EUCHNER

AS-i 3.0 Command Interface

Description of commands

AS-Interface

AS-i 3.0 Specification

Revision date: 2007-12-3

Table of Contents

AS-i 3.0 Command Interface

1 Introduction

The AS-i gateways integrate the AS-i slaves into the upstream fieldbus. Each upstream fieldbus (f.e. Modbus/TCP, CANopen, or PROFIBUS) has its unique possibilities to access cyclically and acyclically data. The gateway polls as an AS-i master all the slaves on the AS-i circuit. The result of these polls the gateway keeps in its internal state RAM as images of the inputs, outputs, parameters, and status. These images are available for use on the upstream fieldbus with their specific access methods. The images of the Modbus/TCP to AS-i gateway are available with Modbus Read and Write function calls on different Modbus registers. The main manual (command: insert cross reference) describes this in detail. CANopen provides this access with PDOs for cyclical access and SDOs for acyclical access.

The access to the images of the gateway is easy to configure on the upstream fieldbus and in most applications sufficient. However, the complete functionality of the gateway is available with the command interface. If you want to read the diagnosis string of an AS-i tuner (slave with 7.4 profile), you will need the command interface to call the WRITE_ACYC_DATA and READ_ACYC_DATA commands.

The command interface is avaible in a special image. A command is called by writing into this image and the command result is avaible with a read to this image.

Note	The manual "AS-i 3.0 Command Interface" describes commands of the AS-i 3.0 Command Interface. A description of an AS-i Master is not included. Please refer to the corresponding manual of your AS-i Master for further information. Please view the documentation of the respective device for further, device-specific information about the kind of the access to the command interface.

2 Structure of the Command Interface

The command interface has the following structure shown in *table 1* and *table 2*.

Table 1

command request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	command							
2	T	O	circuit					
3	request parameter byte 1							
…	…							
36	request parameter byte 34							

*Bit **T*** in the command interface is the **toggle bit**. The toggle bit is only necessary in the case of interfaces which transfer the data cyclically.

The execution of a command of the command interface is declined, if the number of the transferred parameters is too small, this could happen when the command interface is too small or the tegram is too short.

Circuit selects the AS-i circuit. Circuit = 0 selects the first circuit.

*Bit **LO*** is the list order bit. The commands for reading and writinig slave lists support two different sorting schemas.
LO = 0 selects the Euchner schema.
LO = 1 selects the Siemens schema (the sequence of the bits in the slave lists bytes is inverse).

Parameter byte n is the n^{th} parameter of the command. The number of parameters is different for different commands. It is not necessary to set the additional parameter bytes to 0 in the command interface, if a command does not use the maximum number of parameter bytes (36)

Table 2

command response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	command (mirrored)							
2	T	result						
3	response byte 1							
…	…							
36	response byte 34							

There is the reflected command byte and the toggle bit of the request in the response. The execution of the command returns its result in the seven least significant bits of byte 2 of the response. 0 signals execution of the command without an error. The table *result codes* shows all possible result codes.

Note

Please note that possibly some controls can exchange the high and low byte on the field bus with word orientated access to the command interface.

Result codes

Name	***Value***	***Description***
OK	00_{16}	execution without fault
HI_NG	11_{16}	general fault
HI_OPCODE	12_{16}	illegal value in command
HI_LENGTH	13_{16}	length of the command interface is too short[1]
HI_ACCESS	14_{16}	no access right
EC_NG	21_{16}	general fault
EC_SND	22_{16}	slave (source addr) not detected
EC_SD0	23_{16}	slave 0 detected
EC_SD2	24_{16}	slave (target addr) not decteced
EC_DE	25_{16}	delete error
EC_SE	26_{16}	set error
EC_AT	27_{16}	address temporary
EC_ET	28_{16}	extended ID1 temporary
EC_RE	29_{16}	read (extended ID1) error

1. The length of the command interface in the I/O-data area respectively the length of the DPV1 requests is too short

3 List of all Commands

Attention	**!!!** The most of the described commands can be applied to all AS-i 3.0 Masters. Exceptions are indicated in footers.

Values for command

see page	*Command*	*Value*	*Meaning*	*Req Len*	*Res Len*
page 12	**AS-i 16-bit data**				
page 12	RD_7X_IN	50_{16}	read 1 16-bit slave profile in.data	3	10
page 13	WR_7X_OUT	51_{16}	write 1 16-bit slave profile out.data	11	2
page 13	RD_7X_OUT	52_{16}	read 1 16-bit slave profile out.data	3	10
page 14	RD_7X_IN_X	53_{16}	read 4 16-bit slave profile in.data	3	34
page 14	WR_7X_OUT_X	54_{16}	write 4 16-bit slave profile out.data	35	2
page 15	RD_7X_OUT_X	55_{16}	read 4 16-bit slave profile out.data	3	34
page 15	OP_RD_16BIT_IN_CX	$4C_{16}$	read 16 channels 16-bit slave in.data	3	34
page 16	OP_WR_16BIT_IN_CX	$4D_{16}$	write 16 channels 16-bit slave in.data	36	2
page 16	**Commands acc. to Profile S-7.4/S-7.5**				
page 16	WR_74_75_PARAM	$5A_{16}$	write S-7.4/S-7.5-slave parameter	≥6	2
page 17	RD_74_75_PARAM	$5B_{16}$	read S-7.4/S-7.5-slave parameter	4	≥3
page 18	RD_74_75_ID	$5C_{16}$	read S-7.4/S-7.5-slave ID string	4	≥3
page 19	RD_74_DIAG	$5D_{16}$	read S-7.4/S-7.5-slave diagnosis string	4	≥3
page 19	**Acyclic commands**				
page 19	WRITE_ACYC_TRANS	$4E_{16}$	write acyclic transfer	≥7	2
page 21	READ_ACYC_TRANS	$4F_{16}$	read acyclic transfer	5	≥2
page 31	**AS-i Diagnosis**				
page 31	GET_LISTS	30_{16}	get LDS/LAS/LPS flags	2	29
page 33	GET_FLAGS	47_{16}	get flags	2	5
page 34	GET_DELTA	57_{16}	get list of config. diff.	2	10
page 35	GET_LCS	60_{16}	get LCS	2	10
page 35	GET_LAS	45_{16}	get LAS	2	10
page 36	GET_LDS	46_{16}	get LDS	2	10
page 37	GET_LPF	$3E_{16}$	get LPF	2	10
page 37	GET_LOS	61_{16}	get LOS	2	10
page 38	SET_LOS	62_{16}	set LOS	10	2
page 39	GET_TECA	63_{16}	get transm.err.counters	2	34
page 40	GET_TECB	64_{16}	get transm.err.counters	2	34
page 41	GET_TEC_X	66_{16}	get transm.err.counters	4	≥3
! *page 41*	READ_FAULT_DETECTOR[1]	10_{16}	read Fault_Detector	2	4

Values for command

	see page	Command	Value	Meaning	Req Len	Res Len
!	page 42	READ_DUPLICATE_ADDR[2]	11_{16}	read list of duplicate addresses	2	10
	page 43	**Configuration of AS-i Master**				
	page 43	SET_OP_MODE	$0C_{16}$	set Operation_Mode	3	2
	page 44	STORE_CDI	07_{16}	store Actual_Configuration	2	2
	page 45	READ_CDI	28_{16}	read Actual_Configuration	3	4
	page 45	SET_PCD	25_{16}	set Permanent_Config	5	2
	page 45	GET_PCD	26_{16}	get Permanent_Config	3	4
	page 47	SET_LPS	29_{16}	set LPS	11	2
	page 48	GET_LPS	44_{16}	get LPS	2	10
	page 48	STORE_PI	04_{16}	store Actual_Parameter	2	2
	page 49	WRITE_P	02_{16}	write Parameter	4	3
	page 49	READ_PI	03_{16}	read Parameter	3	3
	page 50	SET_PP	43_{16}	set Permanent_Parameter	4	2
	page 50	GET_PP	01_{16}	get Permanent_Parameter	3	3
	page 51	SET_AAE	$0B_{16}$	set Auto_Address_Enable	3	2
	page 53	SLAVE_ADDR	$0D_{16}$	change Slave_Address	4	2
	page 52	WRITE_XID1	$3F_{16}$	write Extended_ID-Code_1	3	2
	page 53	**Other commands**				
	page 53	IDLE	00_{16}	no request	2	2
	page 54	READ_IDI	41_{16}	read IDI	2	36
	page 54	WRITE_ODI	42_{16}	write ODI	34	2
	page 55	READ_ODI	56_{16}	read ODI	2	34
	page 55	SET_OFFLINE	$0A_{16}$	set Off-Line_Mode	3	2
	page 56	SET_DATA_EX	48_{16}	set Data_Exchange_Active	3	2
!	page 56	REWRITE_DPRAM[3]	78_{16}	rewrite DPRAM	3	3
	page 56	BUTTONS	75_{16}	disable push buttons	3	2
	page 57	FP_PARAM	$7D_{16}$	functional Profile Parameter	≥3	≥2
	page 72	language-select	$0E_{16}$	set display language	4	3
	page 73	replacement of safety slaves input data	$0F_{16}$	set safety input slave "interpretation data"	4	2
	page 58	FP_DATA	$7E_{16}$	functional profile data	≥3	≥2
	page 62	"Safety at Work" list	00_{16}	slaves with released safety function, response contains EcFlags	3	8
	page 64	"Safety at Work" list	$0D_{16}$	slaves with released safety function, response doesn‘t contain EcFlags	3	6
	page 64	"Safety at Work" diagno-sis	02_{16}	monitor diagnosis	5	n
	page 70	integrated AS-i sensors: Warnings	03_{16}	sensors with deleted D1 bit	3	10

Values for command

	see page	Command	Value	Meaning	Req Len	Res Len
	page 71	Integrated AS-i sensors: Availability	04_{16}	sensors with deleted D2 bit	3	6
	page 72	language-select	$0E_{16}$	read display language	3	3
	page 73	replacement of safety slaves input data	$0F_{16}$	read safety input slave "interpreta-tion data"	3	4
	page 74	list of safety slaves	10_{16}	read addresses of safety slaves	3	6
!	page 58	EXT_DIAG[4]	71_{16}	ExtDiag generation	6	2
!	page 59	RD_EXT_DIAG[5]	$7B_{16}$	read ExtDiag Settings	2	7
	page 60	INVERTER	$7C_{16}$	configure inverter slaves	12	4
	page 60	MB_OP_CTRL_WR_FLAGS	85_{16}	write flags	≥5	2
	page 61	MB_OP_CTRL_RD_FLAGS	86_{16}	read flags	4	≥3
	page 61	RD_MFK_PARAM	59_{16}	read SEW MFK21 parameter	6	≥3

1. The command READ_FAULT_DETECTOR is valid only for the use with masters which support this function. Please refer to the user manual of the master for further information.
2. The command READ_DUPLICATE_ADDR is valid only for the use with masters which support this function. Please refer to the user manual of the master for further information.
3. The command REWRITE_DPRAM is valid only for the use with ***AS-i 3.0 Module OEM Master***
4. The command EXT_DIAG is valid only for the use with ***AS-i 3.0 PROFIBUS Gateways***
5. The command RD_EXT_DIAG is valid only for the use with ***AS-i 3.0 PROFIBUS Gateways***

4 Commands of the Command Interface

4.1 AS-i 16-bit data

4.1.1 Overview of the commands

Values for command

see page	*Command*	*Value*	*Meaning*	*Req Len*	*Res Len*
page 12	RD_7X_IN	50_{16}	read 1 16-bit slave profile in.data	3	10
page 13	WR_7X_OUT	51_{16}	write 1 16-bit slave profile out.data	11	2
page 13	RD_7X_OUT	52_{16}	read 1 16-bit slave profile out.data	3	10
page 14	RD_7X_IN_X	53_{16}	read 4 16-bit slave profile in.data	3	34
page 14	WR_7X_OUT_X	54_{16}	write 4 16-bit slave profile out.data	35	2
page 15	RD_7X_OUT_X	55_{16}	read 4 16-bit slave profile out.data	3	34
page 15	OP_RD_16BIT_IN_CX	$4C_{16}$	read 16 channels 16-bit slave in.data	3	34
page 16	OP_WR_16BIT_IN_CX	$4D_{16}$	write 16 channels 16-bit slave in.data	36	2

4.1.2 Read 1 16-bit Slave in.Data (RD_7X_IN)

With this command, the four 16-bit channels of an AS-i input slave according to the slave profile (S-7.3, S-7.4, S-7.5, S-7.A.8, S.A.9, S-7.A.A) can be read.

Note

A-Slaves map the data on channels 1 and 2.
B-Slaves map the data on channels 3 and 4.
Only values among 1 and 31 can be taken as a slave address.

Request

byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	50_{16}							
2	T	–	circuit					
3	–		0	slave address				

Response

byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	50_{16}							
2	T	result						
3	channel 1, high byte							
…	…							
10	channel 4, low byte							

4.1.3 Write 1 16-bit Slave out. Data (WR_7X_OUT)

With this command, the four 16-bit channels of an AS-i output slave according to the slave profile (S-7.3, S-7.4, S-7.5, S-7.A.8, S.A.9, S-7.A.A) can be written.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	51_{16}							
2	T	–	circuit					
3	–		0	slave address				
4	channel 1, high byte							
…	…							
11	channel 4, low byte							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	51_{16}							
2	T	result						

4.1.4 Read 1 16-bit Slave out. Data (RD_7X_OUT)

With this command, the four 16-bit channels of an AS-i output slave according to the slave profile (S-7.3, S-7.4, S-7.5, S-7.A.8, S.A.9, S-7.A.A) can be read.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	52_{16}							
2	T	–	circuit					
3	–		0	slave address				

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	52_{16}							
2	T	result						
3	channel 1, high byte							
…	…							
10	channel 4, low byte							

4.1.5 Read 4 16-bit Slave in. Data (RD_7X_IN_X)

With this command, the four 16-bit channels of 4 AS-i input slaves with successive addresses according to slave profile (S-7.3, S-7.4, S-7.5, S-7.A.8, S.A.9, S-7.A.A) can be read.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	53_{16}							
2	T	–	circuit					
3	–		0	1st slave address				

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	53_{16}							
2	T	result						
3	1st slave, channel 1, high byte							
…	…							
34	4th slave, channel 4, low byte							

4.1.6 Write 4 7.3 Slave out. Data (WR_7X_OUT_X)

With this command the four 16-bit channels of four AS-i output slaves with successive addresses according to slave profile (S-7.3, S-7.4, S-7.5, S-7.A.8, S.A.9, S-7.A.A) can be written.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	54_{16}							
2	T	–	circuit					
3	–		0	1st slave address				
4	1st slave, channel 1, high byte							
…	…							
35	4th slave, channel 4, low byte							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	54_{16}							
2	T	result						

4.1.7 Read 4 7.3 Slave out. Data (RD_7X_OUT_X)

With this command, the four 16-bit channels of four AS-i output slaves with successive addresses according to slave profile (S-7.3, S-7.4, S-7.5, S-7.A.8, S.A.9, S-7.A.A) can be read.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	55_{16}							
2	T	–	circuit					
3	–		0	1st slave address				

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	55_{16}							
2	T	result						
3	1st slave, channel 1, high byte							
…	…							
34	4th slave, channel 4, low byte							

4.1.8 Read 16 channels 16-bit Slave in. Data (OP_RD_16BIT_IN_CX)

With this command, the 16 channels of the 16-bit input-data for slaves with successive addresses according to slave profile (S-7.3, S-7.4, S-7.5, S-7.A.8, S.A.9, S-7.A.A) can be read.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$4C_{16}$							
2	T	–	circuit					
3	1. slave							
4	1. channel							

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$4C_{16}$							
2	T	result						
3	1. slave, channel 1, high byte							
4	1. slave, channel 1, low byte							
…	…							
33	16. channel, high byte							
34	16. channel, low byte							

4.1.9 Write 16 channels 16-bit slave out. Data (OP_WR_16BIT_IN_CX)

With this command, the 16 channels of the 16-bit input-data for slaves with successive addresses according to slave profile (S-7.3, S-7.4, S-7.5, S-7.A.8, S.A.9, S-7.A.A) can be written.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$4D_{16}$							
2	T	circuit						
3	1. slave							
4	1. channel							
5	1. slave, 1. channel, high byte							
6	1. slave, 1. channel, low byte							
…	…							
35	16. channel, high byte							
36	16. channel, low byte							

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$4D_{16}$							
2	T	result						

4.2 Commands acc. to Profile S-7.4/S-7.5

4.2.1 Overview of the commands

Values for command

see page	*Command*	*Value*	*Meaning*	*Req Len*	*Res Len*
page 16	WR_74_75_PARAM	$5A_{16}$	write S-7.4/S-7.5-slave parameter	≥6	2
page 17	RD_74_75_PARAM	$5B_{16}$	read S-7.4/S-7.5-slave parameter	4	≥3
page 18	RD_74_75_ID	$5C_{16}$	read S-7.4/S-7.5-slave ID string	4	≥3
page 19	RD_74_DIAG	$5D_{16}$	read S-7.4/S-7.5-slave diagnosis string	4	≥3

4.2.2 WR_74_75_PARAM

With this function the parameter string of a slave according to profile S-7.4 is being written or the data transfer with a slave according to profile S-7.5 is started.

By a slave according to profile 7.5, data have to be registered into the buffer in the same form, as they have to be sent by AS-i.

Since the string can be longer than the command interface, it is written into the buffer in parts at first and then it is transferred to the slave.

n is the length of the part of the string which should be written into the buffer from *index i* on.

If i ≡ 0, then the string is being transferred to the slave.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$5A_{16}$							
2	T	–	circuit					
3	slave address							
4	i							
5	n							
6	buffer byte i							
…	…							
n+5	buffer byte i+n-1							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$5A_{16}$							
2	T	results						

4.2.3 RD_74_75_PARAM

With this function the parameter string of a slave according to profile S-7.4 is being read or the slave response according to profile S-7.5 is being read.

If it is about a slave according to profile 7.5, so have the data in the response buffer the following meaning:

FFh 00_{16}: Transfer is still active

FFh xx_{16}: Transfer finished with error

The first byte in the buffer not equal FF_{16}: slave response. The response is in the same form registered in the buffer and transmitted over AS-i.

Since the string can be longer than the command interface, it is written into the buffer. The content of the buffer can read in parts from index i.

The first byte of the buffer is the length of the read string.

If i ≡ 0, the string is being read from the slave, otherwise the function responses out of the memory; the data can be read consistently.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$5B_{16}$							
2	T	–	circuit					
3	slave address							
4	i							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$5B_{16}$							
2	T	result						
3	buffer byte i							
…	…							
n+2	buffer byte i+n-1							

4.2.4 RD_74_75_ID

With this function the ID string of a slave according to profile S-7.4 or the 16-bit slave configuration according to profile 7.5 is being read. Since the string can be longer than the command interface, it is written into the buffer. The content of the buffer can read in parts from index i.

The first byte of the buffer is the length of the read string.

If i $\equiv$ 0, the string is being read from the slave, otherwise the function responses out of the memory, the data can be read consistently.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$5C_{16}$							
2	T	–	circuit					
3	slave address							
4	i							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$5C_{16}$							
2	T	result						
3	buffer byte i							
…	…							
n+2	buffer byte i+n-1							

By a 7.5 slave is the request always 1. The response byte contains the cyclic 16-bit slave configuration according to S-7.5 profile (analog/transparent bits are cancelled). If the response is 08_{16}, that means that the cyclic 16-bit configuration could not be detected.

4.2.5 RD_74_DIAG

With this function the diagnosis string of a slave according to profile S-7.4 is being read. Since the string can be longer than the command interface, it is written into the buffer. The content of the buffer can be read in parts from index i.

The first byte of the buffer indicates the length of the read string.

If i ≡ 0, the string is being read from the slave, otherwise the function responses out of the memory, the data can be read consistently.

<table>
<tr><th colspan="9">Request</th></tr>
<tr><th>byte</th><th>2^7</th><th>2^6</th><th>2^5</th><th>2^4</th><th>2^3</th><th>2^2</th><th>2^1</th><th>2^0</th></tr>
<tr><td>1</td><td colspan="8">$5D_{16}$</td></tr>
<tr><td>2</td><td>T</td><td>–</td><td colspan="6">circuit</td></tr>
<tr><td>3</td><td colspan="8">slave address</td></tr>
<tr><td>4</td><td colspan="8">i</td></tr>
</table>

<table>
<tr><th colspan="9">Response</th></tr>
<tr><th>byte</th><th>2^7</th><th>2^6</th><th>2^5</th><th>2^4</th><th>2^3</th><th>2^2</th><th>2^1</th><th>2^0</th></tr>
<tr><td>1</td><td colspan="8">$5D_{16}$</td></tr>
<tr><td>2</td><td>T</td><td colspan="7">result</td></tr>
<tr><td>3</td><td colspan="8">buffer byte i</td></tr>
<tr><td>…</td><td colspan="8">…</td></tr>
<tr><td>n+2</td><td colspan="8">buffer byte i+n-1</td></tr>
</table>

4.3 Acyclic commands

4.3.1 Overview of the commands

Values for command

see page	*Command*	*Value*	*Meaning*	*Req Len*	*Res Len*
page 19	WRITE_ACYC_TRANS	$4E_{16}$	write acyclic transfer	≥7	2
page 21	READ_ACYC_TRANS	$4F_{16}$	read acyclic transfer	5	≥2

4.3.2 WRITE_ACYCLIC_TRANS

This function starts various types of acyclic transfer (S-7.4, S-7.5 and Safety Monitor). The transfer is performed in the background. The result must be read using READ_ACYC_TRANS. The function is intended to be a replacement for the functions (RD_74_75_PARAM, WR_74_75_PARAM, RD_74_75_ID, RD_74_DIAG and "Safety at Work" monitor diagnostics), as it runs in the background and does not stop the AS-i master during the transfer.

As the data to be transferred can be longer than the command interface, the data is first written to a buffer in sections before the transfer is started.

n is the length of the sub-string that is to be written to the buffer starting from index (i). When i = 0, the transfer is started.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$4E_{16}$							
2	T	circuit						
3	slave address							
4	buffer index (i) high							
5	buffer index (i) low							
6	command[1]							
7	number of (n)							
8	data 0							
…	...							
n+7	data n-1							

1. For a list of all supported commands <see table "Espoused commands", page 20>.

Espoused commands

	see page	***Command***	***Description***
	page 23	**1**	read string S-7.4 ID
	page 23	**2**	read string S-7.4 diag
	page 23	**3**	read string S-7.4 param string
	page 23	**4**	write S-7.4 param string
	page 24	**5**	transfer S-7.5
	page 24	**6**	read S-7.5 cyclic 16-bit slave configuration
	page 24	**7**	read safety monitor *sorted* by OSSD
	page 26	**8**	read safety monitor *unsorted* (all devices) by OSSD
	page 26	**9**	reserved / not defined
!!!	*page 27*	**10**	safety monitor diagnosis
!!!	*page 27*	**11**	shutdown-history, separate for each release circuit
!!!	*page 27*	**12**	safety monitor diagnosis, but the module allocation has been con-sidered
!!!	*page 27*	**13**	safety monitor diagnosis, but the module allocation has been con-sidered
!!!	*page 28*	**14**	diagnosis / shutdown-history, separate for each release circuit
!!!	*page 30*	**15**	safety status

Note

!!!

The commands **10 ... 15** are available only with safety monitors (external and integrated) in the version 2 and higher.

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$4E_{16}$							
2	return							

4.3.3 READ_ACYCLIC_TRANS

With this call the response of the transfer command (started with WRITE_ACYCLIC_TRANS) is read.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$4F_{16}$							
2	T	circuit						
3	slave address							
4	buffer index (i) high							
5	buffer index (i) low							

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$4F_{16}$							
2	T	response						
3	data i							
...	...							
m[1]	data i+(m-2)							

1. command interface response length **m**

The response data have the same format, as by commands RD_74_75_PARAM, RD_74_75_ID and „safety at work"-monitor diagnostics <see chapter 4.7.2 "Safety at Work" Monitor diagnosis, page 64>.

4.3.3.1 Structure of the response buffer

As the string to be transferred can be longer than the command interface, the string is first saved in a buffer that can be read in sections using the buffer index **(i)**.

The first byte in the response buffer defines the current command. FF_{16} signifies transfer still active, FE_{16} signifies transfer interrupted with errors. In the correct case, the command from WRITE_ACYC_TRANS is given here.

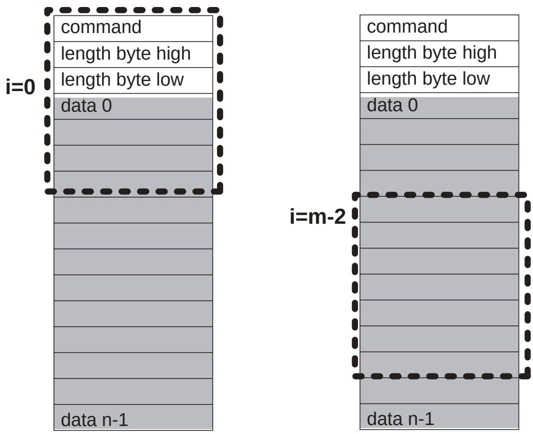
The first sub-section of the string is read using i ≡ 0, the second with i = n-2, etc. The two following bytes (high, low) define the length of the response buffer.

It is recommended to start reading the data always using index $i \equiv 0$. This message also contains the header. The user data length is therefore reduced by 3 bytes.

Note	
	Data with length $i \equiv 0$ can be read successfull only once. Each further read command with length $i \equiv 0$ ist quit with an error. Therefore further read process (sections) must be carried out with i >0!

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	command[1]							
2	length byt[2]e n (high)							
3	length byte n (low)							
4	data 0							
...	...							
n+3	data n-1							

1. FFh signifies transfer still active, FEh signified transfer interrupted with errors. In the correct case the command from WRITE_ACYC_TRANS is given here.
2. Transmit buffer length n



- - - -	read section of the string
m -	command interface response length
n -	transmit buffer length

Note	
	For further information <see chapter 5.4 Example for the readout of the safety monitor with ACYC_TRANS, page 88>

4.3.3.2 Command 1: Read „S-7.4 ID String"

With this call the *ID string* of a slave according to profile S-7.4 can be read.

Response buffer								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	ID string byte 0							
2	ID string byte 1							
...	...							
n	ID string byte n-1							

4.3.3.3 Command 2: Read „S-7.4 Diag String"

With this call the *diag string* of a slave according to profile S-7.4 can be read.

Response buffer								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	diag string byte 0							
2	diag string byte 1							
...	...							
n	diag string byte n-1							

4.3.3.4 Command 3: Read „S-7.4 Param String"

With this call the *param string* of a slave according to profile S-7.4 can be read.

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	param string byte 0							
2	param string byte 1							
...	...							
n	param string byte n-1							

4.3.3.5 Command 4: Write „S-7.4 Param String“

With this call the *param string* of a slave according to profile S-7.4 can be written.

Request buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	param string byte 0							
2	param string byte 1							
...	...							
n	param string byte n-1							

4.3.3.6 Command 5: „Transfer S-7.5“

With this call the *transfer string* of a slave according to profile S-7.5 can be trasferred. The request/response buffer contain the S-7.5 strings in the same form as they are transferred via AS-i.

Request buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	CTT2 command byte (16_{10} - 19_{10})							
2	index							
3	length							
4	data 0							
5	data 1							
...	...							
n	data n-4							

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	CTT2 reponse byte (50_{10} - 52_{10}, 90_{10} - 92_{10})							
2	data 0							
3	data 1							
...	...							
n	data n-2							

4.3.3.7 Command 6: Read „Cyclical S-7.5 16-bit configuration“

With this call the cyclical S-7.5 16-bit configuration can be read, the analog/tranparent bits being deleted in the response.

The cyclical 16-bit configuration cannot be determined if the response is 08_{16}.

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0: no output 1: 1-byte output 2: 1-word output 3: 2-word output 4: 3-word output 5: 4-word output			0: data are valid 1: data are not valid	0: no input 1: 1-byte input 2: 1-word input 3: 2-word input 4: 3-word input 5: 4-word input		

4.3.3.8 Command 7: Read „Safety Monitor sorted acc. to OSSD“

With this command the safety monitor is being read *sorted* acc. to the OSSD.

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	00_{16}							
1	monitor state							

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
2	OSSD 1 state							
3	OSSD 2 state							
4	number of *not green*[1] devices, OSSD 1							
5	number of *not green*[2] devices, OSSD 2							
6	device index 32, OSSD 1							
7	device colour 32, OSSD 1							
8	device index 33, OSSD 1							
…	…							
133	device colour 95, OSSD 1							
134	device index 32, OSSD 2							
…	…							
261	device colour 95, OSSD 2							

1. The maximal value is 7, higher values are limited to 7.
2. The maximal value is 7, higher values are limited to 7.

4.3.3.9 Command 8: Read „Safety Monitor unsorted by OSSD“

With this command the safety monitor is being read *unsorted* by OSSD

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	00_{16}							
1	monitor state							
2	OSSD 1 state							
3	OSSD 2 state							
4	number of *not green*[1] devices, OSSD 1							
5	—							
6	device index 32							
7	device colour 32							
8	device index 33							
…	…							
133	device colour 95							
134	device index 32							
135	assignment of the device 32 to the OSSD							
…	…							
261	assignment of the device 95 to the OSSD							

1. The maximal value is 7, higher values are limited to 7.

Following assignment is possible:

00_{16}: Preprocessing

01_{16}: OSSD 1

02_{16}: OSSD 2

03_{16}: OSSD 1+2

80_{16}: Device doesn‘t exist

Note: See <chapter 6 "Appendix: Code description", on page 94> for a description of the codes used for monitor state, OSSD state, device colours and assignments to OSSDs and the "Safety-at-Work" monitor documentation.

4.3.3.10 Command 9: „reserved“

This command is reserved for future developments.

4.3.3.11 Commands 10 - 13: Safety unit diagnosis and shutdown-history

Note	**!!!** The commands **10 ... 15** are available only with safety monitors (external and integrated) in the version 2 and higher.

In the case of a second generation safety monitor, the shutdown-history can be read additionally to the safety unit diagnosis.

If an OSSD abandons the state *green*, the states of all devices are hold on at this moment. Therefore it is possible to detect the cause for the turning-off läter.

If there has been no turning-off of the related OSSD since the start, all devices are *grey*.

If the ACYC_TRANS slave address is "0", the internal monitor is activated, otherwise the external one.

Command	**Description**
10	safety monitor diagnosis
11	shutdown-history, separate for each release circuit
12	safety monitor diagnosis, but the module allocation has been considered
13	shutdown-history, but the module allocation has been considered

Request buffer (only for commands 10 + 11)								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	OSSD: 0=OSSD 1; 1=OSSD 2							

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	reserved 00_{16}							
1	monitor state[1]							
2	OSSD 1 state[2]							
3	OSSD 2 state							
4	number of *not green*[3] devices							
5	—							
6	device index 32							
7	device[4] colour 32							
8	device index 33							
9	device colour 33							
...								
132	device index 95							

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
133	device colour 95							
134	device index 32							
135	assignment[5] of the device 32 to the OSSD							
...								
260	device index 95							
261	assignment of the device 95 to the OSSD							

1. For code description see table "Monitor state", page 94
2. By means of device colours it is possible to form an opinion about the state of the OSSDs (see table "Colour coding", page 94)
3. The maximal value is 7, higher values are limited to 7
4. By means of device colours it is possible to form an opinion about the state of the OSSDs (see table "Colour coding", page 94)
5. For assignment of the devices to the OSSD see table "Allocation", page 95

4.3.3.12 Command 14: "Diagnosis / shutdown-history"

With this command the "diagnosis / shutdown-history" can be read separate for each release circuit.

Request buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	list selection (0=current diagnosis; >0=diagnosis by OSSD turning-off (past events memory)							
2	number of the OSSD (0=preprocessing)							
3	Fdiagnosis format (0=comlete diagnosis; 1=sorted according to the diagnosis index)							

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	response type (0=device colour; >0=reserved)							
2	monitor state; byte 1 (see table "Monitor state", page 29 for description)							
3	OSSD type (0=internal OSSD; 1=peripheral OSSD)							
4	OSSD info - OSSD number, if internal OSSD (0=preprocessing, 1=OSSD 1, 2=OSSD 2); - Slave address, if peripheral OSSD (address 0 – 63, bit 7 points at the AS-i circuit that is allocated to the AS-i slave; 0=circuit 1, 1=circuit 2)							
5	OSSD state (Bit 0-bit 3 colour of the OSSD; bit 4-bit 7 reserved)							
6	colour device 0 (description see table "State and colour coding", page 29).							
...	...							
261	colour device 255							

Monitor state	
Bit [4 ... 0]	
0 ... 31	reserved
Bit 5	configuration mode
0	monitor *not* in configuration mode
1	monitor in configuration mode
Bit 6	protected mode
0	monitor *not* in protected mode
1	monitor in protected mode
Bit 7	device error
0	no device error
1	fatal device error, RESET or device exchange required

State and colour coding	
Bit [2 ... 0]	State and/or colour coding
0	green permanent light
1	green flashing
2	yellow permanent light
3	yellow flashing
4	red permanent light
5	red flashing
6	grey and/or off
7	reserved
Bit [4 ... 3]	
0 ... 3	reserved
Bit 5	modification
0	*no* device modification by "switch off"
1	device modification by "switch off"
Bit 6	existence
0	device exists
1	device doesn‘t exist
Bit 7	Usage
0	device is used in this OSSD
1	device is *not* used in this OSSD

4.3.3.13 Command 15: "Safety Status"

With this command the status of safety monitors (external and integrated) in the version 2 can be read.

Request buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	reserved 00_{16}							

Response buffer								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	reserved 00_{16}							
2	OSSD 1 state[1]							
3	OSSD 2 state[2]							
...	...							
n	OSSD n-1 state[3]							

1. see table "Coding of status byte", page 30
2. see table "Coding of status byte", page 30
3. see table "Coding of status byte", page 30

Coding of status byte	
Bit [0 ... 2]	state and/or colour
0	green permanent light
1	green flashing
2	yellow permanent light
3	yellow flashing
4	red permanent light
5	red flashing
6	grey and/or off
7	reserved
Bit [6]	state and/or colour
0	*no* device flashes yellow in this OSSD
1	at least *one* device flashes yellow in this OSSD
Bit [7]	state and/or colour
0	*no* device flashes red in this OSSD
1	at least *one* device flashes red in this OSSD

4.4 AS-i Diagnosis

4.4.1 Overview of the commands

Values for command

see page	*Command*	*Value*	*Meaning*	*Req Len*	*Res Len*
page 31	GET_LISTS	30_{16}	Get LDS, LAS, LPS, Flags	2	29
page 33	GET_FLAGS	47_{16}	Get_Flags	2	5
page 34	GET_DELTA	57_{16}	Get list of config. diff.	2	10
page 35	GET_LCS	60_{16}	Get LCS	2	10
page 35	GET_LAS	45_{16}	Get_LAS	2	10
page 36	GET_LDS	46_{16}	Get_LDS	2	10
page 37	GET_LPF	$3E_{16}$	Get_LPF	2	10
page 37	GET_LOS	61_{16}	GET_LOS	2	10
page 38	SET_LOS	62_{16}	SET_LOS	10	2
page 39	GET_TECA	63_{16}	Get transm.err.counters	2	34
page 40	GET_TECB	64_{16}	Get transm.err.counters	2	34
page 41	GET_TEC_X	66_{16}	Get transm.err.counters	4	≥3
page 41	READ_FAULT_DETECTOR[1]	10_{16}	Read Fault Detector	2	4
page 42	READ_DUPLICATE_ADDR[2]	11_{16}	Read List of Duplicate Addresses	2	10

1. The command READ_FAULT_DETECTOR is valid only for the use with masters which support this function. Please see the user manual of the master for further information.
2. The command READ_DUPLICATE_ADDR is valid only for the use with masters which support this function. Please see the user manual of the master for further information.

4.4.2 Get Lists and Flags (Get_LPS, Get_LAS, Get_LDS, Get_Flags) (GET_LISTS)

With this call, the following entries of the AS-i Master can be read:

- The list of active AS-i slaves (**LAS**)
- The list of detected AS-i slaves (**LDS**)
- The list of projected AS-i slaves (**LPS**)
- The flags according to the AS-i slave specification

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	30_{16}							
2	T	O	circuit					

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	30_{16}							
2	T	result						
3	7A	6A	5A	4A	3A	2A	1A	0A

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
…	LAS							
10	31B	30B	29B	28B	27B	26B	25B	24B
11	7A	6A	5A	4A	3A	2A	1A	0A
…	LDS							
18	31B	30B	29B	28B	27B	26B	25B	24B
19	7A	6A	5A	4A	3A	2A	1A	0A
…	LPS							
26	31B	30B	29B	28B	27B	26B	25B	24B
27	–							Pok
28	OR	APF	NA	CA	AAv	AAs	S0	Cok
29	–					AAe	OL	DX

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	30_{16}							
2	T	result						
3	0A	1A	2A	3A	4A	5A	6A	7A
…	LAS							
10	24B	25B	26B	27B	28B	29B	30B	31B
11	0A	1A	2A	3A	4A	5A	6A	7A
…	LDS							
18	24B	25B	26B	27B	28B	29B	30B	31B
19	0A	1A	2A	3A	4A	5A	6A	7A
…	LPS							
26	24B	25B	26B	27B	28B	29B	30B	31B
27	–							Pok
28	OR	APF	NA	CA	AAv	AAs	S0	Cok
29	–					AAe	OL	DX

Pok Periphery_Ok
S0 LDS.0
AAs Auto_Address_Assign
AAv Auto_Address_Available
CA Configuration_Active
NA Normal_Operation_Active
APF APF
OR Offline_Ready
Cok Config_Ok
AAe Auto_Address_Enable
OL Offline
DX Data_Exchange_Active

4.4.3 Get Flags (GET_FLAGS)

With this call, the flags according to the AS-i slave specification can be read.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	47_{16}							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	47_{16}							
2	T	response						
3								Pok
4	OR	APF	NA	CA	AAv	AAs	S0	Cok
5	–					AAe	OL	DX

Pok Periphery_OK
This flag is set when no AS-i slave is signaling a peripheral fault.

S0 LDS.0
This flag is set when an AS-i slave with address 0 exists.

AAs Auto_Address_Assign
This flag is being set when the automatic address programming is possible (in other words, AUTO_ADDR_ENABLE = 1; no "incorrect" slave connected to the AS-i).

AAv Auto_Address_Available
This flag is set when the automatic address programming can be executed, exactly one AS-i slave is currently out of operation.

CA Configuration_Active
The flag is set in configuration mode and reset in protected mode.

NA Normal_Operation_Active
This flag is set when the AS-i master is in normal operation.

APF AS-i Power Fail
This flag is set when the voltage on the AS-i cable is too low.

OR Offline_Ready
The flag is set when the offline phase is active.

Cok Config_O
This flag is set when the desired (configured) and actual configuration match.

AAe Auto_Address_Enable
This flag indicates whether the automatic address programming is enabled (bit = 1) or disabled (bit = 0) by the user.

OL Offline
This flag is set when the mode should be changed to OFFLINE or when this mode has already been reached.

DX Data_Exchange_Active
If the "Data_Exchange_Active" flag is set, the data exchange between AS-i master and slaves is available in the data exchange phase. If this bit is not set the data exchange is not available. The read ID telegrams are transmitted to the slave.
The bit is set if the AS-i master enters the offline phase.

4.4.4 Get Delta List (GET_DELTA)

The delta list contains the list of slave addresses with configuration errors.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	57_{16}							
2	T	0	circuit					

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	57_{16}							
2	T	result						
3	7A	6A	5A	4A	3A	2A	1A	–
…	…							
10	31B	30B	29B	28B	27B	26B	25B	24B

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	57_{16}							
2	T	result						
3	0	1A	2A	3A	4A	5A	6A	7A
…	…							
10	24B	25B	26B	27B	28B	29B	30B	31B

4.4.5 Get list of corrupted Slaves (GET_LCS and GET_LCS_R6 (6CH))

The call GET_LCS_R6 (6CH) differs to the call GET_LCS in the half long LCS list. With the bit 2^5 is selected if the upper (=1) or lower (=0) part of the LCS is read. Read first with 2^5 in order to create a local copy of the LCS. Reading with bit 2^5=1 transmits the upper part of the copy.

With the call GET_LCS, the List of Corrupted Slaves (*LCS*) can be read.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	60_{16}							
2	T	O	circuit					

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	60_{16}							
2	T	result						
3	7A	6A	5A	4A	3A	2A	1A	0A
…	…							
10	31B	30B	29B	28B	27B	26B	25B	24B

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	60_{16}							
2	T	result						
3	0A	1A	2A	3A	4A	5A	6A	7A
…	…							
10	24B	25B	26B	27B	28B	29B	30B	31B

4.4.6 Get list of activated Slaves (GET_LAS)

With this call, the list of activated slaves (*LAS*) can be read.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	45_{16}							
2	T	O	circuit					

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	45_{16}							
2	T	result						
3	7A	6A	5A	4A	3A	2A	1A	0A
…	…							
10	31B	30B	29B	28B	27B	26B	25B	24B

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	45_{16}							
2	T	result						
3	0A	1A	2A	3A	4A	5A	6A	7A
…	…							
10	24B	25B	26B	27B	28B	29B	30B	31B

4.4.7 Get list of detected AS-i Slaves (GET_LDS)

With this call, the list of detected AS-i slaves (*LDS*) can be read.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	46_{16}							
2	T	O	circuit					

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	46_{16}							
2	T	result						
3	7A	6A	5A	4A	3A	2A	1A	0A
…	…							
10	31B	30B	29B	28B	27B	26B	25B	24B

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	46_{16}							
2	T	result						
3	0A	1A	2A	3A	4A	5A	6A	7A
…	…							
10	24B	25B	26B	27B	28B	29B	30B	31B

4.4.8 Get list of peripheral faults (GET_LPF)

With this call, the list of peripheral faults (*LPF*) signaled by the AS-i slaves is read out from the AS-i master. The LPF is updated cyclically by the AS-i master. If and when an AS-i slave signals faults of the attached peripherals (for example broken wire) can be found in the description of the AS-i slave.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$3E_{16}$							
2	T	O	circuit					

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$3E_{16}$							
2	T	result						
3	7A	6A	5A	4A	3A	2A	1A	0A
…	…							
10	31B	30B	29B	28B	27B	26B	25B	24B

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$3E_{16}$							
2	T	result						
3	0A	1A	2A	3A	4A	5A	6A	7A
…	…							
10	24B	25B	26B	27B	28B	29B	30B	31B

4.4.9 Get list of offline Slaves (GET_LOS)

With this call, the list of slaves causing the offline phase when a configuration error occurs in being read out (List of Offline Slaves, *LOS*).

The user can choose the reaction of the master when a configuration error occurs. The master can be switched off line when an important slave causes a configuration error; less important slaves can send an error to the host, AS-i however will not be switched offline.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	61_{16}							
2	T	O	circuit					

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	61_{16}							
2	T	result						
3	7A	6A	5A	4A	3A	2A	1A	0A
…	…							
10	31B	30B	29B	28B	27B	26B	25B	24B

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	61_{16}							
2	T	result						
3	0A	1A	2A	3A	4A	5A	6A	7A
…	…							
10	24B	25B	26B	27B	28B	29B	30B	31B

4.4.10 Set list of offline Slaves (SET_LOS and SET_LOS_R6 (6Dh))

The call **SET_LOS_R6 (6D$_{16}$)** differs to the call GET_LOS in the half long LOS list. With the bit 2^5 is selected if the upper (=1) or lower (=0) part of the LOS is written.

With this call, the list of slaves causing the offline phase when a configuration error occurs in being defined (List of Offline Slaves, *LOS*).

The user can choose the reaction of the master when a configuration error occurs. The master can be switched offline when an important slave causes a configuration error; less important slaves can send an error to the host, AS-i however will not be switched offline.

Request (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	62_{16}							
2	T	O	circuit					
3	7A	6A	5A	4A	3A	2A	1A	0A
…	…							
10	31B	30B	29B	28B	27B	26B	25B	24B

Request (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	62_{16}							
2	T	1	circuit					
3	0A	1A	2A	3A	4A	5A	6A	7A
…	…							
10	24B	25B	26B	27B	28B	29B	30B	31B

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	62_{16}							
2	T	result						

4.4.11 Get transm.err.counters (GET_TECA)

Note	In order to get the real number of transcription errors, multiply the value with 2

With this call the error counters of all single slaves/A-slaves can be read (see chapter: Advanced Diagnostics for AS-i Masters in the manual of your AS-i Master).

With every reading out of the counts, the error counters will be restarted.

The counts are being read out via the corresponding host interface and will be deleted with every read access. The counter´s value is limited to 254. 255 will cause a counter overflow.

The counts could be independent of the counters, which are displayed in the display of the gateway.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	63_{16}							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	63_{16}							
2	T	result						
3	APF							
4	slave 1A							
…	…							
34	slave 31A							

4.4.12 Get transm.err.counters (GET_TECB)

Note	In order to get the real number of transcription errors, multiply the value with 2

With this call, the counts of the error counters for B-slaves are being read out (see chapter: Advanced Diagnostics for AS-i Masters in the manual of your AS-i Master).

With every reading out of the counts, the error counters will be restarted.

The counts are being read out via the corresponding host interface and will be deleted with every read access. The counter´s value is limited to 254. 255 will cause a counter overflow.

The counts could be independent of the counters, which are displayed in the display of the gateway.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	64_{16}							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	64_{16}							
2	T	result						
3	APF							
4	slave 1B							
…	…							
34	slave 31B							

4.4.13 Get transm.err.counters (GET_TEC_X)

Beginning with a definite slave address, the counts of the n error counters are being read out with this call.

With every reading out the counts, the error counters will be restarted.

The counts are being read out via the corresponding host interface and will be deleted with every read access. The counter´s value is limited to 254. 255 will cause a counter overflow.

The counts could be independent of the counters, which are displayed in the display of the gateway.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	66_{16}							
2	T	–	circuit					
3	1. slave address							
4	number of counters							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	66_{16}							
2	T	result						
3	counter 1							
…	…							
n	counter n - 2							

4.4.14 Read fault detector (READ_FAULT_DETECTOR)

Attention	**!!!** The command READ_FAULT_DETECTOR is valid only for the use with masters which support this function. Please see the user manual of the master for further information.

With this call all informations of the AS-i detector are read out. In the first byte are stored the values transferred in the moment, in the second all values since the last deleting. By it is possible to recognize immediate, no more existing before messages also. The second byte is deleted by reading.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	10_{16}							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	10_{16}							
2	T	result						
3	DA	ST	US	ES	24 V	reserved		
4	DA	ST	US	ES	24 V	reserved		

DA duplicate address

ST noise

US over voltage

ES earth fault

24 V failure of the redundant 24V

4.4.15 Read list of duplicate addresses (READ_DUPLICATE_ADDR)

Attention

!!!

The command READ_DUPLICATE_ADDR is valid only for the use with masters which support this function.
Please see the user manual of the master for further information.

With this call the list of slaves with duplicate addresses (the assignment of one address to two slaves) is read out.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	11_{16}							
2	T	O	circuit					

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	11_{16}							
2	T	result						
3	7A	6A	5A	4A	3A	2A	1A	0A
…	…							
10	31B	30B	29B	28B	27B	26B	25B	24B

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	11_{16}							
2	T	result						
3	0A	1A	2A	3A	4A	5A	6A	7A
…	…							
10	24B	25B	26B	27B	28B	29B	30B	31B

Note

Further diagnosis functions for "Safety at Work" and for availability (resp. for warnings) of integrated sensors are detailed explained in the chapter "Functional Profiles" (chapter 4.7).

4.5 Configuration of AS-i Master

4.5.1 Overview of the commands

Values for command

see page	*Command*	*Value*	*Meaning*	*Req Len*	*Res Len*
page 43	SET_OP_MODE	$0C_{16}$	Set_Operation_Mode	3	2
page 44	STORE_CDI	07_{16}	Store_Actual_Configuration	2	2
page 45	READ_CDI	28_{16}	Read_Actual_Configuration	3	4
page 45	SET_PCD	25_{16}	Set_Permanent_Config	5	2
page 45	GET_PCD	26_{16}	Get_Permanent_Config	3	4
page 47	SET_LPS	29_{16}	SET_LPS	11	2
page 48	GET_LPS	44_{16}	Get_LPS	2	10
page 48	STORE_PI	04_{16}	Store_Actual_Parameter	2	2
page 49	WRITE_P	02_{16}	Write_Parameter	4	3
page 49	READ_PI	03_{16}	Read_Parameter	3	3
page 50	SET_PP	43_{16}	Set_Permanent_Parameter	4	2
page 50	GET_PP	01_{16}	Get_Permanent_Parameter	3	3
page 51	SET_AAE	$0B_{16}$	Set_Auto_Address_Enable	3	2
page 53	SLAVE_ADDR	$0D_{16}$	Change_Slave_Address	4	2
page 52	WRITE_XID1	$3F_{16}$	Write_Extended_ID-Code_1	3	2

4.5.2 Set operation mode (SET_OP_MODE: Set_Operation_Mode)

This call switches between configuration mode and protected mode. In protected mode, only AS-i slaves entered in the LPS and whose expected and actual configurations match, are being activated.

In other words: The slaves are being activated if the I/O configuration and the ID codes of the detected AS-i slaves are identical to the configured values.

In configuration mode, all detected AS-i slaves (except for AS-i slave "0") are activated. This also applies to AS-i slaves for which there are differences between the expected and actual configuration.

The “OPERATION MODE” bit is stored permanently; in other words, it is retained after a cold/warm restart.

When you change from configuration mode to protected mode, the AS-i master will do a warm restart (change to the offline phase followed by a change to the online mode).

Note	If an AS-i Slave with address "0" is entered in the LDS, the AS-i Master cannot change from configuration mode to protected mode.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$0C_{16}$							
2	T	–	circuit					
3	operation mode							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$0C_{16}$							
2	T	result						

Meaning of bit operation mode:
- 0 = protected mode
- 1 = configuration mode

4.5.3 Store actual configuration (STORE_CDI)

With this call, the (actual) configuration data (I/O configuration, ID code, extended ID1 code and extended ID2 code) of all AS-i slaves are stored permanently in the EEPROM as the (expected) configuration data. The list of activated AS-i slaves (*LAS*) is adopted in the list of permanent AS-i slaves (*LPS*).

When this command is executed, the AS-i master changes to the offline phase and then changes back to the normal mode (warm restart on the AS-i master).

This command can only be executed in the configuration mode.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	07_{16}							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	07_{16}							
2	T	result						

4.5.4 Read actual configuration (READ_CDI)

With this call, the following configuration data of an addressed AS-i slave obtained by the AS-i master on the AS-i are read.

- I/O configuration
- ID code
- Extended ID1 code
- Extended ID2 code

The configuration data are specified by the manufacturer of the AS-i slave.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	28_{16}							
2	T	–	circuit					
3	–		B	slave address				

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	28_{16}							
2	T	result						
3	xID2				xID1			
4	ID				I0			

Meaning of bit B:

B = 0 Single AS-i slave or A-slave
B = 1 B-slave

4.5.5 Set permanent configuration (SET_PCD)

This call sets the following configuration data for the addressed AS-i slave:

- I/O configuration
- ID code
- Extended ID1 code
- Extended ID2 code

The configuration data are stored permanently on the EEPROM of the AS-i Master and are used as the expected configuration by the AS-i master in the protected mode. The configuration data are specified by the manufacturer of the AS-i slave.

If the addressed AS-i slave does not support an extended ID code 1/2, the value F_{hex} must be specified.

When this command is executed, the AS-i master changes to the offline phase and then changes back to the normal mode (warm restart).

This command can only be executed in the configuration mode.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	25_{16}							
2	T	–	circuit					
3	–		B	slave address				
4	xID2				xID1			
5	ID				I0			

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	25_{16}							
2	T	result						

Meaning of bit B:

B = 0Single AS-i slave or A-slave

B = 1B-slave

4.5.6 Get extended permanent configuration (GET_PCD)

This call reads the following configuration data (configured data) of an addressed AS-i slave stored on the EEPROM of the AS-i master:

- I/O configuration
- ID code
- Extended ID1 code
- Extended ID2 code

The configuration data are specified by the manufacturer of the slave.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	26_{16}							
2	T	–	circuit					
3	–		B	slave address				

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	26_{16}							
2	T	result						
3	xID2				xID1			
4	ID				I0			

Meaning of bit B:

B = 0   Single AS-i slave or A-slave
B = 1   B-slave

4.5.7 Set list of projected slaves (SET_LPS and SET_LPS_R6 (6Bh))

The command **SET_LPS_R6 (6Bh)** differs from the command **SET-LPs** in:

- no empty byte (3)
- half so long LPS list

With the bit 2^5 is selected if the upper (=1) or lower (=0) part of the LCS is read.

With this call, the list of configured AS-i slaves is transferred for permanent storage in the EEPROM of the master.

When this command is executed, the AS-i master changes to the offline phase and then changes back to the normal mode (warm restart).

This command can only be executed in the configuration mode.

Request (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	29_{16}							
2	T	0	circuit					
3	00_{16}							
4	7A	6A	5A	4A	3A	2A	1A	–
…	…							
11	31B	30B	29B	28B	27B	26B	25B	24B

Request (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	29_{16}							
2	T	1	circuit					
3	00_{16}							
4	–	1A	2A	3A	4A	5A	6A	7A
…	…							
11	24B	25B	26B	27B	28B	29B	30B	31B

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	29_{16}							
2	T	result						

4.5.8 Get list of projected slaves (GET_LPS)

With this call, the list of projected AS-i slaves (*LPS*) is read out of the AS-i Master.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	44_{16}							
2	T	O	circuit					

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	44_{16}							
2	T	result						
3	7A	6A	5A	4A	3A	2A	1A	0A
…	…							
10	31B	30B	29B	28B	27B	26B	25B	24B

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	44_{16}							
2	T	result						
3	0A	1A	2A	3A	4A	5A	6A	7A
…	…							
10	24B	25B	26B	27B	28B	29B	30B	31B

4.5.9 Store actual parameters (STORE_PI)

With this call, the configured parameters stored on the EEPROM are overwritten with the current, permanently stored (actual) parameters; in other words, the current parameters of all AS-i slaves are stored.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	04_{16}							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	04_{16}							
2	T	result						

4.5.10 Write parameter (WRITE_P)

The AS-i slave parameter value transferred with the command is passed on to the addressed AS-i slave.

The parameter is stored in the AS-i Master only temporarily and is not stored as a configured parameter in the EEPROM!

The AS-i slave transfers its current parameter value in the response (parameter echo). This can deviate from the value that has just been written according to the AS-i master specification.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	02_{16}							
2	T	–	circuit					
3	–		B	slave address				
4	–				parameter			

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	02_{16}							
2	T	result						
3	–				slave response			

Meaning of bit B:

B = 0 Single AS-i slave or A-slave

B = 1 B-slave

4.5.11 Read parameter (READ_PI: Read_Parameter)

This call returns the current parameter value (actual parameter) of an AS-i slave sent by the AS-i Master. This value must not be confused with the parameter echo that is supplied by the AS-i slave as a response to the write_p job.

This command can not be used for a directly reading of an AS-i parameter out of an AS-i slave.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	03_{16}							
2	T	–	circuit					
3	–		B	slave address				

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	03_{16}							
2	T	result						
3	–				PI			

Meaning of bit B:

B = 0 Single AS-i slave or A-slave
B = 1 B-slave

4.5.12 Set permanent parameter (SET_PP)

With this call, a parameter value for the specified AS-i slave is configured. The value is stored permanently in the EEPROM of the gateway.

The configured parameter value is transferred only when the AS-i slave is activated after turning on the power supply on the AS-i Master.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	43_{16}							
2	T	–	circuit					
3	–		B	slave address				
4	–				PP			

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	43_{16}							
2	T	result						

4.5.13 Get permanent parameter (GET_PP)

With this call, a slave-specific parameter value stored on the EEPROM of the AS-i Master is read.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	01_{16}							
2	T	–	circuit					
3	–		B	slave address				

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	01_{16}							
2	T	result						
3	–				PP			

Meaning of bit B:

B = 0 Single AS-i slave or A-slave
B = 1 B-slave

4.5.14 Set auto address enable (SET_AAE)

This call can enable or disable the "automatic address programming" function.

The AUTO_ADDR_ENABLE bit is stored permanently; in other words, it is retained after a warm/hot restart on the AS-i master.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$0B_{16}$							
2	T	–	circuit					
3	Auto_Address_Enable							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$0B_{16}$							
2	T	result						

4.5.15 Change slave address (SLAVE_ADDR)

With this call, the AS-i address of an AS-i slave can be modified.

This call is mainly used to add a new AS-i slave with the default address "0" to the AS-Interface. In this case, the address is changed from "AS-i slave address old" = 0 to "AS-i slave address new”.

This change can only be made when the following conditions are fulfilled:

1. An AS-i slave with “AS-ii slave address old” exists.
2. If the old AS-i slave address is not equal to 0, an AS-i slave with address "0" cannot be connected at the same time.
3. The "AS-i slave address new" must have a valid value.
4. An AS-i slave with "AS-i slave address new" must not exist.

Note

When the AS-i slave address is changed, the AS-i slave is not reset, in other words, the output data of the AS-i slave are retained until new data are received at the new address.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$0D_{16}$							
2	T	–	circuit					
3	–		B	source address				
4	–		B	target address				

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$0D_{16}$							
2	T	result						

Meaning of bit B:

B = 0 Single AS-i slave or A-slave

B = 1 B-slave

4.5.16 Write AS-i slave extended ID1 (WRITE_XID1)

With this call, the extended ID1 code of an AS-i slave with address "0" can be written directly via the AS-i cable. The call is intended for diagnostic purposes and is not required in the normal master mode.

The AS-i master passes the extended ID1 code on to the AS-i slave without any plausibility check.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$3F_{16}$							
2	T	–	circuit					
3	–				xID1			

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$3F_{16}$							
2	T	result						

4.6 Other commands

4.6.1 Overview of the commands

Values for command

	see page	Command	Value	Meaning	Req Len	Res Len
	page 53	IDLE	00_{16}	No request	2	2
	page 54	READ_IDI	41_{16}	Read IDI	2	36
	page 54	WRITE_ODI	42_{16}	Write ODI	34	2
	page 55	READ_ODI	56_{16}	Read ODI	2	34
	page 55	SET_OFFLINE	$0A_{16}$	Set_Off-Line_Mode	3	2
	page 56	SET_DATA_EX	48_{16}	Set_Data_Exchange_Active	3	2
!	page 56	REWRITE_DPRAM[1]	78_{16}	Rewrite DPRAM	3	3
	page 56	BUTTONS	75_{16}	Disable Pushbuttons	3	2
	page 57	FP_PARAM	$7D_{16}$	„Functional Profile“ Param.	≥3	≥2
	page 58	FP_DATA	$7E_{16}$	„Functional Profile“ Data	≥3	≥2
!	page 58	EXT_DIAG[2]	71_{16}	ExtDiag generation	6	2
!	page 59	RD_EXT_DIAG[3]	$7B_{16}$	Read ExtDiag Settings	2	7
	page 60	INVERTER	$7C_{16}$	Configure Inverter Slaves	12	4
	page 60	MB_OP_CTRL_WR_FLAGS	85_{16}	Write Flags	≥5	2
	page 61	MB_OP_CTRL_RD_FLAGS	86_{16}	Read Flags	4	≥3
	page 61	RD_MFK_PARAM	59_{16}	Read SEW MFK21 Parameter	6	≥3

1. The command REWRITE_DPRAM is valid only for the use with ***AS-i 3.0 Module OEM Master***
2. The command EXT_DIAG is valid only for the use with ***AS-i 3.0 PROFIBUS Gateways***
3. The command RD_EXT_DIAG is valid only for the use with ***AS-i 3.0 PROFIBUS Gateways***

4.6.2 IDLE

When the value of "command" is zero, no request will be fulfilled.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	00_{16}							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	00_{16}							
2	T	result						

4.6.3 Read input data image (READ_IDI)

With this call, the input data values of all AS-i slaves are read out of the AS-i Master in addition to the cyclic data exchange. Though the command READ_IDI transmits all execution control flags (byte 3 and byte 4).

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	41_{16}							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	41_{16}							
2	T	result						
3	–							Pok
4	OR	APF	NA	CA	AAv	AAs	s0	Cok
5	–				slave 1A			
6	slave 2A				slave 3A			
…	…							
36	slave 30B				slave 31B			

Pok Periphery_Ok
S0 LDS.0
AAs Auto_Address_Assign
AAv Auto_Address_Available
CA Configuration_Active
NA Normal_Operation_Active
APF APF
OR Offline_Ready
Cok Config_Ok

4.6.4 Write output data image (WRITE_ODI)

With this call the output data values of all AS-i slaves are written in addition to the cyclic data exchange.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	42_{16}							
2	T	–	circuit					
3	–				slave 1A			
4	slave 2A				slave 3A			
…	…							
34	slave 30B				slave 31B			

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	42_{16}							
2	T	result						

4.6.5 Read output data image (READ_ODI)

With this call, the output data values of all AS-i slaves is being read out of the AS-i Master.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	56_{16}							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	56_{16}							
2	T	result						
3	–				slave 1A			
	slave 2A				slave 3A			
…	…							
34	slave 30B				slave 31B			

4.6.6 Set offline mode (SET_OFFLINE)

This call switches between online and offline mode.

The online mode is the normal operating state for the AS-i master. The following jobs are processed cyclically:

- During the data exchange phase, the fields of the output data are transferred to the slave outputs for all AS-i slaves in the LAS. The addressed AS-i slaves submit the values of the slave inputs to the master when the transfer was free of errors.
- This is followed by the inclusion phase in which existing AS-i slaves are searched and newly added AS-i slaves are entered in the LDS or LAS.
- In the management phase, jobs by the user such as writing parameters are executed.

In the offline mode, the AS-i Master processes jobs by the user only. (Jobs that involve the immediate addressing of an AS-i slave are rejected with an error). There is no cyclic data exchange with the AS-i slaves.

When offline, the AS-i circuit is in a safe state.

The OFFLINE = TRUE bit is not permanently stored; in other words, following a cold/warm restart, the AS-i Master is once again in the online mode.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$0A_{16}$							
2	T	–	circuit					
3	Off-Line							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$0A_{16}$							
2	T	result						

The master changes to the offline phase, if there is a 1 written in byte 3.

The master will change to online mode if there is a 0 written in byte 3.

4.6.7 Release data exchange (SET_DATA_EX)

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	48_{16}							
2	T	–	circuit					
3	Data_Exchange_Active							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	48_{16}							
2	T	result						

4.6.8 Rewrite DPRAM (REWRITE_DPRAM)

Attention

!!!

The command REWRITE_DPRAM is valid only for the use with AS-i 3.0 Module OEM Master.

This command is used for the rewriting of the DPRAM.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	78_{16}							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	78_{16}							
2	T	result						

4.6.9 BUTTONS

With this call, the use of the buttons can be enabled/disabled.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	75_{16}							
2	T	–	circuit					
3	Buttons disabled							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	75_{16}							
2	T	result						

4.6.10 FP_PARAM

This command is used for parametrization of "functional profiles".

The content of the request and response bytes depends on the called function (see chapter 4.7 Functional Profiles, page 62).

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7D_{16}$							
2	T	–	circuit					
3	function							
4	request byte 1							
…	…							
n	request byte n-3							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7D_{16}$							
2	T	result						
3	response byte 1							
…	…							
n	response byte n-2							

4.6.11 FP_DATA

This command is used for the data exchange with "functional profiles".

The content of the request and response bytes depends on the called function (see chapter 4.7 Functional Profiles, page 62).

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	–	circuit					
3	function							
4	request byte 1							
…	…							
n	request byte n-3							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	result						
3	reponse byte 1							
…	…							
n	response byte n-2							

4.6.12 EXT_DIAG

Attention

!!!

The command EXT_DIAG is valid only for the use with AS-i 3.0 PROFIBUS Gateways

With this call, the conditions when to set the ExtDiag bit can be selected.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	71_{16}							
2	T	–	circuit					
3	CF							
4	APF							
5	PF							
6	CS							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	71_{16}							
2	T	result						

CF ExtDiag is set, if ConfigError ≡ 1
APF ExtDiag is set, if APF ≡ 1
PF ExtDiag is set, if PeripheryFault ≡ 1
CS ExtDiag is set, if LCS is not empty

4.6.13 RD_EXT_DIAG

Attention

!!!

The command RD_EXT_DIAG is valid only for the use with AS-i 3.0 PROFIBUS Gateways

With this call, the conditions when the ExtDiag bit is set can be read.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7B_{16}$							
2	T	–	circuit					

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	75_{16}							
2	T	result						
3	CF							
4	APF							
5	PF							
6	CS							
7	FD							

CF ExtDiag is set by ConfigError ≡ 1
APF ExtDiag is set by APF ≡ 1
PF ExtDiag is set by PeripheryFault ≡ 1
CS ExtDiag is set, if LCS is not empty
FD Diagnosis will be updated only if this is dictated by the PROFIBUS norm. Diagnosis date are not up to date when in doubt.

4.6.14 Inverter

With this call, an AS-i slave for frequency inverters is switched from cyclical mode to the transmission mode of four 16-bit values, in order to operate again with the selected AS-i destination parameter.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7C_{16}$							
2	T	–	circuit					
3	slave address							
4	destination parameter							
5	value 1, high byte							
6	value 1, low byte							
7	value 2, high byte							
8	value 2, low byte							
9	value 3, high byte							
10	value 3, low byte							
11	value 4, high byte							
12	value 4, low byte							

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7C_{16}$							
2	T	result						

4.6.15 Write Flag

Use this command to write the flag of a control program.

The control program of devices with control functions takes on data from the PB interface.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	85_{16}							
2	T	–	circuit					
3	introductory address							
4	number n							
5	number 1							
...	...							
n	number n							

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	85_{16}							
2	T	result						

4.6.16 Read Flag

Use this command to read out the flags of a control program.

The control program of devices with control functions takes on data from the superior fieldbus interface.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	86_{16}							
2	T	–	circuit					
3	introductory address							
4	number n							

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	86_{16}							
2	T	result						
3	data 1							
...								
n	data n							

4.6.17 READ_MFK_PARAM

Use this command to read multiple commands of a SEW MFK21 slave.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	59_{16}							
2	T	–	circuit					
3	slave							
4	index high							
5	index low							
6	number (n)							

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	59_{16}							
2	T	result						
3	prm byte (index)							

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
4	prm byte (index+1)							
n+2	prm byte (index+n-1)							

4.7 Functional Profiles

4.7.1 "Safety at Work" List 1

Note	This function has been implemented only for reasons of the downwards compatibility. By AS-i 3.0 Masters, the state of the "safety input slaves" is specified on the image of the input data (0000 released).

4.7.1.1 Slave list with EcFlags

(Function: $00_{16)}$

List of "safety-directed input slaves" ("AS-i Safety at Work"), whose safety function is released.

Safety-directed input slaves have the profile S-7.B or S-0.B (IO = 0 or 7, ID = B, see chapter 4.5.4 Read actual configuration (READ_CDI), page 45: Read Actual Configuration).

The "Safety at Work" list 1 is a bit list which contains a bit for each possible slave address (1 - 31). This list is written in the bytes 5 until 8 in the response of the command of the command interface. Additionally, the reponse contains the ec-flags of the AS-i master in the bytes 3 and 4 (see chapter 4.4.3 Get Flags (GET_FLAGS), page 33: "Get Flags").

The bits of the "Safety at Work" list 1 are set if the safety function of the slave is activated (e.g. emergency button pressed). The bit is only set at security slaves when both contacts are released, otherwise the bits have the value 0. "Normal" (non-security) slaves also have the value 0.

Since the safety monitor is also being activated when a safety slave is missing or if the AS-i circuit is shut off (offline active), the ec-flags will also be transmitted. It is sufficient however to monitor the group error message Cok (configuration error). As long as no configuration error, the list of the "safety-directed input slaves" can be used.

Configured safety slaves which are not available, and available slaves sending a wrong coder order, will not be entered in this list.

With the bit "O", the sequence of the bits within the "Safety at Work" list 1 can be chosen.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	O	circuit					
3	00_{16}							

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	result						
3	–							Pok
4	OR	APF	NA	CA	AAv	AAs	S0	Cok
5	7	6	5	4	3	2	1	-
6	15	14	13	12	11	10	9	8
7	23	22	21	20	19	18	17	16
8	31	30	29	28	27	26	25	25

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	result						
3	–							Pok
4	OR	APF	NA	CA	AAv	AAs	S0	Cok
5	-	1	2	3	4	5	6	7
…	…							
8	24	25	26	27	28	29	30	31

Cok Config_Ok
S0 LDS.0
AAs Auto_Address_Assign
AAv Auto_Address_Available
CA Configuration_Active
NA Normal_Operation_Active
APF APF
OR Offline_Ready
Pok Periphery_Ok

Example for O ≡ 0:

Configuration OK,
periphery OK (no peripheral fault,
2 safety slaves with released safety function,
AS-i addresses 4 and 10

1 safety slave with unreleased safety function,
AS-inferface address 5.

Reponse: 7E 00 01 25 10 04 00 00

4.7.1.2 Slave list without EcFlags

(Function: $0D_{16}$)

There is a function $0D_{16}$ in addition to the function 00_{16} . The function $0D_{16}$ has no EcFlags in the response. The response falls short for 2 bytes.

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	O	circuit					
3	0Dh							

Response (by O ≡ 0)								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	response						
3	7	6	5	4	3	2	1	–
4	15	14	13	12	11	10	9	8
5	23	22	21	20	19	18	17	16
6	31	30	29	28	27	26	25	24

Response (by O ≡ 1)								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	response						
3	–	1	2	3	4	5	6	7
4	8	9	10	11	12	13	14	15
5	16	17	18	19	20	21	22	23
6	24	25	26	27	28	29	30	31

4.7.2 "Safety at Work" Monitor diagnosis

Function: 02_{16}

Since the "Safety at Work" monitor can generate more than 32 Byte diagnosis data, these must be read with several command interface calls. The byte 5 declares the start index in the field of the diagnosis data.

If the start index is 0, new data is fetched from the monitor. Otherwise, the function will respond out of the memory; the data can be read consistently.

4.7.3 Setting of the AS-i diagnosis

Note	The function ***unsorted diagnosis*** is available only with monitors in the version 2.0 and higher. The function ***sorted diagnosis*** is available with all monitors.

The setting of the AS-i diagnosis takes place in the window "*Information about monitor and bus"* of the *configuration software* ***asimon*** for the AS-i safety monitor.

- Call up the menu *Edit/Information about monitor and bus*

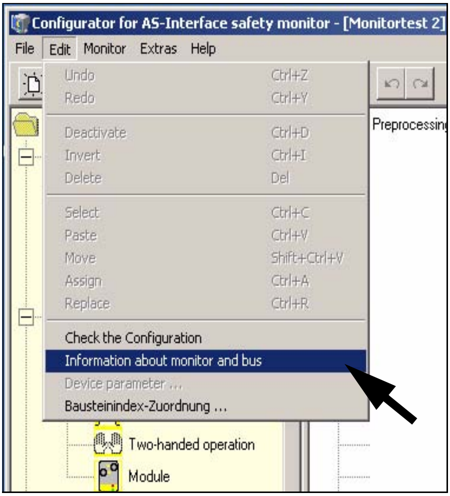


Fig. 1. Calling of Information about monitor and bus

- Set *the function range* in the window *Information about monitor and bus*

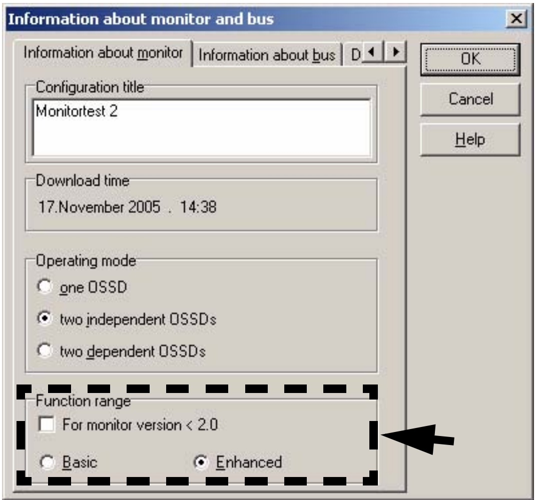


Fig. 2. Setting of function range

- Select in the window *Information about monitor and bus* the tab *Diagnosis/Service*
- Select within the range *Data selection* ***sorted*** (sorted by OSSD) or ***unsorted*** (all devices)

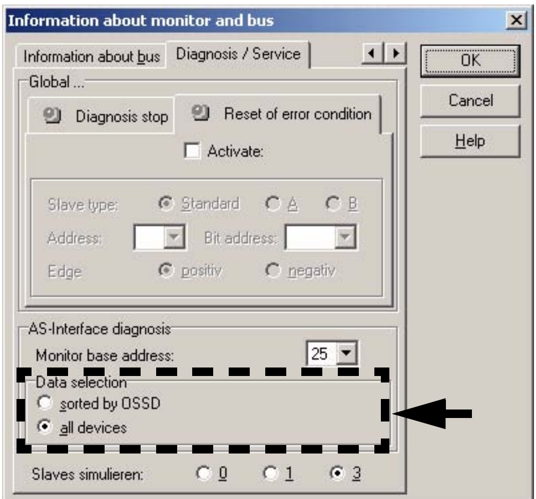


Fig. 3. Data selection (sorted/unsorted)

4.7.4 Enhanced diagnosis

Since the "Safety at Work“ monitor diagnosis is longer than the maximum size of the command interface, it must be read with several adjacent requests.

The byte 5 (‘index’) declares the start index in the array of diagnostic data. If this start index is 0, the whole diagnosis is fetched from the monitor and stored to an internal buffer. Otherwise, the AS-i Master will respond out of the internal buffer. Thus, even though several requests are necessary to read the whole buffer, data integrity is maintained.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	L[1]	U[2]	circuit				
3	02_{16}							
4	slave address							
5	index							

1. L=1 long diagnosis for advanced monitor
2. U=1 unsorted diagnosis (all devices)

Response								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	result						
3	diagnosis byte #index+0							
4	diagnosis byte #index+1							
…	…							
n	diagnosis byte #index+n-3							

The diagnosis array is set up as follows:

Safety Monitor Diagnosis Array "*basic function range*" and "*sorted by OSSD*"								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	00_{16}							
1	state of monitor							
2	state of OSSD 1							
3	state of OSSD 2							
4	number of devices *not green*, OSSD1							
5	number of devices *not green*, OSSD2							
6	device index 32, OSSD 1							
7	color of device 32, OSSD 1							
8	device index 33, OSSD 1							
9	color of device 33, OSSD 1							

Safety Monitor Diagnosis Array "***basic function range***" and "***sorted by OSSD***"								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
…	…							
68	device index 63, OSSD 1							
69	color of device 63, OSSD 1							
70	device index 32, OSSD 2							
71	color of device 32, OSSD 2							
…	…							
132	device index 63, OSSD 2							
133	color of device 63, OSSD 2							

Safety Monitor Diagnosis Array "***enhanced function range***" and "***sorted by OSSD***"								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	00_{16}							
1	state of monitor							
2	state of OSSD1							
3	state of OSSD2							
4	number of devices "not-green", OSSD1							
5	number of devices "not-green", OSSD2							
6	device index 32, OSSD1							
7	color of device 32, OSSD1							
8	device index 33, OSSD1							
…	…							
133	color of device 95, OSSD1							
134	device index 32, OSSD2							
…	…							
261	color of device 95, OSSD2							

Safety Monitor Diagnosis Array "***basic function range***" and "***all devices***"								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	00_{16}							
1	state of monitor							
2	state of OSSD1							
3	state of OSSD2							
4	number of devices "not-green"							
5	—							
6	device index 32							
7	color of device 32							

Safety Monitor Diagnosis Array "***basic function range***" and "***all devices***"								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
8	device index 33							
9	color of device 33							
…	…							
68	device index 63							
69	color of device 63							
70	device index 32							
71	assignment of device 32 to OSSD							
…	…							
132	device index 63							
133	assignment of device 63 to OSSD							

Safety Monitor Diagnosis Array "***enhanced function range***" and "***all devices***"								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	00_{16}							
1	state of monitor							
2	state of OSSD1							
3	state of OSSD2							
4	number of devices number of devices "not-green"							
5	—							
6	device index 32							
7	color of device 32							
8	device index 33							
…	…							
133	color of device 95							
134	device index 32							
135	assignment of device 32 to OSSD2							
…	…							
261	assignment of device 95 to OSSD							

Possible assignment:

00_{16}: preprocessing

01_{16}: OSSD 1

02_{16}: OSSD 2

03_{16}: OSSD 1+2

80_{16}: device does not exist

Note	See <chapter 6 "Appendix: Code description", on page 94> for a description of the codes used for monitor state, OSSD state, device colours and assignments to OSSDs and the "Safety-at-Work" monitor documentation.

4.7.5 Integrated AS-i Sensors: Warnings

(Function: 03_{16})

List of integrated AS-i sensors according to profile S-1.1 (without extended addressing) or profile S-3.A.1 (with extended addressing), by which the input data bit D1 ("Warning") being deleted.

For creating of this list CDI and IDI are used only. Integrated AS-i slaves which are projected but not existing therefore are not entered here.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	O	circuit					
3	03_{16}							

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	result						
3	7A	6A	5A	4A	3A	2A	1A	0
…	…							
10	31B	30B	29B	28B	27B	26B	25B	24B

Response if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	result						
3	0	1A	2A	3A	4A	5A	6A	7A
…	…							
10	24A	25A	26A	27A	28A	29A	30A	31A

4.7.6 Integrated AS-i sensors: Availability

(Function: 04_{16})

List of the integrated slaves according to profile S-1.1 whose input data bits D2 ("Availability") are deleted.

For creating this list, CDI and IDI are used only. Integrated AS-i slaves which are projected but not existing therefore are not entered here.

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	O	circuit					
3	04_{16}							

Response (if O ≡ 0)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	result						
3	7	6	5	4	3	2	1	0
…	…							
6	31	30	29	28	27	26	25	24

Response (if O ≡ 1)								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	result						
3	0	1	2	3	4	5	6	7
…	…							
6	24	25	26	27	28	29	30	31

4.7.7 Language-select

(Function $0E_{16}$)

Use this function to set the display language.

Set:

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7D_{16}$							
2	T	–	circuit					
3	$0E_{16}$							
4	language[1]							

1. Value: 0= default (no changes), 1= english, 2= german, 3= french, 4= italian, 5= spain.

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7D_{16}$							
2	T	result						

Read:

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	–	circuit					
3	$0E_{16}$							

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	result						
3	language[1]							

1. Value: 0= default (no changes), 1= english, 2= german, 3= french, 4= italian, 5= spanish.

4.7.8 Replacement of Safety Slaves input data

(Function $0F_{16}$)

Use this function to replace safety slaves input data with "interpretation data". If the function is active, so have safety slaves input data the following meaning:

Bit 0, 1: 00=channel 1 has released, 11=channel 1 has not released.

Bit 2, 3: 00=channel 2 has released, 11=channel 2 has not released.

Note	This command replaces the old command MB_FP_LSS_ENABLE

Set:

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7D_{16}$							
2	T	–	circuit					
3	$0F_{16}$							
4	safety slaves[1]							

1. Value: 0= no substitute value, 1=substitute value for safety slaves

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7D_{16}$							
2	T	result						

Read:

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	–	circuit					
3	$0F_{16}$							

Response								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7E_{16}$							
2	T	result						
4	safety slaves[1]							

1. Value: 0= no substitute value, 1=substitute value for safety slaves

4.7.9 List of Safety Slaves

(Function 10_{16})

Use this function to find out the addresses of safety slaves.

Read:

Request								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7D_{16}$							
2	T	O[1]	circuit					
3	10_{16}							

1. O = orientation

Response (by O ≡ 0)								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7D_{16}$							
2	T	result						
3	7	6	5	4	3	2	1	0
…	…							
6	31	30	29	28	27	26	25	24

Response (bei O ≡ 1)								
Byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	$7D_{16}$							
2	T	result						
3	0	1	2	3	4	5	6	7
…	…							
6	24	25	26	27	28	29	30	31

5 Command Interface Examples

5.1 Reading 16-bit input values

Command RD_7X_IN: Reading of 16-bit input values.

Note	PROFIBUS: PROFIBUS DP V0: cyclic data exchange Used ID/module in the GSD file: 12-byte management

Meaning of the bytes:

Request: RD_7X_IN	
Byte 1	50_{hex} (RD_7X_IN)
Byte 2	00_{hex} (master 1, single master)
Byte 3	$1D_{hex}$ (slave address 29)
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
Byte 3	00_{hex} (or old values)
Byte 4	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

The call of the command interaface has not been answered with the valid values since the toggle bit has not been set.

<u>Set of toggle bit:</u>

Request	
Byte 1	50_{hex}
Byte 2	80_{hex} (toggle bit, result)
Byte 3	$1D_{hex}$ (slave address 29)
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Result: See chapter 4 "Commands of the Command Interface"

Response	
Byte 1	50_{hex}
Byte 2	80_{hex} (toggle bit, master1)
Byte 3	16-bit channel 1 high $byte_{hex}$
Byte 4	16-bit channel 1 low $byte_{hex}$
Byte 5	16-bit channel 2 high $byte_{hex}$
Byte 6	16-bit channel 2 low $byte_{hex}$
Byte 7	16-bit channel 3 high $byte_{hex}$
Byte 8	16-bit channel 3 low $byte_{hex}$
Byte 9	16-bit channel 4 high $byte_{hex}$
Byte 10	16-bit channel 4 low $byte_{hex}$
Byte 11	00_{hex} not used
Byte 12	00_{hex} not used

To get the input data again, the T-bit has to be reset again.

Note	PROFIBUS: If a command of the command interface with DP V1 is being carried out, setting the toggle bit is not necessary

5.2 Store current configuration to the AS-i master

1. Switch master to configuration mode
2. Write the current slave configuration to the master
3. Switch master to protected mode
4. Wait until master is in normal (protected) operation mode

12-byte management

1. Switch master to config mode

Request: SET_OP_MODE	
Byte 1	$0C_{hex}$ (SET_OP_MODE)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	01_{hex} (= config mode)
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

No result because toggle bit = 0.

Set the toggle bit:

Request: SET_OP_MODE	
Byte 1	$0C_{hex}$ (SET_OP_MODE)
Byte 2	80_{hex} (T = 1, master 1, single master)
Byte 3	01_{hex} (= config mode)
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	$0C_{hex}$
Byte 2	80_{hex} (T = 1, result = 0)
Byte 3	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

Master is now in configuration mode.

Result = 0 $\Rightarrow$ No error, for other result codes see chapter 4 Commands of the Command Interface, page 12 "Commands of the Command Interface".

2. Write the actual slave configuration to the master

Request: STORE_CDI	
Byte 1	07_{hex} (STORE_CDI)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

No result because toggle bit = 0.

Set the toggle bit:

Request: STORE_CDI	
Byte 1	07_{hex} (STORE_CDI)
Byte 2	80_{hex} (T = 0, master 1, single master)
Byte 3	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex}
Byte 2	80_{hex} (T = 1, result = 0)
Byte 3	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

The current configuration data has been written.

3. Set master to protected mode

Request: SET_OP_MODE	
Byte 1	$0C_{hex}$ (SET_OP_MODE)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	00_{hex} (= protected mode)
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

No result because toggle bit = 0.

Set the toggle bit:

Request: SET_OP_MODE	
Byte 1	$0C_{hex}$ (SET_OP_MODE)
Byte 2	80_{hex} (T = 1, master 1, single master)
Byte 3	00_{hex} (= protected mode)
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	$0C_{hex}$
Byte 2	80_{hex} (T = 1, result = 0)
Byte 3	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

The master has now been ordered to switch to the protected mode. It must be maintained now until the master changes into the operation mode.

4.Wait until master is in normal operation mode (and protected mode)

Reading out the flags until NA (Normal Operation Active) has been set.

Request: GET_FLAGS	
Byte 1	47_{hex} (GET_FLAGS)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

No result because toggle bit = 0.

Setting the toggle bit:

Request: GET_FLAGS	
Byte 1	47_{hex} (GET_FLAGS)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	00_{hex}
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response								
Byte 1	47_{hex}							
Byte 2	80_{hex} (T = 1, result = 0)							
Byte 3	-	-	-	-	-	-	-	POK
Byte 4	OR	APF	NA	CA	AAv	AAs	S0	COK
Byte 5						AAe	OL	DX
Byte 6	00_{hex}							
…								
Byte 12	00_{hex}							

The flag NA has to be set before the application is started. In case it is not set, the flags have to be read out until this flag has been set to 1.

The flag NA indicates that the master is in normal operation mode.

Normal operation mode is necessary to run the application safely.

5.3 Store new configuration for all slaves

1. Switch master in configuration mode
2. Write slave configuration to master
3. Write new list of projected slaves (*LPS*)
4. Write permanent parameter (*PP*) to master
5. Switch master to protected mode
6. Wait until master is in normal operation Mode (and protected mode)

12-byte management

1. Set master in config mode

Request: SET_OP_MODE	
Byte 1	$0C_{hex}$ (SET_OP_MODE)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	01_{hex} (= config mode)
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
Byte 3	00_{hex} (or old values)
Byte 4	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

No result because toggle bit = 0.

Set the toggle bit:

Request: SET_OP_MODE	
Byte 1	$0C_{hex}$ (SET_OP_MODE)
Byte 2	80_{hex} (T = 1, master 1, single master)
Byte 3	01_{hex} (= config mode)
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	$0C_{hex}$
Byte 2	80_{hex} (T = 1, result = 0)
Byte 3	00_{hex} (or old values)
Byte 4	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

The master is now in configuration mode.

Result: See chapter 4 "Commands of the Command Interface".

2. Write single configuration to master

Writing a configuration of an AS-i slave to the master.

For example:

16 bit input 4 CH at address 4 (Slave data sheet)
ID: 3_{hex}
ID2: E_{hex}
IO: 7_{hex}
ID1: F_{hex}

Request: SET_PCD	
Byte 1	25_{hex} (SET_PCD)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	04_{hex} (slave address to write to master)
Byte 4	EF_{hex} (ID + IO to configurate)
Byte 5	37_{hex} (xID2 + xID1 to configurate)
Byte 6	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
Byte 3	00_{hex} (or old values)
Byte 4	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

No result because toggle bit = 0.

Set the toggle bit:

Request: SET_PCD	
Byte 1	$0C_{hex}$ (SET_PCD)
Byte 2	80_{hex} (T = 1, master 1, single master)
Byte 3	04_{hex} (slave address to write to master)
Byte 4	EF_{hex} (ID + IO to configurate)
Byte 5	37_{hex} (ID + IO to configurate)
Byte 6	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	25_{hex}
Byte 2	80_{hex} (T = 1, result = 0)
Byte 3	00_{hex} (or old values)
Byte 4	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

The single slave configuration for the 16-bit module is written.

This command must be repeated for all 31 A-slaves and all 31 B-slaves. If you don't connect a slave to an address, write F_{hex} for ID, IO, ID1, ID2.

3. Write new list of projected slaves

Write the complete LPS of your AS-i circuit.

Every bit in the LPS corresponds to one slave after the following scheme:

Byte0/Bit 0: slave 0/0A - can not be set!
Byte1/Bit 1: slave 1/1A
...
Byte3/Bit 7: slave 31/31A
Byte4/Bit 0: slave 0B - can not be set!
Byte4/Bit 1: slave 1B
...

Byte7/Bit 7: slave 31B
The slave is projected if the bit is set.

Example above: 16-bit module at address 4 ⇒ Set bit 4/byte 0:

Request: SET_LPS	
Byte 1	29_{hex} (SET_LPS)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	00_{hex}
Byte 4	10_{hex} (LDS byte 0)
Byte 5	00_{hex} (LDS byte 1)
…	…
Byte 11	00_{hex} (LDS byte 7)
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

No result because toggle bit = 0.

Setting the toggle bit:

Request: SET_LPS	
Byte 1	29_{hex}
Byte 2	80_{hex} (T = 1, master 1, single master)
Byte 3	00_{hex}
Byte 4	10_{hex} (LDS byte 0)
Byte 5	00_{hex} (LDS byte 1)
…	…
Byte 11	00_{hex} (LDS byte 7)
Byte 12	00_{hex}

Response	
Byte 1	29_{hex}
Byte 2	80_{hex} (T = 1, result = 0)
Byte 3	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

The new list of protected slaves (LPS) is written.

4. Write permanent parameter (power on parameter) to master

Example as above: 16 bit module at address 4 with PP = 07_{hex}

Request: SET_PP	
Byte 1	43_{hex} (SET_PP)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	04_{hex} (slave address to write to master)
Byte 4	07_{hex} (PP to write (use low nibble))
Byte 5	00_{hex} (LDS byte 1)
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

No result because toggle bit = 0

Setting the toggle bit:

Request: SET_PP	
Byte 1	43_{hex} (SET_PP)
Byte 2	80_{hex} (T = 0, master 1, single master)
Byte 3	04_{hex} (slave address to write to master)
Byte 4	07_{hex} (PP to write (use low nibble)
Byte 5	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	43_{hex}
Byte 2	80_{hex} (T = 1, Result = 0)
Byte 3	00_{hex}
…	…
Byte 12	00_{hex}

The permanent parameter for the 16 bit module is written.

This command must be repeated for all 31 A-slaves and all 31 B-slaves. If you don't connect a slave to an address, write the default value to the master (F_{hex}) as a permanent parameter.

5. Switch Master to Protected Mode

Request: SET_OP_MODE	
Byte 1	$0C_{hex}$ (SET_OP_MODE)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	00_{hex} (= protected mode)
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

No result because toggle bit = 0.

Setting the toggle bit:

Request: SET_OP_MODE	
Byte 1	$0C_{hex}$ (SET_OP_MODE)
Byte 2	80_{hex} (T = 1, master 1, single master)
Byte 3	00_{hex} (= protected mode)
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	$0C_{hex}$
Byte 2	80_{hex} (T = 1, result = 0)
Byte 3	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

The master has now been ordered to switch to protected mode.

6. Wait until master is in normal (protected) operation mode

Read out the flags, until the NA (Normal Operation Active) has been set.

Request: GET_FLAGS	
Byte 1	47_{hex} (GET_FLAGS)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	00_{hex}
…	…
Byte 12	00_{hex}

Response	
Byte 1	00_{hex} (or old values)
Byte 2	00_{hex} (or old values)
…	…
Byte 12	00_{hex} (or old values)

No result because toggle bit = 0.

Setting the toggle bit:

Request: GET_FLAGS	
Byte 1	47_{hex} (GET_FLAGS)
Byte 2	00_{hex} (T = 0, master 1, single master)
Byte 3	00_{hex}
Byte 4	00_{hex}
…	…
Byte 12	00_{hex}

Response								
Byte 1	47_{hex}							
Byte 2	80_{hex} (T = 1, result = 0)							
Byte 3	-	-	-	-	-	-	-	POK
Byte 4	OR	APF	NA	CA	AAv	AAs	S0	COK
Byte 5						AAe	OL	DX
Byte 6	00_{hex}							
…								
Byte 12	00_{hex}							

The flag NA has to be set before the application is started. In case it is not set, the flags have to be read out until this flag has been set to 1.

The flag NA indicates that the master is in normal operation mode.

Normal operation mode is necessary to run the application safely.

Note	PROFIBUS: If a command of the command interface is used via PROFIBUS DP V1, it is not necessary to use the toggle bit.

The flag NA indicates that the master is in the normal operating mode which is necessary for the application to run safely.

5.4 Example for the readout of the safety monitor with ACYC_TRANS

Command interface length = 2+36

1. Start request:

request	
byte 1	0x4E (WR_ACYC_TRANS)
byte 2	0x00 (master 1)
byte 3	0x00 (integrated safety monitor)
byte 4	0x00 (puffer index high)
byte 5	0x00 (puffer index low)
byte 6	0x0A (safety monitor diagnostics)
byte 7	0x00 (number of bytes to send)

response	
byte 1	0x4E (WR_ACYC_TRANS)
byte 2	0x00 (OK)

2. Poll for the response (busy):

request	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (Master 1)
byte 3	0x00 (integrated safety monitor)
byte 4	0x00 (puffer index high)
byte 5	0x00 (puffer index low)

response	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (OK)
byte 3	0xFF (busy -> refresh)

3. Read response (data part 1):

request	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (master 1)
byte 3	0x00 (integrated safety monitor)
byte 4	0x00 (puffer index high)
byte 5	0x00 (puffer index low)

response	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (OK)
byte 3	0x0A (safety monitor diagnostics)
byte 4	0x01 (length of the response buffer *high*)
byte 5	0x06 (length of the response buffer *low*) 262
byte 6	0x00 (fixed)
byte 7	state of monitor
byte 8	state of OSSD1
byte 9	state of OSSD2
byte 10	number of devices "not-green"
byte 11	reserved
byte 12	0x20 (device index 32)
byte 13	device colour 32
byte 14	0x21 (device index 33)
byte 15	device colour 33
...	
byte 36	0x2C (device index 44)
byte 37	device colour 44
byte 38	0x2D (device index 45)

4. Read response (data part 2):

request	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (master 1)
byte 3	0x00 (integrated safety monitor)
byte 4	0x00 (puffer index high)
byte 5	0x24 (puffer index low) 36

response	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (OK)
byte 3	device colour 45
byte 4	0x2E (device index 46)
byte 5	device colour 46
...	
byte 36	0x3E (device index 62)
byte 37	device colour 62
byte 38	0x3F (device index 63)

5. Read response (data part 3):

request	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (master 1)
byte 3	0x00 (integrated safety monitor)
byte 4	0x00 (puffer index high)
byte 5	0x48 (puffer index low) 72

response	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (OK)
byte 3	device colour 63
byte 4	0x40 (device index 64)
byte 5	device colour 64
...	
byte 36	0x50 (device index 80)
byte 37	device colour 80
byte 38	0x51 (device index 81)

6. Read response (data part 4):

request	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (master 1)
byte 3	0x00 (integrated safety monitor)
byte 4	0x00 (puffer index high)
byte 5	0x6C (puffer index low) 108

response	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (OK)
byte 3	device colour 81
byte 4	0x52 (device index 82)
byte 5	device colour 82
...	
byte 30	0x5F (device index 95)
byte 31	device colour 95
byte 32	0x20 (device index 32)
byte 33	device alocation 32
byte 34	0x21 (device index 33)
byte 35	device alocation 33
byte 36	0x22 (device index 34)
byte 37	device alocation 34
byte 38	0x23 (device index 35)

7. Read response (data part 5):

request	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (master 1)
byte 3	0x00 (integrated safety monitor)
byte 4	0x00 (puffer index high)
byte 5	0x90 (puffer index low) 144

response	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (OK)
byte 3	device alocation 35
byte 4	0x24 (device index 36)
byte 5	device alocation 36
...	
byte 36	0x34 (device index 52)
byte 37	device alocation 52
byte 38	0x35 (device index 53)

8. Read response (data part 6):

request	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (Master 1)
byte 3	0x00 (integrated safety monitor)
byte 4	0x00 (puffer index high)
byte 5	0xB4 (puffer index low) 180

response	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (OK)
byte 3	device alocation 53
byte 4	0x36 (device index 54)
byte 5	device alocation 54
...	
byte 36	0x46 (device index 70)
byte 37	device alocation 70
byte 38	0x47 (device index 71)

9. read response (data part 7):

request	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (master 1)
byte 3	0x00 (integrated safety monitor)
byte 4	0x00 (puffer index high)
byte 5	0xD8 (puffer index low) 216

response	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (OK)
byte 3	device alocation 71
byte 4	0x48 (device index 72)
byte 5	device alocation 72
...	
byte 36	0x58 (device index 88)
byte 37	device alocation 88
byte 38	0x59 (device index 89)

10.Read response (data part 8):

request	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (master 1)
byte 3	0x00 (integrated safety monitor)
byte 4	0x00 (puffer index high)
byte 5	0xFC (puffer index low) 252

response	
byte 1	0x4F (RD_ACYC_TRANS)
byte 2	0x00 (OK)
byte 3	device alocation 89
byte 4	0x5A (device index 90)
byte 5	device alocation 90
byte 6	0x5B (device index 91)
byte 7	device alocation 91
byte 8	0x5C device index 92)
byte 9	device alocation 92
byte 10	0x5D (device index 93)
byte 11	device alocation 93
byte 12	0x5E (device index 94)
byte 13	device alocation 94
byte 14	0x5F (device index 95)
byte 15	device alocation 95

6 Appendix: Code description

Note	See also the separate "Safety-at-Work" monitor documentation for a description of the codes used for monitor state, OSSD state, device colours and assignments to OSSDs.

Table 1: Monitor state

Monitor state	
Code	significance
0	protective mode, everything OK (output circuits that are not installed, not configured or dependent output circuits are indicated as OK)
1	protective mode, output circuit 1 off.
2	protective mode, output circuit 2 off
3	protective mode, both output circuits off.
4	configuration mode: power on
5	configuration mode
6	reserved / not defined
7	configuration mode: fatal device error, RESET or device replacement necessary

Table 2: Colour coding

Colour coding		
Code	colour	meanining
0	green	block is in the ON state (switched on)
1	green flashing	block is in the ON state (switched on), but already in the transition to the OFF state, e.g. shutdown delay
2	yellow	block is ready, but is still waiting for a further condition, e.g. local acknowledgement or Start button
3	yellow flashing	time condition exceeded, action must be repeated, e.g. synchronization time exceeded
4	red	block is in the OFF state (switched off)
5	red flashing	the error interlock is active, clear using one the following actions: > Acknowledge using the ESC/Service button > Power OFF/ON > AS-i OFF/ON
6	grey	OSSD not used / no communication with the AS-i slave

Table 3: Allocation

Allocation								
Value	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	device exists	Device state has *not* changed itself since the last turning-off			device assigned to the preprocessing			
1	device doesn't exist	Device state has changed itself since the last turning-off			device assigned to the OSSD 1			
2					device assigned to the OSSD 2			
3					device assigned to the OSSD 1 and OSSD 2			

7 We Are Interested in Your Opinion!

Please tell us your suggestions, wishes and comments concerning this document.

We will review every note or comment and - if possible - include it in future issues.

Please fill out the form on the following page and fax it to us, or send your comments, suggestions etc. to the following address:

Euchner GmbH + Co. KG
Kohlhammerstr. 16
D - 70771 Leinfelden-Echterdingen

Phone: +49 (0)711-7597 0
Fax: +49 (0) 711-7533 16
eMail: info@euchner.de

Fax-Reply

Euchner GmbH + Co. KG

Date: ____________________

Fax No: +49 (0) 711-7533 16
eMail: info@euchner.de

From:

Company:	____________	Name:	____________
	____________	Department:	____________
Address:	____________	Job Function:	____________
City:	____________	Phone:	____________
ZIP Code:	____________	Fax:	____________
Country:	____________	eMail:	____________

Document:

Designation: ____________________ Issue Date: ____________

My Opinion of the Document

Form	**Yes**	**In part**	**No**
Is the table of contents arranged clearly?	❐	❐	❐
Are the pictures/graphics easy to understand/helpful?	❐	❐	❐
Are the written explanations of the images adequate?	❐	❐	❐
Does the quality of the images meet your expectations/needs?	❐	❐	❐
Does the layout of the document allow you to find a piece of information?	❐	❐	❐

Contents	**Yes**	**In part**	**No**
Is the phraseology/terminology easy to understand?	❐	❐	❐
Are the examples practice-oriented?	❐	❐	❐
Is the document easy to handle?	❐	❐	❐
Is any important information missing? If yes, what?	❐	❐	❐

____

____

Other Comments:

____

____

____

____