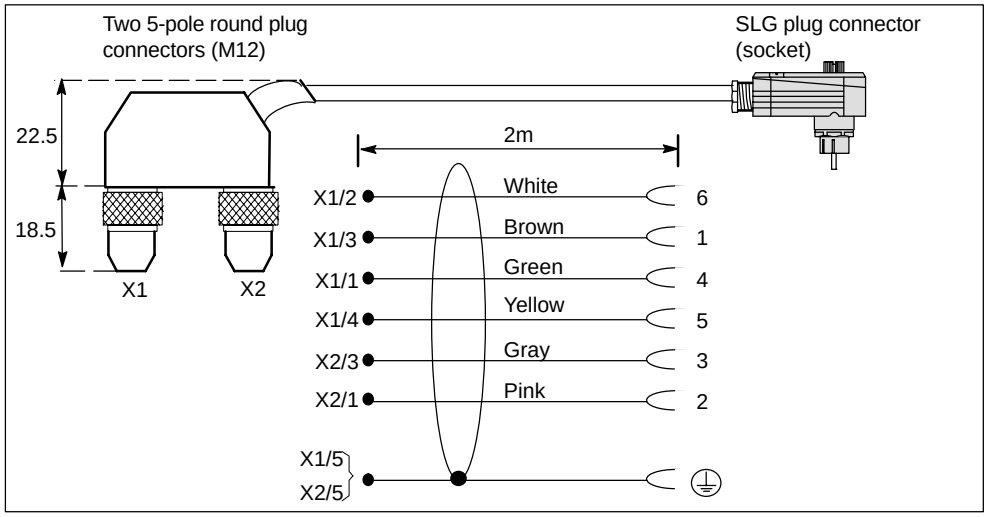
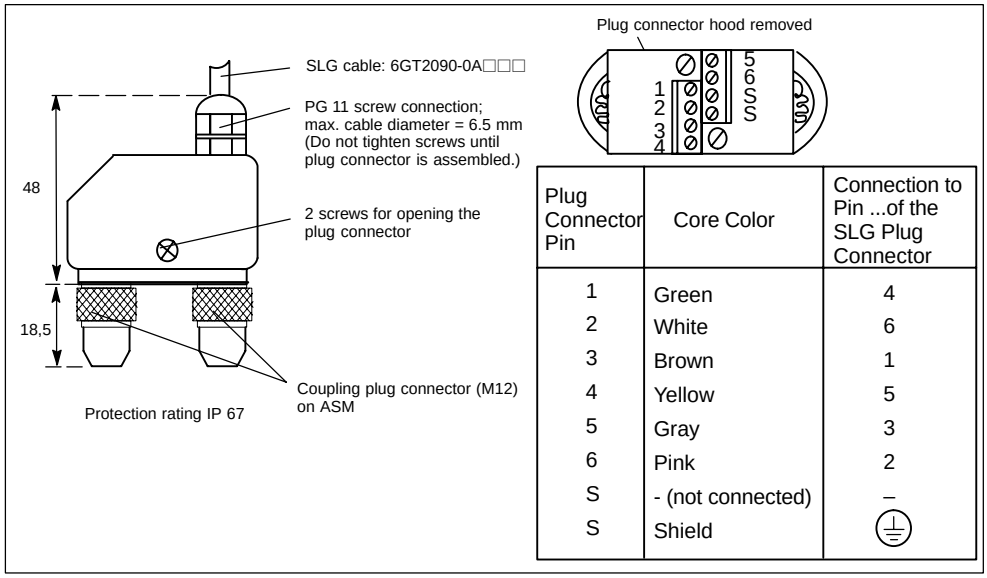


Figure 2-6 ASM ↔ SLG connection cable (6GT2091-1CH20)

With cable made by you

An SLG plug connector with screw terminals is available for users who want to make their own cables. Cable and SLG plug connector can be ordered from the MOBY catalog.



Plug Connector Pin	Core Color	Connection to Pin ...of the SLG Plug Connector
1	Green	4
2	White	6
3	Brown	1
4	Yellow	5
5	Gray	3
6	Pink	2
S	- (not connected)	–
S	Shield	⏚

Figure 2-7 Plug connector ASM ↔ SLG (6GT2 090-0BC00)

Block Specifications

3

3.1 Technical Data

Block number:	FC 46
Block name:	MOBY_FH
Block length:	4.2 kbytes
Data blocks:	BEDB and DATDB (if DATDB not equal BEDB)
Markers used:	None
Counters used:	None
Local data used:	64 bytes
Call:	Cyclic
System functions:	SFC 58, SFC 59

Table 3-1 Typical run times of FC 46 (cycle load of the AS in msec)

	S7-CPU	Idle Cycle	Read File	Write File
FC 46	315-2 DP	0.9	2.73 + n x 0.04	2.73 + n x 0.04
	414-2 DP	0.2	0.6 + n x 0.008	0.6 + n x 0.008

n = Amount of user data to be processed per cycle
Number of bytes to be written (WRITE command: maximum of 215 bytes of a file per cycle)
Number of bytes to be read (for example, READ: maximum of 215 bytes of a file per cycle)
This means that the cycle time does increase for files longer than 215 bytes since FC 46 divides the data stream into several blocks and distributes these blocks over several cycles.

3.2 Configuration Diagram

Table 3-2 Configuration box

LAD Box	Parameter	Data Type	Permissible Values/ Characters	Description
MOBY_FH ADR BEDB EAKO ECC RWD SLG scanning_time dili_multitag field_ON_con_tim	ADR	INT	≥ 128[1]	Address of the cyclic input/output byte on the SIMATIC
	BEDB	INT	≥ 1	Command block
	EAKO	INT	0, 1, 4, 5	Incoming/outgoing monitor
	ECC	INT	0, 1	Error correction code (special driver; MOBY I)
	RWD	CHAR	R, W, D	Access rights
	SLG	WORD	0001...FFFE (FFFF = test)	Number of the SLG station
	scanning_time[2]	BYTE	00..FF	Scan time or standby time
	dili_multitag[3]	WORD	0000	distance_limiting and multitag
	field_ON_con_tim[4]	WORD	0000	field_ON_control and field_ON_time

1 Depends on the CPU. The value must be located outside the process image.
2 Corresponds to the ABTA parameter of older releases of the FC 46
3 Corresponds to the OPT1 parameter of older releases of the FC 46
4 Corresponds to the OPT2 parameter of older releases of the FC 46

ADR:

The ASM uses 1 (ASM 451) or 2 (ASM 452) input and output byte(s) whose address is specified with ADR. Remember that input and output bytes use the same address. The address must be parameterized outside the process image of the controller. The size of the process image depends on the CPU and can be scanned with the ONLINE command 'target system → module status.' ADR must agree with the PROFIBUS configuration in HW Config.

BEDB:

Command data block. Each ASM requires 350 data words (i.e., 700 data bytes) in a block. A separate BEDB must be parameterized for each ASM.

EAKO:

Control of entry/exit checking (see section 5).

“0”= Next mode. “NEXT” command must be programmed by the user. Continuous
MDS control.

“1”= With time-out: An MDS must be present in the transmission window before the
start of the command. “NEXT” command is not mandatory. No continuous MDS
control.

”4” = No entry/exit control by FC and ASM. “NEXT” command is not mandatory. No
continuous MDS control.

”5” = No entry/exit control by FC and ASM. “NEXT” command is not permitted. No
continuous MDS control.

ECC: Error Correction Code (only MOBY I)

Special driver for increased data security. On each access or each write to the MDS the ECC driver checks/produces redundant data.
If at any time1 bit of data is lost from MDS memory (e.g. with a very frequently written EEPROM MDS), the ECC driver is able to reconstruct the lost data bit. The user can be sure the correct data will be returned. The user is able to interrogate and evaluate this reconstruction process by examining bit 11 in BEST “error correction carried out” (e.g. to initiate an early replacement of a “used” MDS). The fact that data correction has been performed is also stored in a counter in the MDS. The counter can be interrogated by the “MDS STATUS” command (see section 4.7). There is a very high probability that large-scale data corruption will be recognized and the user notified through an error message.

1 = ECC driver enabled

0 = ECC driver enabled

Note

The time taken to access MDS data increases when the ECC driver is enabled and the net capacity of the MDS shrinks. Performing a data correction can delay the result by up to 1 second.

RWD:

SLG access rights in respect to the data carrier (see section 4.6)

“R” = Read only; only read commands permitted

”W” = Read + Write; only read and write commands permitted

”D” = Read + Write + Delete; all commands allowed

SLG:

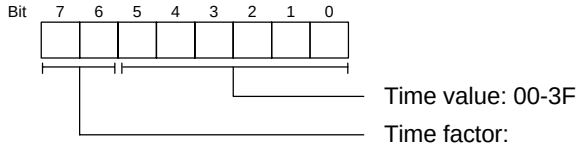
Number of the SLG station; this number is written to the MDS before each processing operation. This enables the file handler to recognize if a “new MDS” has really entered the field.
The user must ensure that every SLG station has a different value for the “SLG” parameter in every application.

0001 ... FFFE: normal operating mode
FFFF: **est function:** the SLG will not be evaluated by the file handler. The file handler’s (or user program’s) entry/exit monitoring can thus only be tested with one SLG station and one MDS. The same MDS can return to the transmission window of an SLG time and time again and will be processed.

scanning_time:

MOBY I: Scan time for MDS 507 operation
Normal operation is configured with scanning_time = 0.

The following diagram applies to the scanning_time.



Time value: 00-3F

Time factor:
00 = 0.01 sec.
01 = 0.1 sec.
10 = 1 sec.
11 = 10 sec.

Example:

Scan time of 1 second for the scanning_time parameter = B#16#81.

MOBY U:

Scanning_time is the standby time for the MDS. If the MDS receives an additional command before scanning_time expires, the command can be executed immediately. If the MDS receives a command after scanning_time expires, command execution is delayed by sleep_time of the MDS.

00 hex = No standby time (default)
01 hex = 7 msec standby time
02 hex = 14 msec standby time
and so on

Remember:

Scanning_time affects the lifespan of the battery. The greater the value of scanning_time the shorter the lifespan of the battery. For more detailed calculations, see the MOBY U manual for configuration, installation and service.

dili_multitag:

This parameter consists of two values – distance_limiting (MSB) and multitag (LSB).

distance_limiting	MOBY U:	Range limit		
		05 hex	=	0.5 m
		0A hex	=	1.0 m
		0F hex	=	1.5 m
		14 hex	=	2.0 m
		19 hex	=	2.5 m
		1E hex	=	3.0 m
		23 hex	=	3.5 m
multitag	MOBY U:	Maximum number of MDSs in the field which can be processed at the same time. Permissible values: 1.		

field_ON_con_tim:

This parameter consists of two values – field_ON_control (MSB) and field_ON_time (LSB).

field_ON_control

MOBY U: Proximity switch mode — Antenna field automatically switched on and off. The command “antenna ON/OFF” takes precedence over proximity switch mode.

00 hex	=	No proximity switches
01 hex	=	One or two proximity switches The proximity switches are logically OR-linked. The field is on as long as a proximity switch is active.
02 hex	=	One or two proximity switches The 1st proximity switch turns on the field. The 2nd proximity switch turns the field off. If there are two proximity switches and field_ON_time is parameterized, the field is automatically turned off if the 2nd proximity switch does not react within this proximity switch time. If no field_ON_time is parameterized, the field remains on until the 2nd proximity switch is activated.

field_ON_time

MOBY U: Time for proximity switch mode (field_ON_control = 02)

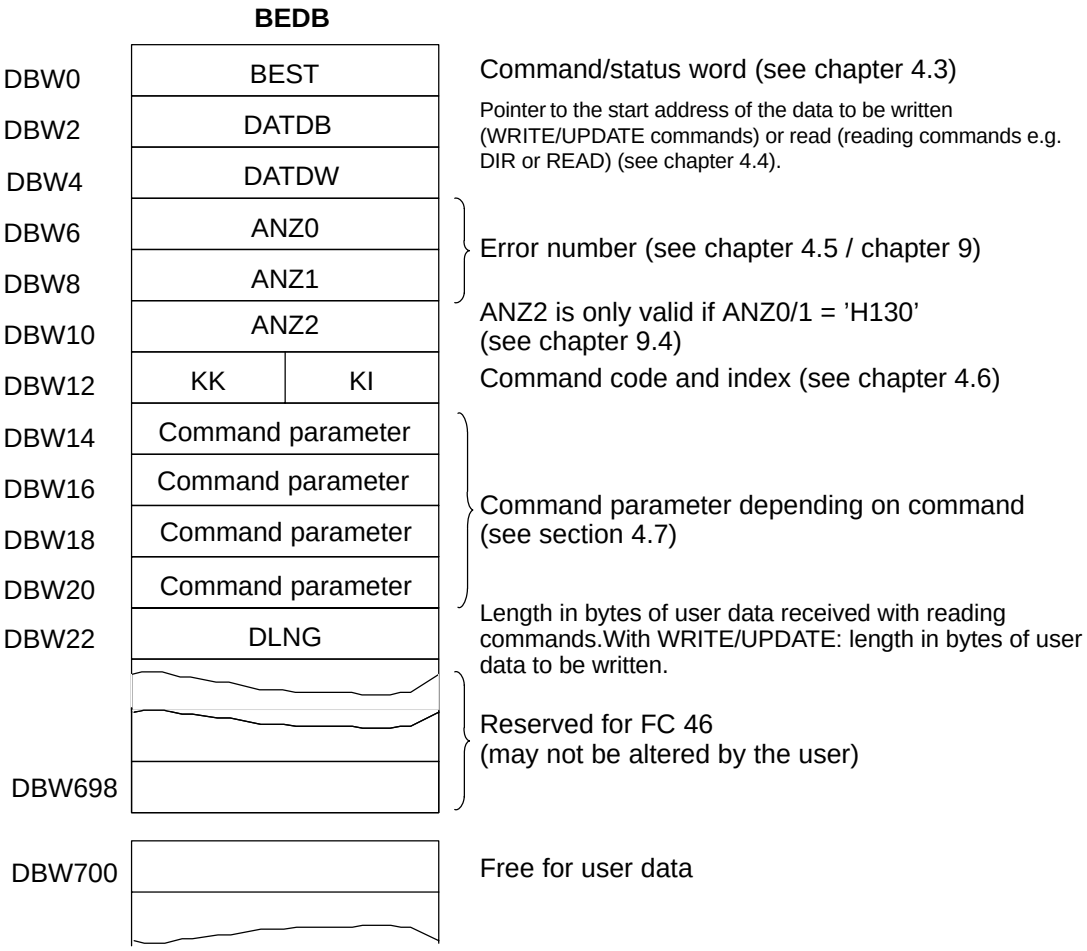
00 hex	=	Time monitoring is off. Field deactivation requires the 2nd proximity switch.
01 hex to FF hex	=	1 to 255 sec switchon time for the SLG field

4 Setting the Parameters

4.1 General Description of the BEDB Command Data Block

The FC 46 requires a BEDB command data block to enable it to function. All information is placed in the BEDB, e.g. data field pointer (DATDB/DATDW), error messages and status bits. The BEDB is always updated at the beginning of each cycle.
The minimum length of 350 data words (DBW 0 - DBW 698) must be maintained.

4.2 Structure of the BEDB

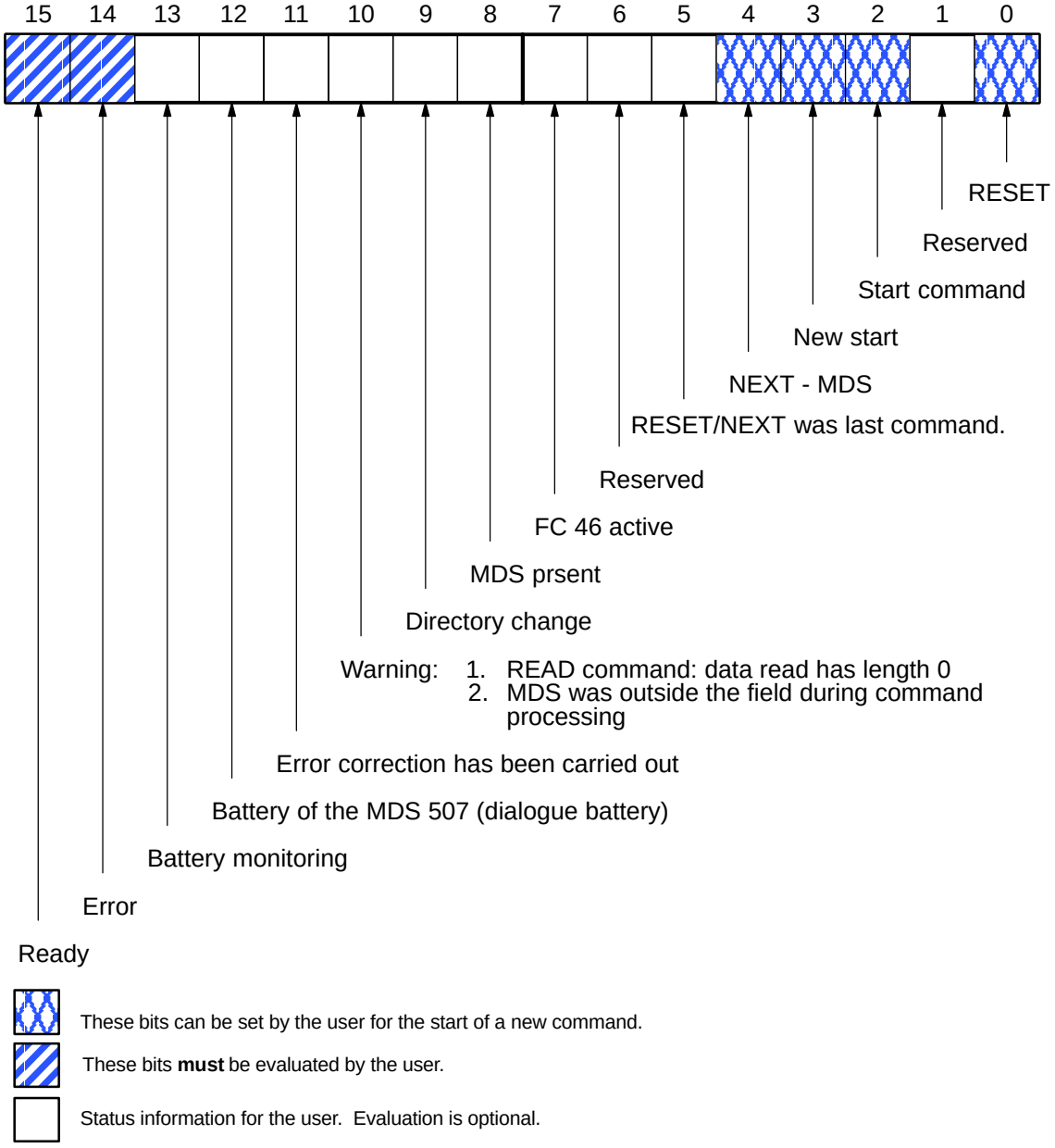


Data words DBW 0 - DBW 22 in the BEDB are accessible by the user. By writing, reading or entering parameters in these data words, individual commands can be transmitted to the ASM or messages displayed.

4.3 Command and Status Word “BEST”

DBW 0 = BEST

BEST is always accessible and can be interrogated by the user at any time.
In an error condition (bit 14 =1) the evaluation of error indications ANZ0 and ANZ1 has absolute priority.



Remarks:

The BEST parameter should be assigned the value 0008H in the cold start branch and the restart branch of the relevant OBs (see SIMATIC S7 equipment manual).

– OB 100 for new start

A new start command must be performed as the first command during commissioning. This checks the parameters and determines the address.

The same applies when the fixed parameters “ADR”, “BEDB”, “EAKO”, “ECC”, “RWD”, “SLG” are modified, or when BEDB or FC 46 are reloaded.

BIT 0

RESET

Terminating a started command (i.e., BEST BIT 7 = 1) has the same function as “new start” (BIT 3) except that the ASM is not initialized again at the PROFIBUS level.

BIT 1

In reserve

BIT 2

Start commamd

Start signal for the FC
The command specified in DBW 12 - DBW 22 (depending on command, see section 4.7) will be started.

BIT 3

New start

A new start should be initiated following initial commissioning and after every modification of FC parameters (ADR, BEDB, EAKO, ECC, RWD, SLG) or reloading of BEDB or FC 46, so that parameter checking, address calculation and command structuring in the BEDB can be performed again. A “RESET” or “new start” command automatically executes the “NEXT MDS” function (EAKO 0 only, see chapter 5).

BIT 4

NEXT - MDS Starts the “NEXT MDS” command

BIT 5

RESET/NEXT was last command

BIT 5 is set when the “RESET” or “new start” or “NEXT-MDS” commands are started.

BIT 6

In reserve

BIT 7

FC 46 active

The FC is currently processing a command or awaiting the result from the file handler.

BIT 8

Presence

This bit is set/reset when a data carrier enters or leaves the MDS transmission window respectively. The presence bit is valid for all EAKO modes (chapter 5). (During the reading of the directory and FAT, presence is already indicated at the ASM by the green LED. The presence bit only appears later in the FC 46).

1 = MDS in SLG field (green LED on ASM illuminated) (Exception: If the “connection to the SLG” is interrupted, ANW will still be shown. The user can then send the command to the ASM and will receive the appropriate error message in the acknowledgement).

0 = no MDS in SLG field

Note

When a RESET/new start command is executed, the ANW bit is reset briefly even when an MDS is located in the field.

BIT 9

Directory change

The check sums of the file handler and the MDS to be processed are not identical. The file handler must read the whole directory and FAT of the MDS before executing the command. The command execution time increases accordingly (see also BIT 14 = error).

BIT 10

Warning

1. The file read has length 0. This warning is reported for READ or QUEUE-READ.
2. The MDS was outside the transmission window during command processing or a change of MDS has occurred (one MDS has left and another has entered the transmission window).

Caution

An MDS leaves the transmission window and then reenters: the command will be processed correctly. All data on the MDS is OK.

One MDS leaves the transmission window and another enters. Error condition; the user must assume the data read is corrupt. In the case of a write command, it must be assumed that a file is corrupt on both MDSs or that the total file structure on the second MDS is destroyed.

BIT 11

Error correction

MDS data has been corrected (in ECC mode). The data is fully valid.

BIT 12

Battery of the MDS 507

The dialogue battery of the MDS 507 is not functioning properly. The MDS 507 may no longer reach its specified limit distance.
→ The battery pack of the MDS must be replaced.

BIT 13

Battery monitor

The MDS battery monitor has tripped. Although continued operation is possible for a short period at room temperature, we recommend the MDS be replaced by a new one as soon as possible.
This bit is only valid for read or write access to the MDS.
With an EEPROM MDS this bit is always set (memory requires no battery back-up).

BIT 14

Error

If a fault occurs during command processing, the “error” bit is set. The exact cause of the fault is given in the data words ANZ 0, ANZ 1 and ANZ 2 (DBW 6, 8 and 10 of the BEDB).

Note

An error message can flag the current MDS directory and FAT in the file handler as invalid. This always happens if the fault occurred while processing the directory. It may be necessary to read the directory and FAT when the next MDS is processed. If this delay is to be avoided, the user must send the MDS directory and FAT to the file handler by issuing the LOAD command after every error message.

BIT 15

Ready

The current command has ended. The user can generate and start a new command or evaluate the FC 46 parameters “ANZ0”, “ANZ1”, “ANZ2”,“BEST”, “DLNG” etc.

Note:
The “ready” bit does not need to be set at the start of a “RESET” command.

4.4 Data Field Pointer DATDB / DATDW

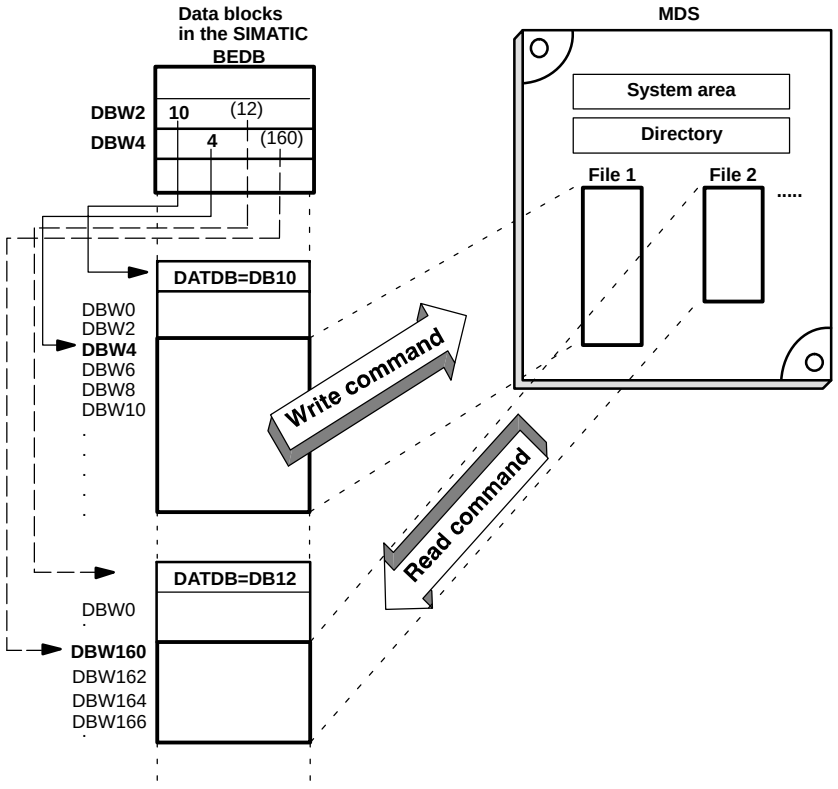
The entire instruction set of the FC 46 can basically be divided into read and write commands. If a read command is issued (e.g. READ, TRACE, DIR), the incoming data must be placed in a data block (DATDB). With the write command (WRITE) the data to be written to the MDS is taken from a data block. This means that when issuing a command, the user must specify a DATDB (generate before on the controller) to or from which the appropriate data is to be transferred. The DATDW determines the start address of the data in the DATDB.

Note

DATDB can also be BEDB so long as DATDW ≥ DBW 700
When issuing any command (including RESET) a DATDB and DATDW must be specified in DBW 2 and 4 of BEDB.

The user can assign any length to DATDB.
The start pointer DATDW should be adjusted to the user data.

The length of DATDB should match the anticipated amount of user data (e.g. with READ).



4.5 Error Indicators ANZ0 and ANZ1

DBW 6 = ANZ0 and DBW 8 = ANZ1. **Both ANZ0 and ANZ1 are in ASCII code.** ANZ0 and ANZ1 are always valid if the ready bit is set in BEST or if the ready bit and error bit are set. An error message only appears in the form of an acknowledgement telegram to a command that was issued.

Table 4-1 Error indicators

ERR LED Flashes	File Handler - Error Message	Typ of Error
20x	**A0 06** Unknown error, the command code CC is invalid **A0 11** DBN not equal to 1 in first command block; CC or DBN incorrect for subsequent blocks **A0 16** Command from another user being processed – – – – Internal ASM overflow **A0 17** The Data block of the SLG is too long. **A0 18** MOBY driver is active when a new command is sent.	PROTOCOL ERROR
3x	**B0 01** Fault in connection to the SLG **B0 02** 1. Command when EAKO = 1; but no MDS present (Timeout) 2. New MDS in transmission window; no NEXT command but command started (EAKO= 0) 3. New MDS has left station without NEXT (EAKO = 0) **B0 08** Antenna not on **B0 09** Buffer overflow on MOBY driver of the ASM/SLG **B0 10** Something is wrong with communication between filehandler and MDS driver (AB byte). **B0 11** The MDS driver reports a RESET termination. **B0 12** Unmotivated startup message of the MDS driver	SLG-ERROR
4x 2x 5x 6x 7x 8x 9x 10x 11x 12x 13x 14x 15x	**C0 02** RAM error in MDS **C0 06** Proximity error **C0 07** Incorrect parameters in TRACE or FORMAT / command cannot be interpreted **C0 08** Too many sync attempts **C0 09** Too many send errors **C0 10** CRC send error **C0 11** FORMAT, CRC receive error **C0 12** FORMAT, cannot initialise MDS **C0 13** FORMAT, timeout **C0 14** FORMAT, not initialised **C0 15** CMD address error **C0 16** ECC error **C0 17** General driver error **C0 18** Operating system error (AMOS mailbox) **C0 19** More than one MDS is in the field. **C0 20** The MOBY driver doesn't recognize the command from the filehandler. **C0 21** Watchdog error on ASM/SLG	MDS-ERROR
1x	**D0 01** Only RESET command permissible **D0 05** Invalid parameter with FORMATf, CREATE, WRITE, UPDATE, ATTRIB, QUEUE-WRITE, OCER, or QUEUE-READ **D0 07** 1. The directory specified in the LOAD command is invalid or load command parameter incorrect 2. MOVE command cannot be executed, check sum does not correspond to DIR + FAT **D0 09** Incorrect parameter in RESET command **D0 14** CREATE and WRITE: the user data area in the MDS is full **D0 15** Only FORMAT command possible; MDS not identified **D0 18** Start address in the command is outside the data area (start address > file length) **D0 22** Directory and/or FAT change access not permissible on an MDS protected by COVER **D0 23** COVER: invalid MDS name	TASK RELATED ERROR

Table 4-1 Error indicators

ERR LED Flashes	File Handler - Error Message	Typ of Error
	E0 01 The MDS type is incorrect or unsuitable for the selected mode of operation (ECC). **E0 02** CREATE command; no more directory entries available **E0 03** CREATE command; file already exists in directory **E0 05** 1. FAT block sequence error detected in READ or WRITE; FAT is corrupt 2. Invalid address given for TRACE command	DIRECTORY RELATED ERROR
	F0 01 Specified file not present **F0 05** WRITE/UPDATE/DELETE command to a write-protected file **F0 07** QUEUE-READ: Specified file length is less than the file length. **F0 08** QUEUE-READ: The skip calculated by the file handler is larger than 0FFF hex (4095 in decimal notation)	FILE RELATED ERROR
	H1 01 FB parameter or DATDB/DATDW invalid **H1 02** Length of BEDB < 50 DW **H0 03** Command index “CI” not permissible **H0 04** Command code “CC” not permissible **H0 05** Invalid SLG priority for this command **H0 06** Parameter “DLNG” of the command too long **H1 07** DATDB not present in PLC **H1 08** Acknowledgement block user data length not permissible **H1 10** ASM has received a hardware reset **H1 11** Invalid acknowledgement **H1 12** “CC” of command and acknowledgement do not correspond **H1 13** First command block not correctly acknowledged **H1 14** Synchro error when reading/writing cache memory **H1 15** DATDW parameter invalid **H1 16** Internal bookkeeping error (command and acknowledgement ADBs not identical) **H1 17** Internal bookkeeping error (command and acknowledgement DBNs not identical) **H1 18** DATDB address changed during command processing **H1 19** DATDB too short **H1 20** Compression procedure carried out in controller (absolute address of BEDB and DATDB changed) **H1 21** Only RESET command permissible **H0 27** QUEUE-READ: QUDW pointer is outside the DB specified in QUDB. **H0 28** QUEUE-READ: DBs missing in the controller or are too small to read in the user data **H1 30** System error: Non-permissible acknowledgement from the DP master. Word ANZ2 contains the exact cause of the error.	ERROR MESSAGES FROM FC 46
	Kx xx 1. xxx = Number of data entry (see QUEUE WRITE) in error 2. QUEUE WRITE command has incorrect parameters 3. The file to be created already exists.	QUEUE PARAMETE-RIZATION

4.6 FC 46 Command Set

Command	Function	Command code CC	Command index CI	Access rights RWD *)
FORMAT	Format the MDS	“I”	“I”	D
CREATE	Create a new file on the formatted MDS	“B”	“I”	W, D
QUEUE WRITE	Set up a complete data carrier	“Q”	“I”	D
QUEUE READ	Read several files with one command	“E”	“I”	R, W, D
UPDATE	Write data to file (revise file length)	“U”	“I”	W, D
WRITE	Write data to file	“W”	“I”	W, D
READ	Read data from file	“R”	“I”	R, W, D
DELETE	Delete file from MDS	“D”	“I”	D
ATTRIB	Assign attribute to file	“Y”	“I”	W, D
COVER	Protect MDS structure	“C”	“I”	W, D
DIR	Read MDS directory	“G”	“I”	R, W, D
MDS-STATUS	Request MDS status	“F”	“I”	R, W, D
END	Conclude communication with MDS	“K”	“I”	R, W, D
TRACE	Absolute read of MDS	“T”	“I”	R, W, D
MOVE	Place system data (DIR + FAT + check sum) in data block	“M”	“I”	R, W, D
LOAD	Transfer system data to ASM	“O”	“I”	R, W, D
SET-ANT	Turn SLG antenna on or off.	“A”	“I”	R, W, D
ASM-STATUS	Request ASM status	“S”	“I”	R, W, D
RESET	Reset ASM/file handler	“X”	“I”	R, W, D
NEXT	Process the next MDS	“N”	“I”	R, W, D

*) Cf. chapter 3.2

The command code “CC” is a mnemonic for the command (defines the command as such). The command index “CI” identifies the command explicitly as a “file handler” command.

The individual commands are described in detail below.

FORMAT: Format (initialise) a data carrier

After a data carrier is received from SIEMENS, it must be formatted before entering the manufacturing process (or it can be formatted in the manufacturing process itself).

- The MDS is completely erased. All the user data (file directory and data) will therefore be lost (option = 00H or 01H , see section 4.7).
- The directory + FAT remain. Only the user data area is deleted (option = 02H or 03H, see section 4.7).
- The directory + FAT are deleted. The user data area remains unchanged (option = 04H or 05H, see section 4.7).
- The MDS type is reported.
- The file handler code is entered onto the MDS.
- The operating mode is set with/without ECC.
- FORMAT command is only permissible on MDSs with access rights “D”.

CREATE: Create new file on formatted MDS

- The new file is entered in the directory (file index).
- Memory is reserved on the MDS for this file.

QUEUE-WRITE: Set up a complete data carrier

This command enables individual files to be loaded very quickly onto the data carrier, together with their corresponding contents and attributes. The individual files to be created (consisting of file name, attribute, file length and data) are processed sequentially by the file handler. This means that the file handler internally first creates a file, then writes data to it and assigns it an attribute. The next file to be created is then processed in the same way.
This function can only be performed after the specified MDS has been successfully formatted.

QUEUE-READ: Read several files with one command

The QUEUE-READ command can be used to read up to 15 files from the MDS with one command. As an option, the entire MDS with all files can be read.

UPDATE: Write data to an existing file

- A defined quantity of bytes is written to the existing file (parameter DLNG = DBW 22 in BEDB) from the DATDB (from start address DATDW).
- In contrast to the WRITE command, the file handler updates the file length, i.e. the file length corresponds to the DLNG parameter (DBW 22 in BEDB) in bytes.
- The memory area reserved for the file will not be released (FAT is not updated).

WRITE: Write data to an existing file

- A defined quantity of bytes is written to the existing file (parameter DLNG = DBW 22 in BEDB) from the DATDB (from start address DATDW).
- The WRITE command overwrites existing file data from logical address 0 with length DLNG (DBW 22 in BEDB). If the data length DLNG specified in the WRITE command is shorter than the already existing file length, the old data remains starting at the logical address DLNG to the end of the file.

READ: Read data from a file

The complete file (see WRITE) is read into the specified data block DATDB, starting at data word DATDW.

DELETE: Delete existing file

- The specified file (file name) is removed from the directory.
- All file data is lost.
- The DELETE command is only permissible on SLGs with access rights “D”.
- A delete-protected file cannot be deleted. Use the ATTRIB command before deletion.

ATTRIB: Assign attributes (write or delete protection) to file

- A write or delete protection attribute can be assigned to the specified file. This file cannot then be overwritten or deleted.
- An attribute assigned to a file can be removed (attribute = 0). The file can then be overwritten or deleted again.
- The attribute assigned to a file is stored in the directory.

COVER: Protect MDS structure

The data structure of a data carrier can be protected against “unauthorized access”. An “unauthorized access” occurs if any user (e.g. supplier) wants to execute the following write operations on the MDS.

- delete file or add data
- increase or decrease file length
- format data carrier, if already formatted
- delete or assign attribute

A data memory layout (i.e., a specified DIR + FAT structure) specified by the authorized user (i.e., super user) cannot be changed or destroyed.

DIR: Read directory

The parameters of all existing files are read into DATDB/DATDW.

MDS-STATUS: Read MDS status (information about the operating status of the MDS)

Reads MDS specific data into DATDB/DATDW.

END: Conclude communication with the MDS (only MOBY U)

The command is recommended when a scanning_time other than 0 (standby time) is parameterized. Execution of this command optimizes the lifespan of the MDS battery. Execution of an additional MDS command will now be delayed until the sleep_time of the MDS expires.
Sleep_time can be poled with the MDS-STATUS command.

TRACE: Absolute MDS read

The contents of MDS memory are read into DATDB/DATDW from a physical start address and with a given length.
(Test function)

MOVE: Save system data from the file handler (directory + FAT + checksum)

Directory + FAT + checksum of the corresponding SLG station is saved in a data block.

- After a power failure, this information can be transmitted directly to the ASM (file handler) using a LOAD command. The checksum of the incoming MDS will correspond again with the internal checksum in the file handler. The process can continue immediately at full transport speed.
- Invoking the MOVE command is recommended after a corresponding modification to a directory (checksum change: flag in bit 9 in BEST). Saving system data with MOVE is only recommended if MOVE/LOAD is in general use.

LOAD: Transmit system data to ASM

This command can be invoked by the user to transmit data saved with MOVE to the file handler. Following a power failure, a LOAD command should be issued immediately after the RESET command (only if LOAD/MOVE is in general use).

SET-ANT: Turn antenna of the SLG on or off (only MOBY U)

This command is not needed during normal operation since the antenna is always on after an SLG is turned on.
The antenna must be turned off when two sensitive SLGs are to be located very close to each other. The application software must ensure that only one antenna is on at once.

ASM-STATUS: Read ASM status (information about the operating status of the ASM and SLG)

Reads ASM specific data into DATDB/DATDW.
(System command; no dialog takes place with the MDS).

RESET: Reset ASM and SLG

- The connection between FC 46 and ASM is reinitialized. The file handler parameters are set according to the FC 46 call parameters.
- If an **MDS is present** in the SLG field during “RESET”, all subsequent commands will refer to this (current) MDS.
- If **no MDS is present** in the SLG field during “RESET”, the commands will only be processed with the next MDS (for EAKO 0 with the new/next MDS, see also chapter 5).
- RESET can be invoked at any time. All current SLG actions will be immediately terminated with the risk of defective data structures on the MDS

Caution

A RESET command declares the current MDS directory and FAT in the file handler to be invalid. The directory and FAT of the next MDS to be processed must always first be read from the MDS (indicated by bit 9 in BEST). If this delay in reading the directory is to be avoided, the MDS directory must be transmitted to the file handler by the user through a LOAD command immediately after the RESET command.
(System command; no dialog with the MDS).

(System command; no dialog with the MDS.)

NEXT: Process the next MDS (see chapter 5)

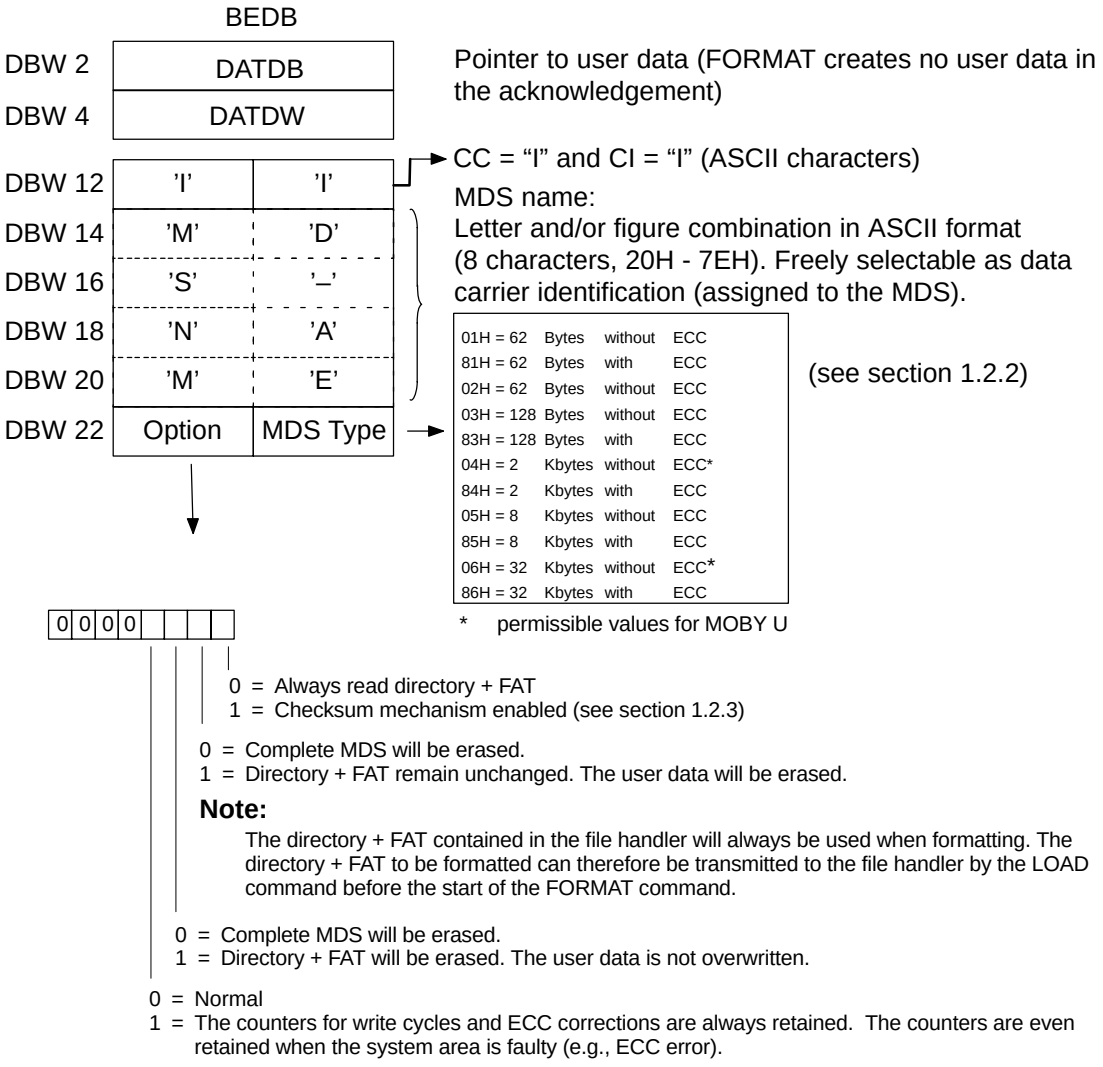
NEXT concludes the processing of an MDS (current MDS is still in the SLG field or has already left). This processing can consist of several commands. The next/following command will not be processed until the next MDS enters the field.

4.7 Setting the command parameters

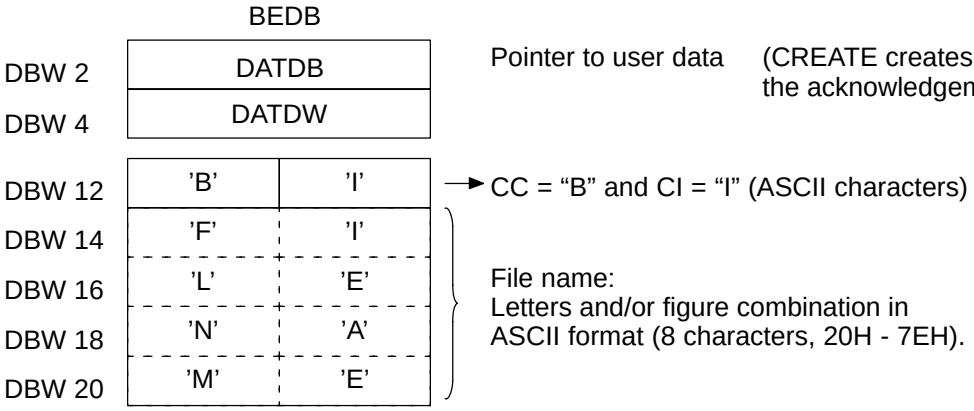
Before the above commands can be issued, the appropriate command parameters (DBW 12 - DBW 22) must be entered in BEDB**. Entering the parameters and issuing the command is only permissible if the "Ready Bit" (Bit 15 in BEST) is set.**

After a command has been started, FC 46 sets the ready bit to 0. The result of the started command is valid when the ready bit is set again and the start bit has been reset by the FC.

FORMAT



CREATE


BEDB
DBW 2
DATDB
DBW 4
DATDW
Pointer to user data
(CREATE creates no user data the acknowledgement)
DBW 12
'B'
'I'
CC = "B" and CI = "I" (ASCII characters)
DBW 14
'F'
'I'
DBW 16
'L'
'E'
DBW 18
'N'
'A'
DBW 20
'M'
'E'
File name:
Letters and/or figure combination in ASCII format (8 characters, 20H - 7EH).

QUEUE-WRITE

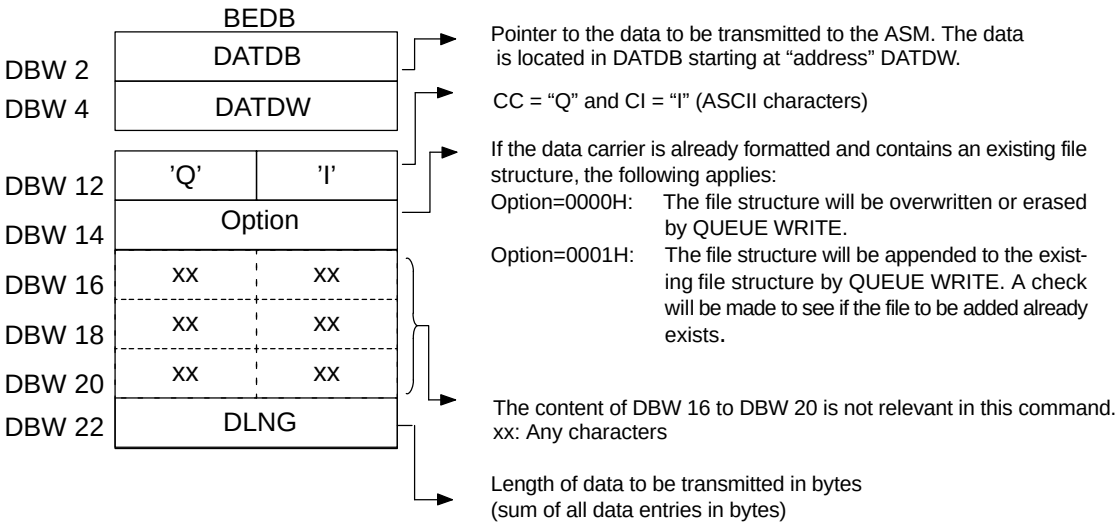


Table of data in DATDB:

DATDW:		
	'F'	'I'
	'L'	'E'
	'0'	'0'
	'0'	'1'
	Attribute	Length (H)
	Length (I)	Length (L)
	In reserve	Skip
	Data	Data
	Data	Data
	'F'	'I'
	'L'	'E'
	'0'	'0'
	'0'	'2'
	Attribute	Length (H)
	Length (I)	Length (L)
	In reserve	Skip
	Data	Data
	Data	Data
	Data	xx
	'F'	'I'
	'L'	'E'
	'0'	'0'
	'0'	'3'

File entry 1

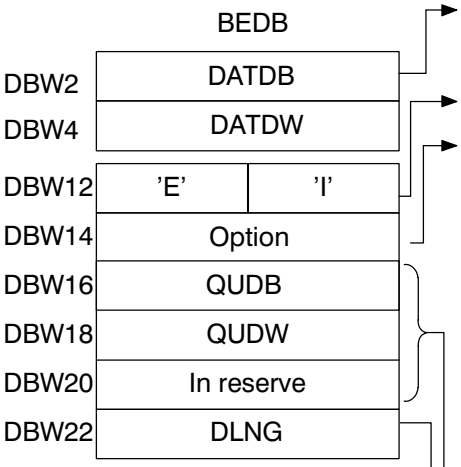
File entry 2

File entry 3

etc.

File name: (as before)
Attribute: Attribute for this file. The attribute will only be set after the data has been written.
In reserve: 1 byte reserved.
Skip: 'Skip' indicates by how many bytes the next file follows after the end of the previous file.
Example:
Skip = 0 hex (i.e.,the file name of the next file entry
immediately follows the data of the previous file.)
Skip = 1 hex can be set when the amount of data is an odd number. The next file name then starts with the "left data word".
Length: Number of bytes of data to be written to the specified file. The data is located using the "skip" parameter.
Data: The data to be written. This can also be a dummy entry. The amount of data in bytes must correspond to the "length" parameter.

QUEUE-READ



Pointer to user data which are to be transferred to the ASM. The command data are located in the DATDB starting with the "address" DATDW. The result is located in QUDB starting with data word QUDW.

CC = E"; KI = I" (ASCII)

Option byte

0000	No option. The data contain the files to be read.
0001	(Reserved for QUEUE-WRITE)
0002	All files longer than 0 are read by the MDS. DLNG must be preset with 0000.*
0004	No error message is issued if the actual file length is longer
than	the file length reserved in the data.
0008	The user data contain the files which are not to be read (i.e., except for the files specified in the command, all files of the data memory are read). The parameter length in the file entry can be disregarded for this option.
0010	The file data read are organized by word and placed directly behind one another (i.e., if the file length is an odd number, a skip byte with the value 20 hex is added to the data). The "skip" parameter then has the value 0001 in the acknowledgement. When this option is used, the "length" must always be set to FFFFFF in the file entries.
0022	All MDS files are read. In contrast to option 0002, blank files are also read (file length = 0). This option reads a complete data medium and then writes it to another MDS with QUEUE-WRITE.*

The options are coded by bit. Several options can also be set.
Example: 000C, 0012, 0018 and so on

Description of the data blocks for the acknowledgement for QUEUE-READ
QUDB: ≥ 1 = no. of the data block
QUDW: Start address of the result within the QUDB

Length in bytes of the data to be transferred

The maximum length for this command is 210 (i.e., 00D2) which results in a maximum of 15 file entries in the DATDB.

* Remarks for table A-1 (see appendix A.6) (QUEUE-READ):
No more data starting at byte 25 of the command.
Length (bytes 22 to 24) is 000000.

Table of data in DATDB for this command:

DATDW:		
	’F’	’I’
	’L’	’E’
	’0’	’0’
	’0’	’1’
	Reserved	Length (H)
	Length (l)	Length (L)
	In reserve	In reserve
	’F’	’I’
	’L’	’E’
	’0’	’0’
	’0’	’2’
	Reserved	Length (H)
	Length (l)	Length (L)
	In reserve	In reserve
	’F’	’I’
	’L’	’E’
	’0’	’0’
	’0’	’3’

File entry 1

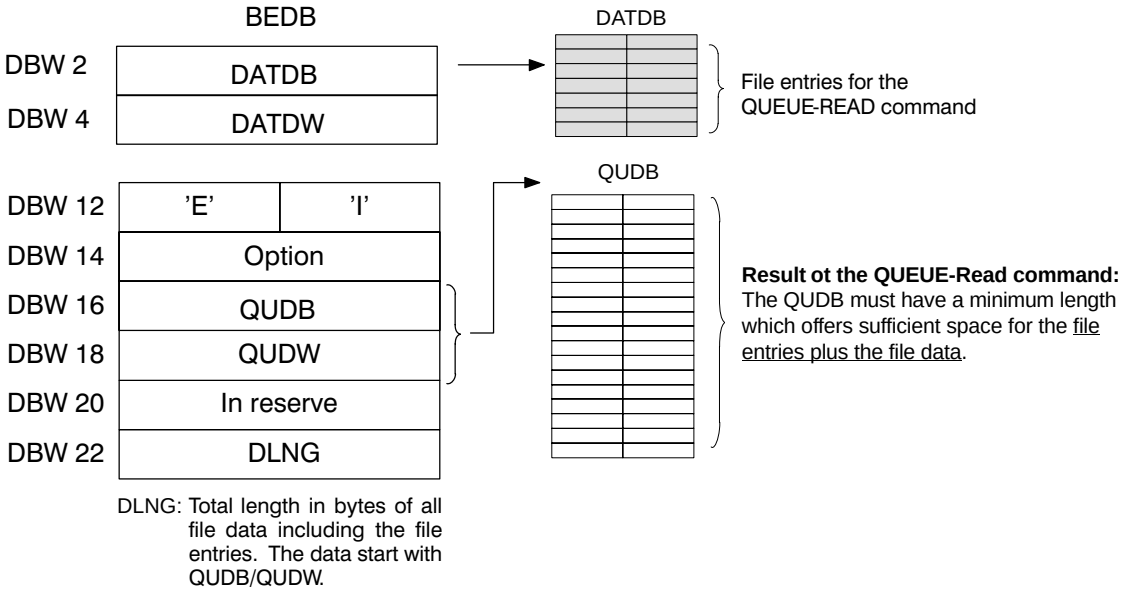
File name: File name (8 characters in ASCII format)
Length: Length in bytes of the data to be read for the specified file
FFFFFF = Complete file
xxxxxx = Length of the data in the result DB

File entry 2

File entry 3

And so on, up to a maximum of 15

Location of the read data for QUEUE-READ:

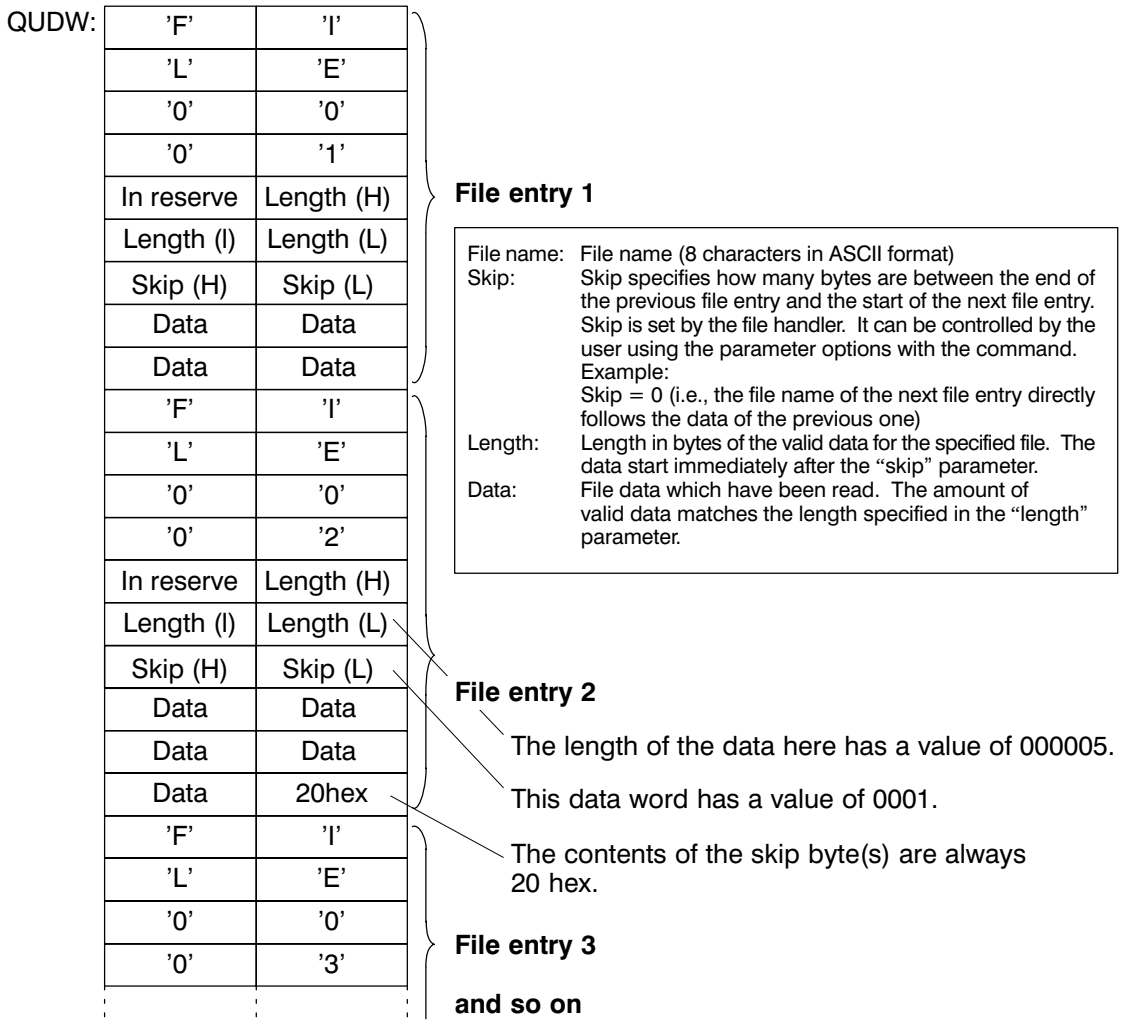


DLNG: Total length in bytes of all file data including the file entries. The data start with QUDB/QUDW.

Data in QUDB for the result:

The data from QUEUE-READ are stored in the QUDB starting with dataword QUDW, and not the DATDB parameterised in the command. The data files of the command (DATDB) are not affected by the result. The file entries of the command are transferred to the result and provided with additional parameters.

QUDB:



WRITE / UPDATE

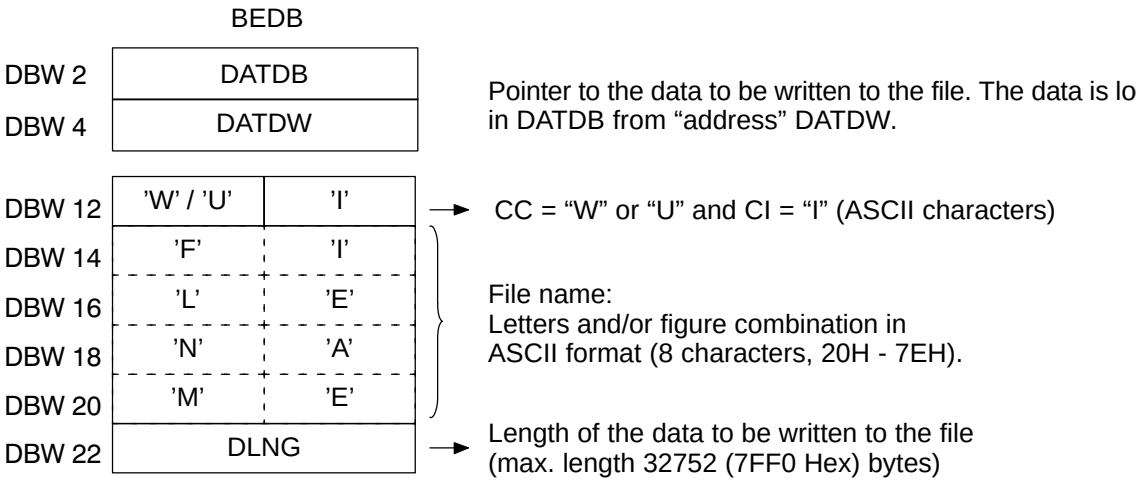


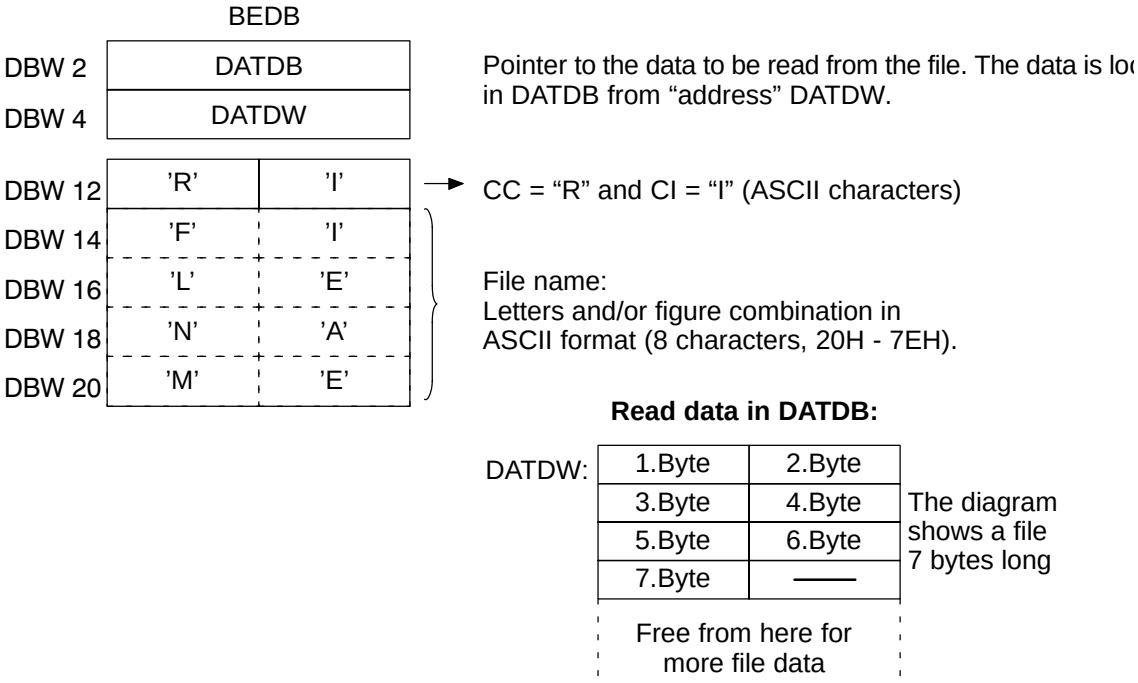
Table of the write data in DATDB:

DATDW:

1.Byte	2.Byte
3.Byte	4.Byte
5.Byte	6.Byte
7.Byte	——
Free from here for more file data	

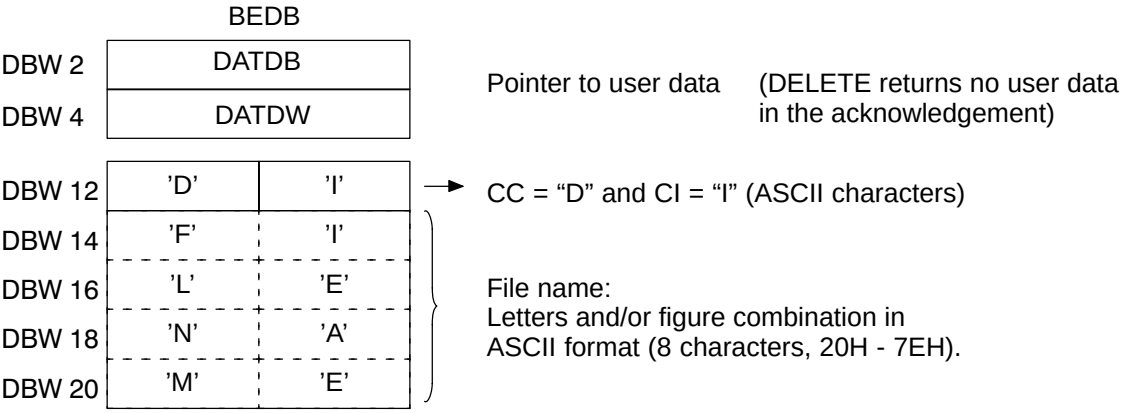
The diagram shows a file 7 bytes long

READ

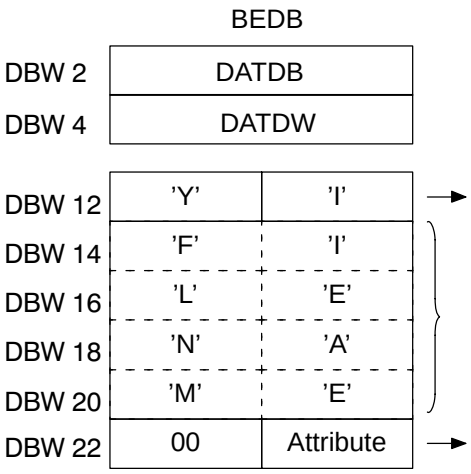


The read file data are located in DATDB starting at the "address" DATDW. The DLNG parameter (DBW 22 in BEDB) indicates how much data was read.

DELETE



ATTRIB



Pointer to user data (ATTRIB creates no user data in the acknowledgement)

CC = "Y" and CI = "I" (ASCII characters)

File name:
Letters and/or figure combination in ASCII format (8 characters, 20H - 7EH).

Attribute:

00H: no attribute for the data entered or delete existing attribute

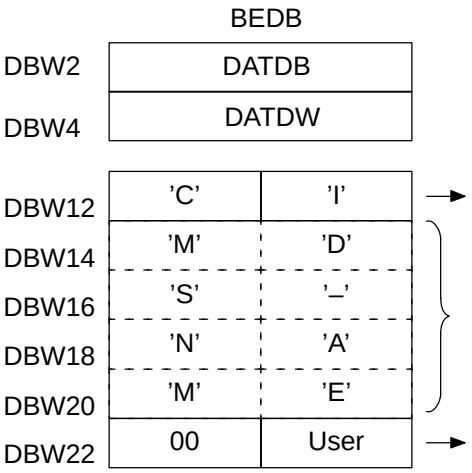
01H: attribute "read only"
The file can only be read (no DELETE or WRITE or UPDATE permissible).

02H: attribute "write once"
The file can be write-accessed once and then only read-accessed (no DELETE or WRITE or UPDATE permissible).

04H: attribute "fixed length"
The length of the file cannot be changed by write commands such as UPDATE, WRITE or APPEND.

05H: Same as 01H.

COVER



Pointer to user data (COVER returns no user data in the acknowledgement)

CC = “C” and CI = “I” (ASCII characters)

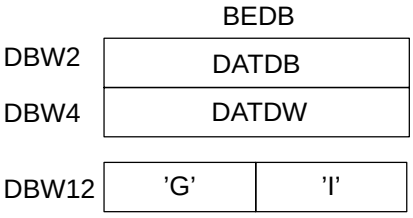
File name:
Letters and/or figure combination in ASCII format (8 characters, 20H - 7EH).
Corresponds to the MDS name given by FORMAT
.

User:

00H: Change to the MDS file structure permissible again

01H: The MDS is protected against “unauthorized access”.

DIR



Pointer to the data contained in the acknowledgement telegram. Data is located in DATDB starting at “address” DATDW.

CC = “G” and CI = “I” (ASCII characters)

The desired data are located in DATDB starting at “address” DATDW. The DLNG parameter (DBW 22 in BEDB) indicates how much data was read.

Table showing directory in DATDB:

Significance	Length in bytes	Type	Explanation
MDS name Checksum Free space	8 2 4	ASCII Binary Binary	MDS name Checksum Remaining free memory space (user data) on the MDS in bytes
For each file:			
File name File length Attribute	8 3 1	ASCII Binary Binary	File name (8 characters) Absolute length of file in bytes File attribute

MDS-STATUS

BEDB

DBW2	DATDB
DBW4	DATDW

Pointer to the data contained in the acknowledgement telegram. Data is located in DATDB starting at “address” DATDW.

DBW12	’F’	’I’
DBW14	00	Mode
DBW16	KW	Year

→ CC = “F” and CI = “I” (ASCII characters)

→ 00H: Read logical MDS parameters
01H: Read physical MDS parameters (only MOBY U)

→ KW = 1 to 53 (current calendar week, hex, 14hex ≙ 20th week)
Year = 1 to 99 (current year, hex, 01 ≙ 2001)

The desired data are located in DATDB starting at “address” DATDW. The DLNG parameter (DBW 22 in BEDB) indicates how much data was read.

Mode 00H: Table of logical MDS parameters in DATDB:

Meaning	Length in Bytes	Type	Explanation
MDS name	8	ASCII	MDS name (ASCII characters)
MDS type	1	Binary	MDS type specified in FORMAT command
MDS capacity	3	Binary	Total user data area
Free capacity	3	Binary	User data area still free
Free directory entries	2	Binary	This number of files can still be set up.
Battery indicator	1	Binary	Bit 2 = 0: Battery OK Bit 2 = 1: Battery poor Bit 3 = 1: Battery in MDS 507 is empty. Bit 7 = 0: MDS is unprotected. Bit 7 = 1: MDS is protected (COVER).
Write cycles	3	Binary	Number of SLG stations that have processed the MDS
ECC corrections	1	Binary	Number of ECC corrections

Mode 01H: Table of physical MDS parameters in DATDB:

Meaning	Length in Bytes	Type	Explanation
MDS no.	4	Binary	MDS number
Physical MDS type	1	Binary	84 hex = MDS with 2 Kbytes without ECC 86 hex = MDS with 32 Kbytes with ECC
Sum of station accesses	4	Binary	Sum of station accesses
Sum of search_mode accesses	2	Binary	The upper 16 bits of a 32-bit value showing the number of search states before the last time sleep_time was changed
Date last time sleep_time was changed	2	Binary	1st byte: Calendar week 2nd byte: Calendar year (without century)
battery_left	2	Binary	Remaining battery life in days
sleep_time	1	Binary	Sleep_time value set on MDS 04 = Default = 320 msec

END (only MOBY U)

BEDB

DBW2	DATDB	
DBW4	DATDW	
DBW12	'K'	'I'
DBW14	00	Mode

Pointer to user data (END doesn't create user data in the acknowledgment)

→ KK = "K" and KI = "I" (ASCII characters)

→ 00H: Processing with MDS is finished.
01H: Processing pause with MDS
Only recommended when a scanning_time other than 0 is parameterized. Execution of this command optimizes the life of the MDS battery.

TRACE

BEDB

DBW2	DATDB	
DBW4	DATDW	
DBW12	'T'	'I'
DBW14	00	Addr.(High)
DBW16	Addr.(Inner)	Addr.(Low)
DBW18	Length (High)	Length (Low)

Pointer to the data read. The data is located in DATDB starting at "address" DATDW

→ CC = "T" and CI = "I" (ASCII characters)

Start address on the MDS (Hex)

→ Length of data to be read (max. length 7FF0 Hex)

Table of READ data in DATDB:

DATDW:		
	1.Byte	2.Byte
	3.Byte	4.Byte
	5.Byte	6.Byte
	7.Byte	——

The diagram shows a block of data that has been read which is 7 bytes in length.

The desired data are located in DATDB starting at "address" DATDW. The DLNG parameter (DBW 22 in BEDB) indicates how much data was read.

MOVE

BEDB

DBW2	DATDB	Pointer to the data contained in the acknowledgement telegram. Data is located in DATDB starting at "address" DATDW.
DBW4	DATDW	
DBW12	'M' 'I'	→ CC = "M" and CI = "I" (ASCII characters)

The desired data are located in DATDB starting at "address" DATDW. The DLNG parameter (DBW 22 in BEDB) indicates how much data was read.

Data in DATDB:

DATDW:

1st byte	**2nd byte**
3rd byte	4th byte
5th byte	6th byte
7th byte	8th byte
9th byte	10th byte
11th byte	——

→ **1st byte** and **2nd byte** correspond to parameter DLNG when issuing the LOAD command
The table shows the structure of the user data in DATDB.

Note: After executing the MOVE command the corresponding DATDB may no longer be changed (no write access to the data in DATDB). This data block must be parameterized in DBW 2 with the LOAD command.

LOAD

BEDB

DBW2	DATDB	Pointer to the user data to be transmitted to the ASM. Data is located in DATDB starting at "address" DATDW.
DBW4	DATDW	
DBW12	'O' 'I'	→ CC = "O" and CI = "I" (ASCII characters)
DBW14	xx xx	The content of DBW 12 - DBW 20 is not relevant in this command xx: Any characters
DBW16	xx xx	
DBW18	xx xx	
DBW20	xx xx	
DBW22	DLNG	→ Content of DATDW: **1st byte** and **2nd byte** (corresponds to the length of the data to be transmitted).

Caution: The DATDB specified in DBW 2 must correspond to the MOVE command data block.

Data in DATDB:

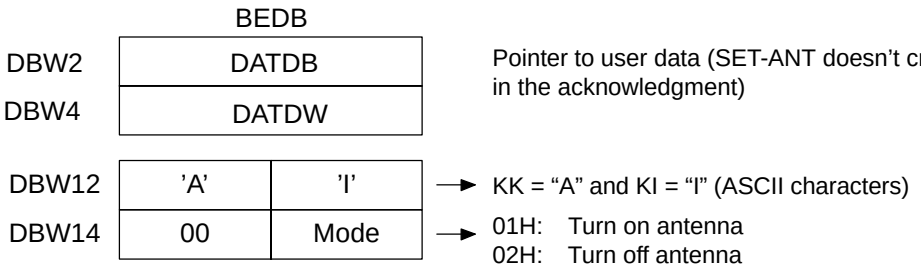
DATDW:

1st byte	**2nd byte**
3rd byte	4th byte
5th byte	6th byte
7th byte	8th byte
9th byte	10th byte
11th byte	——

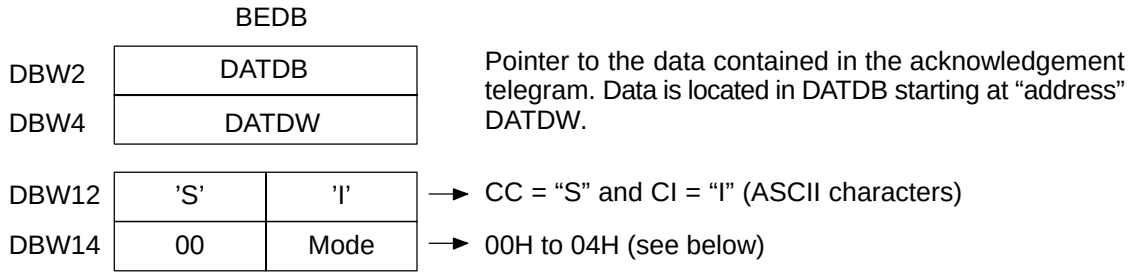
→ Enter in DLNG

The table shows the structure of the data in DATDB.

SET-ANT (only MOBY U)



ASM-STATUS



The desired data are located in DATDB starting at "address" DATDW. The DLNG parameter (DBW 22 in BEDB) indicates how much data was read.

Mode 00H: Table of data in DATDB:

Meaning	Length in Bytes	Type	Explanation
Version number	8	ASCII	Version number
Last instruction	26	ASCII/ binary	Last instruction to ASM
Last acknowledgement	26	ASCII/ binary	Last acknowledgement from ASM
Connection status between SLG and ASM	1	Binary	00 hex = connection OK 01 hex = connection not OK

Mode 01H: Table of SLG parameters in DATDB:

Meaning	Length in Bytes	Type	Explanation
status_info	1	Binary	01 hex: Mode = SLG status
hardware	1	ASCII	HW model
hardware_version	2	Binary	HW version
loader_version	2	Binary	Bootstrap loader version
firmware	1	ASCII	FW model
firmware_version	2	Binary	FW version

Meaning	Length in Bytes	Type	Explanation
driver	1	ASCII	Driver model; ’1’ = 3964R
driver_version	2	Binary	Driver version
interface	1	Binary	RS 232/RS 422 interface
baud	1	Binary	Baud rate
reserved	3	Binary	3 bytes in reserve
distance_limiting_SLG	1	Binary	Range limit (see chapter 3.2)
multitag_SLG	1	Binary	Number of MDSs in the antenna field which can be processed
field_ON_control_SLG	1	Binary	Proximity switch mode (see chapter 3.2)
field_ON_time_SLG	1	Binary	Proximity switch time (see chapter 3.2)
sync_SLG	1	Binary	Semaphore control (synchronization with SLG, 01 = yes)
status_ant	1	Binary	Status of antenna: 01 hex = antenna on
stand_by	1	Binary	Standby time of MDS (see chapter 3.2)
MDS_control	1	Binary	Presence (see RESET) 0 = Operation without presence 1 = Operation with presence (see ANW-STATUS telegram)

Mode 02H: Table of the SLG parameters in DATDB:

Meaning	Length in Bytes	Type	Explanation
status_info	1	Binary	Status information 02 hex: Mode = SLG diagnosis I
number_functions	1	Binary	Number of functions last called, 1 to 33
function_number [1]	7	Binary	1st function Example: Function recognition, identification of whether MDS or SLG function, and so on
...		Binary	...
function_number [n]	7	Binary	No. of function (function data with length of x + 1 bytes)

Mode 03H: Table of the SLG parameters in DATDB:

Meaning	Length in Bytes	Type	Explanation
status_info	1		
number_errors	1		

Mode 04H: Table of the SLG parameters in DATDB:

Meaning	Length in Bytes	Type	Explanation
status_info	1	Binary	Status information 04 hex: Mode = SLG diagnosis III
number_MDS	1	Binary	Number of MDSs last identified, 1 to 24
MDS_number [1]	4	Binary	Value of 2^0 to 2^{31}
...		Binary	...
MDS_number [n]	4	Binary	Value of 2^0 to 2^{31}

New Start

This function allows the connection between FC 46 and file handler to be reinitialized and synchronized. In addition, the FC 46 calculates the absolute addresses of BEDB and the data field indicator (DATDB/DATDW) and places these in BEDB. A new start command must therefore be executed each time the FC parameters change. In addition, with EAKO 0 the new start/RESET command contains the NEXT function (see section 4.6).

The new start command is issued by setting bit 3 in BEST (DBW 0).

RESET

Same function as new start except that the ASM is not initialized again. The RESET command should be used during running operation.

The RESET command is started by setting bit 0 in BEST (DBW 0).

NEXT

This function enables the user to terminate processing of the MDS, which may have consisted of a number of commands. The file handler ensures that the command following the NEXT command is not executed on the MDS which is now being processed or is still present, but on the following/new MDS (see chapter 5). In time-critical applications, this function offers the user the ability to transmit a command for the next MDS, even while the old MDS is still in the transmission window or is just leaving, and thus save time.

The NEXT command is issued by setting bit 4 in BEST (DBW 0).

MDS Entry/Exit Checking (EAKO) in an SLG Field

5

This section describes various ways of detecting the presence of an MDS in the transmission window of an SLG. The user can select the most suitable method for the application using the FB parameter “EAKO”.

Table 5-1 Entry/exit checking of an MDS

EAKO	Recognition of an MDS	NEXT Command	MDS Flow Control	Command Processing
0	Scan field. ANW bit in BEST is available to the user for scanning.	Mandatory	Continuous monitoring via the file handler	Command intermediately stored by the file handler until a permissible MDS enters the transmission window of the SLG.
1	User program: Limit switch or ANW bit in BEST	Yes, but not mandatory	Not continuous	Operating mode with timeout. Command is executed immediately. An error message is generated if no MDS is located in the transmission window.
4	User program: Limit switch of ANW bit in BEST	Yes, but not mandatory	Not continuous	Command intermediately stored by the file handler until a permissible MDS enters the transmission window of the SLG.
5	User program: Limit switch of ANW bit in BEST	no	Not continuous	Command intermediately stored by the file handler until a permissible MDS enters the transmission window of the SLG. (Test operation)

5.1 Definition of Terms

Each SLG that processes an MDS writes its **station-specific SLG number to the system area of the MDS.** The SLG number is written automatically before the first command to be executed on the MDS. The SLG number can be used to detect whether it is a new or an old data carrier that is currently passing the SLG station.

New MDS:

SLG numbers (SLG no. on MDS and station-specific SLG no.) do not correspond

Old MDS:

SLG numbers correspond

Current MDS:

All active commands apply to this MDS

Next MDS bit:

Internal status flag in the file handler

The old MDS remains current (all commands relate to this MDS) until it is closed by a NEXT command. All subsequent commands on this SLG refer to the next new MDS. This is achieved by having the file handler set the “next MDS bit” when it receives a NEXT command. If the old MDS reenters the SLG transmission window as the next MDS (SLG numbers correspond), no command is issued (the “next MDS bit” remains set). When a new MDS enters the transmission window, the “next MDS bit” will be reset and the station-specific SLG number will be written to the system area of the MDS (the new MDS thus becomes the old/current MDS).

This mechanism is based on the assumption that all SLG stations in the system have different SLG numbers (set by the “SLG” parameter; see section 3.2).

5.2 Entry/Exit Checking Mode 0 (EAKO 0)

After receiving a command from the FC 46, the file handler continuously attempts to establish communication with the MDS via the connected SLG. As soon as an MDS appears in the transmission window, the file handler begins to process the command. With this method, the user can issue a command to the file handler at any time. This command is stored until it can be processed on a valid MDS.

The NEXT command plays a decisive role in this type of control. The issuing of the NEXT command means that:

- the processing of the old/current MDS is terminated
- immediately after NEXT, a command can be issued that will not be processed until the arrival of a “new” MDS.
- the file handler ensures that a command will only be executed on the next “new” MDS. Thus, if the old (already completely processed) MDS returns as the new MDS into the SLG transmission window, the command that was issued will not be executed.

Issuing the NEXT command is always permissible if:

- the old/current MDS is still in the SLG transmission window
  the old/current MDS has already left (no MDS in SLG transmission window)
- new/next MDS is in the SLG transmission window (the new MDS thus becomes the current one)

This mode of EAKO operation allows a continuous monitoring of the MDS flow control. If a new MDS enters the SLG transmission window before the old/current MDS is terminated by a NEXT command, the following occurs:

1. If a command has been issued (not NEXT), then an error message is output. The user has not terminated the old/current MDS with NEXT or an MDS for which this command is not intended has entered the transmission window.
2. If the NEXT command has been issued, the old/current MDS will be terminated and the new MDS, which is now in the SLG transmission window, becomes the current MDS (and thus also the old MDS). All further commands will be executed on this current MDS until a further NEXT terminates the processing.
3. An error situation arises if the new MDS leaves the SLG transmission window again, since no actions were performed on the new MDS while it was in the window. The error message will not appear until a further command is issued.

5.3 Entry/Exit Checking Mode 1 (EAKO 1)

If a command is issued to the file handler, the user must ensure that an MDS is present in the SLG transmission window. The command will be executed immediately. If there is no MDS in the SLG transmission window, an appropriate error message is output (timeout error). This procedure can be used in all cases in which the user recognizes the presence of an MDS (e.g. by limit switches). Use of the NEXT command is possible, but not mandatory. The user handles part of the MDS control in his program.

5.4 Entry/Exit Checking Mode 4 (EAKO 4)

In this mode of operation absolutely no checking of MDS entry/exit is carried out by the file handler. The MDS control must be handled entirely by the user.

If a command is issued to the file handler, it will be stored until an MDS arrives in the transmission window of the SLG. The file handler then commences the processing of the command. Use of the NEXT command is possible, but not mandatory.

5.5 Entry/Exit Checking Mode 5 (EAKO 5)

In this mode of operation absolutely no checking of MDS entry/exit is carried out by the file handler. The MDS control must be handled entirely by the user.

If a command is issued to the file handler, it will be stored until an MDS arrives in the transmission window of the SLG. The file handler then commences the processing of the command. When executing commands that do not change the DIR + FAT structure (i.e. no change to the checksum), the system area is not updated. This increases the processing speed in especially time-critical applications, (e.g., ECC oper-ation of EEPROM MDS). Write-accesses to an EEPROM MDS are reduced.

Operating with EAKO 5 automatically precludes the use of the NEXT command.

Caution

The use of EAKO 5 is a special mode of operation and it should therefore be regarded as test operation.

5.6 Diagram showing possible MDS states on an SLG

Notes for the understanding and use of the following diagram

The following section describes all types of status which are important for correct processing. The significance of the remaining types of status will be self-evident from an understanding of those described here.

The source point of all considerations is the RESET command. In principle, this can be issued if:

– there is no MDS in the SLG transmission window. Status 11 is then valid, i.e. “MDS not present, Next MDS bit set, command not present”.

Note

The “Next MDS bit” is set in the file handler since the NEXT function is implemented in the RESET command (EAKO 0 only). The “Next MDS bit” is not set in the case of EAKO 1 or EAKO 4. Status 5 is then valid, i.e. “MDS not present, Next MDS bit not set, command not present”.

– an MDS (old or new) is in the MDS field. Status 3 is then valid, i.e. “old/current MDS present,
Next MDS bit not set, command not present”.

Note

The “Next MDS bit” is not set, since the file handler always considers the MDS, which is present when the RESET command is issued, the old/current MDS (whether it is new or old) and thus resets the “Next MDS bit”.

In status 11 the following can occur.

– a new MDS enters the transmission window. Since the Next MDS bit is set, this MDS immediately becomes the old/current MDS and the Next MDS bit is reset. Status 3 is valid.

– an old MDS enters the transmission window. Since the Next MDS bit is set (processing of the old MDS is terminated), no further command can be issued to this MDS. Status 9 is valid.

– a NEXT command is issued by the FC 46. Since the Next MDS bit is already set, the file handler remains in status 11.

– a command is issued by the FC 46. No MDS is present, the Next MDS bit is set (do not process this command until the next/new MDS). The command is temporarily stored by the file handler. Status 23 is valid.

In status 23 the following can occur.

- an old MDS enters the transmission window. Since the Next MDS bit is set, no processing of the temporarily stored command occurs. Status 21 is valid.
- a new/next MDS enters the transmission window. This immediately becomes the old/current MDS and the Next MDS bit is reset. The command is issued and status 27 is valid. Once the command has been processed, status 3 is valid again (another command can be issued for this MDS).

In status 3 the following can occur.

- another command can be issued by the FC 46. Status 27 is then valid.
- a NEXT command can be issued by the FC 46. Processing of the current/old MDS is terminated. Status 9 is valid.
- the old/current MDS leaves the transmission window. Status 5 is then valid.

In status 5 the following can occur

- the old MDS returns to the transmission window. Status 3 is valid.
- a NEXT command is issued by the FC 46. The file handler sets the Next MDS bit internally and goes into status 11 (waiting for the next/new MDS).
- a command is issued by the FC 46. Since no MDS is present, this command will be temporarily stored by the file handler (not with EAKO 1). Status 17 is valid.
- a new MDS enters the SLG transmission window. Status 1 is valid.

Note

Continuous MDS control is effective in status 1 (EAKO 0 only).

The old/current MDS left the field in status 3 (status 5). The new MDS enters the field, therefore status 1. The following can now occur:

- a command can be issued by the FC 46. This is an error condition, since the old/current MDS has not been terminated by a NEXT command and a command for the new MDS has already been issued.
- the new MDS leaves the SLG transmission window again. This is also an error condition, since the new MDS has not been processed by a command (the new MDS “slips through”). An error message is output when the file handler issues the next command.

The codes shown below are used in the diagram on the following page:

These MDS states are possible at an SLG:

$\overline{\text{MD}}$ MDS not present

MDN new/next MDS present

MDA current or old (no longer current) MDS present

BD command present (has been issued by the FC 46)

$\overline{\text{BD}}$ command not present (has not yet been issued by the FC 46)

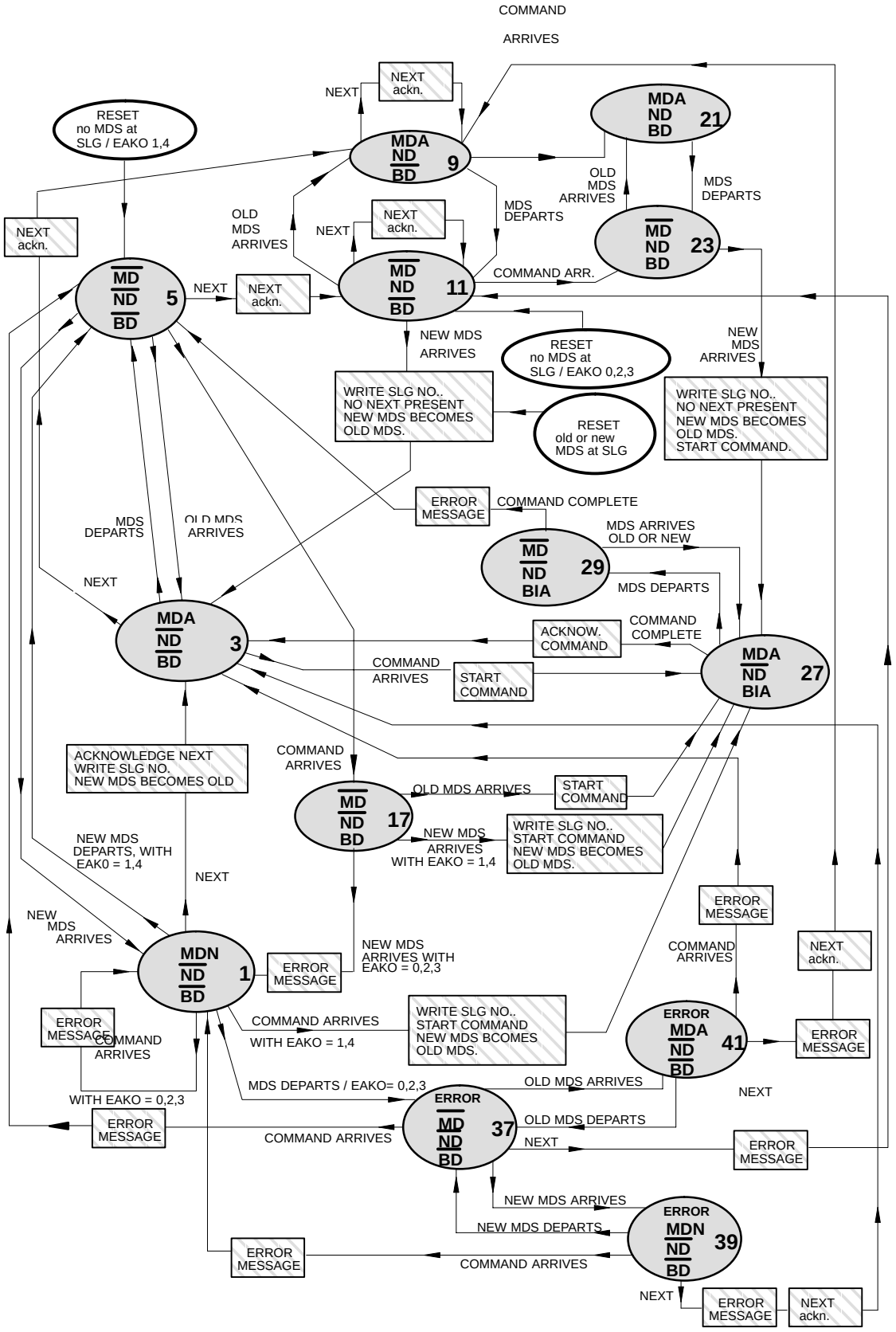
BIA command in progress (a command is now being executed)

ND “Next MDS bit” set in the file handler (NEXT command had been issued by the FC 46). The processing of all subsequent commands takes place only with the next/new MDS.

$\overline{\text{ND}}$ “Next MDS bit” not set in the file handler (NEXT command has not yet been issued by the FC 46).


File handler action


Status at an SLG station

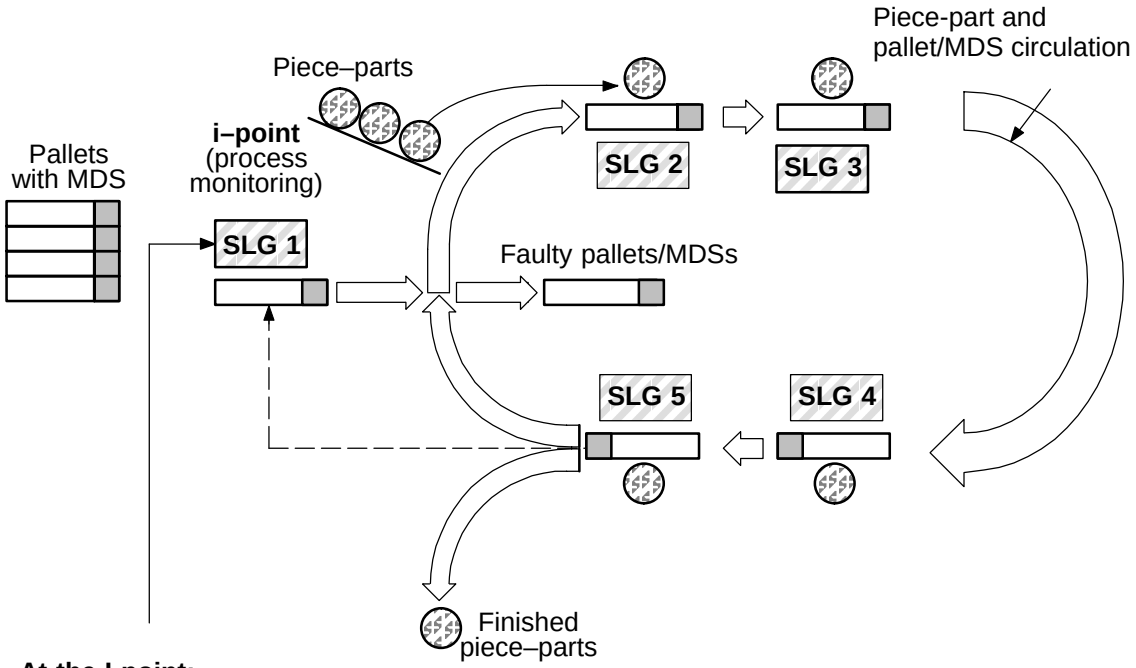

COMMAND
ARRIVES
NEXT
ackn.
NEXT
RESET
no MDS at
SLG / EAKO 1,4
MDA
ND
BD
9
MDA
ND
BD
21
OLD
MDS
ARRIVES
MDS
DEPARTS
NEXT
ackn.
OLD
MDS
ARRIVES
NEXT
ackn.
MDS
DEPARTS
MD
ND
BD
23
NEXT
ackn.
MD
ND
BD
5
NEXT
NEXT
ackn.
MD
ND
BD
11
COMMAND ARR.
NEW MDS
ARRIVES
RESET
no MDS at
SLG / EAKO 0,2,3
NEW
MDS
ARRIVES
WRITE SLG NO..
NO NEXT PRESENT
NEW MDS BECOMES
OLD MDS.
RESET
old or new
MDS at SLG
WRITE SLG NO..
NO NEXT PRESENT
NEW MDS BECOMES
OLD MDS.
START COMMAND.
ERROR
MESSAGE
COMMAND COMPLETE
MDS
DEPARTS
OLD MDS
ARRIVES
MDS ARRIVES
OLD OR NEW
MD
ND
BIA
29
NEXT
MDS DEPARTS
MDA
ND
BD
3
ACKNOW.
COMMAND
COMMAND
COMPLETE
COMMAND
ARRIVES
START
COMMAND
MDA
ND
BIA
27
ACKNOWLEDGE NEXT
WRITE SLG NO.
NEW MDS BECOMES OLD
COMMAND
ARRIVES
OLD MDS ARRIVES
START
COMMAND
MD
ND
BD
17
NEW MDS
ARRIVES
WITH EAKO = 1,4
WRITE SLG NO..
START COMMAND
NEW MDS BECOMES
OLD MDS.
NEW MDS
DEPARTS, WITH
EAK0 = 1,4
NEXT
ERROR
MESSAGE
NEW
MDS
ARRIVES
MDN
ND
BD
1
ERROR
MESSAGE
NEW MDS
ARRIVES WITH
EAKO = 0,2,3
COMMAND
ARRIVES
NEXT
ackn.
ERROR
MESSAGE
COMMAND
ARRIVES
COMMAND ARRIVES
WITH EAKO = 1,4
WRITE SLG NO..
START COMMAND
NEW MDS BCOMES
OLD MDS.
ERROR
MDA
ND
BD
41
ERROR
MESSAGE
WITH EAKO = 0,2,3
MDS DEPARTS / EAKO= 0,2,3
OLD MDS ARRIVES
NEXT
ERROR
MESSAGE
COMMAND ARRIVES
ERROR
MD
ND
BD
37
OLD MDS DEPARTS
NEXT
ERROR
MESSAGE
NEW MDS ARRIVES
NEW MDS DEPARTS
ERROR
MDN
ND
BD
39
ERROR
MESSAGE
COMMAND ARRIVES
NEXT
ERROR
MESSAGE
NEXT
ackn.

System Configuration 6

6.1 Example of a Fixed File Structure in a Manufacturing Process

A fixed file structure in the manufacturing process is distinguished by the fact that the individual SLGs only perform READ or WRITE/UPDATE functions on the MDS. Commands such as DIR, MDS/ASM-STATUS, DELETE, FORMAT or CREATE, are only carried out at “i-points” (information points) in the system in order to check the incoming and outgoing MDSs. Further advantages of this system configuration:

- Time-consuming operations such as FORMAT occur at the “i-point” and not in the production process.
- Since the same file structure is present on every MDS, this can be written to the MDS at the “i-point”.
- If the file structures on the MDS are structured accordingly, the checksum (and thus also the directory + FAT) always remains the same. By using the LOAD and MOVE commands an MDS can also be processed dynamically.
- At the end of the production process (SLG 5 in the following figure), an MDS can be processed again at the “i-point” if required.



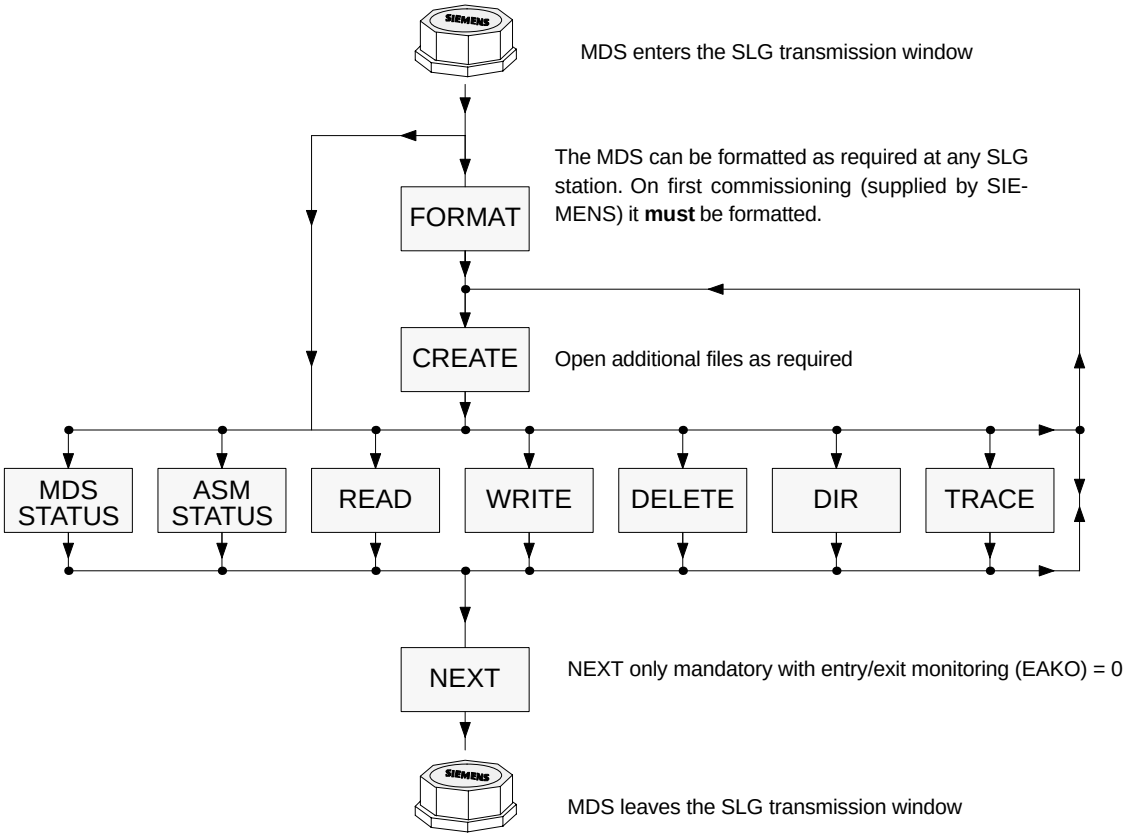
At the I-point:

- the MDS is formatted
- file structure is created (CREATE)
- faulty MDSs are rejected
- production data is monitored

6.2 Example of a Variable File Structure in a Production Process

A variable file structure in the manufacturing process is distinguished by the fact that any commands can be processed by any SLG. For instance:

- additional files can be opened
- existing files can be deleted
- MDS can be formatted as required
- data of completely differing lengths can be written to files



Commissioning Instructions for ASM with FC 46

7

Inserting the module

- Set the desired PROFIBUS address using the DIP switch (see section 2.1)
- Connect the module to PROFIBUS-DP and 24 V DC.
- Connect the SLG

Configuring PROFIBUS

- Link the GSD file SIEM804D.GSD (for ASM 451) or SIEM80B6.GSD (for ASM 452) to the configuration tool (e.g., WINCOM), and configure the address area. The ASM 451 requires 1 byte and the ASM 452 requires 2 bytes of input and output for the cyclic image.
- **Only ASM 452:**
  The basic parameterization of the ASM is done with the GSD file. The basic parameterization can usually be found under “Eigenschaften” of the ASM in the hardware configuration. The settings permitted with FC 46 are “MOBY I Filehandler” or “MOBY U Filehandler.”
  When the “MOBY U Filehandler” setting is used, the baud rate to the ASM can also be selected.

Loading the FC 46

- Load the project with FC 46 to the automation system.

Setting the default OB values

- Preset the “BEST” parameters in the OBs for cold start and restart as follows:
  for restart 0008 Hex
  for cold start 0008 Hex (see section 4.3 and section 8.5)

Note: When operating several SLGs from one automation system, “BEST” must be preset in every BEDB.

Creating the data blocks

- Create BEDB for every channel (length ≥ DBW 700) (see sections 4 and 8.6)
- Create DATDB if DATDB different to BEDB (see sections 4 and 8.6)

Calling up the FC 46 in the user program

- Always call up FC 46 absolutely (CALL FC 46)
- Determine the “ADR”, “BEDB”, “EAKO”, “ECC”, “RWD”, “SLG” FB parameters (see section 3.2)
- **Only MOBY U:**
  The parameters scanning_time, dili_multitag and field_ON_con_tim must also be supplied with values. Cf. chapter 3.2.

Program processing

- Call up the user program e.g. in OB 1 (cyclic call).

7.1 PROFIBUS Process Image

Communication between FC 46 and ASM has already been described in section 1.3. The cyclic I/O bytes will be presented here. This information can be useful when trouble shooting in a complex environment. The PAE byte (i.e., cyclic byte of ASM) shown below can be indicated with the “status/control” command.

0 0	No ASM is present under this I/O byte. — DP bus has not started up. — ASM is not configured correctly on the master. — Slave is not on bus (i.e., no power for the slave).
8 8	PROFIBUS has started up. ASM reports a startup. ASM is ready for a first command.
1 0	First file handler command is being processed on the ASM.
3 0	ASM has transferred the acknowledgment for the first FH command.
3 8	Second file handler command is being processed on the ASM.
5 8	ASM has transferred the acknowledgment for the second FH command.
4 0	Third file handler command is being processed on the ASM.
6 0	ASM has transferred the acknowledgment for the third FH command.
6 8	Fourth file handler command is being processed on the ASM.
0 8	ASM has transferred the acknowledgment for the fourth FH command.
1 0	Fifth file handler command is being prcessed on the ASM.

and so forth

Combinations of the I/O byte other than shown here are also possible.
Bit 0 = ANW is of special importance. See also appendix A.1.

7.2 Configuring PROFIBUS

Table 7-1 Default values for PROFIBUS with ASM 451 and ASM 452 (see also GSD files)

Values for Master Module											
ASM 451	Baud rate (kbit/sec)	9.6	19.2	45.45	93.75	187.5	500	1500	3000	6000	12000
	T_{SL} (T_{bit})	100	100	100	100	100	200	300	400	600	1000
	$MinT_{SDR}$ (T_{bit})	11	11	11	11	11	11	11	11	11	11
	$MaxT_{SDR}$ (T_{bit})	20	20	20	20	20	20	30	30	60	120
ASM 452	Baud rate (kbit/sec)	9.6	19.2	45.45	93.75	187.5	500	1500	3000	6000	12000
	T_{SL} (T_{bit})	100	100	600	100	100	200	300	400	600	1000
	$MinT_{SDR}$ (T_{bit})	11	11	11	11	11	11	11	11	11	11
	$MaxT_{SDR}$ (T_{bit})	60	60	250	60	60	100	150	250	450	800

T_{SL}: Slot time
Maximum wait time after the last bit of a call telegram has been sent until the receipt of the first telegram character of the acknowledgment.
The setting is made in the PROFIBUS master. The default values set there do not need to be changed for normal operation.

$MinT_{SDR}$: Minimum time a slave must wait before it replies

$MaxT_{SDR}$: Maximum time after which a slave must respond

The change is made via WINCOM or an appropriate configuration program

During startup, the ASM always uses $MinT_{SDR}$ = 11 T_{bit}

The following table shows the maximum length of a PROFIBUS segment.
Use of up to 3 repeaters increases the maximum PROFIBUS cable length to up to four times the length stated in the table.

Table 7-2 PROFIBUS cable length based on the baud rate

Baud rate in kbit/sec	9.6	19.2	93.75	187.5	500	1500	3000	6000	12000
Cable length in m	1200	1200	1200	1000	400	200	100	100	100

Possible optimization on PROFIBUS

Data communication on PROFIBUS can be optimized with the following settings.

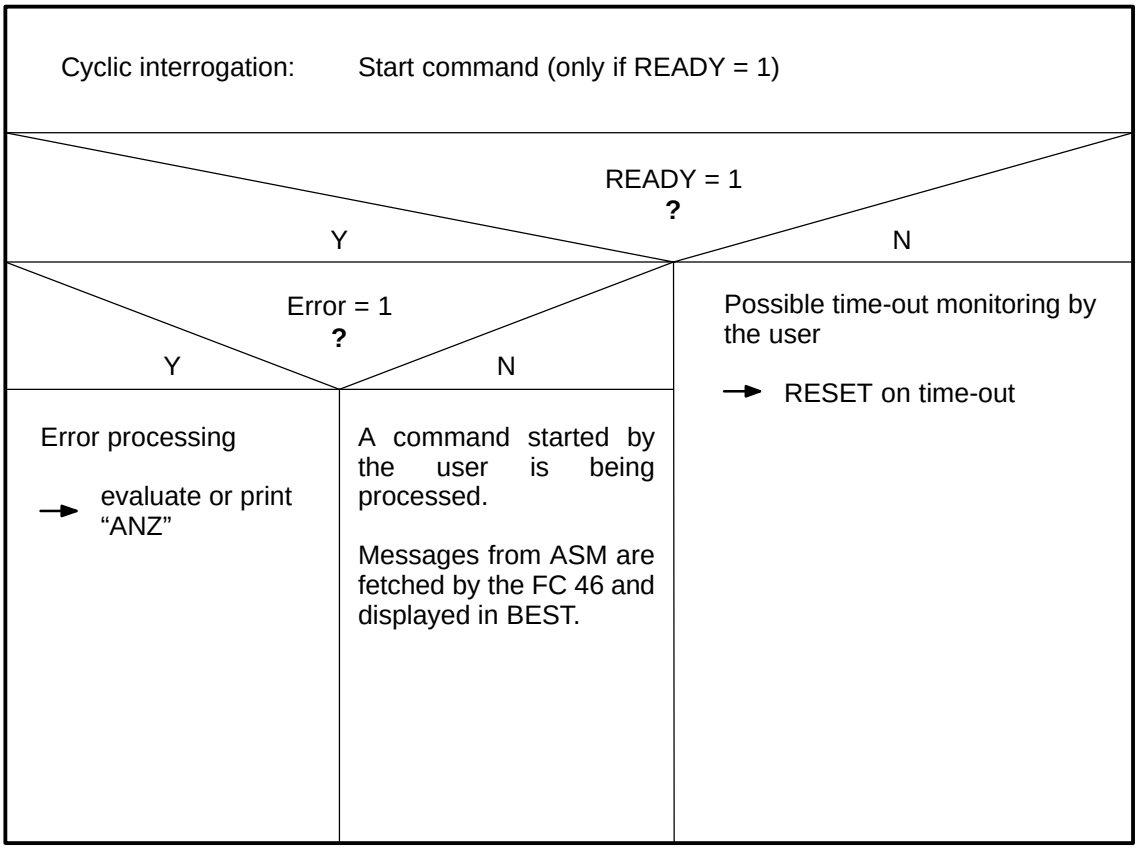
- MinT_{SDR} as short as possible
- Polling frequency of the bus stations (GAP) as low as possible
- T_{QUI} (switchover time for repeaters/modulators) as short as possible or “0” if there are no repeaters/modulators
- T_{RDY} (time after which the master is ready to receive) as short as possible (less than MinT_{SDR})
- High baud rates based on cable lengths (see table 7-2)
- EMC-compatible setup with appropriately dimensioned terminating resistances to avoid telegram repetitions

Programming with Examples

8

8.1 FC 46 - Interrogation by the User

Interrogation of the FC 46 is performed via the command/status word BEST (see section 4.3)

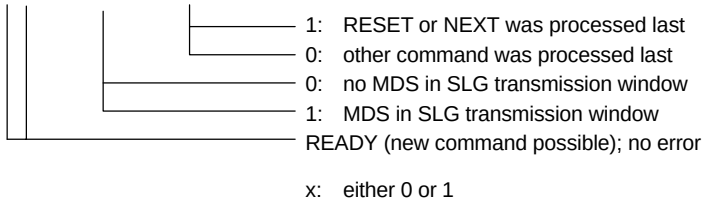


8.2 Function Monitoring

Function monitoring can be carried out by using the “BEST” and “ANZ0”, “ANZ1” parameters. The parameters can be displayed and changed at the programming unit via the “status/control” setting.

8.2.1 Program Process Running, No Command Being Processed

”BEST”: 10xxxxx0 00100000 or
10xxxxx1 00000000



“ANZ 0” → “00”
“ANZ 1” → “00”

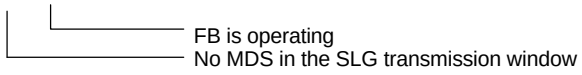
If the status of both parameters differs from that shown above, a RESET command should be issued. If the content of both parameters still differs from that shown above, the error should be investigated further.
Note: No RESET command is necessary after most errors (error bit = 1)

Note

Nach den meisten Fehlern (Fehler-Bit = 1) ist kein RESET-Befehl notwendig.

8.2.2 Command Has Been Issued and Transmitted to the ASM (EAKO 0/4)

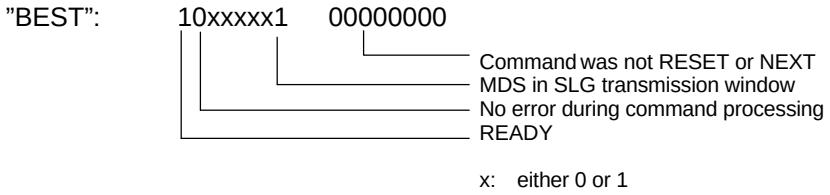
”BEST”: 10000000 00000100 (Start with “status/control” on the PG), and then:
00000000 10000000



“ANZ 0” → “00”
“ANZ 1” → “00”

The status of the “BEST” parameter remains the same until an MDS enters the SLG field and the command is correctly processed on the MDS.

8.2.3 Command Executed



“ANZ 0” → “00”
“ANZ 1” → “00”

After execution of the command, both parameters return to the basic setting (8.2.1). If an error is detected and one of the parameters differs from the diagram shown, the error should be investigated further.

8.3 Cyclic Call-Up of FC 46

The following sample program starts an already issued command again and again. Processing stops after an error.

Assumption: **DB100 is BEDB**

```
                CALL FC 46      Absolute call
ADR:            3
BEDB:           100
EAKO:           4
ECC:            1
RWD:            'D'
SLG:            W#16# 4711
ABTA:           B#16#00
OPT1:           W#16#0000
OPT2:           W#16#0000
                .
                .
                C DB 100        Open BEDB
                L DBW 0         Load BEST
                T FW 250
                AN F250.7       Ready?
                BEC             Wait until Ready is set
                .
                A F 10.0        Is a RESET to be issued?
                JC=REST         Branch after issuing RESET
                .
                A F 250.6       Error?
                JC=ERR          Branch after error evaluation
                .
                .
                S F 251.2       Set start bit to start a command (command parameters must
                L FW 250        already have been entered) / load flag word 250 and
                T DBW 0         transfer to  BEST
                BEU
                .
REST:           S F 251.0       Set RESET bit to start RESET command
                L FW 250        Load flag word
                T DBW 0         Transfer to  BEST
                R M 10.0        Reset the RESET auxiliary flag !
                BEU
                .
ERR:            L DBW 6         ANZ0
                T FW 100        Save in flag word 100
                L DBW 8         ANZ1
                T FW 102        Save in flag word 102
                                NOTE:   If an error bit is set, the error number must be
                                        interrogated immediately. This has highest
                .                       priority.
                .
                BE
```

8.4 Formulate and Start a Command

The following sample program issues a complete write command and then starts it.

Assumption: DB 100 is BEDB; DB 10 is DATDB

```
          CALL FC 46
ADR:      3
BEDB:     100
EAKO:     4
ECC:      1
RWD:      'D'
SLG:      W#16# 4711
ABTA:     B#16#00
OPT1:     W#16#0000
OPT2:     W#16#0000
          :
          C DB 100      Call up BEDB
          L DBW 0       Load BEST
          T FW 250      Store temporarily
          AN F 250.7    Ready?
          BEC           Wait until Ready
          A F 250.6     Error?
          JC=ERR        After error evaluation
          .
          .
          L 10
          T DBW 2       DATDB is DB10
          L 0
          T DBW 4       DATDW is DBW 0. The data is written
                        to file "FILENAME" starting at this address.
          L 'WI'        Command code and index / WRITE command
          T DBW 12      ENTER in command parameter
          L 'FI'        'FI'
          T DBW 14
          L 'LE'        'LE'
          T DBW 16                  } FILE NAME
          L 'NA'        'NA'
          T DBW 18
          L 'ME'        'ME'
          T DBW 20      Enter FILENAME in DBW 14 to DBW 20
          L +373        Length in bytes of user data to be written
          T DBW 22      Enter in DBW 22 (DLNG)
          S F 251.2     Set start bit  to start the command (located  starting in DBW 12)
          L FW 250      Flag word
          T DBW 0       Transmit to BEST
          BEU
          .
          .
ERR:      L DBW 6       Store ANZ0 and ANZ1 temporarily
          T FW 100      If error bit is set, the error number must be interrogated
          L DBW 8       immediately. This has the highest priority.
          T FW 102
          BEU
          :
          BE
```

8.5 Programming New Start and Restart

Assumption: DB 100 is BEDB; DB 10 is DATDB

OB 100

```
C DB 100        Open BEDB
L W#16#0008     Set new start bit
T DBW 0         in BEST
L 10                                  }
T DBW2          DATDB is DB 10        }  can be omitted if BEDB
L 0                                   }  was specified directly. See
T DBW4          DATDW is DBW 0        }  section 8.6.
BE
```

OB 101

```
C DB 100        Open BEDB
L W#16#0008     Set new start bit
T DBW 0         in BEST
L 10                                  }
T DBW2          DATDB is DB 10        }  can be omitted if BEDB
L 0                                   }  was specified directly. See
T DBW4          DATDW is DBW 0        }  section 8.6.
BE
```

8.6 Setting Up BEDB

Example 1: DB 100 is BEDB and DATDB.

DB 100

0	W#16#0008	New start is set.
2	100	DATDB = DB 100
4	700	DATDW = DBW 700
6	W#16#0000	**Remember: DBW 700 is the smallest value for DATDW. Larger values are possible.**
	.	
	.	
	.	**DBW 2/4 can also be prespecified - appropriately in OB 100 and OB 101.**
.	.	
.	.	
.	.	
.	.	
694	W#16#0000	
696	W#16#0000	
698	W#16#0000	The DB is occupied by FC 46 up to DBW 698.
700	'MOBY I'	These are data to be written.
	'SIEMENS'	
	'DATEN'	

Example 2: DB 100 is BEDB, and DB 10 is DATDB.

DB 100

0	W#16#0008	New start command is set.
2	10	DATDB = DB 10
4	12	DATDW = DBW 12
6	W#16#0000	**Remember: DBW 2/4 can also be prespecified**
8	W#16#0000	**appropriately in OB 100 and OB 101.**
:		
698	W#16#0000	Minimum length of the BEDB

DB 10

0	W#16#0000	DBW 0 to DBW 10 are reserved by the user.
2	W#16#0000	
4	W#16#0000	
6	W#16#0000	
8	W#16#0000	
10	W#16#0000	
12	'MOBY I'	These are the data which
14	'SIEMENS'	are to be written or were read.
16	'DATEN'	

9 Causes of Errors with ASM and FC 46

9.1 PROFIBUS Diagnosis

"ON" LED is not on or is flashing.

If the "ON" LED is not on, this means that either no supply voltage or too low voltage is available to the ASM. Possible causes include a bad fuse or missing/too low network voltage. An LED which is not on or is flashing may indicate a defective module.

Diagnosis with LEDs

The following table lists possible error indications with their meanings and provides remedies.

Table 9-1 LED indication

"BF" LED	"SF" LED	Cause of Error	Error Correction
ON	*	• ASM is starting up.	–
		• The connection to the DP master has failed. • ASM does not detect a baud rate.	• Check the PROFIBUS-DP connection. • Check the DP master.
		• Bus interruption • DP master is not in operation.	• Check all cables in your PROFIBUS-DP network. • Check to determine whether the plug connectors for PROFIBUS-DP are securely connected to the ASM.
Off	On	• The PROFIBUS address set on the ASM is illegal.	• Change the PROFIBUS address set on the ASM.
Flashing	On	• The configuration data sent by the DP master to the ASM do not correspond to the setup of the ASM.	• Check the configuration of the ASM (i.e., input/output and PROFIBUS address). • Did you use the right GSD file? Check switch 8 on ASM (see chapter 2.1).
Flashing	Off	• ASM has detected the baud rate but is not addressed by the DP master. • ASM was not (correctly) configured.	• Check the PROFIBUS address set on the ASM or in the configuration software. • Check the configuration of the ASM (i.e., station type). • Check the bus parameters as described in chapter 7.2. The standard values of PROFIBUS-DP must be changed.
ON	Flashing	• ASM has a hardware defect.	• Replace the ASM.

* Status not relevant

System diagnosis

ASM 451 and ASM 452 support the standard system diagnosis of PROFIBUS with a length of 6 bytes.

9.2 Evaluating the ERR LED

The ERR LED flashes to indicate filehandler errors which may indicate defective hardware of ASM, SLG or MDS.

Table 9-2 Evaluation of the ERR LED

Flashing ERR LED	Filehandler Error Message	
1x	**D0 01**	Only RESET command permitted
2x	**C0 06**	Presence error
3x	**B0 01**	Error in connection to the SLG
4x	**C0 02**	Error in RAM of the MDS
5x	**C0 07**	Parameterization error with TRACE or FORMAT/command cannot be interpreted.
6x	**C0 08**	Too many sync attempts
7x	**C0 09**	Too many sending errors
8x	**C0 10**	CRC sending error
9x	**C0 11**	FORMAT, CRC error during receiving
10x	**C0 12**	FORMAT, MDS cannot be initialized.
11x	**C0 13**	FORMAT, timeout
12x	**C0 14**	FORMAT, not initialized
13x	**C0 15**	CMD address error
14x	**C0 16**	ECC error
15x	**C0 17**	General driver error
18x	– – – –	Internal ASM communication errror →Hardware defective →Perform new start.
20x	– – – –	Internal ASM overflow; stack overflow; PLC memory overflow; diagnosis does not work →Send RESET or restart →Turn interface module off and o →Check bus parameterization
21x	– – – –	ASM parameterization incorrect →Check parameterization in HW Config.
30x	– – – –	Corrupt SLG telegram

9.3 Evaluating the Error Indicators ANZ0 and ANZ1

A0 06:

The command code for the issued command is not valid (not defined). The correct CC must be entered (see section 4.6).

A0 11:

The telegram check parameters (DBN or CC) are not coming in the correct order. Two or more telegrams have been written to the same ASM. The settings of the FC call parameter “ADR” must be checked.
Do not execute command start with “control variable” function.

A0 16:

The file handler is currently processing another command. Execution of a RESET command is mandatory.

A0 17:

The data block of the SLG is too long and cannot be transferred with PROFIBUS.

- The “block length” parameter for the RESET command is too long (FC error or user error).
- Program sequence error on the SLG
- Perform new start of the ASM and start command again.

A0 18:

Communication error. MOBY driver is active while a new command is being sent.

- Check command sequences in the application.
- Perform new start of the ASM.

B0 01:

Fault in connection to the SLG.

- Cable between ASM and SLG is incorrectly wired or cable is damaged.
- 24 V supply is not connected or is switched off.
- Automatic fuse on the ASM has blown.
- Hardware defect

This error does not occur at the start of the system commands
(RESET, NEXT, ASM STATUS).

B0 02:

EAKO 1:

- A command has been issued, but there is no MDS in the transmission window of the SLG.
- The dialog battery of the MDS 507 is dead.
  ("Batt 2-Bit" is not set. Measure battery voltage.)

EAKO 0:

- The old/current MDS has moved out of the transmission window and the next/new MDS has entered the transmission window. A command has been issued (not NEXT). This command refers to the new MDS, but the old/current MDS has not been terminated with NEXT.
- A new MDS entered the transmission window of the SLG and left it again without any command being processed (MDS has "slipped through").

B0 08:

Antenna not on or SET-ANT = ON with antenna already turned on

- User error. Adhere to command sequence.

B0 09:

Buffer overflow on MOBY driver of the ASM/SLG. Internal system error.

- Perform new start of the ASM.

B0 10:

Driver error. Communication between filehandler and MDS driver has malfunctioned (AB byte).

- Perform new start of the ASM.

B0 11:

Communication error between filehandler and MDS driver. The MDS driver reported a RESET termination although the filehandler did not process a RESET.

- Perform new start of the ASM.

B0 12:

Unmotivated startup message of MDS driver

- Perform new start of the ASM.

C0 02:

Memory error message from MDS. The MDS has not yet been written or has lost its memory contents due to battery failure (not with EEPROM MDS). Therefore:

- Change the MDS (if the battery monitoring bit is set)
- Initialize the MDS with the STG 4F
- Format the MDS using the FORMAT command

C0 06:

During certain important operations (e.g. writing system area of MDS, formatting MDS) the MDS must not leave the transmission window of the SLG, as otherwise the command will terminate with this error. Therefore:

- Issue the command again
- MDS is situated on the edge of the SLG transmission window

C0 07:

- The FORMAT or TRACE commands have been issued with incorrect parameters. The physical address requested does not exist in the MDS (MDS memory is smaller than specified in the command)
- With READ/WRITE/UPDATE: the pointer in the FAT is incorrect; it is pointing to a block that does not exist in the MDS.

C0 08:

Field interference at the SLG. The SLG is being affected by interference from its surroundings. Some examples are listed below.

- External interference field; the interference field can be detected by the “inductive field indicator” of the STG.
- The distance between two SLGs is too small and does not conform to the design guidelines.
- The connection cable to the SLG is subject to interference, is too long or does not conform to specification.

Or the dialog battery of the MDS 507 is dead.

- Check “Batt 2-Bit”.
- Measure battery voltage.

C0 09:

Too many send errors have occurred. The MDS could not receive the command or the data from the ASM correctly, despite several attempts

- The MDS is standing exactly on the edge of the transmission window.
- Data transmission to the MDS is being affected by external interference.

C0 10:

- CRC send error. The monitoring system of the file handler has detected a data transmission error. Cause of error same as **C0 08**.
- The MDS reports CRC errors very frequently. (The MDS may be positioned in the boundary area, or the MDS or SLG is defective.)

C0 11:

See **C0 08.**

C0 12:

The MDS cannot carry out the FORMAT command. The MDS is defective.

C0 13:

The MDS must be within the SLG transmission window when formatting, otherwise a timeout error occurs, i.e.:

- The MDS is situated right on the edge of the transmission window.
- The MDS is using too much current (defective).
- An EEPROM type MDS has been given incorrect parameters during FORMAT.

Or the dialog battery of the MDS 507 is dead.

- Check “Batt 2-Bit”.
- Measure battery voltage.

C0 14:

The MDS memory cannot be written

- The MDS has a smaller memory than specified in the FORMAT command, i.e. enter the correct parameters for this type of MDS.
- The MDS memory is defective.
- An EEPROM MDS has been written too often and has reached the end of its life.

C0 15:

Address error. The address area of the MDS has been exceeded

- The MDS is not of the correct type.

C0 16:

An ECC error has occurred. Data cannot be read from the MDS.

- The MDS has lost its data (MDS defective.)
- The MDS was not formatted by the ECC driver. Reformat the MDS.
- An EEPROM MDS has reached the end of its life. The data has been lost. Replace the MDS
- The MDS was moved out of the field while being write-accessed. The MDS is not positioned correctly. (**NOTE:** The system area of the MDS is automatically write-accessed at every SLG station.)

C0 17:

The file handler is not working correctly.

- Check command format and command sequence.
- The ASM hardware (firmware) is defective.

C0 18:

Operating system error (AMOS mailbox)

- Perform new start of the ASM.

C0 19:

More than one MDS is located in the field. The number of MDSs in the field is greater than the number of MDSs parameterized in “multitag.”

- With the FC 46 only 1 MDS can be processed in the field at a time.
- Remove all other MDSs from the field.
- Configuration of dili (distance_limiting) is set incorrectly.
- Search the surroundings of the SLG for an MDS that is located in the field by coincidence.

C0 20:

Communication error between filehandler and MDS driver. The MOBY driver doesn’t recognize the command from the filehandler.

- Perform new start of the ASM.

C0 21:

Operating system error. Watchdog error on ASM/SLG.

- Perform new start of the ASM.

D0 01:

The file handler will only accept RESET commands.

- The file handler has not yet been initialized by a RESET command.
- This situation can only be remedied by issuing a RESET command.

D0 05:

The FORMAT, CREATE, WRITE, ATTRIB, UPDATE, COVER, QUEUE READ or QUEUE WRITE commands have been issued with incorrect parameters

- FORMAT with invalid MDS name or MDS type
- CREATE with invalid file name
- WRITE/UPDATE with length 0 (DLNG = 0)
- Attribute not permissible
- QUEUE WRITE or QUEUE READ with invalid option
- COVER with invalid user (only 0 or 1 permissible)

D0 07:

- The system data transmitted with the LOAD command is incorrect.
- DLNG parameters incorrect with LOAD (see section 4.7).
- Data block given is incorrect or has incorrect parameters (see section 4.7).
- The MOVE command cannot be executed. The checksum does not correspond to DIR + FAT. The MDS has apparently left the transmission window while system operations were being performed (e.g. write DIR + FAT) .

D0 09:

A RESET command has been issued by the FC 46 with invalid parameters. Cause of fault lies in user program.

- Check the FC 46 parameters.

D0 14:

WRITE command:

There is no longer enough memory available in the MDS. Not all the data has been written to the MDS.

CREATE command:

No data blocks can be reserved when creating a file. No more blocks are free.

D0 15:

The MDS could not be identified by the file handler. Reformat the MDS.

D0 18:

The logical address requested lies outside the file. There is an error in the FAT. Reformat the MDS.

D0 22:

The MDS is protected by the COVER command. A write command (e.g. UPDATE, CREATE) must not modify the memory structure and is therefore rejected.

D0 23:

COVER command:

The MDS name specified in the command does not correspond to the actual MDS name.

E0 01:

- The type of MDS present at the SLG does not correspond to the ECC operating mode selected. The MDS must be reformatted to correspond to the desired ECC operating mode.
- The MDS is not a file handler MDS. Format the MDS.

E0 02:

No more directory entries available. The file specified in the CREATE command cannot be created (see section 1.2.2).

E0 03:

The file specified with the CREATE command already exists in the directory (no duplicate names).

E0 05:

- A FAT block sequence error has been detected during a READ or WRITE command. The file allocation table (FAT) is incorrect. The MDS must be reformatted.
- Invalid address in TRACE command

F0 01:

- The file specified in a command (e.g. WRITE) does not exist in the directory. The filemust be created with CREATE.
- Check file name (perhaps not in ASCII format)
- The QUEUE READ command is to read one or more files which do not exist on the MDS. Valid data are not transferred to the user.

F0 05:

Write attempt (WRITE, UPDATE or DELETE) to a file that may not be changed (protected by an appropriate attribute).

F0 07:

QUEUE READ: The file length specified is shorter than the file length.

F0 08

QUEUE READ: The “skip” calculated by the file handler is greater than 0FFF hex (4095 in decimal notation).

H1 01:

The FC 46 input parameters or DATDB / DATDW have been incorrectly specified in an absolute call (see section 3.2).
Change FC parameters in calling program and issue a RESET command.

H1 02:

The length of the loaded BEDB is less than 350 data words. The FC 46 does therefore not have enough space for the internal FC parameters. A new BEDB of the correct length must be loaded. Then issue a RESET command.

H0 03:

The command index is invalid. Change the command index (section 4.6).

H0 04:

This command code and therefore this command is not known to the FC 46. Check the command code against the table in section 4.6.

H0 05:

The access rights of the corresponding SLG do not permit this command. If, for instance, the access rights “R” (Read only) have been allocated to the SLG, no WRITE command can be issued to this SLG. Therefore either the FC parameter “RWD” must be changed (then issue a RESET command to effect the change) or a valid command should be issued.

H0 06:

The WRITE/UPDATE/LOAD/QUEUE-WRITE or QUEUE-READ command parameter specified in DBW 22 (DLNG) of BEDB is invalid. Only a data length of 7FF0 hex (32752 in decimal notation) bytes or a maximum of 210 (decimal) bytes for QUEUE READ is permitted. Change DLNG accordingly.

H1 07:

The data block specified in DBW 2 (BEDB) does not exist. The appropriate data block must be loaded. A RESET command must then be issued so the absolute addresses can be calculated.

H1 08:

This is a software error that cannot occur in normal operation.

H1 10:

The ASM has carried out a hardware reset. The reason for this can be, for example, a voltage dip on the module rack or a plug contact fault. The user must issue a RESET command to reparameterize the SLG.

H1 11:

The acknowledgement has absolutely nothing to do with the current operation. This is a pure software or synchronization error that cannot occur in normal operation.

H1 12:

The command code and the respective acknowledgement do not correspond. This is a software or synchronization error that cannot occur in normal operation.

H1 13:

The first command block has not been correctly acknowledged, i.e. the telegram check parameters do not correspond. This is a pure software or synchronization error that cannot occur in normal operation.

H1 14:

An error was detected when reading the interface check register. This means there is no longer any synchronization between the writing of the command blocks and reading of the respective acknowledgements. This is usually the result of a plug contact fault. A RESET command must be issued to reestablish synchronous operation.

H1 15:

The data start address pointer calculated from the parameters DATDB and DATDW (DBW 2 and 4 in BEDB) lies outside the specified data block. Either DATDW must be made smaller or the specified data block (DATDB) extended. Then issue a RESET command.

H1 16:

The telegram check parameters of the command and acknowledgement blocks do not cor-respond. This is a pure software or synchronization error that cannot occur in normal operation

H1 17:

See error **H1 16**

H1 18:

The data start address pointer (calculated from DATDB and DATDW) has been changed during processing of the current command (Ready bit not yet set). The absolute addresses are thus no longer correct. A RESET command must be issued to recalculate the absolute addresses.

H1 19:

The absolute address used during the write or read operation (from/to the data block) lies outside the data block. Either the data block must be extended or the data start address pointer (DATDB and DATDW) must be corrected accordingly (make more space in the data block). A RESET command must then be issued.

H1 20:

The PLC memory has been compressed or the absolute length of the data block (BEDB and/or DATDB) has been changed during the current operation (cyclic call-up of FC 46). The absolute addresses are thus no longer correct. A RESET command must be issued.

H1 21:

This indication tells the user that only a RESET command is permitted as the next command. All other commands will be rejected.

H0 27

QUEUE-READ: QUDW pointer is outside the DB or DX specified in QUDB.

H0 28

QUEUE-READ: QUDB is missing in the automation system or is too small to read in the user data.

H1 30:

The FC 46 has found a system error. The acknowledgement from the file handler or PROFIBUS-DP master is not permitted.

- DP master is overloaded.
- Firmware release is outdated.

The exact error code is indicated in ANZ2 (i.e., DBW 10). The error codes are listed in the description of SFC 58/59 in the S7 manual.

Kx xx:

QUEUE WRITE parameter incorrect (DATDB / DATDW or DLNG)

Option 0000 Hex:

The file entry with number xxx or xxx + 1 specified in DATDB is incorrect. The method of counting the file entries in DATDB begins with 1.

Option 0001 Hex:

The file entry with number xxx or xxx + 1 specified in DATDB contains a file name that already exists on the MDS. The method of counting the file entries in DATDB begins with 1.

Note: The data entries are counted in decimal format.

9.4 Evaluation of the ANZ2 Indication

Table 9-3 Evaluation of the ANZ2 indication

Error Code (W#16#...)	Description
800A	ASM is not ready (temporary message). → This message is received by users who are not using the FC 46 and who pole the ASM non-cyclically in very rapid succession.
8x7F	Internal error in parameter x. Cannot be corrected by the user.
8x22 8x23	Area length error while reading a parameter Area length error while writing a parameter This error code indicates that parameter x is either completely or partially outside the operand range or the length of a bit field in an ANY parameter is not divisible by 8.
8x24 8x25	Area length error while reading a parameter Area length error while writing a parameter This error code indicates that parameter x is in an area that is not permitted for the system function.
8x26	The parameter contains a number of a time cell which is too large.
8x27	The parameter contains a number of a counter cell which is too large.
8x28 8x29	Offset error while reading a parameter Offset error while writing a parameter The reference to parameter x is an operand whose bit address is not 0.
8x30 8x31	The parameter is located in the write-protected global DB. The parameter is located in the write-protected instance DB.
8x32 8x34 8x35	The parameter contains a DB number which is too large. The parameter contains an FC number which is too large. The parameter contains an FB number which is too large.
8x3A 8x3C 8x3E	The parameter contains the number of a DB which is not loaded. The parameter contains the number of an FC which is not loaded. The parameter contains the number of an FB which is not loaded.
8x42 8x43	An access error occurred while the system was trying to read a parameter from the I/O area of the inputs. An access error occurred while the system was trying to write a parameter to the I/O area of the outputs.
8x44 8x45	Error during nth (n > 1) read access after an error occurred Error during nth (n > 1) write access after an error occurred
8090	Specified logical base address is invalid. No allocation in SDB1/SDB2x or the address is not a base address.
8092	A type other than BYTE was specified in an ANY reference.
8093	The area ID given when the logical address was configured is not permitted for these SFCs. The following are permitted. • 0 = S7-400 • 1 = S7-300 • 2, 7 = DP modules

Table 9-3 Evaluation of the ANZ2 indication

Error Code (W#16#...)	Description
80A0	Negative acknowledgment while reading from module. FC fetches acknowledgment although there is no acknowledgment to be fetched. A user who is not using the FC 46 would like to fetch DR 101 (or DR 102 to DR 104) but an acknowledgment is not yet available. → Perform an init_run to synchronize ASM and application again.
80A1	Negative acknowledgment while writing to the module. FC sends command although ASM cannot receive a command.
80A2	DP protocol error for layer 2. Hardware may be defective.
80A3	DP protocol error for direct data link mapper or user interface/user. Hardware may be defective.
80B0	• SFC not possible for module type • Data record is unknown to module. • Data record number ≥ 241 is not permitted. • Data records 0 and 1 are not permitted for SFC 58 “WR_REC.”
80B1	The length in the RECORD parameter is wrong.
80B2	The configured slot is not filled.
80B3	Actual module type is not “should be” module type in SDB1.
80C0	• RDREC: The module has the data record but no read data. • WRREC: ASM is not ready to accept new data. → Wait for cyclic counter to count.
80C1	The data of the previous write job on the module for the same data have not yet been processed by the module.
80C2	The module is processing the maximum possible number of jobs for a CPU.
80C3	Required resources (memory, etc.) are in use at the moment. The FC 46 does not report this error. If this error occurs, the FC 46 waits until resources become available to the system again.
80C4	Communication error • Parity error • SW-Ready not set • Error in block length • Checksum error on CPU side • Checksum error on module side
80C5	Distributed I/O not available

9.5 Further Fault Causes

Fault:

The program does not work following cold start or restart

Cause:

- The organization blocks for restart and cold start have not been preset in accordance with the FB description
- No PROFIBUS connection. The bus is not in RUN status.

Fault:

The automation system goes into STOP after loading the MOBY block.

Cause:

BEDB and/or data block (DATDB) not present in the automation system or is too short.

- Check FC parameter settings, particularly the ADR parameter.
- Check PROFIBUS-DP master paramterization.

Fault:

The automation system goes into STOP after start or execution of a command.

Cause:

- DATDB not present or deleted or too small
- Read/write from/to ASM not possible
- New start not carried out after loading BEDB and/or data block
- New start not carried out after modifying FC parameters

Programming the ASM on PROFIBUS-DP

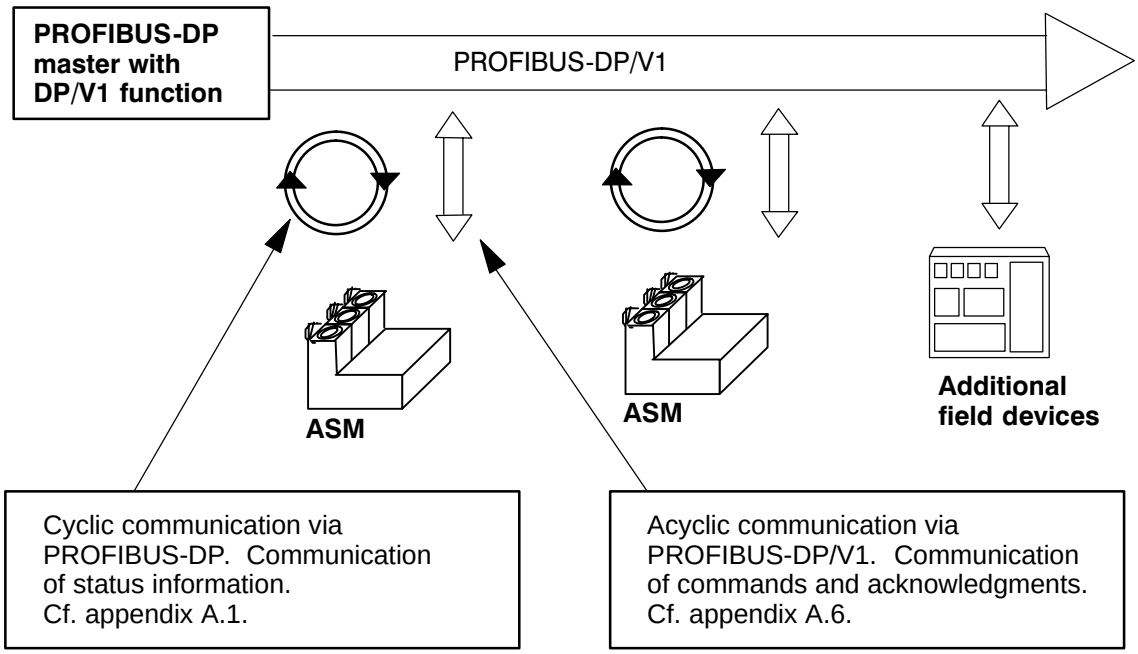
A

Who should read this appendix?

SIMATIC users can skip this section. This appendix has been written for programmers of PCs and controllers of other manufacturers. It shows programmers how to create their own function block or driver for the ASM.

Communication between ASM and PROFIBUS master

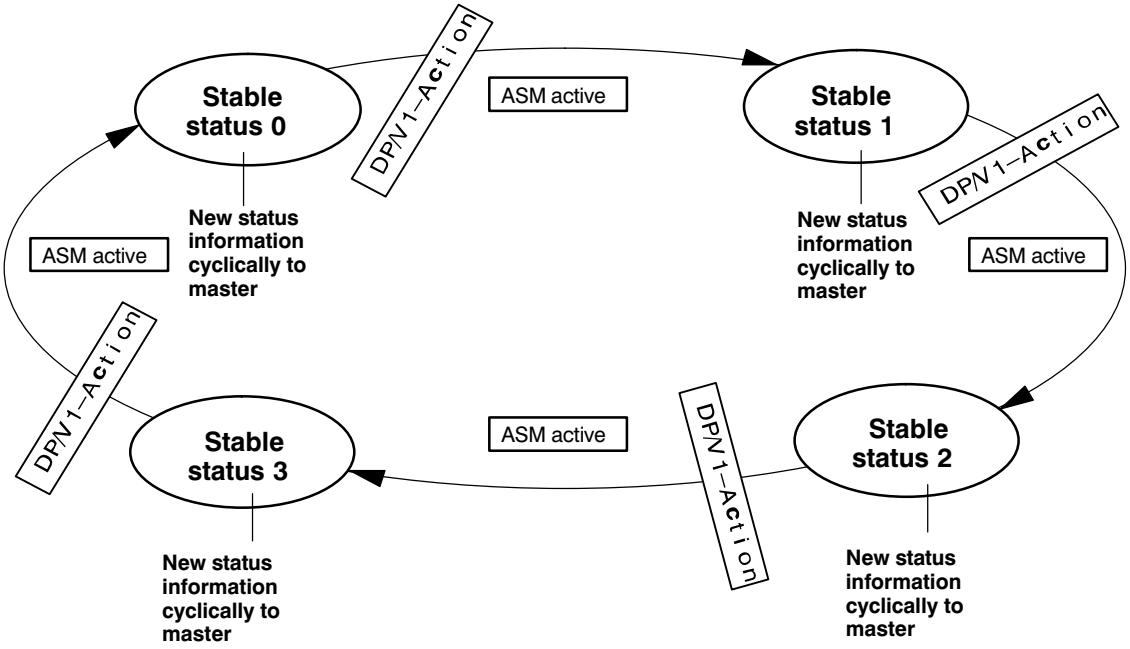
Both the cyclic (DP) and acyclic data (DP/V1) must be able to be transferred via PROFIBUS-DP.



New commands cannot be sent from master to slave (i.e., ASM) unless the ASM 451 is ready. ASM readiness is indicated by status information. Similarly, **new acknowledgments** can only be fetched from the ASM when a new acknowledgment is actually present (i.e., has not yet been read in). This information is also indicated by appropriate status information.

Two indications are defined in the status information. The PROFIBUS-DP master uses these two indications to determine whether DP/V1 telegram execution can be performed from or to the ASM.

Flow chart of the control of acyclic communication via command and acknowledgment counters



The above diagram shows that the change from one stable status to the next is triggered by a DP/V1 action. A new DP/V1 action is not permitted until the next stable status has been achieved. A DP/V1 action is either a command to the ASM or an acknowledgment from the ASM.

For this reason, it is important to inform the master as to whether a new DP/V1 action may be performed. Each status is coded in 2 bits and counted as shown in the diagram above. These are called **status bits or status counters**.

The status bits are continuously transferred to the master via PROFIBUS-DP. The user must evaluate them in his/her program. When the status bit changes, a new stable status (i.e., new status = old status + 1) has been achieved, and the next DP/V1 action is permitted.

Two states must be coded.

1. Command status (command counter) to tell the user whether a new/next command may be transferred to the ASM.
2. Acknowledgment status (acknowledgment counter) to show the user whether a new acknowledgment is available from the ASM.

The user must assign a higher priority to the evaluation of the acknowledgment status (i.e., if the user wants to send a telegram to the ASM but a telegram from the ASM is ready to be fetched at the same time, fetching the telegram from the ASM has the higher priority).

Both the command and the acknowledgment status have two-bit coding. Both states are stored in one byte. See appendix A.1.

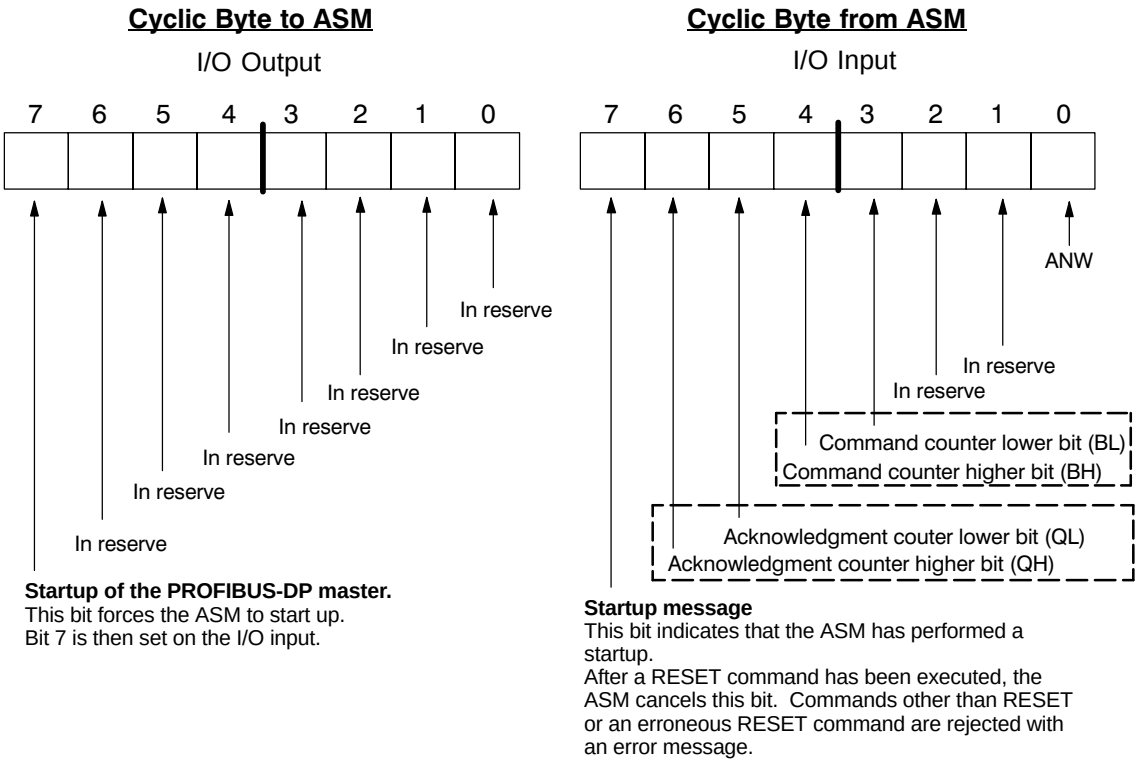
A.1 Data Communication between Master and ASM

Cyclic control byte

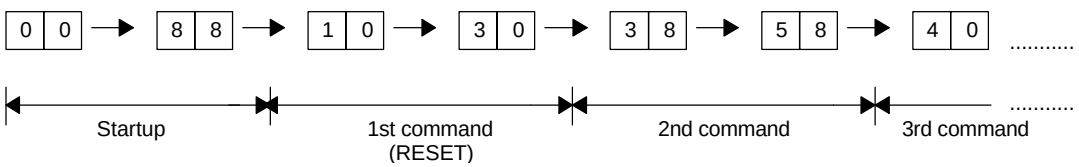
Telegram communication between master (FB) and slave (ASM) is synchronized with the cyclic control bytes. The actual acyclic command and acknowledgment telegrams via DP/V1 may not be started until contents of the cyclic byte are changed by the ASM in the command or acknowledgment counter.

Note

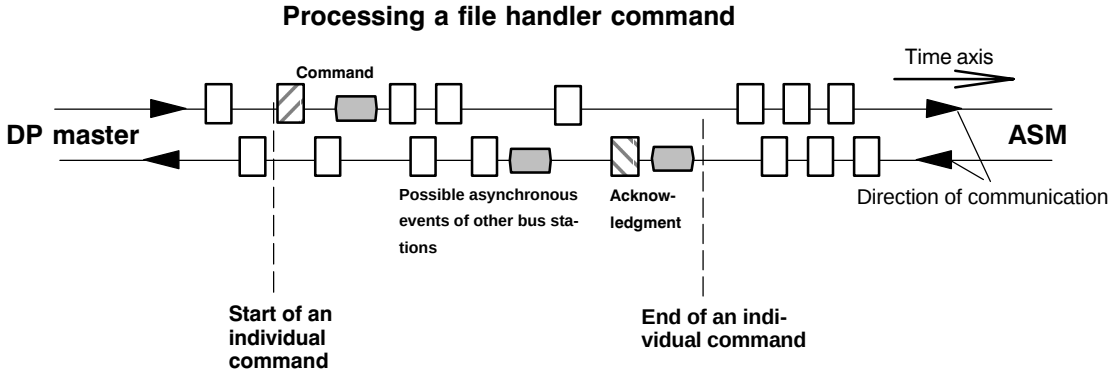
The ASM 452 requires 2 bytes of input/output. When the FC 46 is used, only 1 byte is needed. That is why this chapter states that only the one byte is written.



After startup, the “cyclic byte of ASM” succession looks like this.
See also chapter 7.1:



The diagram below shows the sequence of command and acknowledgment communication between user and ASM.



Acyclic DP/V1 telegram with DP/V1 response
Start is only permitted after a change of the command and/or acknowledgment counter

Change of the command counter. New status = old status + 1 (cyclic byte)

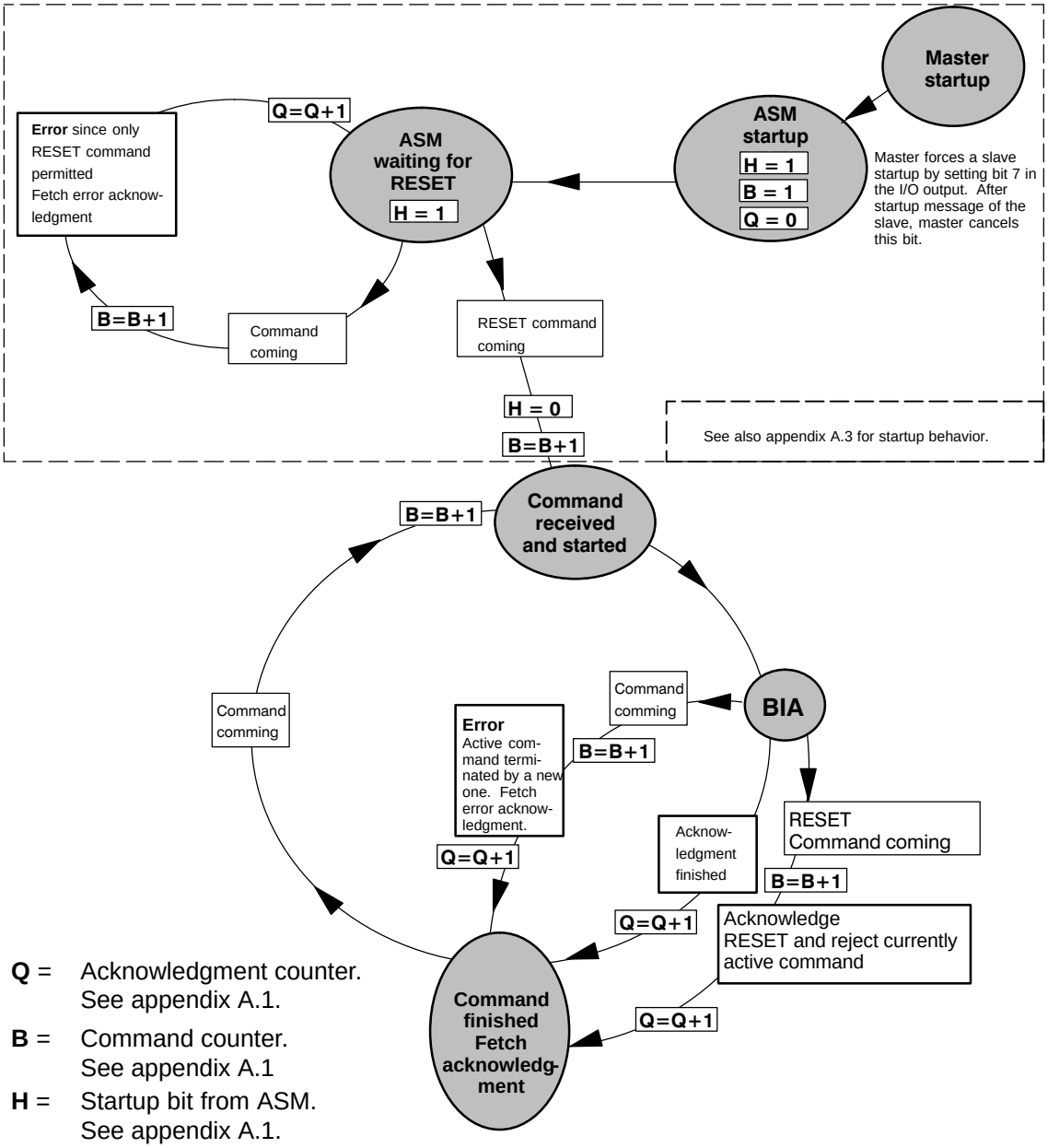
Change of the acknowledgment counter. New status = old status + 1 (cyclic byte)

No change in the cyclic data

Parameterization of non-cyclic telegrams

- DP/V1 bit must be set in the configuration/parameterization.
- Slot number: Disregard
- Data record number (identifier): 2 to 199. Use of data record number 111 is recommended.
- Length of the data records: $\leq$ 240 bytes

A.2 Status Diagram for Programming a Function Block



Q = Acknowledgment counter.
See appendix A.1.

B = Command counter.
See appendix A.1

H = Startup bit from ASM.
See appendix A.1.

BIA = Command being processed

Command = Command or RESET command

RESET command = Explicitly the RESET command

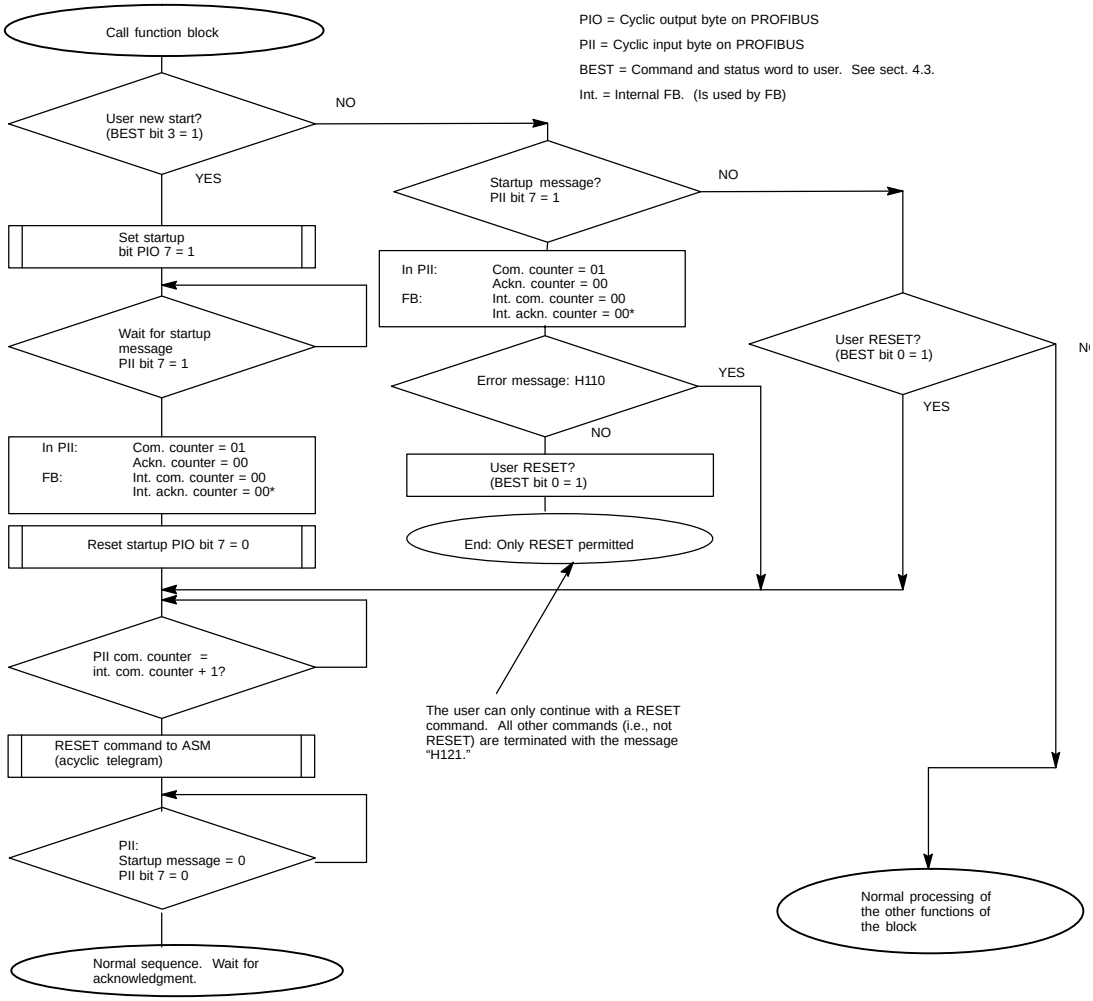
A.3 Startup Behavior

After power returns, the ASM performs extensive self tests and then reports that it is ready for operation with the “startup message.” The status counters are put into a defined state so that the user can use synchronization.

The master can inform the slave of its startup by setting the startup bit in the cyclic byte. The slave then performs a startup also.

The following flow chart shows how startup and RESET are programmed in the user program (e.g., function block).

The user must program a new start during startup when the controller (PC) is turned on.



*) During startup, the ASM reports the following in PII.

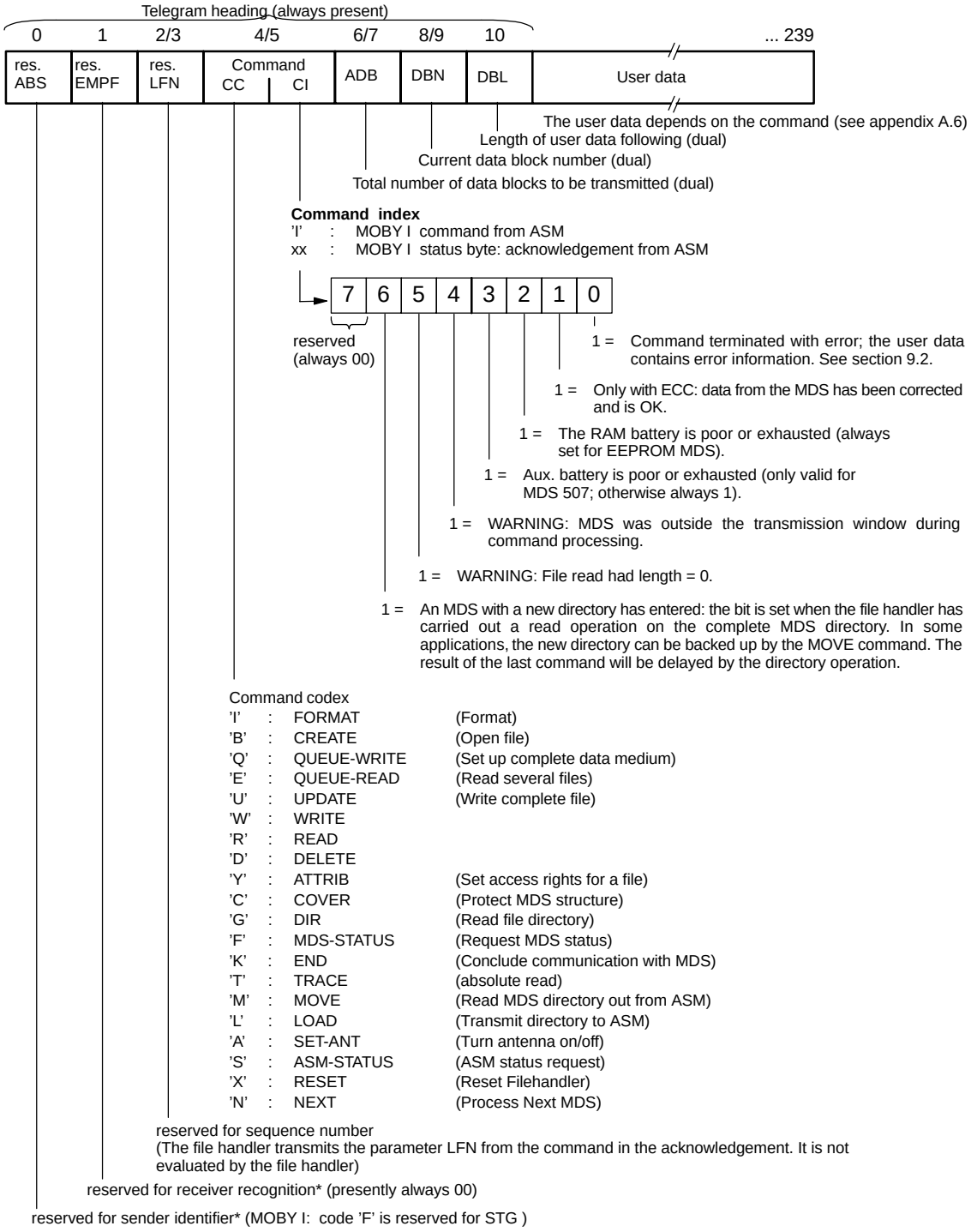
- Command counter with 01
- Acknowledgement counter with 00

This permits the FB to synchronize its internal counters with the counters in PII.

A.4 Telegram Format for the Filehandler

The user communicates with the file handler via telegrams. Telegrams from the user are checked, interpreted and processed by the file handler. The user then receives an acknowledgement telegram containing user data and status information.

The following schematic shows the general telegram format:



* The file handler interchanges sender and receiver identifiers.

A.5 Blocking of Long Data Sequences

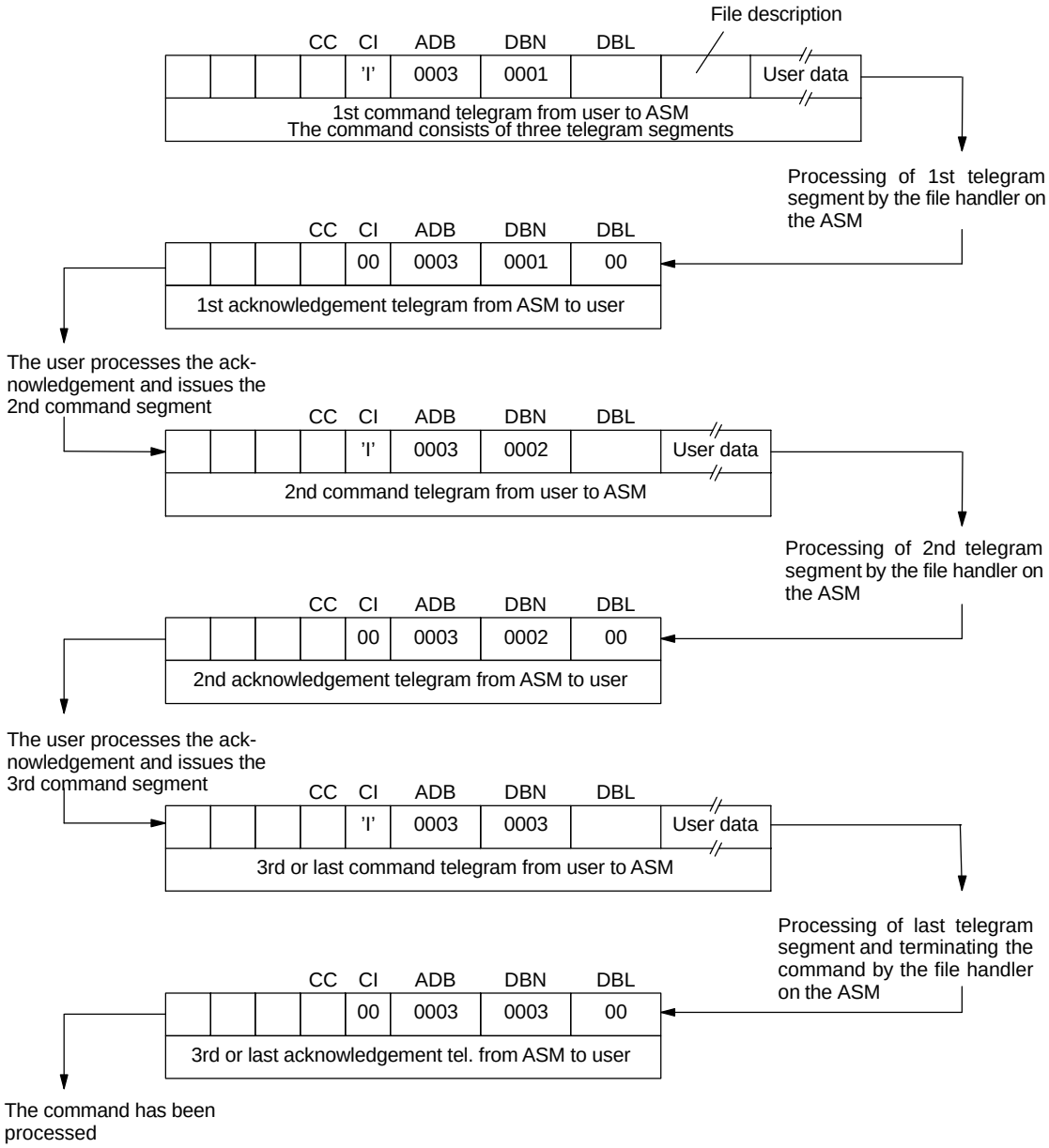
Since it is not possible to work with telegrams of indefinite length, the telegram is formed into blocks, i.e. telegrams are transmitted in segments.
Blocking can be used with READ, WRITE, UPDATE, QUEUE-WRITE, QUEUE-READ, LOAD, MOVE, DIR and TRACE commands. Blocking is controlled by the parameters ADB (no. of data blocks) and DBN (current data block no.) in the telegram heading.
The file description immediately follows the DBL parameter in the first command telegram to the file handler (DBN = 1). It contains the file name, start address and length of the file. The file description is omitted from subsequent telegram segments (DBN > 1).

The maximum block length (maximum length of a telegram segment) can be set using the RESET command. The default block length is 128. The maximum block length which can be set is 240.

The blocking sequence can be aborted using the RESET command. The blocking sequence will be ended. No further acknowledgement of the aborted command will be issued.

All other commands within a blocking sequence will be rejected by the ASM with an error message. The present status of the blocking sequence remains unchanged.

Schematic sequence (example WRITE):



A.6 The File Handler Commands

The following table provides an overview of all file handler commands. See the description of the ASM 421 (order no. 6GT2 097-3AH00-0DA2) for a detailed description of the individual data bytes.

Table A-1 File handler commands

Command	KK	Meaning	Prio*	Telegram Layout
FORMAT	I	Format and initalize MDS	D	Command: 0 1 2/3 4 5 6/7 8/9 10 11/18 19 20 21 22 KK KI ADB DBN DBL xx 00 0000 'I' 'I' 0001 0001 0C Volume MDS type OPTIONS INIT mode INIT value Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'I' 00 0001 0001 00
CREATE	B	Set up file on MDS	WD	Command: 0 1 2/3 4 5 6/7 8/9 10 11/18 19 20/22 KK KI ADB DBN DBL xx 00 0000 'B' 'I' 0001 0001 0C F name Attribute File length of be res. Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'B' 00 0001 0001 00
QUEUE-WRITE	Q	Set up complete data medium	D	Command: 0 1 2/3 4 5 6/7 8/9 10 11–12 13-21 22–24 25 ... KK KI ADB DBN DBL xx 00 0000 'Q' 'I' ADB 0001 DBL Option In reserve Length of file entries File entries 8 bytes 1 byte 3 bytes 1 byte 1 byte Number of bytes = "length" F name Attribute Length data In reserve Skip Data Data entry Other data entries Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'Q' 00 ADB 0001 00

Table A-1 File handler commands

Command	KK	Meaning	Prio*	Telegram Layout
QUEUE-READ	E	Read several files with one command	RWD	Command: 0 1 2/3 4 5 6/7 8/9 10 11/12 13/21 22/24 25..239 KK KI ADB DBN DBL xx 00 0000 'E' 'I' 0001 0001 DBL Option In reserve Length Data 8 bytes 1 byte 3 bytes 2 bytes File name Res. = 00 Length In reserve Other file entries Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 11...239 KK Status ADB DBN DBL Data (max. of 229 bytes) 00 xx 0000 'E' 00 xxxx 0001 DBL xx xx xx xx xx xx xx xx xx 8 bytes 1 byte 3 bytes 2 bytes File name Res. = 00 Length Skip (hex) File data Skip bytes Other file entries
UPDATE	U	Write complete file	WD	Command: 0 1 2/3 4 5 6/7 8/9 10 11/18 19/21 22/24 25..239 KK KI ADB DBN DBL xx 00 0000 'U' 'I' ADB 0001 DBL File name Start addr. Length Data Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'U' 00 ADB 0001 00
WRITE	W	Write complete or partial file	WD	Command: 0 1 2/3 4 5 6/7 8/9 10 11/18 19/21 22/24 25..239 KK KI ADB DBN DBL xx 00 0000 'W' 'I' ADB 0001 DBL File name Start addr. Length Data Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'W' 00 ADB 0001 00
READ	R	Read file from file	RWD	Command: 0 1 2/3 4 5 6/7 8/9 10 11/18 19/21 22/24 KK KI ADB DBN DBL xx 00 0000 'R' 'I' 0001 0001 0E File name Start addr. Length Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 11...239 KK Status ADB DBN DBL Data 00 xx 0000 'R' 00 xxxx 0001 DBL xx xx xx xx xx xx xx xx xx

Table A-1 File handler commands

Command	KK	Meaning	Prio*	Telegram Layout
DELETE	D	Delete file	D	Command: 0 1 2/3 4 5 6/7 8/9 10 11/18 KK KI ADB DBN DBL xx 00 0000 'D' 'I' 0001 0001 08 File name Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'D' 00 0001 0001 00
ATTRIB	Y	Assign file attribute	WD	Command: 0 1 2/3 4 5 6/7 8/9 10 11/18 19 KK KI ADB DBN DBL xx 00 0000 'Y' 'I' 0001 0001 09 File name Attribute Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'Y' 00 0001 0001 00
COVER	C	Protect MDS structure	WD	Command: 0 1 2/3 4 5 6/7 8/9 10 11/18 19 KK KI ADB DBN DBL xx 00 0000 'C' 'I' 0001 0001 09 Volume User Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'C' 00 0001 0001 00
DIR	G	Read directory of MDS	RWD	Command: 0 1 2/3 4 5 6/7 8/9 10 KK KI ADB DBN DBL xx 00 0000 'G' 'I' 0001 0001 00 Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 11/18 19/20 21/24 25/32 33/35 36 KK Sta–tus ADB DBN DBL 00 xx 0000 'G' 00 0001 0001 xx 'MDS name' xxxx 00000015 ' FILE 1' 00000A 00 Other file entries

Table A-1 File handler commands

Command	KK	Meaning	Prio*	Telegram Layout
MDS-STATUS	F	Read MDS status	RWD	Command: 0 1 2/3 4 5 6/7 8/9 10 KK KI ADB DBN DBL xx 00 0000 'F' 'I' 0001 0001 00 Or: xx 00 0000 'F' 'I' 0001 0001 01 Mode Acknowledgement (mode = 00): 0 1 2/3 4 5 6/7 8/9 10 11/18 19 20/22 23/25 26/27 28 29/31 32 KK Sta–tus ADB DBN DBL 00 xx 0000 'F' 00 0001 0001 16 Vol. Type Chap. prot. Chap. free DIR Anz. BZ ECC Acknowledgement (mode = 01): 0 1 2/3 4 5 6/7 8/9 10 11/14 15 16/23 24/25 26 KK Sta-tus ADB DBN DBL 00 xx 0000 'F' 00 0001 0001 10 MDS no. P type Diagnosis Battery ST
END	K	Conclude communi-cation with the MDS	RWD	Command: 0 1 2/3 4 5 6/7 8/9 10 11 KK KI ADB DBN DBL xx 00 0000 'K' 'I' 0001 0001 01 Mode Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'K' 00 0001 0001 00
TRACE	T	Read MDS uncondi-tionally	RWD	Command: 0 1 2/3 4 5 6/7 8/9 10 11/13 14/16 KK KI ADB DBN DBL Start addr. Length xx 00 0000 'T' 'I' 0001 0001 06 xxxxxxH xxxxxxH Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 11...239 KK Status ADB DBN DBL Data 00 xx 0000 'T' 00 0001 0001 xx Data

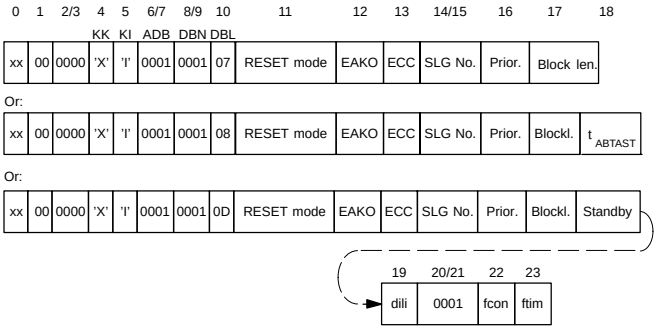
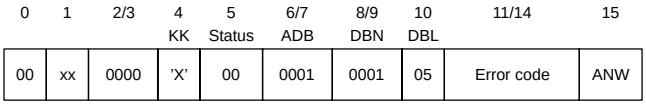
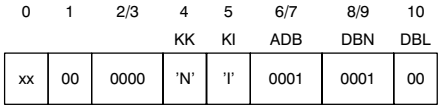
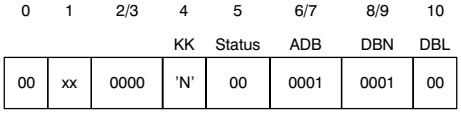
Table A-1 File handler commands

Command	KK	Meaning	Prio*	Telegram Layout
MOVE	M	Read MDS directory from ASM	RWD	Command: 0 1 2/3 4 5 6/7 8/9 10 KK KI ADB DBN DBL xx 00 0000 'M' 'I' 0001 0001 00 Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 11/12 13 ... KK Status ADB DBN DBL Data 00 xx 0000 'M' 00 xxxx 0001 DBL Length DIR + FAT
LOAD	L	Transfer directory to ASM	RWD	Command: 0 1 2/3 4 5 6/7 8/9 10 11/12 13 ... KK KI ADB DBN DBL Data xx 00 0000 'L' 'I' ADB 0001 DBL Length DIR + FAT Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'L' 00 ADB 0001 00
SET-ANT	A	Turn SLG antenna on or off	RWD	Command: 0 1 2/3 4 5 6/7 8/9 10 11 KK KI ADB DBN DBL xx 00 0000 'A' 'I' 0001 0001 01 Mode Acknowledgement: 0 1 2/3 4 5 6/7 8/9 10 KK Status ADB DBN DBL 00 xx 0000 'A' 00 0001 0001 00

Table A-1 File handler commands

Command	KK	Meaning	Prio*	Telegram Layout
ASM-STATUS	S	Read ASM status	RWD	Command: 0 1 2/3 4 5 6/7 8/9 10 KK KI ADB DBN DBL xx 00 0000 'S' 'I' 0001 0001 00 Or: xx 00 0000 'S' 'I' 0001 0001 01 Mode Acknowledgement (mode = 00): 0 1 2/3 4 5 6/7 8/9 10 11/18 19/44 45/70 71 KK Status ADB DBN DBL 00 xx 0000 'S' 00 0001 0001 3DH Ver. no. Last job Last acknowl. Connection status Acknowledgement (mode = 01): 0 1 2/3 4 5 6/7 8/9 10 11 12/35 KK Status ADB DBN DBL 00 xx 0000 'S' 00 0001 0001 19 01 SLG status information (see chap. 4.7) Acknowledgement (mode = 02): 0 1 2/3 4 5 6/7 8/9 10 11 12 13 ... KK Status ADB DBN DBL 00 xx 0000 'S' 00 0001 0001 DBL 02 No. of functions Functions Acknowledgement (mode = 03): 0 1 2/3 4 5 6/7 8/9 10 11 12 13 ... KK Status ADB DBN DBL 00 xx 0000 'S' 00 0001 0001 DBL 03 No. of errors Last error Acknowledgement (mode = 04): 0 1 2/3 4 5 6/7 8/9 10 11 12 13 ... KK Status ADB DBN DBL 00 xx 0000 'S' 00 0001 0001 DBL 04 No. of MDSs MDS IDs

Table A-1 File handler commands

Command	KK	Meaning	Prio*	Telegram Layout
RESET	X	RESET command to ASM	RWD	Command:  Acknowledgement: 
NEXT	N	Process next MDS	RWD	Command:  Acknowledgement: 

* The priority of access rights of the SLG is specified with the RWD parameter during configuration. See section 3.2.

Warnings

B

English

Warning

Hazardous voltages are present in this equipment during operation.

To ensure safe operation of the equipment, maintenance shall only be performed by qualified personnel in accordance with the instructions in the MOBY catalog[1] and technical description.

Failure to observe these instructions can result in death, severe personal injury or substantial damage to property.

The following instructions and those on all product labels must be followed when carrying out any maintenance work.

- Always disconnect and earth the equipment before starting any maintenance.
- Use only spare parts authorized by the manufacturer.
- The servicing intervals as well as the instructions for repair and replacement shall be duly observed.
- A lithium battery is contained in mobile data memories with RAM. The following instructions must be observed:

  To avoid the risk of fire, explosion and severe burns, the battery should not be recharged, dismantled, exposed to heat over 100 degrees Celsius, ignited, or brought into contact with water.

The special instructions must be followed when using heat-resistant data storage media.

1 Should you not be in possession of the MOBY catalog, it can be obtained through your local Siemens office.

Deutsch

Warnung

Beim Betrieb elektrischer Geräte stehen zwangsläufig bestimmte Teile dieser Geräte unter gefährlicher Spannung.

Sicherer Betrieb der Geräte setzt voraus, daß diese von qualifiziertem Personal sachgemäß unter Beachtung der im MOBY-Katalog[1] und der technischen Beschreibung enthaltenen Hinweise eingesetzt werden.

Bei Nichtbeachtung können Tod, schwere Körperverletzung oder erheblicher Sachschaden die Folge sein.

Beachten Sie daher auch bei Instandhaltungsmaßnahmen an diesem Gerät alle hier und auf dem Produkt selbst aufgeführten Hinweise.

- Vor Beginn jeglicher Arbeiten ist das Gerät vom Netz zu trennen und zu erden.
- Es dürfen nur vom Hersteller zugelassene Ersatzteile verwendet werden.
- Die vorgeschriebenen Wartungsintervalle sowie die Anweisungen für Reparatur und Austausch sind unbedingt einzuhalten.
- Bei einem mobilen Datenspeicher mit RAM ist eine Lithiumbatterie integriert, hierzu sind folgende Hinweise zu beachten:

  Vermeiden Sie das Risiko von Feuer, Explosionen und schweren Verbrennungen. Die Batterie darf nicht nachgeladen, auseinandergebaut, über 100˚ Celsius erwärmt, entzündet oder ihr Inhalt mit Wasser in Berührung gebracht werden.

Beim hitzefesten Datenträger sind die besonderen Hinweise zu berücksichtigen.

1 Sollten Sie nicht im Besitz des MOBY-Katalogs sein, so kann er über jede örtliche SIEMENS-Niederlassung bestellt werden.

Français

Attention

Le fonctionnement d'un équipement électrique implique nécessairement la présence de tensions dangereuses sur certaines de ses parties.

L'exploitation sûre de cet équipement implique qu'il soit mis en oeuvre de façon adéquate par des personnes qualifiées, en respectant les consignes de sécurité figurant au catalogue MOBY[1] et aux descriptions techniques.

Le non-respect des consignes de sécurité peut conduire à la mort, à des lésions corporelles graves ou à un dommage matériel important.

Ne procéder à l'entretien que dans le plus grand respect des règles de sécurité énoncées ici ou figurant sur le produit.

- Avant toute intervention, mettre l'appareil hors tension et à la terre.
- N'utiliser que des pièces de rechange autorisées.
- Respecter la périodicité d'entretien et les instructions de réparation et de remplacement.
- Les mémoires embarquées (RAM) sont équipées d'une pile au lithium.

  Ne pas exposer la pile au feu, danger d'explosion et de lésions graves. La pile ne doit pas être rechargée, ouverte exposée à des températures supérieures à 100˚ C ou exposée au feu. Son contenu ne doit pas entrer en contact avec de l'eau.

En ce qui concerne les supports de données résistants à la chaleur, respecter les consignes spécifiques.

1 Si vous ne disposez pas ou du catalogue MOBY, ce peuvent être commandés auprès de votre agence SIEMENS.

Italiano

Pericolo

Durante il funzionamento di apparecchi elettrici, determinate parti di tali apparecchi si trovano inevitabilmente sotto tensione pericolosa.

Per un funzionamento sicuro di questi apparecchi è necessario che essi vengano adoperati, nel modo opportuno, solo da personale qualificato, che osservi le indicazioni contenute nel catalogo[1] per gli apparecchi MOBY e nella descrizione tecnica.

In caso di non osservanza si possono verificare la morte, gravi lesioni alle persone o notevoli vanni alle cose.

Per questo motivo è necessario che le avvertenze riportate qui e sul prodotto stesso vengano rispettate anche nel caso di misure di manutenzione degli apparecchi.

- Prima di iniziare qualsiasi lavoro è necessario staccare l'apparecchio dalla rete ed effettuare una messa a terra.
- Possono essere utilizzati solo pezzi di ricambio prodotti dal costruttore.
- E'assolutamente necessario rispettare i tempi di manutenzione previsti e le indicazioni riguardanti il ricambio e la riparazione.
- In una memoria dati mobile dotata di RAM è integrata una batteria al litio; in questo caso è necessario osservare le seguenti indicazioni:

  evitare il pericolo di incendio, di esplosioni e di gravi ustioni. E'vietato ricaricare, smontare, riscaldare oltre i 100˚ C o incendiare la batteria, oppure mettere il suo contenuto a contatto con acqua.

Nel caso di un supporto dati resistente al calore è necessario osservare le indicazioni speciali al riguardo.

1 Se non doveste essere in possesso del catalogo MOBY, potete ordinarlo presso qualsiasi filiale SIEMENS di zona.

Español

Precaución

Durante el funcionamiento de los equipos eléctricos hay partes de los mismos que se encuentran forzosamente bajo tensión peligrosa.

Un funcionamiento seguro de los equipos presupone que han sido instalados correctamente por personal calificado observando las indicaciones contenidas en el Catálogo[1] de los equipos MOBY y la Descripción técnica.

La no observación de dichas indicaciones puede provocar la muerte, lesiones corporales graves o daños materiales considerables.

Por este motivo es preciso observar también durante las operaciones de mantenimiento y reparación en dicho equipo todas las indicaciones que figuran aquí y en el producto.

- Antes de comenzar cualquier trabajo es preciso seccionar de la red el equipo y ponerlo a tierra.
- Solo deben utilizarse repuestos homologados por el fabricante.
- Es imprescindible observar los intervalos de mantenimiento especificados asi como las instrucciones de reparación y reemplazo.
- Las memorias de datos móviles con RAM tienen integrada una batería de litio; al respecto es preciso observar las indicaciones siguientes:

  Evite riesgos de fuego, explosiones y quemaduras graves. La batería no debe ser recargada, desmontada, calentada a mas de 100 grados centígrados, inflamada: su contenido no deberá ponerse en contacto con agua.

En los soportes de datos con protección térmica es preciso observar las indicaciones particulares respectivas.

1 Si no dispone del catálogo MOBY, estos pueden pedirse a través de cualquier sucursal local de SIEMENS.

Svensk

Varning

Vid drift av elektrisk utrustning ligger det alltid en farlig spänning på vissa delar av utrustningen.

Säker drift av utrustningen förutsätter att den utförs av kvalificerad personal med uppmärksamhet på anvisningarna i MOBY-katalogen[1] samt de anvisningarsom ges i den tekniska beskrivningen.

Om dessa anvisningar ej beaktas kan följden bli dödsfall, svår kroppskada eller avsevärda materielskador.

Uppmärksamma vid underhållsarbete också anvisningar som ges här och på själva produkten.

- Före allt arbete skall utrustningen skiljas fran nätet och jordas.
- Bara reservdelar som godkänts av tillverkaren får användas.
- lakttag alltid föreskrivna underhållsintervall samt de anvisningar som givits rörande reparation och utbyte.
- Det mobila dataminnet med RAM innehåller et litiumbatteri. För detta gäller följande anvisningar:

  Undvik risk för öppen låga, explosioner och förbränning. Batteriet får inte efterladdas, tas isär, värmas upp över 100˚ C eller tändas på., och dess innehåll får ej komma i beröring med vatten.

För värmebeständiga datamedier gäller speciella anvisningar, som måste beaktas.

1 Om Ni inte har ett exemplar av MOBY-katalogen så kan den beställas från närmaste SIEMENS-kontor.

TO:

Siemens AG

A&D SE ES4

PO Box 2355

D-90713 Fuerth

FROM:

Your name: _

Your title: _

Your company: _

Street: _

City: _

Telephone: _

Please tick your branch.

- ❑ Automotive industry
- ❑ Chemical industry
- ❑ Electrical industry
- ❑ Foodstuffs
- ❑ Process control technology
- ❑ Mechanical engineering
- ❑ Petrochemistry
- ❑ Pharmaceutics industry
- ❑ Plastics processing
- ❑ Paper industry
- ❑ Textiles industry
- ❑ Transportation industry
- ❑ Other _ _ _ _ _ _ _ _ _ _ _ _ _

Remarks/Suggestions

Your remarks and suggestions help us to improve the quality of our documentation. Please complete this questionnaire as soon as you have time, and return it to Siemens.

Title of your manual:_ _

Order number of your manual:_ _

Please enter your personal evaluation from 1 (good) to 5 (poor).

1. Do the contents of this manual meet your requirements ? ☐
2. Is it easy to find the information you need ? ☐
3. Is the information written in an easy-to-understand manner ? ☐
4. Does the amount of technical detail meet your requirements ? ☐
5. What is your opinion of the figures and tables ? ☐

If you encountered concrete problems, please use this space to explain.

_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _