



AS-Interface Master OEM Module
according to Spec. 2.1
User Manual

Table of Contents

1	The Symbols Used	6
2	Safety	7
2.1	Intended Use	7
2.2	General Safety Information	7
3	Features	8
4	Installation of the AS-i Master OEM Module	9
4.1	LED Indicators	9
4.2	Connection of the AS-i Master OEM Module	9
4.3	Power Supply and AS-i Connection Techniques	10
5	Operating the AS-i Master OEM Module	13
5.1	Master Start-Up	13
5.2	Configuration Mode	13
5.3	Protected Operating Mode	14
5.3.1	Switching to Protected Operating Mode	14
5.3.2	Configuration Errors in Protected Operating Mode	14
5.4	Assigning an AS-i Address in Configuration Mode	15
5.4.1	Assigning a Slave Address	15
5.4.2	Erasing the Slave Address	15
5.5	Programming the Address in Case of Configuration Errors	16
5.5.1	Automatic Address Assignment	16
5.5.2	Manual Address Assignment	16
5.6	Error Messages	17
6	Advanced Diagnostics for AS-i Masters	18
6.1	List of Corrupted AS-i Slaves (LCS)	18
6.2	Protocol Analysis: Counters of Corrupted Data Telegrams	18
6.3	Offline Phase on Configuration Errors (LOS)	18
7	Data Exchange via the Dual Port RAM (DPRAM)	20
7.1	Principle	20
7.2	DPRAM Access	20
7.2.1	Structure of Input Data Image IDI and Output Data Image ODI	21
7.2.2	Structure of LAS, LDS, LPF and Delta List	22
7.3	AS-i Master Execution Control Lists	22
7.4	Command Interface	23
7.4.1	Construction	23
7.4.1.1	Values for Command	24
7.4.1.2	Values for Results	26
7.4.2	Commands of the Command Interface	26
7.4.2.1	IDLE	26
7.4.2.2	Read Input Data Image (READ_IDI)	27

7.4.2.3	Write Output Data Image (WRITE_ODI)	27
7.4.2.4	Set Permanent Parameter (SET_PP)	28
7.4.2.5	Get Permanent Parameter (GET_PP)	28
7.4.2.6	Write Parameter (WRITE_P)	29
7.4.2.7	Read Parameter (READ_PI)	29
7.4.2.8	Store Actual Parameters (STORE_PI)	30
7.4.2.9	Set Permanent Configuration (SET_PCD)	30
7.4.2.10	Get Extended Permanent Configuration (GET_PCD)	31
7.4.2.11	Store Actual Configuration (STORE_CDI)	31
7.4.2.12	Read Actual Configuration (READ_CDI)	32
7.4.2.13	Set List of Projected Slaves (SET_LPS)	32
7.4.2.14	Get List of Projected Slaves (GET_LPS)	33
7.4.2.15	Get List of Activated Slaves (GET_LAS)	34
7.4.2.16	Get List of Detected AS-i slaves (GET_LDS)	35
7.4.2.17	Get Flags (GET_FLAGS)	35
7.4.2.18	Set Operation Mode (SET_OP_MODE: Set_Operation_Mode)	36
7.4.2.19	Set Offline Mode (SET_OFFLINE)	37
7.4.2.20	Release Data Exchange (SET_DATA_EX)	38
7.4.2.21	Change Slave Address (SLAVE_ADDR)	38
7.4.2.22	Set Auto Address Enable (SET_AAE)	39
7.4.2.23	Get list of peripheral faults (GET_LPF)	40
7.4.2.24	Write AS-i Slave Extended ID1 (WRITE_XID1)	40
7.4.2.25	Read 1 7.3-Slave in.Data (RD_7X_IN)	41
7.4.2.26	Write 1 7.3-Slave out.Data (WR_7X_OUT)	41
7.4.2.27	Read 1 7.3-Slave out.Data (RD_7X_OUT)	42
7.4.2.28	Read 4 7.3-Slave in.Data (RD_7X_IN_X)	42
7.4.2.29	Write 4 7.3-Slave out.Data (WR_7X_OUT_X)	43
7.4.2.30	Read 4 7.3-Slave out.Data (RD_7X_OUT_X)	43
7.4.2.31	Read Output Data Image (READ_ODI)	44
7.4.2.32	Get Delta List (GET_DELTA)	44
7.4.2.33	WR_74_PARAM	45
7.4.2.34	RD_74_PARAM	45
7.4.2.35	RD_74_ID	46
7.4.2.36	RD_74_DIAG	46
7.4.2.37	Get Lists and Flags (Get_LPS, Get_LAS, Get_LDS, Get_Flags) (GET_LISTS)	47
7.4.2.38	Get List of Corrupted Slaves (GET_LCS)	49
7.4.2.39	Get List of Offline Slaves (GET_LOS)	49
7.4.2.40	Set List of Off-line Slaves (SET_LOS)	50
7.4.2.41	Get transm.err.counters (GET_TECA)	51
7.4.2.42	Get transm.err.counters (GET_TECB)	51
7.4.2.43	GET_TEC_X	52
7.4.2.44	Rewrite DPRAM (REWRITE_DPRAM)	53
7.4.2.45	FP_PARAM	53
7.4.2.46	FP_DATA	54
7.4.3	Functional profiles	54
7.4.3.1	„Safety at Work“ List 1	54
7.4.3.2	„Safety at Work“ Monitor Diagnosis	56
7.4.3.3	Integrated AS-i Sensors: Warnings	57

7.4.3.4	Integrated AS-i Sensors: Availability	58
7.4.4	Command Interface Examples	59
7.4.4.1	Reading analog Input Values	59
7.4.4.2	Store current Configuration to the AS-i Master	60
7.4.4.3	Store new Configuration for all Slaves	64
7.5	Watchdog	71
7.5.1	Config_ok Delay	72
7.6	Interrupts	72
8	Serial Communication using RXD and TXD	73
8.1	Telegrams of the Serial Communication	73
8.1.1	Message Structure	73
8.1.2	Synopsis of the Command Bytes	74
8.1.3	Message Descriptions	76
8.1.4	Representation of Information in the User Data Bytes	82
9	Appendix: Codes indicated by the Display	87
10	We Are Interested in Your Opinion!	89

Declaration of Conformity

according to 89/336/EEC and 73/23/EEC

Bihl+Wiedemann GmbH, Mannheim, Germany, hereby declares under its sole responsibility that the products mentioned below are according to the listed harmonized standards or normative documents and (where necessary) a competent body has been released.

Specification: AS-Interface Master OEM Module

Article-no.: 1554

Applied harmonized standards:

EN50295 (AS-i Complete Specification)

Applied national standards or normative documents:

Prüfungsordnung für AS-i Master

Manufacturer: Bihl+Wiedemann GmbH

Date: 02.06.03

Bernhard Wiedemann

1 The Symbols Used



Warning

This symbol warns the user of possible danger. Not following this warning can lead to personal injury or death and/or destruction of the equipment.



Attention

This symbol warns the user of a possible failure. Not following this warning can lead to total failure of the device or any other connected equipment.



Note

This symbol draws the user's attention to important information.

2 Safety

2.1 Intended Use



Warning

The protection of operating personnel and the system against possible danger is not guaranteed if the control interface unit is not operated in accordance with its intended use.

The device may only be operated by appropriately qualified personnel in accordance with this operating manual.

2.2 General Safety Information



Warning

Safety and correct functioning of the device cannot be guaranteed if any operation other than that described in this operation manual is performed.

The connecting of the equipment and any maintenance work to be carried out with voltage applied to the equipment must exclusively be performed by appropriately qualified electrotechnical personnel. In case a failure cannot be repaired, the device must be taken out of operation and kept from inadvertently being put back into operation. Repair work is to be carried out by the manufacturer only. Additions or modifications to the equipment are not allowed and will void the warranty.



Note

The operator is responsible for the observance of local safety standards.

3 Features

The AS-i Master OEM Module is supporting the following features:

- +5 V operating voltage
- 8 Bit data bus
- 10 Bit address bus
- Interrupt-capable
- AS-i connection through COMBICON connectors on the front
- AS-i connection also on 2.54 mm connection
- 2 x 15 pin wiring pin for the DPRAM interface and the serial interface (TTL)
- AS-i master according AS-i specification 2.1

The activated watchdog sets the Master to the offline phase, if it is not triggered by a host program. Advanced AS-i diagnostics to detect occasional occurring configuration errors and judge the quality of the AS-i communication are implemented.

Normally there is no need for an interrupt, but the AS-i Master OEM Module is capable to generate interrupts cyclically with every AS-i cycle or led by configuration errors or changes in input data. The AS-i Master OEM Module uses a DPRAM interface for data exchange. The DPRAM interface is consuming 1 kbytes. The DPRAM interface is easy to use, especially with any operating systems and with different programming languages. The AS-i Master OEM Module can be implemented as embedded AS-i Master into specific controllers.

4 Installation of the AS-i Master OEM Module

On the one side of the AS-i Master OEM Module is a terminal with COMBICON connectors for connection of the AS-i circuits (X1). It is possible to connect an AS-i circuit via X2 on the bottom side of the module.

4.1 LED Indicators

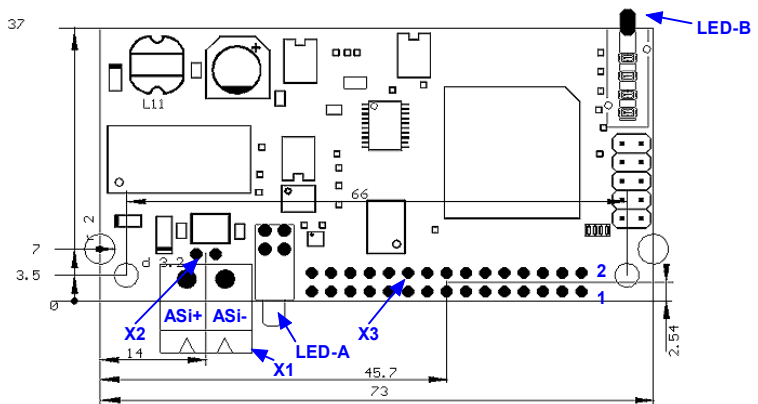
LEDs in LED-A (Top to bottom):

LED	Color	Description
U-ASi	green	The AS-i circuit is sufficiently powered
Config Error	red	Configuration error At least one configured slave is missing, or at least one detected slave is not projected or for at least one projected and detected slave the actual configuration data does not match the nominal configuration data.

LEDs in LED-B (Top to bottom):

LED	Color	Description
PWR	green	The master's power supply is sufficient
U-ASi	green	The AS-i circuit is sufficiently powered
Config. mode	yellow	The AS-i master is in configuration mode
Config Error	red	Configuration error At least one configured slave is missing, or at least one detected slave is not projected or for at least one projected and detected slave the actual configuration data does not match the nominal configuration data.

4.2 Connection of the AS-i Master OEM Module



The terminals of X1 and X2 have the following functions:

AS-i+ "AS-i +", Actuator Sensor Interface, positive terminal

AS-i- "AS-i -", Actuator Sensor Interface, negative terminal

Pin connection of the bus plug X3:

Pin	Name	Pin	Name
1	+5V	2	+5V
3	/CS	4	/WR
5	/Busy	6	/INT
7	/RD	8	A0
9	A1	10	A2
11	A3	12	A4
13	A5	14	A6
15	A7	16	A8
17	A9	18	nc
19	D7	20	D6
21	D5	22	D4
23	D3	24	D2
25	D1	26	D0
27	RXD, input	28	TXD, output
29	GND	30	GND

Pin 1, 2, 29, 30 are supply pins.

Pin 3 ... 26 are the DPRAM-Interface. For the timing refer to datasheets of industry-standard 1 k DPRAMs (55 ns), e. g. Cypress CY7C131-55.

Pin 27 and 28 are the serial interface with TTL level.

4.3 Power Supply and AS-i Connection Techniques

The AS-i Master OEM Module does not need a voltage supply of their own. It can be powered completely out of the AS-i line (the power consumption is about 70 mA from AS-i). An additional 24 V voltage source is not necessary. The AS-i master merely requires the connection to the AS-i line. When the 5 V power supply is adjacent at bus plug, the master starts to operate.



Note

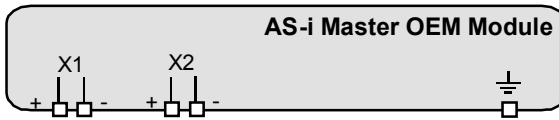
All AS-i terminals of the AS-i Master OEM Module are connected. So Slaves and AS-i powersupply can be connected to any of the terminals.



Warning

Pay attention to correct polarity by the connection of the AS-i circuit to the terminals of the AS-i Master OEM Module.

Connection samples for the AS-i power supply:

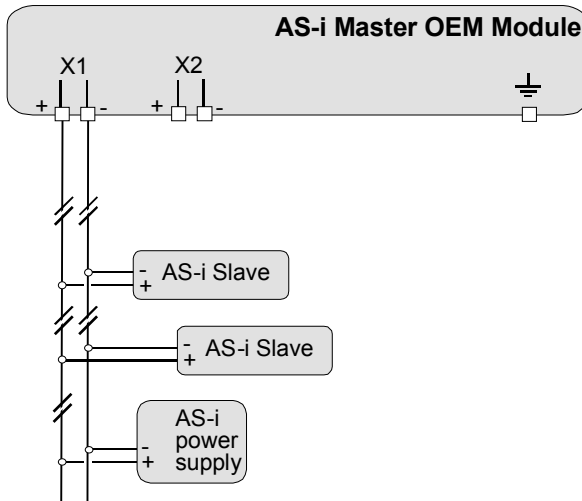


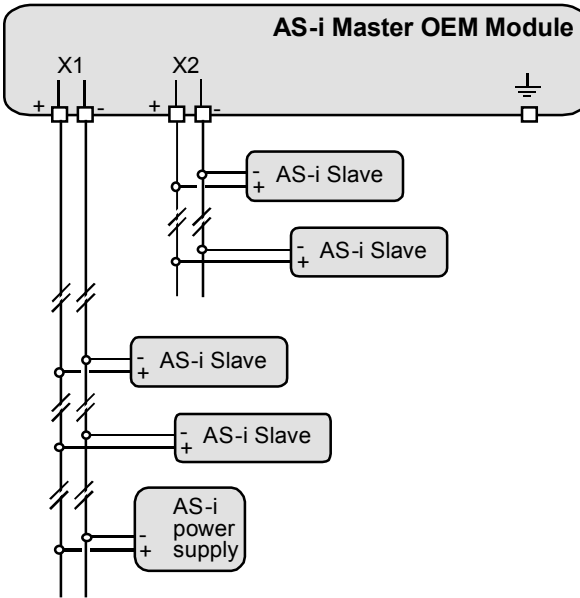
The terminals of X1 and X2 have the following functions:

AS-i+ "AS-i +", Actuator Sensor Interface, positive terminal

AS-i- "AS-i -", Actuator Sensor Interface, negative terminal

Connection variations for the AS-i power supply and the AS-i slaves:





In the wiring schemes above the current through the AS-i master must not exceed 2.8 A.

5 Operating the AS-i Master OEM Module

Since the AS-i Master OEM Module has no actual push-buttons nor a display, the terms push-button and display refer to the respective addresses in the DPRAM (0CB_{hex} and 0CC - 0CE_{hex}, see chapter 7).

5.1 Master Start-Up

After starting up, all segments of the figure display and all LEDs light up for approximately one second (self-test). Afterwards, the LC display the condition of their respective flags. The LC display shows the state of the master:

OFF Offline Phase

The AS-i master initializes - there is no data communication happening on the AS-i..



Attention

If the AS-i circuit is insufficiently powered (“U AS-i” does not light up) or there is no communication relationship between the master and gateway.

SEA Detection Phase

Start-up phase, in which the system looks for slaves located on the AS-i. The master remains in the detection phase until it finds at least one slave.

42¹ Activation Phase

End of the start-up operation when the parameters are transmitted to all connected and recognized slaves. This enables access to the AS-i slaves' data connections.

43² Start of Normal Operation

The AS-i master can exchange data with all active slaves. It transmits management messages and looks for and activates newly connected slaves. During normal operation, the system keeps the maximum cycle time of 5 milliseconds.

5.2 Configuration Mode

The configuration mode serves to configure the AS-i circuit.



Attention

In the configuration mode, all recognized slaves are activated even when the desired and actual configurations do not match.

Pressing the “mode” button for at least five seconds switches the AS-i master to configuration mode. While in configuration mode, the yellow “prj mode” LED lights up.

1. Activation phase and the start of normal operation maybe so short that the numbers can not be seen in the display.
2. Activation phase and the start of normal operation maybe so short that the numbers can not be seen in the display.

If the display is empty, no slaves have been connected to the AS-i circuit. In configuration mode, all recognized slaves are activated except for slave zero. The AS-i master is in normal operation. Data exchange between the AS-i master and all AS-i slaves has been detected by the master, regardless of whether the detected AS-i slaves have been projected before.



Attention

When delivered the device is in configuration mode.

5.3 Protected Operating Mode



Note

Unlike the configuration mode, the protected mode allows data exchange between the AS-i master and the projected AS-i slaves only.



Attention

If there is no communication between the host and the AS-I master, the AS-i master clears the output data of all slaves.

5.3.1 Switching to Protected Operating Mode

The configuration mode can be left by pressing the “mode” button.

Pressing the button shortly:

Exits the configuration mode without saving the current AS-i configuration.

Pressing the button for more than five seconds:

Exits the configuration mode and projects the actual AS-i configuration. Simultaneously the actual AS-i configuration is stored as nominal configuration in the EEPROM.



Note

If the system detects an AS-i slave with address zero on the AS-i, it can not leave the configuration mode.

In the protected operating mode, only AS-i slaves which are projected and whose actual configurations match the nominal configurations will be activated.

5.3.2 Configuration Errors in Protected Operating Mode

As long as there is no configuration error, the numeric display is turned off while in protected operating mode. Otherwise, the address with the faulty assignment is displayed. A faulty assignment occurs when a slave has been recognized or projected but cannot be activated.

If there are more than one faulty assignments the one that was first detected is displayed. Pressing the “set” button shortly displays the next higher faulty address.

Shortly appearing configuration errors are stored in the device (advanced AS-i diagnosis). The last error that occurred can be displayed by pressing the “set” button. If a short AS-i power failure is responsible for the configuration error the display will show a “39”.

5.4 Assigning an AS-i Address in Configuration Mode

To assign a slave with an address unequal zero to a different address unequal zero, please follow the following instructions in reverse order:

5.4.1 Assigning a Slave Address

(assigning an available address to a slave with address zero)

In configuration mode, the addresses of all detected slaves are displayed in succession. To display the next higher available operating address, press the “set” button shortly. Each time you press the “set” button, the next available address is displayed.

Choose the displayed address as your target address by pressing the “set” button for more than five seconds. The address display flashes. The master is ready for programming; pressing the “set” button again addresses the connected slave with address zero to the target (flashing address).

Any errors will be displayed by their error codes according to chapter 9. Otherwise, the detected slaves are displayed again as described in chapter 5.2..



Note

Only slaves with address 0 can get a new address by the master.



Attention

There must not be two AS-i slaves with the same address on the AS-i circuit, since this would cause malfunctions.

5.4.2 Erasing the Slave Address

(assigning address zero to a detected slave)

In configuration mode, the addresses of all recognized slaves are displayed in succession. By pressing the “set” button repeatedly, the master will display the next available address. Pressing the button more than five seconds while the address of a detected slave is displayed, this slave will get the address zero and the display will show “0”.

When you release the button, the display continues to display the detected slaves.

5.5 Programming the Address in Case of Configuration Errors

5.5.1 Automatic Address Assignment



Note

One of AS-i's major advantages is the automatic address assignment. If a slave fails, it can be replaced by another one of the same type with the address zero. The master will detect the replacement and automatically address the new slave with the address of the faulty one.

For automatic programming, the following requirements must be met:

1. The AS-i master must be in the protected operating mode.
2. The "Auto_Address_Assign"¹ release flag must be set.
3. Only one of the projected slaves may not be detected.

If these requirements are met, the AS-i master's "**prg enable**" LED lights up and a slave with address zero will be automatically be assigned to the operating address of the missing slave. The "Automatic Address Assignment" can be activated and deactivated with the software "AS-i Control Tools".



Note

Only slaves with address 0 can get a new address by the master, since this would cause malfunctions.



Attention

If the two slaves have different configuration data, i.e. are not of the same type as far as AS-i is concerned, the automatic address assignment will not be carried out.

5.5.2 Manual Address Assignment



Note

If several slaves fail, they cannot be replaced automatically by the AS-i master. These addresses have to be set manually. If this should not be done with the host interface or with a handheld addressing device, the slave addresses can also be changed by using the push-buttons and the LC display of the device.

In protected operating mode, wrong assignments are displayed as errors (see chapter 5.3). By pressing the "set" button all faulty assignments will be displayed in succession. By pressing the "set" button for more than five seconds the currently displayed address will be selected as a potential target address, and the display starts to flash.

If the faulty slave was previously replaced by a slave with address zero, the new slave can now be programmed for the blinking address by pressing the "set" key again. As a requirement, the new slave's configuration data must match the configuration data for the flashing address.

1. By deleting the flag "Auto_Address_Assign", the user can deactivate "automatic addressing".

After the address has been successfully set, the next faulty assignment is displayed and the address assignment can be carried out again. Otherwise, the system displays an error code (chapter 9). When all faulty assignments are eliminated, the display will be empty.

5.6 Error Messages



The system displays error codes for error messages that do not point to faulty assignments on the AS-i circuit. The code numbers are larger than 50 and therefore outside the slave address range. These codes are described in the appendix, chapter 9.

6 Advanced Diagnostics for AS-i Masters

The advanced AS-i diagnostics serve to locate occasionally occurring errors and to judge the quality of data transmission on AS-i without additional diagnostics tools.

6.1 List of Corrupted AS-i Slaves (LCS)

The LCS contains the history of the delta list. Besides the list of projected slaves (LPS), the list of detected slaves (LDS) and the list of activated slaves (LAS), a fourth list, the **list of corrupted slaves (LCS)**, is being created by AS-i masters with advanced diagnostics in order to locate occasionally occurring short-time configuration errors. This list contains entries of all AS-i slaves which were responsible for at least one configuration error since powering up the AS-i master or reading the list. Short-time AS-i power failures are represented in the LCS at the position of AS-i slave with address 0.



Note

With every read access the LCS will be deleted.

6.2 Protocol Analysis: Counters of Corrupted Data Telegrams

The AS-i master with advanced diagnostics has a counter of telegram repetitions for each AS-i slave, which increases every time a corrupted data telegram has been found. This makes possible to judge the quality of the AS-i network, even if only a few corrupted telegrams occurred and the AS-i slave did not cause any configuration errors.



Note

The counter values can be read via the host interface and will be deleted with every read access. The counter value is limited to 254. 255 will cause a counter overflow.

6.3 Offline Phase on Configuration Errors (LOS)

The AS-i master with advanced diagnostics offer the possibility to put themselves into the offline phase when a configuration error on the AS-Interface occurs. In this way the security of the application can be ensured. The reaction to a configuration error is very fast and the host can be relieved from this task. If there are any problems on the AS-i network, the AS-interface can be switched to a secure state.

There are two different ways to parameterize the AS-i master for this feature:

- Every configuration error during normal operation in protected mode releases the off-line phase.
- For each slave address, it can be chosen whether a configuration error on this address will cause the offline phase or not. This information is stored in the list of offline slaves (LOS).

The user himself can decide how the system reacts to a configuration error on

the AS-Interface. The AS-i master can release the off-line phase in critical situations, i. e. only with certain slave addresses, whereas in less critical situations (if one of the other AS-i slaves has a configuration error) only the error message is sent to the host, but AS-i is still running.

7 Data Exchange via the Dual Port RAM (DPRAM)

7.1 Principle

The AS-i (Control) data is stored in a DPRAM. That means that the device driver is very simple and can easily be written in any language and for any operating system.

7.2 DPRAM Access

The access on the RAM occurs over a field with a range of 1024 byte.

The base address of the module results from the /CS Logic on the carrier board.

DPRAM Address Map			
Address	Size	Data	Access
000 _{hex}	01 _{hex}	watchdog enable	r/w
001 _{hex}	01 _{hex}	watchdog counter	r/w
002 _{hex}	20 _{hex}	input data image <i>IDI</i>	r/-
022 _{hex}	02 _{hex}	execution control flags <i>ec-flags</i>	r/-
024 _{hex}	20 _{hex}	output data image <i>ODI</i>	r/w
044 _{hex}	01 _{hex}	host interface flags <i>hi-flags</i>	r/w
045 _{hex}	01 _{hex}	lock RS232 write	-/w
046 _{hex}	08 _{hex}	list of active slaves <i>LAS</i>	r/-
04E _{hex}	08 _{hex}	list of detected slaves <i>LDS</i>	r/-
056 _{hex}	08 _{hex}	list of slaves with periph faults <i>LPF</i>	r/-
05E _{hex}	08 _{hex}	delta list	r/-
070 _{hex}	01 _{hex}	command interface request	-/w
071 _{hex}	23 _{hex}	command interface buffer request	-/w
0A0 _{hex}	01 _{hex}	command interface response	r/-
0A1 _{hex}	23 _{hex}	command interface buffer response	r/-
0CB _{hex}	01 _{hex}	push buttons, mode=bit 0, set = bit 1	r/w
0CC _{hex}	03 _{hex}	3 digit display, ASCII, for displaying only	r/-
0CF _{hex}	01 _{hex}	config OK delay	r/w
0D0 _{hex}	18 _{hex}	name	r/-
0E8 _{hex}	08 _{hex}	software date code	r/-
0F0 _{hex}	08 _{hex}	feature string	r/-
0F8 _{hex}	04 _{hex}	serial number	r/-
3FB _{hex}	01 _{hex}	interrupt event clear	-/w
3FC _{hex}	01 _{hex}	interrupt event	r/-
3FD _{hex}	01 _{hex}	interrupt enable	r/w

DPRAM Address Map			
Address	Size	Data	Access
3FF _{hex}	01 _{hex}	interrupt request clear	r/-

7.2.1 Structure of Input Data Image *IDI* and Output Data Image *ODI*

byte	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
0	flags				slave 1/1A			
	F3	F2	F1	F0	D3	D2	D1	D0
1	slave 2/2A				slave 3/3A			
2	slave 4/4A				slave 5/5A			
3	slave 6/6A				slave 7/7A			
4	slave 8/8A				slave 9/9A			
5	slave 10/10A				slave 11/11A			
6	slave 12/12A				slave 13/13A			
7	slave 14/14A				slave 15/15A			
8	slave 16/16A				slave 17/17A			
9	slave 18/18A				slave 19/19A			
10	slave 20/20A				slave 21/21A			
11	slave 22/22A				slave 23/23A			
12	slave 24/24A				slave 25/25A			
13	slave 26/26A				slave 27/27A			
14	slave 28/28A				slave 29/29A			
15	slave 30/30A				slave 31/31A			
16	reserved				slave 1B			
17	slave 2B				slave 3B			
18	slave 4B				slave 5B			
19	slave 6B				slave 7B			
20	slave 8B				slave 9B			
21	slave 10B				slave 11B			
22	slave 12B				slave 13B			
23	slave 14B				slave 15B			
24	slave 16B				slave 17B			
25	slave 18B				slave 19B			
26	slave 20B				slave 21B			
27	slave 22B				slave 23B			
28	slave 24B				slave 25B			
29	slave 26B				slave 27B			
30	slave 28B				slave 29B			
31	slave 30B				slave 31B			

Flags		
	input data	output data
F0	ConfigError	Off-line
F1	APF	LOS-master-bit
F2	PeripheryFault	→ ConfigurationMode
F3	ConfigurationActive	→ ProtectedMode

ConfigError: 0 = ConfigOK, 1 = ConfigError
 APF: 0 = AS-i-Power OK, 1 = AS-i-Power Fail
 PeripheryFault: 0 = PeripheryOK, 1 = PeripheryFault
 ConfigurationActive: 0 = ConfigurationActive, 1 = ConfigurationInactive
 Offline: 0 = OnLine, 1 = Offline
 LOS master bit 0 = Off-Line by ConfigError deactivated
 1 = Off-Line by ConfigError deactivated

A rising edge of the "LOS master bit" effects that all bits in the LOS are set. A falling edge effects that all bits are deleted.

A rising edge of F2 and F3 switch the master to the desired mode.

7.2.2 Structure of LAS, LDS, LPF and Delta List

The list of activated slaves (LAS), list of detected slaves (LDS), list of slaves with peripheral faults (LPF) and delta list have following structure.

byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	7A	6A	5A	4A	3A	2A	1A	0A
...				...				
7	31B	30B	29B	28B	27B	26B	25B	24B

7.3 AS-i Master Execution Control Lists

All data of the AS-i master may be read at any time out of the DPRAM.

The output data image *ODI* and host interface flags (*hi-flags*) are read cyclically by the AS-i master.

The AS-i master writes cyclically the input data image *IDI*, the execution control flags (*ec-flags*), and the list of slaves lists.

Due to the internal processing of the output data image *ODI*, the user has to store it inverted to the DPRAM.

The *ec-flags* are stored in the DPRAM in reverse order than in the command interface.



Note

In the case that the flags in the ODI are changed and this activates the offline phase, the output data for the slave 1 (and all other slaves) are set to f_{hex} (low nibble) also if an alternate value has been written.

Therefore it is advisable to set flags and output data in succession with delay.

**Note**

In the case that the hi-flags are changing (if there is a stage during the offline phase), the bit *DataExchangeActive* will be set also if it has been deleted explicitly.

Therefore it is advisable to set the bits in succession with delay.

**Note**

The "name" and "serial number" will be written in the DPRAM only on power-up of the AS-i Master OEM Module and by execution of the function "update DPRAM".

7.4 Command Interface

The byte *command interface request* (byte 070_{hex} of the DPRAM address array) is used, to send commands to the host-interface of the AS-i Master OEM Module. If necessarily, in the *command interface buffer request* (DPRAM address array byte 071_{hex} to 093_{hex}) are entered all further data for this command (see chapter 7.4.2).

After the entry of the command in the byte *command interface buffer request* it is executed at once. This means, but that the parameters belonging to the command must be entered in the *command interface buffer request* BEFORE:

After execution of the command the command interface request byte (070_{hex}) is copied to the command interface response byte (0A0_{hex}).

The response of the successful executed command or the error description of the failed command can be read out in the *command interface buffer response* (byte 0A1_{hex} - 0C3_{hex} of the DPRAM).

7.4.1 Construction

If an AS-i slave is addressed in a command or in a response, the address is structured as shown below:

Request								
byte	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
1	command							
2	T	–	circuit					
3	request parameter byte 1							
...	...							
36	request parameter byte 34							

Response								
byte	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
1	command							
2	T	result						
3	response parameter byte 1							
...	...							
34	response parameter byte 32							

Command byte and T-bit are always part of the response. The T-bit is necessary to operate the command interface. This way the same command of the command interface can be used two twice repeatedly, possibly with different parameters.

Circuit = 0 If an AS-i gateway with one AS-i master or the master 1 of an AS-i gateway with 2 masters should be chosen.

Circuit = 1 If master 2 of a double master should be chosen.

The commands for reading and writing exist in two variations. At the first variation the bits in the slave lists are arranged as usually with Bihl+Wiedemann products: Data for slave with lower address appear in the lower bits. The second variation is compatible to Siemens masters: The sequence of the bits in the slave lists bytes are inverse.

Switching between the two variations can be done with bit 2^6 in byte 2 of the request. If it is deleted, the Bihl+Wiedemann arrangement is selected, otherwise the Siemens compatible arrangement is selected.

The coding of requests for commands to reading and writing is following therefore:

Request								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	command							
2	T	0	circuit					
3	Request parameter byte 1							
...	...							

7.4.1.1 Values for Command

Values for command				
command	value	meaning	Req Len	Res Len
IDLE	00 ₁₆	No order	2	2
READ_IDI	41 ₁₆	Read IDI	2	364
WRITE_ODI	42 ₁₆	Write_ODI	344	2
SET_PP	43 ₁₆	Set_Permanent_Parameter	4	2
GET_PP	01 ₁₆	Get_Permanent_Parameter	3	3
WRITE_P	02 ₁₆	Write_Parameter	4	3
READ_PI	03 ₁₆	Read_Parameter	3	3
STORE_PI	04 ₁₆	Store_Actual_Parameter	2	2
SET_PCD	25 ₁₆	Set_Permanent_Config	5	2
GET_PCD	26 ₁₆	Get_Permanent_Config	3	4
STORE_CDI	07 ₁₆	Store_Actual_Configuration	2	2
READ_CDI	28 ₁₆	Read_Actual_Configuration	3	4
SET_LPS	29 ₁₆	SET_LPS	11	2
GET_LPS	44 ₁₆	Get_LPS	2	10

Values for command				
command	value	meaning	Req Len	Res Len
GET_LAS	45 ₁₆	Get_LAS	2	10
GET_LDS	46 ₁₆	Get_LDS	2	10
GET_FLAGS	47 ₁₆	Get_Flags	2	5
SET_OP_MODE	0C ₁₆	Set_Operation_Mode	3	2
SET_OFFLINE	0A ₁₆	Set_Offline_Mode	3	2
SET_DATA_EX	48 ₁₆	Set_Data_Exchange_Active	3	2
SLAVE_ADDR	0D ₁₆	Change_Slave_Address	4	2
SET_AAE	0B ₁₆	Set_Auto_Address_Enable	3	2
GET_LPF	3E ₁₆	Get_LPF	2	10
WRITE_XID1	3F ₁₆	Write_Extended_ID-Code_1	3	2
RD_7X_IN	50 ₁₆	Read 1 7.3-slave in.data	3	10
WR_7X_OUT	51 ₁₆	Write 1 7.3-slave out.data	11	2
RD_7X_OUT	52 ₁₆	Read 1 7.3-slave out.data	3	10
RD_7X_IN_X	53 ₁₆	Read 4 7.3-slaves in.data	3	34
WR_7X_OUT_X	54 ₁₆	Write 4 7.3-slaves out.data	35	2
RD_7X_OUT_X	55 ₁₆	Read 4 7.3-slaves out.data	3	34
READ_ODI	56 ₁₆	Read ODI	2	34
GET_DELTA	57 ₁₆	Get list of config. diff.	2	10
WR_74_PARAM	5A ₁₆	Write S-7.4-slave parameter	≥6	2
RD_74_PARAM	5B ₁₆	Read S-7.4-slave parameter	4	≥3
RD_74_ID	5C ₁₆	Read S-7.4-slave ID string	4	≥3
RD_74_DIAG	5D ₁₆	Read S-7.4-slave diagnosis string	4	≥3
GET_LISTS	30 ₁₆	Get LDS, LAS, LPS, Flags	2	29
GET_LCS	60 ₁₆	Get LCS	2	10
GET_LOS	61 ₁₆	GET_LOS	2	10
SET_LOS	62 ₁₆	SET_LOS	10	2
GET_TECA	63 ₁₆	Get transm.err.counters	2	34
GET_TECB	64 ₁₆	Get transm.err.counters	2	34
GET_TEC_X	66 ₁₆	Get transm.err.counters	4	≥3
REWRITE_DPRAM	78 ₁₆	Rewrite DPRAM	3	3
FP_PARAM	7D ₁₆	„Functional Profile“ Param.	≥3	≥2
FP_DATA	7E ₁₆	„Functional Profile“ Data	≥3	≥2

7.4.1.2 Values for Results

Values for result			
	value	place	meaning
OK	00 ₁₆	–	execution without fault
HI_NG	11 ₁₆	HI	general fault
HI_OPCODE	12 ₁₆	HI	illegal value in command
HI_LENGTH	13 ₁₆	HI	length of the command interface is too short
HI_ACCESS	14 ₁₆	HI	no access right
EC_NG	21 ₁₆	EC	general fault
EC_SND	22 ₁₆	EC	„slave (source addr) not detected“
EC_SD0	23 ₁₆	EC	„slave 0 detected“
EC_SD2	24 ₁₆	EC	„slave (target addr) not detected“
EC_DE	25 ₁₆	EC	„delete error“
EC_SE	26 ₁₆	EC	„set error“
EC_AT	27 ₁₆	EC	„address temporary“
EC_ET	28 ₁₆	EC	„extended ID1 temporary“
EC_RE	29 ₁₆	EC	„read (extended ID1) error“

7.4.2 Commands of the Command Interface

The byte 1 of the request in the tables is according to byte 070_{hex} *command interface request* of the DPRAM table (see chapter 7.2).

The bytes 2 - 23 of the request in the tables are according to byte 071_{hex} - 093_{hex} of the *command interface buffer request* of the DPRAM table (see chapter 7.2).

The byte 1 of the response in the tables is according to byte 0A0_{hex} *command interface response* of the DPRAM table (see chapter 7.2).

The bytes 2 - 23 of the response in the tables are according to byte 0A1_{hex} - 0C3_{hex} of the *command interface buffer response* of the DPRAM table (see chapter 7.2).

7.4.2.1 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						

7.4.2.2 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 OEM Module 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	4 ₁₆							
2	T	–	circuit					
Response								
byte	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
1	4 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

7.4.2.3 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	4 ₂₁₆							
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						

7.4.2.4 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 OEM Module.

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						

7.4.2.5 Get Permanent Parameter (GET_PP)

With this call, a slave-specific parameter value stored on the EEPROM of the AS-i master OEM Module 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

7.4.2.6 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 OEM Module 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

7.4.2.7 Read Parameter (READ_PI)

This call returns the current parameter value (actual parameter) of an AS-i slave sent by the AS-i master OEM Module.

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.

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

7.4.2.8 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						

7.4.2.9 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 OEM Module 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				IO			

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 = 0 Single AS-i slave or A-slave

B = 1 B-slave

7.4.2.10 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 AS-i 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				IO			

Meaning of bit B:

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

B = 1 B-slave

7.4.2.11 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						

7.4.2.12 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-Interface 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

7.4.2.13 Set List of Projected Slaves (SET_LPS)

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						

7.4.2.14 Get List of Projected Slaves (GET_LPS)

With this call, the following entry is read out of the AS-i master OEM Module: The list of projected AS-i slaves (*LPS*).

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

7.4.2.15 Get List of Activated Slaves (GET_LAS)

With this call, the following entry is read out of the AS-i master OEM Module: The list of activated slaves (LAS).

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

7.4.2.16 Get List of Detected AS-i slaves (GET_LDS)

With this call, the following entry is read out of the AS-i master OEM Module: The list of detected AS-i slaves (LDS).

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

7.4.2.17 Get Flags (GET_FLAGS)

With this call, the following entry is read out of the AS-i master OEM Module: 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	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	AA _s	S0	Cok
5	–					AA _e	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_Ok
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.

7.4.2.18 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 OEM Module 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

7.4.2.19 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 OEM Module 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 OEM Module 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.

7.4.2.20 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						

7.4.2.21 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-i 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.



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.

Note

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

7.4.2.22 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						

7.4.2.23 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 \equiv 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 \equiv 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

7.4.2.24 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						

7.4.2.25 Read 1 7.3-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 7.3 can be read.

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							

7.4.2.26 Write 1 7.3-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 7.3 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						

7.4.2.27 Read 1 7.3-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 7.3 can be read out of the AS-i Master OEM Module.

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							

7.4.2.28 Read 4 7.3-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 7.3 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							

7.4.2.29 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 7.3 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						

7.4.2.30 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 7.3 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							

7.4.2.31 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 OEM Module.

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 3A	
	slave 2A			slave 3A				
...				...				
34	slave 30B			slave 31B				

7.4.2.32 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

7.4.2.33 WR_74_PARAM

With this function the parameter string of a slave according to profile S-7.4 is written. Since the string can be longer than the command interface, it will partly be written into the buffer and then be 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 $i \equiv 0$, 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						

7.4.2.34 RD_74_PARAM

With this function the parameter string 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 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	$5B_{16}$							
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						
	buffer byte i							
...	...							
n+2	buffer byte i+n-1							

7.4.2.35 RD_74_ID

With this function the ID 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 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 responds 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						
	buffer byte i							
...	...							
n+2	buffer byte i+n-1							

7.4.2.36 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 $i \equiv 0$, 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	5D 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	5D ₁₆							
2	T	result						
	buffer byte i							
...	...							
n+2	buffer byte i+n-1							

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

With this call, the following entries are read out of the AS-i master OEM Module:

- 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
...	LAS							
10	31B	30B	29B	28B	27B	26B	25B	24B
11	7A	6As	5A	4A	3A	2A	1A	0A
...	LDS							
19	31B	30B	29B	28B	27B	26B	25B	24B
20	7A	6As	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							
19	24B	25B	26B	27B	28B	29B	30B	31B
20	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

7.4.2.38 Get List of Corrupted Slaves (GET_LCS)

The LCS contains the history of the delta list.

With this call, the List of Corrupted Slaves (LCS) is read out of the AS-i master OEM Module.

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

7.4.2.39 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

7.4.2.40 Set List of Off-line Slaves (SET_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						

Issue date: 20.11.2003

7.4.2.41 Get transm.err.counters (GET_TECA)

With this call the error counters of all single slaves/A-slaves can be read (see chapter 6).

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							

7.4.2.42 Get transm.err.counters (GET_TECB)

With this call, the counts of the error counters for B-slaves are being read out (see chapter 6).

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.

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							

7.4.2.43 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 correspondending host interface and will be deleted with every read access. The counter's value is limited to 254. 255 will cause a counter overflow.

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							

7.4.2.44 Rewrite DPRAM (REWRITE_DPRAM)

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						

7.4.2.45 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 7.4.3).

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							

7.4.2.46 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 7.4.3).

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							

7.4.3 Functional profiles

7.4.3.1 „Safety at Work“ List 1

Function: 00₁₆

List of "safety-directed input slaves" ("AS-Interface 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 7.4.2.12: 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-Interface master in the bytes 3 and 4 (see chapter 7.4.2.17: "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 security monitor is also being activated if a security 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-Interface addresses 4 and 10
 1 safety slave with unreleased safety function,
 AS-Interface address 5.

Reponse: 7E 00 01 25 10 04 00 00

7.4.3.2 „Safety at Work“ Monitor Diagnosis

Function: 02₁₆

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

If the start index ist 0, new data is fetched from the monitor. Otherwise, the function will respond 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	7E ₁₆							
2	T	-	circuit					
3	02 ₁₆							
4	slave address							
5	index							

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 data field of the safety monitor is structured as follows:

Request								
byte	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
0	00 ₁₆							
1	monitor state							
2	state circuit 1							
3	state circuit 2							
4	number circuit 1							
5	number circuit 2							
6	device index 32, circuit 1							
7	device color, circuit 1							
8	device index 33, circuit 1							
9	device color, circuit 1							
...	...							
68	device index 63, circuit 1							
69	device color, circuit 1							
70	device index 32, circuit 2							
71	device color, circuit 2							
...	...							
132	device index 63, circuit 2							
133	device color, circuit 2							

7.4.3.3 Integrated AS-i Sensors: Warnings

Function: 03₁₆

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

7.4.3.4 Integrated AS-i Sensors: Availability

Function: 04₁₆

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

7.4.4 Command Interface Examples

7.4.4.1 Reading analog Input Values

Command RD_7X_IN: Reading of analog input values.

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 interface has not been answered with the valid values since the toggle bit has not been set.

Set of toggle bit:

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 7.4.1.2 "Values for Results"

Response	
Byte 1	50 _{hex}
Byte 2	80 _{hex} (toggle bit, master1)
Byte 3	analog channel 1 high byte _{hex}
Byte 4	analog channel 1 low byte _{hex}
Byte 5	analog channel 2 high byte _{hex}
Byte 6	analog channel 2 low byte _{hex}
Byte 7	analog channel 3 high byte _{hex}
Byte 8	analog channel 3 low byte _{hex}
Byte 9	analog channel 4 high byte _{hex}
Byte 10	analog 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. If a command of the command interface with DP V1 is being carried out, setting the toggle bit is not necessary.

7.4.4.2 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.

issue date: 20.11.2003

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 ⇒ No error, for other result codes see chapter 7.4.1.2 "Values for Results".

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 protected mode.

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

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.

7.4.4.3 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 7.4.1.2 "Values for Results".

2. Write single configuration to master

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

For example:

Analog input 4 CH at address 4 (Slave datasheet)

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 analog 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: Analog 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: Analog 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 analog 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.

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

7.5 Watchdog

If the contents of the byte *watchdog enable* is not equal to zero, the watchdog is enabled and the user has to write cyclically a value different to zero to the byte *watchdog counter*. The AS-i master decrements the watchdog counter every 10 ms. If Zero is reached, it changes to the off-line phase.

This way, the maximum watchdog time is written in units of 10 ms. That allows supervising times from 10 ms up to 2.55 seconds.

To disable the watchdog, the user has to write 00h to the byte *watchdog enable*.

Timing out may be recognized by *watchdog enable* \neq 0 and *watchdog counter* = 0.

The watchdog for the serial interface and DPRAM is the same. Therefore it is recommendable to use the watchdog only either of the serial interface or of the DPRAM.

7.5.1 Config_ok Delay

It is recommended to read the execution control flags every time the input data image *IDI* is read. Only if the *Config_ok* flag is set, the user can be sure that all input data is valid.

The PC needs a certain time to read the input data image and the execution control flags out of the DPRAM (in addition, the PC may be interrupted by other tasks while reading), and the AS-i master is able to alter the DPRAM at any time.

To guarantee the input data read by the PC is valid when it reads 'configuration O.K.', the 0-to-1 transition of the *Config_ok* flag is delayed. The user has to read both, input data and execution control flags before this delay runs out.

The DPRAM byte *Config_ok delay* holds the maximum time for this delay in units of 10 ms. If the user is sure reading of *IDI* and flags does not take more than 10 ms, he does not need to change the default value of 2 (20 ms).

7.6 Interrupts

The AS-i Master OEM Module is able to release interrupts of type A (software-end-of-interrupt). Each interrupt source has to be enabled by setting the corresponding bit in the word (or memory location) *interrupt enable*. It is possible to activate several interrupt sources at one time.

The register *interrupt event* holds the source(s) of the pending interrupt and should be cleared by the users interrupt routine.

Interrupt enable and *interrupt event* are coded as follows:

Interrupt Source								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0		WD		MB		CY	IDI	CFG

WD Watchdog time is over

MB Command of the command interface is worked aout

CY AS-i cycle

IDI Changes in the input data image *IDI*

CFG Changes in *Config_ok*

Interrupt acception processing::

1. Store the value of $3FC_{\text{hex}}$ (interrupt event) in an interim variable.
2. Write the value of the interim variable to $3FB_{\text{hex}}$ (interrupt event clear).
3. Read $3FF_{\text{hex}}$ (interrupt request clear).
4. Evaluate the interim variable.

8 Serial Communication using RXD and TXD

Besides the DPRAM interface, the AS-Interface Master OEM Module has a serial interface, using the B+W standard protocol.

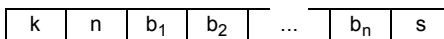
The communication parameters are:

19200 Baud, 8 data bits, no parity, 1 stop bit.

8.1 Telegrams of the Serial Communication

8.1.1 Message Structure

The messages have the following structure:



Command byte k:

Message ID character.

User data length n:

Number of user data bytes (zero to 17).

User data bytes b_i:

If user data length $n \equiv 00_{\text{hex}}$, this field is not used

Checksum s:

The lowest eight bits of the sum of all previously sent bytes are transmitted as the checksum. The checksum can also be calculated with the formula:

$$s = (k + n + \sum_{i=1}^n b_i) \bmod 256$$

The AS-i master responds to a host message with a message of the same type but normally of different length, or it responds with an error message (command byte 75_{hex}, 1 byte usable data).

Example:

For a change of the operating address from 7 to 26, the messages would look like this:

Host message:

command byte k: 6E_{hex}

user data length n: 02_{hex}

user data byte b₁: old slave address = 7E_{hex}

user data byte b₂: new slave address = 1A_{hex}

checksum s: 6E + 02 + 07 + 1A = 91_{hex}

Master message (master responds with "O.K.):

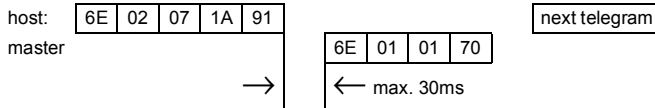
command byte k: 6A_{hex}

user data length n: 01_{hex}

user data byte b₁: status = "O.K." = 00_{hex}

checksum s: 6A + 01 + 00 = 6B_{hex}

maximum reaction time of the master: 30ms



8.1.2 Synopsis of the Command Bytes

k	Message	AS-i Specification		B+W
		2.04	2.1	Extensions
01 _{hex}	data exchange of all input and output data			✓
02 _{hex}	read output data			✓
03 _{hex}	write AS-i flags			✓
10 _{hex}	read input data		✓	
11 _{hex}	write output data		✓	
12 _{hex}	write configured parameters		✓	
13 _{hex}	read configured parameters		✓	
14 _{hex}	write actual parameters		✓	
15 _{hex}	read actual parameters		✓	
16 _{hex}	store actual parameters		✓	
17 _{hex}	write configuration data		✓	
18 _{hex}	read configuration data		✓	
19 _{hex}	store actual configuration		✓	
1A _{hex}	read actual configuration		✓	
1B _{hex}	write LPS		✓	
1C _{hex}	read LPS		✓	
1D _{hex}	read LAS		✓	
1E _{hex}	read LDS		✓	
1F _{hex}	read AS-i flags		✓	
29 _{hex}	set operating mode		✓	
2A _{hex}	write offline		✓	
2B _{hex}	write data exchange active		✓	
2C _{hex}	change slave address		✓	
2D _{hex}	write auto address enable		✓	
2F _{hex}	execute AS-i command		✓	
36 _{hex}	read LPF		✓	
37 _{hex}	write extended ID code 1		✓	
40 _{hex}	read 16 bit data			✓

issue date: 20.11.2003

k	Message	AS-i Specification		B+W
		2.04	2.1	Extensions
41 _{hex}	write 16 bit data			✓
42 _{hex}	16 bit data transmission control			✓
50 _{hex}	read LCS			✓
51 _{hex}	read error counters			✓
52 _{hex}	read LOS			✓
53 _{hex}	write LOS			✓
55 _{hex}	reserved for baud rate search			
61 _{hex}	write configured parameters	✓		
62 _{hex}	read configured parameters	✓		
63 _{hex}	write actual parameters	✓		
64 _{hex}	read actual parameters	✓		
65 _{hex}	store actual parameters	✓		
66 _{hex}	write configuration data	✓		
67 _{hex}	read configuration data	✓		
68 _{hex}	store actual configuration	✓		
69 _{hex}	read actual configuration	✓		
6A _{hex}	write LPS	✓		
6B _{hex}	read LPS	✓		
6C _{hex}	read LAS	✓		
6D _{hex}	read LDS	✓		
6E _{hex}	change slave address	✓		
6F _{hex}	execute AS-i command	✓		
71 _{hex}	read input data	✓		
70 _{hex}	write output data	✓		
72 _{hex}	read execution control flags	✓		
73 _{hex}	set operating mode	✓		
74 _{hex}	write host interface flags	✓		
75 _{hex}	error telegram			✓
76 _{hex}	exchange all input and output data			✓
77 _{hex}	write selected output data			✓
78 _{hex}	read selected output data			✓
79 _{hex}	disable automatic programming	✓		
7A _{hex}	watchdog test			✓
7B _{hex}	set watchdog			✓

k	Message	AS-i Specification		B+W
		2.04	2.1	Extensions
7C _{hex}	lock front panel operation			✓
7D _{hex}	read master version			✓
7E _{hex}	activate master			✓
7F _{hex}	download AS-i control program			✓
80 _{hex}	start AS-i control program			✓
81 _{hex}	read output data			✓
82 _{hex}	change master address			✓
83 _{hex}	upload AS-i control program			✓
84 _{hex}	read user memory (flags)			✓
85 _{hex}	write user memory (flags)			✓
88 _{hex}	advanced diagnostics			✓
89 _{hex}	write LOS			✓
8A _{hex}	read LOS			✓
8B _{hex}	exchange all process data			✓
8C _{hex}	write actual parameter			✓
8D _{hex}	read configuration data of all AS-i circuits			✓
8E _{hex}	configure all AS-i circuits			✓

8.1.3 Message Descriptions

In tables of the following pages are listed for each communication message the command byte k, the content of the data byte b_i for host and master message and the maximum reaction time t_{max} of the master.

The master returns the status byte, if there would otherwise be no user data. Normally, it takes on only one of the two following values:

- status = 0: error while executing a host request
- status = 1: no error while executing a host request

The recommendable communication messages are printed bold.

Commands according to the previous AS-i Master Specification (2.04)				
message	k	b _i (host message)	b _i (master message)	t _{max}
read input data	71 _{hex}	-	b ₁ ...b ₁₆ : input data	10ms
write output data	70 _{hex}	b ₁ ...b ₁₆ : output data	b ₁ : status	10ms
write configured parameters	61 _{hex}	b ₁ : slave address b ₂ : parameters	b ₁ : status	30ms
read configured parameters	62 _{hex}	b ₁ : slave address	b ₁ : parameters	20ms
write actual parameters	63 _{hex}	b ₁ : slave address b ₂ : parameters	b ₁ : counter-read parameters (inverted in case of error)	20ms
read actual parameters	64 _{hex}	b ₁ : slave address	b ₁ : parameters	20ms

Issue date: 20.11.2003

Commands according to the previous AS-i Master Specification (2.04)				
message	k	b ₁ (host message)	b ₁ (master message)	t _{max}
store actual parameters	65 _{hex}	-	b ₁ : status	200ms
write configuration data	66 _{hex}	b ₁ : slave address b ₂ : configuration data	b ₁ : status	30ms
read configuration data	67 _{hex}	b ₁ : slave address	b ₁ : configuration data	10ms
store actual configuration	68 _{hex}	-	b ₁ : status	200ms
read actual configuration	69 _{hex}	b ₁ : slave address	b ₁ : configuration data	10ms
write LPS	6A _{hex}	b ₁ ... b ₄ : LPS	b ₁ : status	30ms
read LPS	6B _{hex}	-	b ₁ ... b ₄ : LPS	10ms
read LAS	6C _{hex}	-	b ₁ ... b ₄ : LAS	10ms
read LDS	6D _{hex}	-	b ₁ ... b ₄ : LDS	10ms
read execution control flags	72 _{hex}	-	b ₁ : execution control flags	10ms
set operating mode	73 _{hex}	b ₁ = 0: protected operating mode b ₁ = 1: configuration mode	b ₁ : status	100ms
write host interface flags	74 _{hex}	b ₁ : host interface flag	b ₁ : status	30ms
change slave address	6E _{hex}	b ₁ : old slave address b ₂ : new slave address	b ₁ : status b ₁ =1: no error b ₁ =2: slave whose address should be changed not detected b ₁ =3: slave with address 0 detected b ₁ =4: address to which the slave should be programmed is already occupied. b ₁ =5: slave could not be programmed to address 0 b ₁ =6: slave could not be set for new operating address b ₁ =7: new operating address could not be stored in slave's EEPROM	30ms
execute AS-i command	6F _{hex}	b ₁ : slave address b ₂ : information part of the master request	b ₁ : response from slave b ₂ : status	30ms

Additional Commands beyond the AS-i Master Specification 2.04				
message	k	b ₁ (host message)	b ₁ (master message)	t _{max}
exchange all input and output data ^a	76 _{hex}	b ₁ ...b ₁₆ : output data	b ₁ : execution control flags b ₂ ...b ₁₇ : input data	10ms
write selected output data ^b	77 _{hex}	b ₁ : first slave address b ₂ : amount of slaves b ₃ ...b ₁₈ : output data	b ₁ : status	10ms

Additional Commands beyond the AS-i Master Specification 2.04				
message	k	b ₁ (host message)	b ₁ (master message)	t _{max}
read selected input data ^b	78 _{hex}	b ₁ : first slave address b ₂ : amount of slaves	b ₁ : execution control flags b ₂ ...b ₁₇ : input data	10ms
read output data	81 _{hex}	-	b ₁ ...b ₁₆ : output data	10ms
write parameter field	8C _{hex}	b ₁ : slave address b ₂ : actual parameters	b ₁ : status	10ms
read configured data of all AS-i circuits	8D _{hex}	b ₁ : number of the AS-i circuit b ₂ : slave address	b ₁ : status b ₂ : configured parameter b ₃ : configured data	10ms
configure all AS-i circuits	8E _{hex}	<u>Request 1 (start):</u> b ₁ ...b ₂ : FF _{hex} b ₃ ...b ₄ : 00 _{hex} <u>Request 2 (data):</u> b ₁ : number of the AS-i circuit b ₂ : slave address b ₃ : parameter of the slave b ₄ : configured data of the slave <u>Request 3 (commit):</u> b ₁ ...b ₂ : FF _{hex} b ₃ ...b ₄ : 01 _{hex}		300ms
	8D _{hex}	-	b ₁ : status	-
read master version	7D _{hex}	b ₁ : = 0: versions number (8 Bytes) b ₁ : = 1: master name part 1 (17 Bytes) b ₁ : = 2: master name part 2 (17 Bytes) b ₁ : = 3: master version (17 Bytes) b ₁ : = 4: installed software and host interface flags (17 Bytes)	b ₁ : version information (8 or 17 bytes)	10ms
activate/deactivate watchdog ^c for serial communication	7B _{hex}	b ₁ : 0: deaktiviert watchdog b ₁ : 1: watchdog timeout * 10ms	b ₁ : status	10ms
read watchdog status for serial communication	7A _{hex}	-	b ₁ : 0: watchdog not aktive b ₁ : 1: max. watchdog time * 10ms	10ms
lock/unlock front panel operation	7C _{hex}	b ₁ : 0: front panel operation enabled b ₁ : 1: front panel operation disabled	b ₁ : status	10ms

Additional Commands beyond the AS-i Master Specification 2.04				
message	k	b _i (host message)	b _i (master message)	t _{max}
error message	75 _{hex}	only sent by the AS-i master!	b ₁ : error code Bit 0: checksum error Bit 1: time-out Bit 2: unknown command Bit 3: illogical message length Bit 4: illogical number of user data bytes Bit 5: watchdog timer expired Bit 6: command execution error	-

- Recommended command because of least overhead: the AS-i master only has to wait once for the response of the slaves.
- The commands "write selected output data" and "read selected input data" will only be executed, if the AS-i master is in normal operation mode.
- If the watchdog has been activated, AS-i will go into the offline phase. By sending this message again AS-i leaves the off-line phase.

Commands according to the new AS-i Master Specification (2.1)				
message	k	b _i (host message)	b _i (master message)	t _{max}
read input data	10 _{hex}	-	b ₁ : status b ₂ , b ₃ : execution control flags b ₄ ...b ₃₅ : input data	
write output data	11 _{hex}	b ₁ ...b ₃₂ : output data	b ₁ : status	
write configured parameter	12 _{hex}	b ₁ : slave address b ₂ : parameter	b ₁ : status	
read configured parameter	13 _{hex}	b ₁ : slave address	b ₁ : status b ₂ : parameter	
write actual parameter	14 _{hex}	b ₁ : slave address b ₂ : parameter	b ₁ : status b ₂ : counter-read parameter (inverted in case of error)	
read actual parameter	15 _{hex}	b ₁ : slave address	b ₁ : status b ₂ : parameter	
store actual parameters	16 _{hex}	-	b ₁ : status	
write configuration data	17 _{hex}	b ₁ : slave address b ₂ , b ₃ : configuration data	b ₁ : status	
read configuration data	18 _{hex}	b ₁ : slave address	b ₁ : status b ₂ , b ₃ : configuration data	
store actual configuration	19 _{hex}	-	b ₁ : status	
read actual configuration	1A _{hex}	b ₁ : slave address	b ₁ : status b ₂ , b ₃ : configuration data	
write LPS	1B _{hex}	b ₁ ... b ₈ : LPS	b ₁ : status	
read LPS	1C _{hex}	-	b ₁ : status b ₂ ... b ₉ : LPS	
read LAS	1D _{hex}	-	b ₁ : status b ₂ ... b ₉ : LAS	

Commands according to the new AS-i Master Specification (2.1)				
message	k	b ₁ (host message)	b ₁ (master message)	t _{max}
read LDS	1E _{hex}	-	b ₁ : status b ₂ ... b ₉ : LDS	
read AS-i flags	1F _{hex}	-	b ₁ : status b ₂ , b ₃ : execution control flags b ₄ : host interface flags	
set operating mode	29 _{hex}	b ₁ = 0: protected mode b ₁ = 1: configuration mode	b ₁ : status	
set offline	2A _{hex}	b ₁ = 0: leave offline-phase b ₁ = 1: switch to offline-phase	b ₁ : status	
activate data exchange	2B _{hex}	b ₁ = 0: deactivate data exchange b ₁ = 1: activate data exchange	b ₁ : status	
change slave address	2C _{hex}	b ₁ : old slave address b ₂ : new slave address	b ₁ : status b ₁ =1: no error b ₁ =2: slave whose address should be changed not detected b ₁ =3: slave with address 0 detected b ₁ =4: address to which the slave should be programmed is already occupied. b ₁ =5: slave could not be programmed to address 0 b ₁ =6: slave could not be set for new operating address b ₁ =7: new operating address could not be stored in slave's EEPROM b ₁ =0: other error	
automatic address assigning	2D _{hex}	b ₁ = 0: disable automatic address assigning b ₁ = 1: enable automatic address assigning	b ₁ : status	
execute AS-i command	2F _{hex}	b ₁ : slave address b ₂ : information part of the master request	b ₁ : response from slave b ₂ : status	
read LPF	36 _{hex}	-	b ₁ : status b ₂ ... b ₉ : LPF	
write extended ID code 1 of slave 0	37 _{hex}	b ₁ : extended ID code 1	b ₁ : status b ₁ = 1: no error b ₁ = 2: slave with address 0 not detected b ₁ = 6: error with setting extended ID code 1 b ₁ = 8: extended ID code 1 stored only temporarily b ₁ = 0: other error	

Additional Commands beyond the AS-i Master Specification (for Masters according to Specification 2.1)				
message	k	b _i (host message)	b _i (master message)	t _{max}
exchange all input and output data ^a	01 _{hex}	b₁: host interface flags 2 ⁰ : Data_Exchange_Active 2 ¹ : Off-Line 2 ² : Auto_Address_Enable b₂...b₃₃: output data	b₁, b₂: execution control flags b ₁ , 2 ⁰ : Config_OK b ₁ , 2 ¹ : LDS.0 b ₁ , 2 ² : Auto_Address_Assign b ₁ , 2 ³ : Auto_Address_Available b ₁ , 2 ⁴ : Configuration_Active b ₁ , 2 ⁵ : Normal_Operation_Active b ₁ , 2 ⁶ : AS-i Power Fail b ₁ , 2 ⁷ : Offline_Ready b ₂ , 2 ⁰ : Periphery_OK b₃...b₃₄:input data	
output data lesen	02 _{hex}	-	b ₁ ...b ₃₂ : output data	
write AS-i flags	03 _{hex}	b₁: host interface flags 2 ⁰ : Data_Exchange_Active 2 ¹ : Off-Line 2 ² : Auto_Address_Enable	-	
error telegram	75 _{hex}	only sent by the AS-i Master!	b₁: error code Bit 0: checksum error Bit 1: time-out Bit 2: unknown command Bit 3: illogical message length Bit 4: illogical number of user data bytes Bit 5: watchdog timer expired Bit 6: command execution error	

a. Recommended command because of least overhead: the AS-i master only has to wait once for the response of the slaves.

Additional Commands for 16 Bit Transmissions (e.g. Analog Input or Output Slaves) (for Masters according to Specification 2.1)				
message	k	b _i (host message)	b _i (master message)	t _{max}
read 16 bit data	40 _{hex}	b ₁ : slave address	b ₁ ...b ₇ : 4 channels with 16 bit data each	
write 16 bit data	41 _{hex}	b ₁ : slave address b ₂ ...b ₈ : 4 channels with 16 bit data each	-	
enable/disable 16 bit transmission	42 _{hex}	b₁: bitfield Bit 0 = 0: start Bit 0 = 1: stop Bit 1 = 1: reset	-	

Commands for Advanced AS-i Diagnostics				
message	k	b ₁ (host message)	b ₁ (master message)	t _{max}
advanced diagnostics	88 _{hex}	b ₁ : selection	n=0: b ₁ -b ₁₅ : slave 1 - 31 n=1: b ₁ -b ₁₅ : slave 0 - 15 n=2: b ₁ -b ₁₅ : slave 16 -31	10ms
write LOS	89 _{hex}	b ₁ ... b ₄ : slaves 0 - 31	b ₁ : error status	10ms
read LOS	8A _{hex}	-	b ₁ ... b ₄ : slaves 0 - 31	10ms

Commands for Advanced AS-i Diagnostics (for Master according to Specification 2.1)				
message	k	b ₁ (host message)	b ₁ (master message)	t _{max}
read LCS	50 _{hex}	-	b ₁ ... b ₈ : LCS	
read error counters	51 _{hex}	b ₁ : choice (a)	choice a=0: b ₁ ... b ₃₂ : slaves 0 - 31 or 0A - 31A choice a=1: b ₁ ... b ₃₂ : slaves 0B -31B	
read LOS	52 _{hex}	-	b ₁ ... b ₈ : LOS	
write LOS	53 _{hex}	b ₁ ... b ₈ : LOS	-	

Commands for Backward Compatibility with Older Master Versions				
message	k	b ₁ (host message)	b ₁ (master message)	t _{max}
enable/disable automatic programming	79 _{hex}	b ₁ ≡ 0: disable b ₂ ≡ 1: enable	b ₁ : status	30ms

8.1.4 Representation of Information in the User Data Bytes

Input and Output Data

For each slave, a four-digit binary number can be entered as input and output data. Input and output data can therefore range from 0 to 15 (or hexadecimal 0 to F).

For serial transmission, the data for two slaves are combined in a single byte. With message "q" (read input data, 71_{hex}), the master therefore sends 32/2 = 16 bytes of user data.

byte 0	byte 1	...	byte 15
slave 0, slave 1	slave 2, slave 3	...	slave 30, slave 31

The entries for low slave addresses are transmitted first. Byte 0, bits 0 through 3 (lower nibble) thus contains the input data of the slave with operating address zero; the upper nibble of the user data byte 15 contains the data of slave 31.

byte								
bit	0	1	2	3	4	5	6	7
slave	slave 0				slave 1			

For the AS-i master according to specification 2.1 the following information applies additionally:

- The bytes 0 to 15 contain data for the slaves 0 to 31 or 0A to 31A.
- The bytes 16 to 31 contain data for the slaves 0B to 31B.

byte 16	byte17	...	byte 15
slave 0B, slave 1B	slave2B, slave 3B	...	slave 30B, slave 31B

Slave Lists

The AS-i slave lists LPS, LDS, LAS, LCS and LOS are built up as follows:

byte	0								1							
bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
slave	0 ^a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

a. LDS and LCS only

byte	2								3							
bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
slave	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Meaning of the lists:

LPS List of Projected Slaves

LDS List of Detected Slaves

LAS List of Activated Slaves

LCS List of Corrupted Slaves

List of those slaves, that have caused a short-time configuration error.

LOS List of Off-line Slaves

List of those slaves, with that in case of configuration error the AS-i master shall switch to the Off-line phase.

For the AS-i master according to specification 2.1 the following information applies additionally:

- The bytes 0 bis 3 contain the entries for the slaves 0 to 31 or 0A to 31A.
- The bytes 4 bis 7 contain the entries for the slaves 0B bis 31B

byte	4								5							
bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
slave	0B ^a	1B	2B	3B	4B	5B	6B	7B	8B	9B	10B	11B	12B	13B	14B	15B

a. LDS and LCS only

byte	6								7							
bit	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
slave	16B	17B	18B	19B	20B	21B	22B	23B	24B	25B	26B	27B	28B	29B	30B	31B

Furthermore there is another list for the AS-i master according to specification 2.1:

LPF List of Peripheral Faults
List of those slaves, where a peripheral occurred.

AS-i Configuration Data

Each AS-i slave informs about its type with the AS-i configuration data. This data consists of one byte, the lower four bits representing the ID code, the upper four bits the I/O code.

byte	0							
bit	0	1	2	3	4	5	6	7
	ID code				I/O code			

For the AS-i master according to specification 2.1 there is an additional second byte for the AS-i configuration data:

In this byte the lower four bits represent the extended ID code 2, the upper four bits the extended ID code 1:

byte	1							
bit	0	1	2	3	4	5	6	7
	ext. ID code 2				ext. I/O code 1			

Execution Control Flags

The execution control flags are transmitted in the diagnosis telegram, if the gateway is operated in the professional mode.

When set (=1), the individual bits have the following meaning:

Bit 0:	<i>Config_OK</i>	no configuration error
Bit 1:	<i>LDS.0</i>	slave with address 0 present
Bit 2:	<i>Auto_Address_Assign</i>	automatic programming permitted
Bit 3:	<i>Auto_Address_Available</i>	automatic programming available
Bit 4:	<i>Configuration_Active</i>	configuration mode active
Bit 5:	<i>Normal_Operation_Active</i>	normal operation active
Bit 6:	<i>APF</i>	AS-i power failure
Bit 7:	<i>Offline_Ready</i>	off-line mode active

For the AS-i master according to specification 2.1 there is an additional second byte for the execution control flags:

Bit 0:	<i>Periphery_OK</i>	no peripheral error
Bit 1-7:		not used

Host Interface Flags

The setting of the host interface flags has the following effects:

Bit 0:	<i>Data_Exchange_Active</i>	The data communication between AS-i master and slaves is active
Bit 1:	<i>Off-line</i>	The AS-i master is set into offline phase
Bit 2:	<i>Auto_Address_Enable</i>	The automatic programming is disabled (This flag is stored non-volatile)

Installed Software/Host Interface Flags (message 7D_{hex})

If message 7D_{hex} ("read master version") is sent with a "4" in the host message's data byte, the AS-i master responds with a 17 bytes long character string (16 letters, zero terminated).

The letters have the following explanations:

- Byte 0 (C/c, D/d, Z/z)
 The responding AS-i master is an AS-i control.
 The capital 'C' means that a controller program is currently being executed. A lower-case 'c' means that either the start flag has not been set or that the AS-i master's status does not permit the execution. If D/d instead of C/c is displayed, it is the newer software version of AS-i Control II.
- Byte 1 (B/b)
 The responding master has a bus-capable RS 485 or RS 422 interface. The messages 7E_{hex} (activate master) and 82_{hex} (change master address) can be processed.
- Byte 2 (F/f)
 The responding AS-i master is featured with an AS-i error counter.
- Byte 3 (E/e)
 The responding AS-i master is featured with an EMC test mode.
- Byte 4 (D/d)
 The responding AS-i master is featured with advanced diagnostics.
- Byte 5 (C/c)
 The responding AS-i master is featured the function off-line by configuration error.
- Byte 6 (. /2)
 The responding AS-i master manages one ('.') or two ('2') AS-i circuits.
- Byte 7 not used
- Byte 8 (D/d)
 The "data_exchange_active" host interface flag is set/erased.

Byte 9	(O/o) The “ <i>off-line</i> ” host interface flag is set/erased.
Byte 10	(A/a) The “ <i>auto_address_enable</i> ” host interface flag is set/erased.
Byte 11	not used
Byte 12	(./A) The AS-i master is according to the new AS-i master specification 2.1 (AAS-i).
Byte 13	not used
Byte 14	(W/w) The serial watchdog was activated/deactivated.
Byte 15	(T/t) The operation of the AS-i master via the front panel buttons is enabled/disabled.

AS-i Control Flags, Start/Stop Code

Bit 0:	<i>start_flag</i>	if bit 0 is set, the controller program is executed as soon as the AS-i master’s status permits (this flag is stored non-volatile).
Bit 1:	<i>reset_bit</i>	the controller program is read from the EEPROM prior to the start. In addition, the user memory (flag bytes) is erased (necessary after each download), not returned as AS-i control flag).
Bit 2:	<i>ignore_config_errors</i>	if bit 2 is erased, the controller program is stopped as soon as an AS-i configuration error occurs (this flag is stored non-volatile).
Bit 3:	<i>auto_start</i>	if bit 3 is set, AS-i control waits for a push on the “set” button before it restarts the controller program (this flag is stored non-volatile).
Bit 4:	<i>counter_map</i>	if bit 4 is set, the counter registers of the 15 counters can be accessed by M 96.0 to M 125.7 (this flag is stored non-volatile).

9 Appendix: Codes indicated by the Display

In the basic state of the configuration mode, the display shows the addresses of all detected slaves at a rate of two per second one after the other. A blank display indicates that the *LDS* is empty, no slaves were detected.

Addresses are displayed in three digits. The first two digits show the address (0 ... 31) and the last digit shows a blank for a Single Slave and 'A' or 'B' for an A/B-Slave.

In the basic state of the protected operating mode, the display is either blank or displays the address of a faulty assignment (see chapter 5.3.2).

During manual address programming, the slave address display has a different meaning (see chapter 5.4 and 5.5).

All displayed numbers bigger than 31 which can not be interpreted as a slave address are status or error messages of the master. They have the following meanings:

39	Advanced AS-i diagnostics: After pressing the 'set'-button a short-time AS-i power failure occurred.
OFP	The AS-i master is in offline phase.
SEA	The AS-i master is in detection phase.
42	The AS-i master is in activation phase.
43	The AS-i master starts the normal operating mode.
E50	Peer Processor Interface is defect.
E70	Hardware error: The AS-i master's EEPROM cannot be written.
E72	Hardware error: The PIC processor does not respond.
E73	Hardware error: The PIC processor does not respond.
E74	Checksum error in the EEPROM.
E75	Error in the external RAM.
E76	Error in the external RAM.
E80	Error while attempting to exit the configuration mode: A slave with address zero exists.
E81	General error while changing a slave address.
E82	The front panel operation is blocked. Until repowering-up the device can only be accessed from the host via the interface.
E83	Program reset of the AS-i Control programm: The AS-i Control programm is being read out of EEPROM and copied into the RAM.
E90	Error while changing a slave address in protected operating mode: No slave with address 0 existing.
E91	Error while changing slave address: Target address is already used.
E92	Error while changing slave address: New address could not be set.
E93	Error while changing slave address: New address could only be stored volatily in the slave.
E94	Error while changing the slave address in protected operating mode: Slave has wrong configuration data.

E95	Error while changing slave address in protected operating mode: The configuration error was caused by a superfluous slave (instead of a missing slave).
888	Display test while starting up the AS-i master

10 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:

Bihl+Wiedemann GmbH
Technical Support
Flosswoerthstr. 41
D - 68199 Mannheim
Phone: +49 (0) 621-33 99 6-0
Fax: +49 (0) 621-33 9 22 39
eMail: mail@bihl-wiedemann.de

Fax-Reply

Bihl+Wiedemann GmbH
Technical Support

Date: _____

Fax No: +49 (0) 621-33 9 22 39

eMail: mail@bihl-wiedemann.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?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Are the pictures/graphics easy to understand/helpful?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Are the written explanations of the images adequate?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Does the quality of the images meet your expectations/needs?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Does the layout of the document allow you to find a piece of information?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Contents

Yes	In part	No
------------	----------------	-----------

Is the phraseology/terminology easy to understand?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Are the examples practice-oriented?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Is the document easy to handle?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Is any important information missing? If yes, what?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Other Comments:

issue date: 20.11.2003