FACTORY AUTOMATION

Manual

VBG-DN-K5-D
VBG-DN-K5-DMD

AS-Interface/DeviceNet Gateway

in acc. to specification 2.11

PEPPERL+FUCHS

Table of Contents

1 Declaration of Conformity 5

2 The Symbols Used 6

3 Safety 7
3.1 Intended Use 7
3.2 General Safety Information 7

4 General Information 8

5 Description 11
5.1 LED Indicators 11
5.2 Power Supply Concepts and AS-i Connection Techniques 11
5.2.1 Single Masters in IP20 with AS-i Power Supply A 11
5.2.1.1 Connections of the AS-i/DeviceNet Gateway with graphic Display 11
5.2.2 Double Master in IP20 with Power Supply A 12
5.2.2.1 Power supply out of AS-i circuit 1 13
5.2.2.2 Operation with separate 24 V DC Power Supply 14
5.3 AS-i Network Connection 15
5.4 DeviceNet Connection 15
5.5 Display and Operating Elements 16
5.5.1 LEDs of the Single Masters and Double Masters with Graphical Display 16
5.5.2 Push-Buttons 16

6 Configuration 17
6.1 Setting DeviceNet Addres and Baud Rate 17
6.2 I/O Data Interpretation 17

7 Operating the AS-i/DeviceNet Gateway 19
7.1 Peculiarities with Double Masters 19
7.2 Master Start-Up 19
7.3 Configuration Mode 20
7.4 Protected Operating Mode 20
7.4.1 Switching to Protected Operating Mode 20
7.4.2 Configuration Errors in Protected Operating Mode 21
7.5 Assigning an AS-i Address in Configuration Mode 21
7.5.1 Assigning a Slave Address 21
7.5.2 Erasing the Slave Address 22
7.6 Programming the Address in Case of Configuration Errors 22
7.6.1 Automatic Address Assignment 22
7.6.2 Manual Address Assignment 23
7.7 Error Messages 23

8 Operating by Full-graphic Display 24
8.1 DeviceNet (Fieldbus Interface) 26

8.1.1 DeviceNet MAC ID 26
8.1.2 DeviceNet Baud Rate 27
8.1.3 DeviceNet Status 27
8.1.4 DeviceNet I/O-Path 27
8.2 Setup (Configuration of the AS-i Circuit) 28
8.2.1 AS-i Circuit 28
8.2.2 AS-i Slave Addr (Change AS-i Slave Address) 28
8.2.3 Force Offline (switch AS-i Master offline) 29
8.2.4 Operation Mode 29
8.2.5 Store Act Cfg (Store Actual Detected Configuration) 30
8.2.6 Permanent Param (Projected Parameter) 30
8.2.7 Permanent Cfg (Projected Configuration Data) 30
8.2.8 Addr. Assistant (Address Assistant) 31
8.2.9 LOS (List of Offline Slaves) 31
8.3 IO + Param. Test (Testing AS-i In- and Outputs as well as reading and writing AS-i Parameters) 32
8.3.1 Binary Inputs 32
8.3.2 Binary Outputs 32
8.3.3 Analog Inputs 33
8.3.4 Analog Outputs 33
8.3.5 Parameter 33
8.4 Diagnosis (Normal AS-i Diagnosis) 34
8.4.1 EC-Flags (Execution Control Flags) 34
8.4.2 Actual Config (Actual Configuration) 35
8.4.3 LPF (List of Periphery Faults) 35
8.4.4 AS-i Master (Info) 36
8.5 Adv. Diagnosis (Advanced AS-i Diagnosis) 36
8.5.1 Error Counters 36
8.5.2 LCS (List of Slaves having caused a Configuration Error) 37
8.6 AS-i Safety 37
8.6.1 Safety Slaves 37
8.6.2 Safety Monitor 38

9 Advanced Diagnostics for AS-i Masters 39
9.1 List of Corrupted AS-i Slaves (LCS) 39
9.2 Protocol Analysis: Counters of Corrupted Data Telegrams 39
9.3 Offline Phase on Configuration Errors (LOS) 39

10 DeviceNet Details 41
10.1 DeviceNet Information 41
10.1.1 DeviceNet Message Types 41
10.1.2 DeviceNet Class Services 41
10.2 Object Modelling 42
10.2.1 Identity Object 42
10.2.2 DeviceNet Object 43
10.2.3 Assembly Object 44
10.2.4 Connection Object 53
10.2.5 Parameter Object 56
10.2.6 AS-i Master Object 57

issue date 11.1.2005

10.2.7 AS-i Slave Object 59
10.2.8 IO Data Object 60
10.2.9 Advanced Diagnostics Object 62
10.2.10 Short Command Interface Object 63
10.2.11 Long Command Interface Object 63

11 Command Interface 64
11.1 Construction 64
11.1.1 List of all Commands 65
11.1.2 Values for Results 67
11.2 Commands of the Command Interface 67
11.2.1 Analog Data 67
11.2.1.1 Overview of the Commands 67
11.2.1.2 Read 1 7.3-Slave in.Data (RD_7X_IN) 68
11.2.1.3 Write 1 7.3-Slave out.Data (WR_7X_OUT) 68
11.2.1.4 Read 1 7.3-Slave out.Data (RD_7X_OUT) 69
11.2.1.5 Read 4 7.3-Slave in.Data (RD_7X_IN_X) 69
11.2.1.6 Write 4 7.3-Slave out.Data (WR_7X_OUT_X) 70
11.2.1.7 Read 4 7.3-Slave out.Data (RD_7X_OUT_X) 70
11.2.1.8 WR_74_PARAM 70
11.2.1.9 RD_74_PARAM 71
11.2.1.10 RD_74_ID 72
11.2.1.11 RD_74_DIAG 72
11.2.2 Diagnosis Data 73
11.2.2.1 Overview of the Commands 73
11.2.2.2 Get Lists and Flags (Get_LPS, Get_LAS, Get_LDS, Get_Flags) (GET_LISTS) 73
11.2.2.3 Get Flags (GET_FLAGS) 75
11.2.2.4 Get Delta List (GET_DELTA) 76
11.2.2.5 Get List of Corrupted Slaves (GET_LCS) 77
11.2.2.6 Get List of Activated Slaves (GET_LAS) 77
11.2.2.7 Get List of Detected AS-i slaves (GET_LDS) 78
11.2.2.8 Get list of peripheral faults (GET_LPF) 79
11.2.2.9 Get List of Offline Slaves (GET_LOS) 79
11.2.2.10 Set List of Off-line Slaves (SET_LOS) 80
11.2.2.11 Get transm.err.counters (GET_TECA) 81
11.2.2.12 Get transm.err.counters (GET_TECB) 81
11.2.2.13 GET_TEC_X 82
11.2.2.14 Functional profiles 82
11.2.3 Functional profiles 83
11.2.3.1 "Safety at Work" List 1 83
11.2.3.2 "Safety at Work" Monitor Diagnosis 84
11.2.3.3 Integrated AS-i Sensors: Warnings 86
11.2.3.4 Integrated AS-i Sensors: Availability 86
11.2.4 Configuration of the AS-i Master 87
11.2.4.1 Overview of the Commands 87
11.2.4.2 Set Operation Mode (SET_OP_MODE: Set_Operation_Mode) 87
11.2.4.3 Store Actual Configuration (STORE_CDI) 88
11.2.4.4 Read Actual Configuration (READ_CDI) 89

11.2.4.5 Set Permanent Configuration (SET_PCD) 89
11.2.4.6 Get Extended Permanent Configuration (GET_PCD) 90
11.2.4.7 Set List of Projected Slaves (SET_LPS) 91
11.2.4.8 Get List of Projected Slaves (GET_LPS) 91
11.2.4.9 Store Actual Parameters (STORE_PI) 92
11.2.4.10 Write Parameter (WRITE_P) 93
11.2.4.11 Read Parameter (READ_PI: Read_Parameter) 93
11.2.4.12 Set_Permanent_Parameter (SET_PP) 94
11.2.4.13 Get_Permanent_Parameter (GET_PP) 94
11.2.4.14 Set Auto Address Enable (SET_AAE) 95
11.2.5 Other Commands 95
11.2.5.1 Overview of the Commands 95
11.2.5.2 IDLE 95
11.2.5.3 Read Input Data Image (READ_IDI) 96
11.2.5.4 Write Output Data Image (WRITE_ODI) 97
11.2.5.5 Read Output Data Image (READ_ODI) 97
11.2.5.6 Change Slave Address (SLAVE_ADDR) 97
11.2.5.7 Write AS-i Slave Extended ID1 (WRITE_XID1) 98
11.2.5.8 Set Offline Mode (SET_OFFLINE) 99
11.2.5.9 Release Data Exchange (SET_DATA_EX) 100
11.2.5.10 BUTTONS 100
11.2.5.11 FP_PARAM 100
11.2.5.12 FP_DATA 101
11.2.5.13 EXT_DIAG 101
11.2.5.14 INVERTER 102
11.3 Command Interface Examples 103
11.3.1 Reading analog Input Values 103
11.3.2 Store current Configuration to the AS-i Master 104
11.3.3 Store new Configuration for all Slaves 108

12 Commissioning Tools and Accessories 116
12.1 Windows Software AS-i Control Tools 116
12.2 DeviceNet Master Simulator 118

13 Appendix: Codes indicated by the Display 120

14 Appendix: The First Commissioning of AS-i 121

issue date 11.1.2005

1 Declaration of Conformity

The AS-i/DeviceNet gateways VAG-DN-K5, VBG-DN-K5-DMD have been developed and produced in accordance with the applicable European standards and directives.

Note

The corresponding of conformity can be requested from the manufacturer.

The manufacturer of the product, Pepperl+Fuchs Group in D-68301 Mannheim, possesses a certified quality assurance system in accordance with ISO 9001.

ISO**9001**

2 The Symbols Used

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.

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.

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

3 Safety

3.1 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.

3.2 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.
Connecting 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.

4 General Information

This operating instruction holds for the following devices of the Pepperl+Fuchs GmbH:

- VBG-DN-K5
- VBG-DN-K5-DMD

The AS-i/DeviceNet-Gateway serves to connect the AS-Interface to a superordinate DeviceNet. The Gateway acts as a complete Master for the AS-Interface and as a slave for the DeviceNet.

New AS-i Specification 2.1

The AS-i/DeviceNet Gateways already fulfil the new AS-i Specification 2.1. This means:

- Up to 62 AS-Interface slaves can be connected per 1 AS-i network
- The transfer of analog signals via AS-i is integrated in the Masters
- All further functions of the new specification as e.g. the diagnosis of the AS-i peripheral fault are implemented.

AS-i Scope Function

Diagnostics, which go far beyond the standard diagnostics facilitate the simple detection of the occassionally occuring configuration errors and further irritations towards the AS-i communication. So in case of an error the down time of machines can be minimized or you can initiate preventive maintenance.

Commissioning and Monitoring

The AS-i/DeviceNet Gateways can be commissioned with the help of the software "AS-i Control Tools" in combination with the DeviceNet Master Simulator. The EDS file is included in the package.

Commissioning, debugging and setting up of the AS-i parameters without the software can only be accomplished with the use of two push-buttons, the display and the LEDs directly on the system.

Gateways with Graphical Display

The AS-i Gateways with Graphical Display are a high-end solution to link AS-Interfaces with a superior DeviceNet system.

Simple and Fast Commissioning

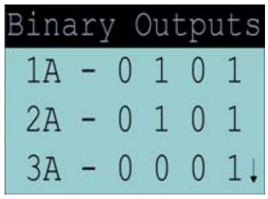
Using the AS-i Gateway with Graphical Display, the entire AS-i network can be commissioned and the connected periphery can be completely tested without DeviceNet Master. The new interactive graphic display also enables the user to complete all tasks which previously required the "AS-i Control Tools" software package. This allows for simpler and faster commissioning.

Addressing Unit within the AS-i Master

With the help of the new graphical display, the hand held unit is now obsolete. The slaves can now be easily addressed directly on the gateway. Slaves with extended address mode are detected automatically and are used only when allowed. This ensures that no two AS-i slaves with the same address will be on the same network.

Testing of Connected Periphery without Additional Test Tools

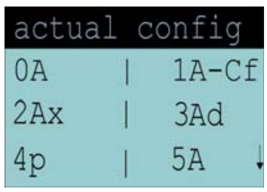
Once the AS-Interface is put into operation, the cabling and the connected sensors and actuators can be tested, inputs can be read and outputs can be set and even analog sensors and actuators can be checked just using the Gateway with Graphical Display.



On-board Diagnostics:

Configuration Fault, Periphery Fault

At a glance, the display shows the configuration faults (missing slave, additional slave detected, wrong slave type) as well as periphery faults, such as a short circuit at a sensor cable. This allows the user to get the proper information to solve the problem in the shortest amount of time.



Detection of Occasional Faults

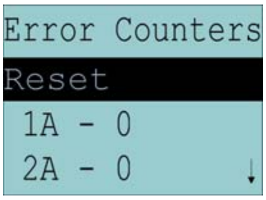
A list of slaves, which have previously caused an error, is also available through the graphical display. This can be very helpful in solving problems.

Scope Functions shown on the Display

While strange phenomena can occur as the AS-Interface gets to its limits (e. g. cable length >100 m, EMC problems), the AS-i Gateway with Graphical Display has on-board diagnostic tools. With the help of the AS-i error counters the user can

issue date 11.1.2005

easily check the quality of AS-i communications. The user can then test the impact of any actions taken.



Accessories

DeviceNet Master Simulator (VAZ-DN-SIM Part.No. 108447)

Software "AS-i Control Tools" (VAZ-SW-ACT32 Part. No. 99519)

5 Description

5.1 LED Indicators

Indicator	Color	Description
power	Green	Gateway power
MNS	Green/Red	Module/Network status
config err	Red	Configuration error
U ASI	Green	AS-i Cable sufficiently powered
ASI aktiv	Green	Normal operation active
prg enable	Green	Automatic addressing enable
prj mode	Yellow	Configuration mode

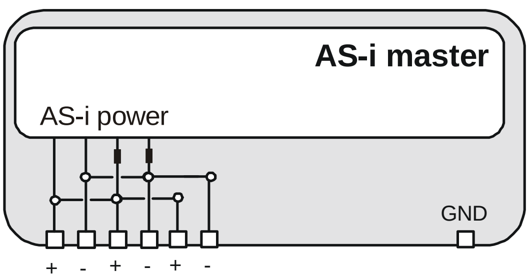
5.2 Power Supply Concepts and AS-i Connection Techniques

Note

The AS-i masters with master power supply A do not need a voltage supply of their own. They can be powered completely out of the AS-i line (the power consumption is about 200 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 AS-i power supply is switched on, the master starts to operate.

5.2.1 Single Masters in IP20 with AS-i Power Supply A

5.2.1.1 Connections of the AS-i/DeviceNet Gateway with graphic Display



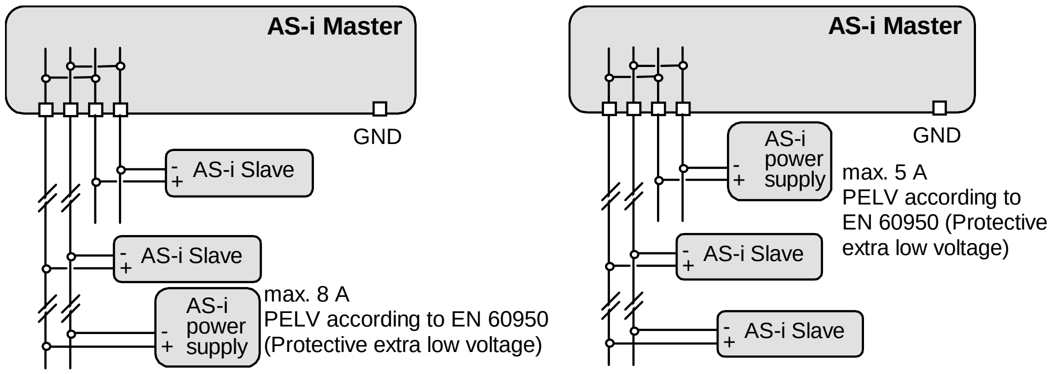
The AS-i/DeviceNet gateway with graphic display, power supply A, has an additional, second AS-i connection.

The terminals have the following functions:

+	"AS-i +", Actuator Sensor Interface, positive terminal
-	"AS-i -", Actuator Sensor Interface, negative terminal
GND	Ground terminal, used for better EMC. Should be connected with a short wire to machine GND.

Warning

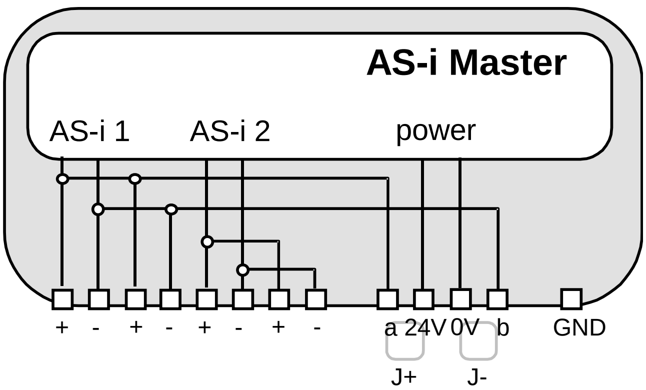
Connection variations of the AS-i power supply:



Attention

When using these circuit variations, a maximum current of 5 A must not be exceeded.

5.2.2 Double Master in IP20 with Power Supply A



The terminals have the following functions:

+	"AS-i +", Actuator Sensor Interface 1 or 2, positive terminal These terminals are connected internally with point a of jumper "J+".
-	"AS-i -", Actuator Sensor Interface 1 or 2, negative terminal These terminals are connected internally with point b of jumper "J-".
24V	Master power supply, positive terminal (18 - 31.6 V DC)
0V	Master power supply, negative terminal
GND	Ground terminal, used for better EMC. Should be connected with a short wire to machine GND.
J+, J-	Jumpers for selecting the power supply of AS-i

issue date 11.1.2005

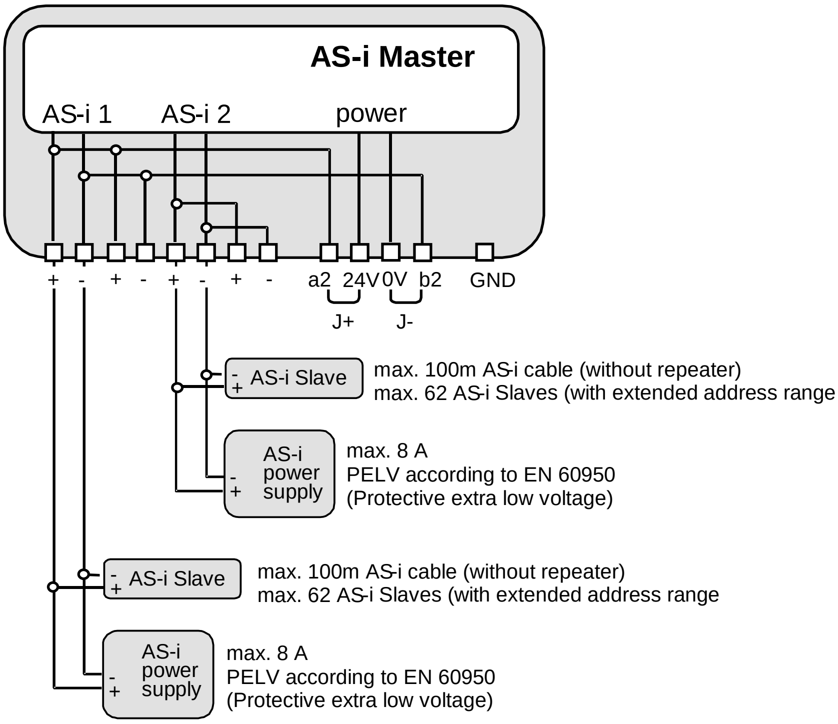
jumpers closed:

The AS-i master is powered out of AS-i circuit 1.
Master power supply and AS-i network are then decoupled with coils.

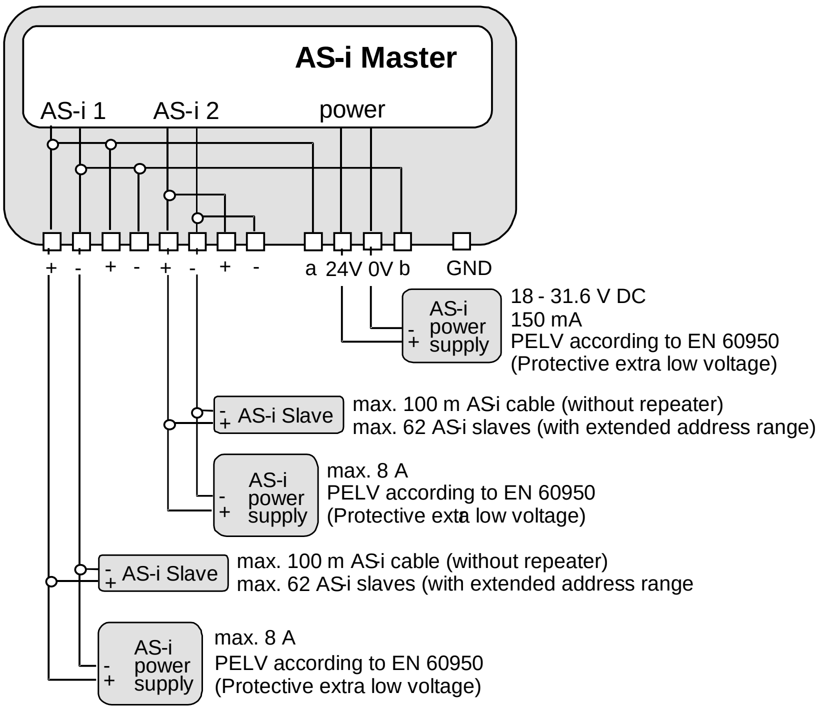
jumpers open:

With the jumpers open (or missing), the AS-i master must be powered by a separate 24 V DC power supply.

5.2.2.1 Power supply out of AS-i circuit 1



5.2.2.2 Operation with separate 24 V DC Power Supply

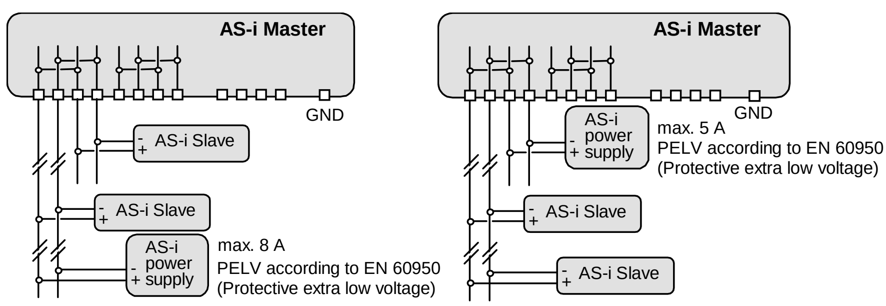


Warning

Do not use a seperate 24 V DC power supply without having removed the jumpers

Connection variations for the AS-i circuits (showed here for one AS-i circuit)

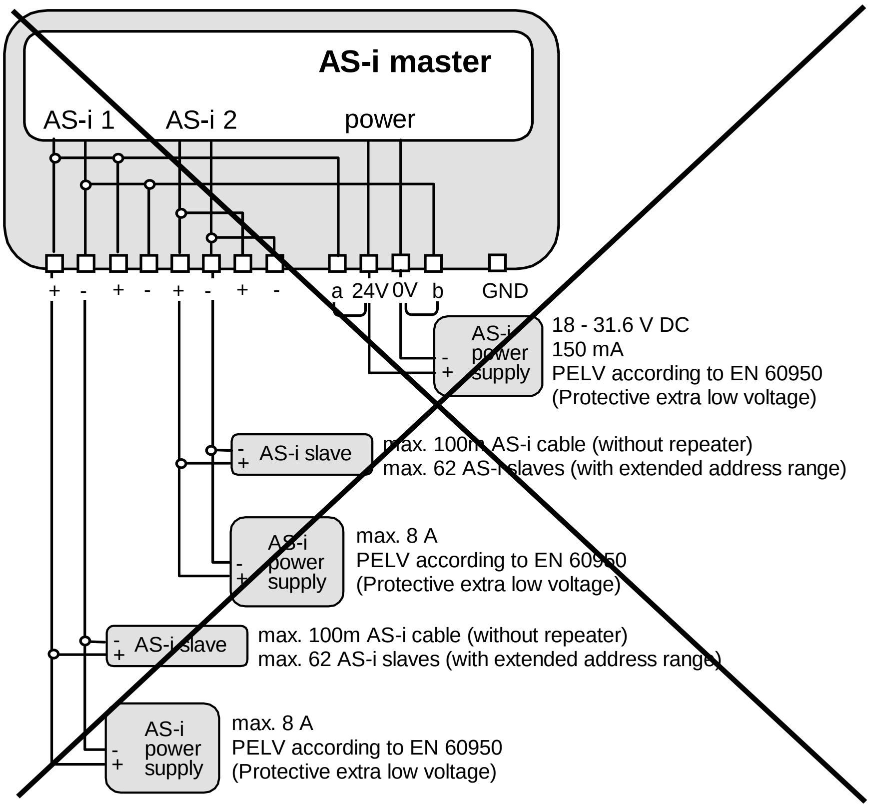
Warning



Attention

In the wiring schemes above the current through the AS-i master must not exceed 5 A.

issue date 11.1.2005



5.3 AS-i Network Connection

At the base of the gateway are 3 terminals for the connection of the AS-i network. The individual terminals are as follows:

Terminal	Description
+ (double)	The positive terminal for the AS-i network
- (double)	The negative terminal for the AS-i network
GND	Ground terminal

5.4 DeviceNet Connection

Terminal	Signal	Function	Color
1	V+	DeviceNet Power	Red
2	CAN_H	Signal High	White
3	SHIELD	Shield	n/a
4	CAN_L	Signal Low	Blue
5	V-	DeviceNet Power	Black

5.5 Display and Operating Elements

5.5.1 LEDs of the Single Masters and Double Masters with Graphical Display

power	The master's power supply is sufficient.	
MNS	Red LED flashes:	no CAN communication in “Pre Operational Mode”
	Green LED flashes:	CAN communication node in “Pre Operational Mode”
	Green LED:	CAN communication node in “Operational Mode”
config err	Configuration error At least one configured slave is missing, 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.	
U ASI	The AS-i circuit is sufficiently powered.	
ASI active	Normal operation active.	
prg enable	Automatic address programming enabled. Exactly one slave is missing in protected operating mode. The slave can be replaced by another slave of the same type with address zero. The master addresses the new slave to the faulty address and thus eliminates the configuration error.	
prj mode	The AS-i master is in configuration mode.	

5.5.2 Push-Buttons

The push-buttons cause the following:

mode	Switching between configuration mode and protected operating mode and saving the current AS-i configuration as the nominal configuration.
set	Selecting and assigning the address to a slave.
OK, ESC	Changing to graphical mode. Have a look at chapter 8 (master with graphical display only).

The detailled operation is described in chapter 7.

issue date 11.1.2005

6 Configuration

6.1 Setting DeviceNet Addres and Baud Rate

To set the DeviceNet address and baud rate, refer to the front panel of the gateway. Locate the two buttons on the front panel marked "*mode*" and "*set*". By pressing both buttons simultaneously for approximately 5 seconds, the actual DeviceNet address will appear on the LCD display. DeviceNet address can be changed by pressing the "*set*" button until the desired DeviceNet address has been reached. To store the DeviceNet address, press the "*mode*" button. Now the LCD screen displays a 0, 1, or 2. See the chart below for the meaning of the code.

CODE	BAUD RATE
0	125k Baud
1	250k Baud
2	500k Baud

Press the "*set*" button until the desired code is reached. Again press the "*mode*" button to store the baud rate. The setting of the DeviceNet address and baud rate is complete.

The default address is 63 and the default baud rate is 125 kBaud.

6.2 I/O Data Interpretation

Input data comes from Assembly Object Instance 100 (single channel) or 118 (double channel).

Output data comes from Assembly Object Instance 118 (single channel) or 154 (double channel).

These bytes of data are as follows:

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			

issue date 11.1.2005

byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
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
Off-Line: 0 = OnLine, 1 = Off-Line
LOS-master-bit 0 = Off-Line by ConfigError deactivated
1 = Off-Line by ConfigError activated

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 reset.

issue date 11.1.2005

7 Operating the AS-i/DeviceNet Gateway

7.1 Peculiarities with Double Masters

Note

*In **protected mode**, the displays of the double masters switch between AS-i circuit 1 and 2 every two seconds.*

In **configuration mode** all detected AS-i slaves are displayed before the display switches to the other AS-i circuit.

The operation of the push-buttons always refers to the currently displayed AS-i circuit (LED AS-i 1/AS-i 2). After the push-button is pressed, the display keeps switched to the respective AS-i circuit until the operation is finished or the operator has not interfered for 10 seconds.

7.2 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:

40 Offline Phase
The AS-i master initializes - there is no data communication happening on theAS-i.

Attention

If the AS-i circuit is insufficiently powered ("U AS-i" does not light up).

As long as there is no communication, the outputs remain to 0. If the communication breaks off and the watchdog is switched on, all outputs will be set to 0 again.

41 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.

42[1] 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.

43[2] 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

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.

slaves. During normal operation, the system keeps the maximum cycle time of 5 milliseconds.

7.3 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 gateway to configuration mode. While in configuration mode, the yellow "prj mode" LED lights up.

All "A" slaves followed by all "B" slaves are being displayed in ascending order, switching every 0,5 seconds. If a "B" slave is displayed, the "AS-i active" LED flashes. 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.

7.4 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.

7.4.1 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.

issue date 11.1.2005

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.

7.4.2 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".

7.5 Assigning an AS-i Address in Configuration Mode

AS-i can be put into operation in a very comfortable manner by using the Windows software AS-i Control Tools (adressing directly or with the the AS-i address assistant, see chapter 12.1).

Furthermore, a handheld addressing device can be used.

If you have neither a PC nor a hand held addressing device, address assigning of the AS-i slaves is also possible with the AS-i/DeviceNet gateway using the push-buttons.

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

7.5.1 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).

issue date 11.1.2005

Any errors will be displayed by their error codes according to chapter 13. Otherwise, the detected slaves are displayed again as described in chapter 7.3.

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 adrress on the AS-i circuit, since this would cause malfunctions.

7.5.2 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.

7.6 Programming the Address in Case of Configuration Errors

7.6.1 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.

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

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.

7.6.2 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 (using the AS-i Control Tools) 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 7.4). 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.

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 13). When all faulty assignments are eliminated, the display will be empty.

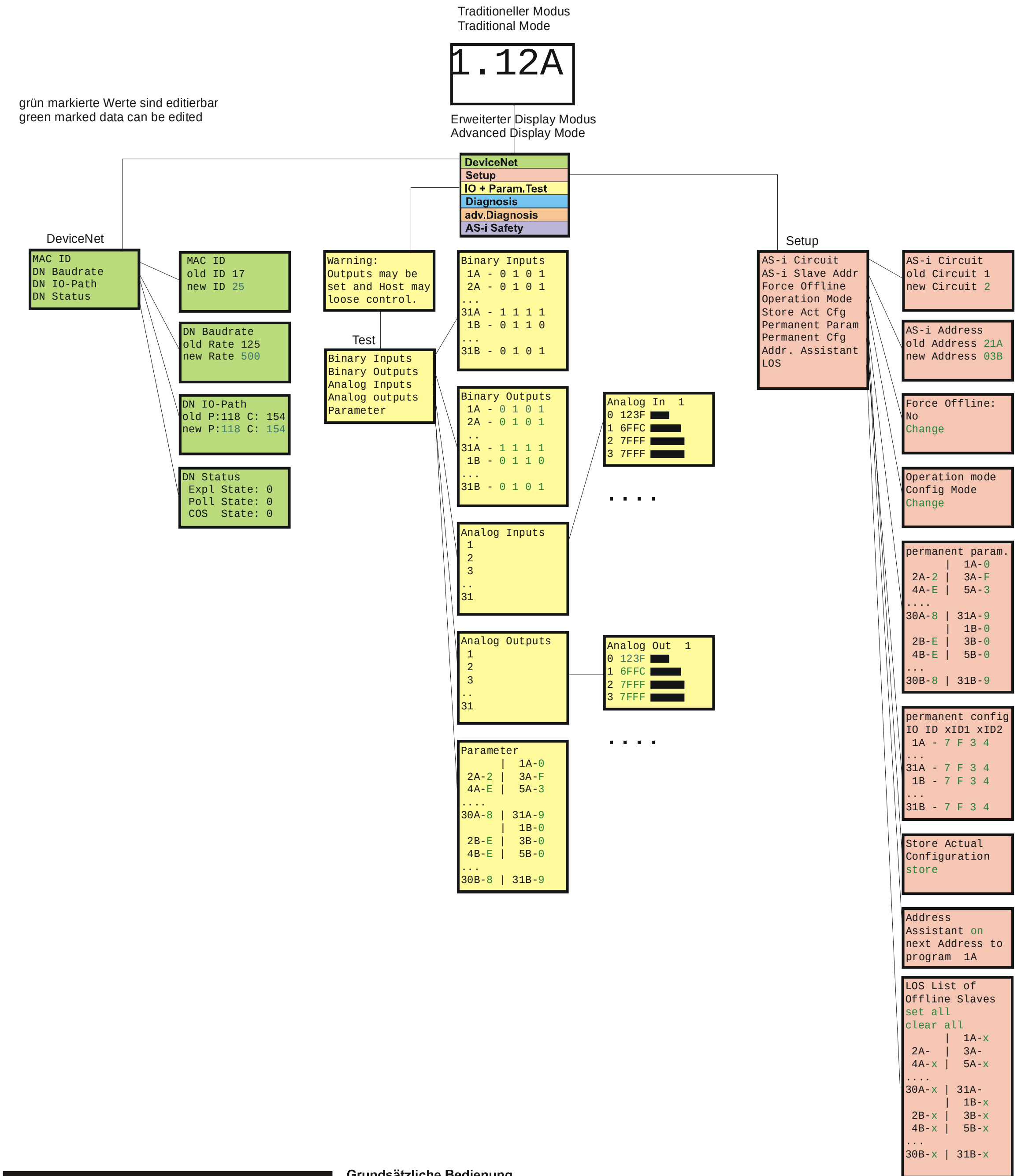
7.7 Error Messages

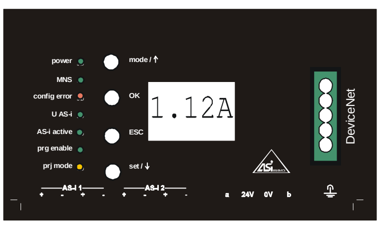
Attention

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 13.

8 Operating by Full-graphic Display

Inbetriebnahme/Commissioning





Grundsätzliche Bedienung

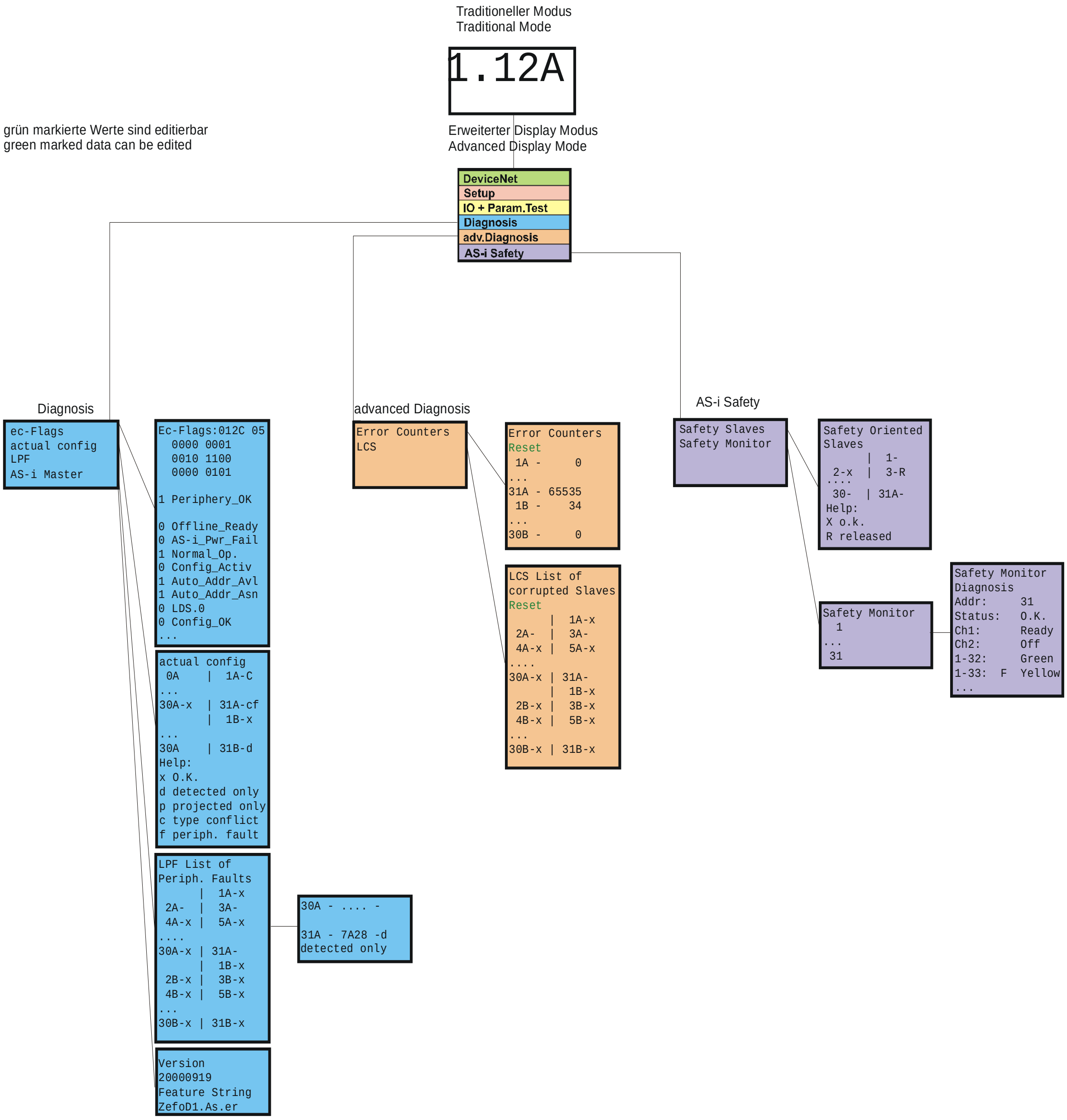
Das Gerät startet im traditionellen Modus. Mit ESC oder OK kann zwischen beiden Modi gewechselt werden.
Im Erweiterten Modus wird ein Cursor mit den beiden Pfeil-Tasten bewegt. OK bringt ins nächsthöhere Menü (in der Zeichnung weiter nach rechts). ESC bringt zurück ins vorherige Menü.
Wenn Werte editiert werden, werden sie zunächst mit dem Cursor markiert, dann mit OK ausgewählt, mit den Pfeiltasten verändert und schließlich mit OK übernommen. ESC bricht das Editieren ab.

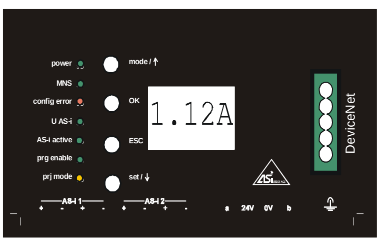
Basic Operation

The device starts in the traditional mode. You can switch between the two modes with ESC or OK.
In the advanced mode the cursor is moved by both arrow buttons. Pushing OK puts you to the superior menue (in the drawing one step to the right side). ESC puts you back to the previous menue.
To edit data you first mark them with the cursor and then select them with OK, change them with the arrow buttons and finally apply them with OK. Pushing ESC cancels the editing.

issue date 11.1.2005

Fehlersuche/Diagnostics





Grundsätzliche Bedienung
Das Gerät startet im traditionellen Modus. Mit ESC oder OK kann zwischen beiden Modi gewechselt werden.
Im Erweiterten Modus wird ein Cursor mit den beiden Pfeil-Tasten bewegt. OK bringt ins nächsthöhere Menü (in der Zeichnung weiter nach rechts). ESC bringt zurück ins vorherige Menü.
Wenn Werte editiert werden, werden sie zunächst mit dem Cursor markiert, dann mit OK ausgewählt, mit den Pfeiltasten verändert und schließlich mit OK übernommen. ESC bricht das Editieren ab.

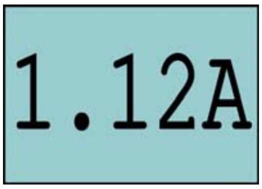
Basic Operation
The device starts in the traditional mode. You can switch between the two modes with ESC or OK.
In the advanced mode the cursor is moved by both arrow buttons. Pushing OK puts you to the superior menue (in the drawing one step to the right side). ESC puts you back to the previous menue.
To edit data you first mark them with the cursor and then select them with OK, change them with the arrow buttons and finally apply them with OK. Pushing ESC cancels the editing.

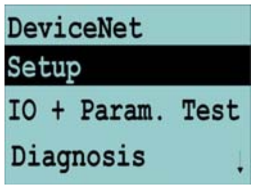
issue date 11.1.2005

Warning

In the classical mode, it is possible to change settings while the device is in operation. This can lead to failure of the plant (e. g. changing the address of an AS-i slave).

In the full-graphic mode, however, the settings are protected, as long as the superior fieldbus (DeviceNet) is running.





The device starts in the classical mode, like the former AS-i Masters with two-digit display (see chapter 7). Press the buttons ESC or OK to switch to the full graphic mode. To return to the classical mode, simply press the ESC-button several times.

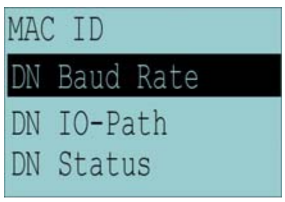
In full graphic mode, the selection can be moved up and down with the arrow buttons.

Pressing OK will switch to the selected function or menu (one step to the right on the drawing, page 24). Pressing ESC will switch back to the previous menu.

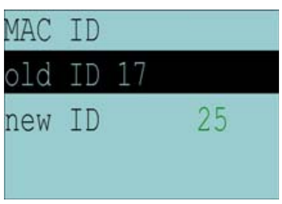
To edit data values hightlight them with the selection bar, press OK, then change them with the arrow-buttons and confirm with OK. The ESC-button cancels the editing process.

All possible addresses are displayed one after the other from 1A to 31A and from 1B to 31B. Data for single slaves are displayed at the addresses 1A - 31A.

8.1 DeviceNet (Fieldbus Interface)



8.1.1 DeviceNet MAC ID



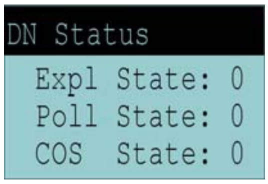
This function is for setting and changing the DeviceNet address.

The number behind "old ID" shows the actual address. By selecting "new ID", this address can be changed.

issue date 11.1.2005

8.1.2 DeviceNet Baud Rate

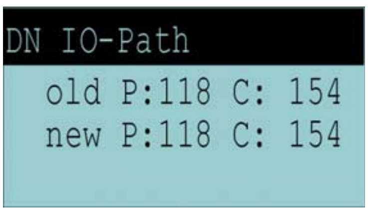
8.1.3 DeviceNet Status



The function DeviceNet status indicates if and how many connections are active on each DeviceNet channel. Following status are indicated:

- 0 = nonexistent
- 1 = configuring
- 2 = waiting of connection ID
- 3 = established
- 4 = timed out
- 5 = deferred delete

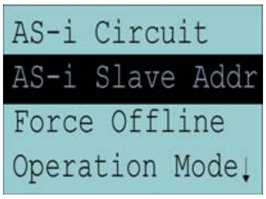
8.1.4 DeviceNet I/O-Path



With this function the DeviceNet POLL Connection Production/Consume Path and the Cyclic/COS Production Path can be easily modified. The displayed values are the assembly instances of the Production and Consume Path. If the current path values are inconsistent for this function the old values are marked with "---".

"P" modifies the Production Path of the POLL and the Cyclic/COS connection. "C" the Consume Path of the POLL Connection.

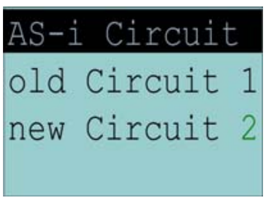
8.2 Setup (Configuration of the AS-i Circuit)



Within the menu "Setup", one of the following submenues can be chosen:

- AS-i Circuit
- AS-i Slave Addr (AS-i Slave Address)
- Force Offline (switch AS-i Master offline)
- Operation Mode
- Store Act Cfg (store actual detected configuration)
- Permanent Param (projected parameter)
- Permanent Cfg (projected configuration data)
- Addr. Assistant (address assistent)
- LOS (list of offline-slaves)

8.2.1 AS-i Circuit



This function is available for devices with two AS-i masters only.

It makes possible to change the AS-i circuit that is currently active for being operated.

The number behind "old Circuit" shows the active AS-i circuit. By selecting "new circuit", the other AS-i circuit can be chosen to be active.

8.2.2 AS-i Slave Addr (Change AS-i Slave Address)

This function sets and changes the addresses of the AS-i slaves. This function replaces the handheld AS-i address programming device.

"Old Address" shows the address of the first detected AS-i slave on the AS-i circuit. Please note that you must have selected the desired AS-i circuit when you operating a device with two AS-i circuits (see chapter 8.2.1).

If "old Address" is selected, the next detected AS-i slave can by selected by pressing the "OK" button. The new address for the AS-i slave has to be set with "new Address".

If an error occurs while addressing a slave, one of the following error messages is displayed for about 2 seconds:

Failed: SND:	slave with old address has not been detected.
Failed: SD0:	slave with address zero has been detected.
Failed: SD2:	slave with new address has been detected.
Failed: DE:	could not delete old address.
Failed: SE:	error setting new address.
Failed: AT:	new address could be stored temporarily only.
Failed: RE:	error reading the extended ID-code 1.

8.2.3 Force Offline (switch AS-i Master offline)

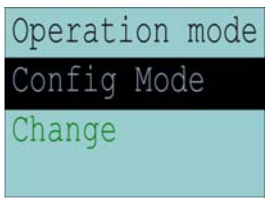
This function shows the current state of the AS-i Master:

Yes:	AS-i Master is offline.
No:	AS-i Master is online.

With "Change", this state can be modified.

Switching the AS-i master offline puts the AS-i circuit into the safe state. The AS-i master has to be offline if an AS-i slave should be addressed via the IR-interface.

8.2.4 Operation Mode



This function shows the current operation mode of the AS-i Master:

Protected Mode:	protected mode
Config Mode:	configuration mode

With "Change" you can switch to the other operation mode.

Only in configuration mode parameters and configuration data can be stored.

8.2.5 Store Act Cfg (Store Actual Detected Configuration)

This function can only be executed in configuration mode.

This function enables you to store the configuration of all slaves wich are connected and detected on the selected AS-i circuit.

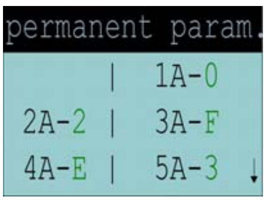
If "Store" was successful, the LED "config error" is off. The configuration is stored, the configuration error has been eliminated.

If one of the connected slaves has a peripheral fault, the LED "config error" will flash.

If the AS-i Master is in protected mode, the following error message will appear: "Failed No Config Mode"

If an AS-i slave with address zero exists, stroring the configuration will be confirmed with "OK". However, the configuration error remains because address zero is not a valid operating address for storing a slave.

8.2.6 Permanent Param (Projected Parameter)



This function allows you to set the permanent parameters. A list of all slaves is displayed. The parameter is shown as a hexadecimal value behind the slave address.

8.2.7 Permanent Cfg (Projected Configuration Data)

With this function, the projected configuration data can be projected. The values for the configuration data are displayed behind the slave address in the following order:

IO (I/O-configuration) ID (ID-configuration) xID1 (extended ID1)
xID2 (extended ID2).

8.2.8 Addr. Assistant (Address Assistant)

The AS-i address assistant helps you in setting up the AS-i circuit quickly. Once you have stored the AS-i configuration, the AS-i address assistant addresses a new AS-i slave with address zero to the desired address.

Selecting "Assistant on" or "Assistant off" switches the AS-i address assistant on or off. The current state of the AS-i address assistant is displayed:

Assistant on: AS-i address assistant is switched on.
Assistant off: AS-i address assistant is switched off.

Procedure:

1. This can be done very comfortably with the Windows software AS-i Control Tools (Master | Write configuration to the AS-i Master ...), or directly with the full graphic display (see chapter 8.2.7).
2. All AS-i slaves have to be addressed to 0 or to the desired address. The slaves must be disconnected from the AS-i circuit.
3. Start the AS-i address assistant.
4. Now connect the AS-i slaves one after the other. The last line of the display of the AS-i address assistant shows which AS-i slave has to be connected next.

8.2.9 LOS (List of Offline Slaves)

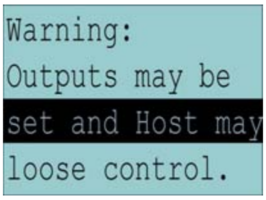
See also "Advanced Diagnostics for AS-i Masters", chapter 9.

With "Clear all" and "Set all" you can delete or set a single bit for each AS-i slave address.

empty field: LOS-bit deleted
X: LOS-bit set

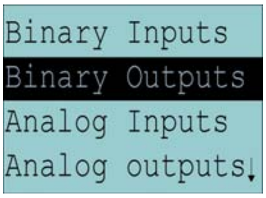
issue date 11.1.2005

8.3 IO + Param. Test (Testing AS-i In- and Outputs as well as reading and writing AS-i Parameters)



Before you can switch to this menu, the following warning will occur:

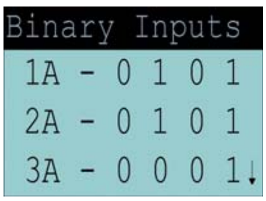
"Warning: Outputs may be set and Host may loose control."



The menu "IO + Param.Test" enables you to choose one of the following submenues:

- Binary Inputs
- Binary Outputs
- Analog Inputs
- Analog Outputs
- Parameter

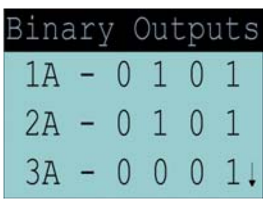
8.3.1 Binary Inputs



This list shows the state of the binary inputs for all AS-i slaves.

0: input deleted
1: input set

8.3.2 Binary Outputs



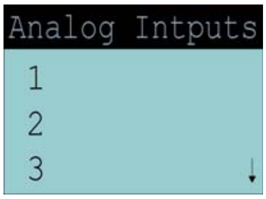
This function shows the state of the binary outputs for all AS-i slaves.

0: output deleted
1: output set

The binary outputs can be changed after selecting the desired AS-i slave.

issue date 11.1.2005

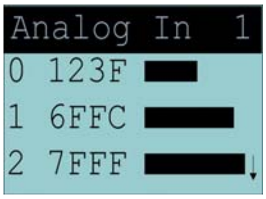
8.3.3 Analog Inputs



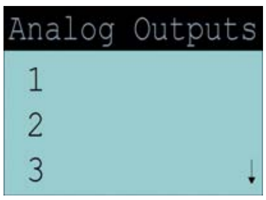
This function shows the state of the analog inputs for all AS-i slaves.

The display is as follows:

AS-i slave address, hexadecimal 16-bit-value, bar display indicating the input or output value



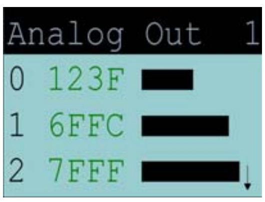
8.3.4 Analog Outputs

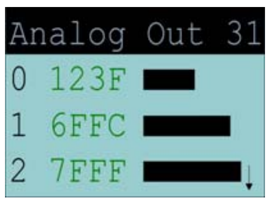


This function shows the state of the analog outputs for all AS-i slaves.

The display is as follows:

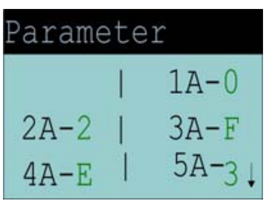
AS-i slave address, hexadecimal 16-bit-value, bar display





The analog outputs can be changed after selecting the desired AS-i slave.

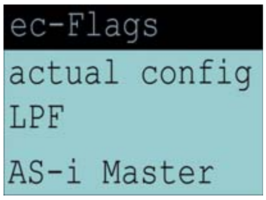
8.3.5 Parameter



This function shows the hexadecimal value of the current AS-i parameters for all AS-i slaves.

The actual AS-i parameters can be changed after selecting the desired slave address.

8.4 Diagnosis (Normal AS-i Diagnosis)



The menu "Diagnosis" enables you to choose one of the following submenues:

- EC-Flags (Execution control flags)
- Actual Config (actual configuration)
- LPF (list of periphery faults)
- AS-i Master (Info)

8.4.1 EC-Flags (Execution Control Flags)

This function shows the EC-flags hexadecimal, binary and as single bits beginning with the lowest-order bit.

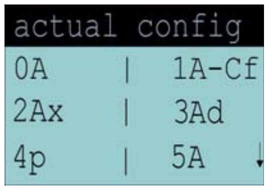
Byte 1:
Bit 0: 1 = Periphery_OK

Byte 2:
Bit 0: 0 = Offline_Ready
Bit 1: 0 = AS-i Pwr Fail
Bit 2: 1 = Normal_Op.
Bit 3: 0 = Config_Active
Bit 4: 1 = Auto_Addr_Avl
Bit 5: 1 = Auto_Addr_Asn
Bit 6: 0 = LDS.0
Bit 7: 0 = Config_OK

Byte 3:
Bit 0: 1 = Auto_Addr_Ena
Bit 1: 1 = Offline
Bit 2: 1 = Data_Exch_Act

issue date 11.1.2005

8.4.2 Actual Config (Actual Configuration)



This function shows the state of the actual configuration of the individual AS-i slaves.

At the end of the list there is a help text describing the abbreviations:

X (O.K.): The configuration data of the detected AS-i slave matches the projected configuration data.

D (Detected Only): An AS-i slave is detected at this address, but not projected.

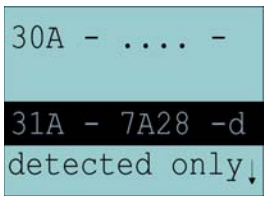
P (Projected Only): An AS-i slave is projected at this address, but not detected.

C (Type Conflict): The configuration data of the detected AS-i slave does not match the projected configuration data. The actual detected configuration of the connected AS-i slave is displayed.

F (Periph. Fault): The AS-i slave has a peripheral fault.

After selecting the desired AS-i slave address the values for the actual configuration data are displayed behind the respective address in the following order:

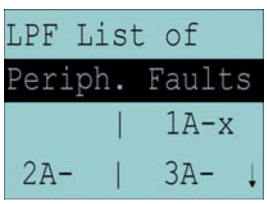
IO (I/O-configuration) ID (ID-configuration) xID1 (extended ID1) xID2 (extended ID2).



Furthermore the state of the configuration is displayed in plaintext.

If no AS-i slave is detected and no AS-i slave is projected at a certain address, four dots instead of the configuration data are displayed.

8.4.3 LPF (List of Periphery Faults)

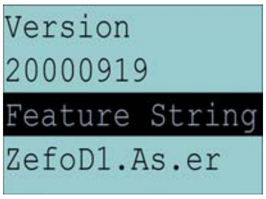


The list shows AS-i slaves, which have released a peripheral fault.

empty field: periphery O.K.
X: peripheral fault

issue date 11.1.2005

8.4.4 AS-i Master (Info)



This function shows information about the version and the features of the AS-i master.

Version xxxxxxxx (date of the firmware)

Feature String xxxxxxxxxxxxxxxx

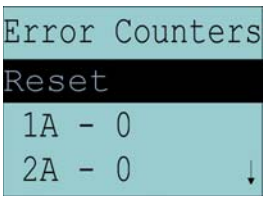
8.5 Adv. Diagnosis (Advanced AS-i Diagnosis)

See also "Advanced Diagnostics for AS-i Masters", chapter 9.

In the menu "Adv. Diagnosis", the following submenues can be found:

- Error Counters
- LCS (list of slaves, that produced a configuration error)

8.5.1 Error Counters



This list shows the error counter for each single AS-i slave.

Furthermore the number of power failures on AS-i (APF) is displayed.

By selecting "Reset", the error counters are reset to 0.

8.5.2 LCS (List of Slaves having caused a Configuration Error)

This list shows for each single AS-i slave wether at least one configuration error was released through an errorneous telegram transmission. This function is especially important if the configuration error only occurs short-time.

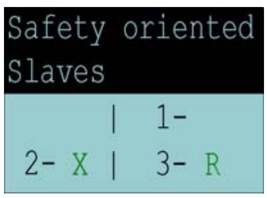
empty field: no error
X: AS-i slave caused a configuration error.

8.6 AS-i Safety

This function shows information about the safety slaves and the safety monitor:

- Safety Slaves
- Safety Monitor

8.6.1 Safety Slaves



This list shows the "safety-directed input slaves" ("AS-i Safety at Work"), by which the safety function is released

empty field
X: o.k.
R: released

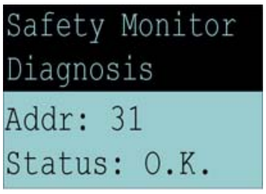
The slaves according to profile S-7.B or S-0.B are listed here, by which all 4 bits in the IDI are deleted. Therefore slaves with 2 contacts are entered only if both contacts are released.

Since the safety function of a safety-directed input slave can be released even if the slave does exchange no data with the AS-i master, the list may be utilized only in combination with the ec-flags.

Only CDI and IDI are used for creating this list. Safety slaves which are projected but not existing, and existing slaves sending a wrong code are not listed here.

This list is not constantly updated; it is created from the image of the digital inputs IDI, if required.

8.6.2 Safety Monitor



The AS-i safety monitor reads the diagnosis data out of the AS-i safety monitor and shows this data in the display. The meaning of the shown diagnosis can be seen in the description of the safety monitor.

issue date 11.1.2005

9 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.

9.1 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 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 listed in the *LCS* at the position of AS-i slave with address 0.

Note

With every read access the LCS will be deleted.

Note

The last short-time configuration error can also be displayed on the AS-i master:
Pressing the "set" button of the AS-i master shows the AS-i slave which was responsible for the last short-time configuration error. If there was a short-time AS-i power failure the display shows "39" after pressing the "set" button.
This function is only available if the device is in the normal operation mode of the protected mode (display empty) or in the off-line-phase.

9.2 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 count up every time a currupted data telegram has been found. This makes possible to judge the quality of the AS-i network, even if only a few corrupted telegrams occured 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.

9.3 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. 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.

10 DeviceNet Details

10.1 DeviceNet Information

The AS-i/DeviceNet gateway operates as a Group 2-Only slave device on the DeviceNet network, supporting polled I/O and explicit messaging. It does not support strobed I/O.

This appendix defines DeviceNet message types, class services and object classes that are supported by the AS-i/DeviceNet gateway.

10.1.1 DeviceNet Message Types

The gateway supports the following message types:

CAN Identifier Field	Group 1 Message Type
01101xxxxxx	Slave's I/O Change of State or Cyclic Message
01111xxxxxx	Slave's I/O Poll Response or Change of State/Cyclic Acknowledge Message

CAN Identifier Field	Group 2 Message Type
10xxxxxx111	Duplicate MAC ID Check Messages
10xxxxxx110	Unconnected Explicit Request Messages
10xxxxxx101	Master I/O Poll Command Message
10xxxxxx100	Master Explicit Request Message
10xxxxxx010	Master's I/O Poll/Change of State/Cyclic Message

Note

xxxxxx = AS-i/DeviceNet gateway node address

10.1.2 DeviceNet Class Services

The gateway supports the following class services and instance services:

Service Name	Service Code
Reset	0x05
Delete	0x09
Get_Attribute_Single	0x0E
Set_Attribute_Single	0x10
Allocate Master/Slave_Connection_Set	0x4B
Release Master/Slave_Connection_Set	0x4C

issue date 11.1.2005

10.2 Object Modelling

According to the DeviceNet philosophy, one should model the properties of the physical device in DeviceNet Objects. For an AS-i Master, the objects could be:

- An "AS-i Master Object" which contains the properties of the Master itself and of the whole AS-i circuit, e.g. ec-flags, lists and functions for slave programming and so on.
- "AS-i Slave Objects", one for every AS-i slave which contains the properties of the individual AS-i slave, e.g. parameters, configuration and so on.

Therefore following Object List ist existing (overview):

Class Code	Object Name	Number of Instances
0x01	Identity	1
0x03	DeviceNet	1
0x04	Assembly	72
0x05	Connection	4
0x15	Parameter Object	1
0x64	AS-i Master	1 for every AS-i Circuit
0x65	AS-i Slave	64 for every AS-i Circuit
0x66	IO-Data	1 for every AS-i Circuit
0x67	Advanced Diagnostics	1 for every AS-i Circuit
0x68	Short Command Interface	1
0x69	Long Command Interface	1

10.2.1 Identity Object

Class Code: 1
Number of Instances: 1

Instance Attributes

Attribute ID	Access Rule	Name	Value
100	Get	Vendor	645
101	Get	Device Type	100
102	Get	Product Code	Single master: AS-i/DeviceNet gateway BW1334 Double master: AS-i/DeviceNet gateway BW1335
103	Get	Revision	2.1
104	Get	Status	see chart below
105	Get	Serial Number	unique 32 bit number
106	Get	Product Name	AS-i/DeviceNet Gateway
109	Get/Set	Heartbeat Interval	

issue date 11.1.2005

Status

bit 0	owned	0 = not owned 1 = owned (group 2 allocated to master)
bit 1	reserved	always 0
bit 2	configured	always 0
bit 3	reserved	always 0
bit 4-7	vendor specific	all 0
bit 8	minor cfg fault	0 = no fault 1 = minor configuration fault
bit 9	minor device fault	0 = no fault 1 = minor device fault
bit 10	major cfg fault	0 = no fault 1 = major configuration fault
bit 11	major device fault	0 = no fault 1 = major device fault
bit 12,13	reserved	always 0
bit 14,15	reserved	always 0

Common Services

Service Code	Class	Instance	Service Name
0x05	No	Yes	Reset
0x10	Yes	Yes	Get_Attribute_Single
0x0E	Yes	Yes	Get_Attribute_Single

10.2.2 DeviceNet Object

Class Code: 3
Number of Instances: 1

Instance Attributes

Attribute ID	Access Rule	Name	DeviceNet Data Type	Data Value
1	Get/Set	MAC ID	USINT	Range 0-63
2	Get/Set	Baud Rate	USINT	Range 0-2
3	Get/Set	BOI	BOOL	Range 0-1
4	Get/Set	Bus-off Counter	USINT	Range 0-255
5	Get	Allocation Information Allocation Choice Byte Master's Node Address	Structure of: BYTE USINT	0-63=Master Address 255=unallocated

Common Services

Service Code	Class	Instance	Service Name
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single
0x4B	No	Yes	Allocate_M/S_Connection_Set
0x4C	No	Yes	Release_M/S_Connection_Set

10.2.3 Assembly Object

Class Code 4

The Assembly Object bundles data from the application objects.

The Assembly Object Instances consist of (in case of a double master):

one of …

- A-slaves and single slaves from circuit 1
- Single, A- and B-slaves (all slaves) from circuit 1
- A-slaves and single slaves from both circuits
- Single, A- and B-slaves (all slaves) from both circuits
- No analog data
- Analog data from slaves 29 … 31 from circuit 1
- Analog data from slaves 29 … 31 from both circuits
- No command interface
- Short command interface
- Long command interface

Attribute ID	Access Rule	Name	Data Value
3		Data Item(s)	

72 Instances. Instances 100 … 135 can only be read, while instances 136 … 171 can be read and written.

Warning

Don't use "Change of State" for Assembly, if you have slaves with fast input change (like 7.x or safety slaves)!

Instance ID	Data Item 1	Data Item 2	Data Item 3	Size (Members, Byte)	Complementary ID
100	AS-i Circuit 1, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64]			1, 16	136
101	AS-i Circuit 1, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64]		Short Command Interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	2, 28	137
102	AS-i Circuit 1, Input Data Image, single and A-slaves, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x64]		Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	2, 52	138
103	AS-i Circuit 1, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64]	AS-i Circuit 1, Analog Input Data slave 29 … 31 path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]		4, 40	139
104	AS-i Circuit 1, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]	Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	5, 52	140
105	AS-i Circuit 1, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]	Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	5, 76	141

Instance ID	Data Item 1	Data Item 2	Data Item 3	Size (Members, Byte)	Complementary ID
106	AS-i Circuit 1, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]		7, 64	142
107	AS-i Circuit 1, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]	Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	8, 76	143
108	AS-i Circuit 1, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]	Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	8, 100	144
109	AS-i Circuit 1, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65]			2, 32	145

Instance ID	Data Item 1	Data Item 2	Data Item 3	Size (Members, Byte)	Complementary ID
110	AS-i Circuit 1, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65]		Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	3, 44	146
111	AS-i Circuit 1, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65]		Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	3, 68	147
112	AS-i Circuit 1, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]		5, 56	148
113	AS-i Circuit 1, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]	Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	6, 68	149
114	AS-i Circuit 1, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]	Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	6, 92	150

Instan-ce ID	Data Item 1	Data Item 2	Data Item 3	Size (Members, Byte)	Comple-mentary ID
115	AS-i Circuit 1, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]		8, 80	151
116	AS-i Circuit 1, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]	Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	9, 92	152
117	AS-i Circuit 1, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]	Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	9, 116	153
118	AS-i Circuit 1 + 2, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64]			2, 32	154

issue date 11.1.2005

Instance ID	Data Item 1	Data Item 2	Data Item 3	Size (Members, Byte)	Complementary ID
119	AS-i Circuit 1 + 2, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64]		Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	3, 44	155
120	AS-i Circuit 1 + 2, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64]		Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	3, 68	156
121	AS-i Circuit 1 + 2, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]		5, 56	157
122	AS-i Circuit 1 + 2, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]	Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	6, 68	158
123	AS-i Circuit 1 + 2, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]	Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	6, 92	159

Instance ID	Data Item 1	Data Item 2	Data Item 3	Size (Members, Byte)	Complementary ID
124	AS-i Circuit 1 + 2, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]		8, 80	160
125	AS-i Circuit 1 + 2, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]	Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	9, 92	161
126	AS-i Circuit 1 + 2, Input Data Image, single and A-slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]	Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	9, 116	162
127	AS-i Circuit 1 + 2, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x65]			4, 64	163

issue date 11.1.2005

Instance ID	Data Item 1	Data Item 2	Data Item 3	Size (Members, Byte)	Complementary ID
128	AS-i Circuit 1 + 2, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x65]		Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	5, 76	164
129	AS-i Circuit 1 + 2, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x65]		Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	5, 100	165
130	AS-i Circuit 1 + 2, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x65]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]		7, 88	166
131	AS-i Circuit 1 + 2, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x65]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]	Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	8, 100	167
132	AS-i Circuit 1 + 2, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x65]	AS-i Circuit 1, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84]	Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	8, 124	168

issue date 11.1.2005

Instance ID	Data Item 1	Data Item 2	Data Item 3	Size (Members, Byte)	Complementary ID
133	AS-i Circuit 1 + 2, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x652] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x65]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]		10, 112	169
134	AS-i Circuit 1 + 2, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x65]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]	Short command interface [0x20, 0x68, 0x24, 0x01, 0x30, 0x64]	11, 124	170
135	AS-i Circuit 1 + 2, Input Data Image, all slaves, path: [0x20, 0x66, 0x24, 0x01, 0x30, 0x64] [0x20, 0x66, 0x24, 0x01, 0x30, 0x65] [0x20, 0x66, 0x24, 0x02, 0x30, 0x64] [0x20, 0x66, 0x24, 0x02, 0x30, 0x65]	AS-i Circuit 1+2, analog Input Data slave 29 … 31, path [0x20, 0x66, 0x24, 0x01, 0x30, 0x86] [0x20, 0x66, 0x24, 0x01, 0x30, 0x85] [0x20, 0x66, 0x24, 0x01, 0x30, 0x84] [0x20, 0x66, 0x24, 0x02, 0x30, 0x86] [0x20, 0x66, 0x24, 0x02, 0x30, 0x85] [0x20, 0x66, 0x24, 0x02, 0x30, 0x84]	Long command interface [0x20, 0x69, 0x24, 0x01, 0x30, 0x64]	12, 148	171

Instances 136 … 171 have the same structure but with analog and binary outputs. They can be read and written.

In case of a single master, only Instances 100 … 105 and 109 … 114 exist.

In case of single master, instance 100 is the default connection path for produced and Instance 136 for consumed data.

In case of double master, instance 118 is the default connection path for produced and instance 154 for consumed data.

10.2.4 Connection Object

Class Code: 5
Number of Instances: 3

Note

If the polled I/O message connection leaves the established state (3) the AS-i output data will be cleared.

Instance 1 Attributes (Explicit Message Connection)

Attribute ID	Access Rule	Name	DeviceNet Data Type	Data Value
1	Get	State	USINT	1 = configuring 2 = waiting for con-nection ID 3 = estabilshed 4 = timed out 5 = deferred delete
2	Get	Instance Type	USINT	0 = explicit message
3	Get	Transport Class Trigger	USINT	83 (hex)
4	Get	Produced Connection ID	UINT	10xxxxxx011 (binary) xxxxxx=node address
5	Get	Consumed Connection ID	UINT	10xxxxxx100 (binary) xxxxxx=node address
6	Get	Initial Comm. Characteristics	USINT	21 (hex)
7	Get	Produced Connection Size	UINT	204 (dec)
8	Get	Consumed Connection Size	UINT	204 (dec)
9	Get/Set	Expected Packet Rate	UINT	0 (msec)
12	Get	Watchdog Timeout Action	USINT	0 = timeout 1 = auto delete 2 = auto reset 3 = deferred delete
13	Get	Produced Connection Path Length	USINT	0
14	Get	Produced Connection Path		null (no data)
15	Get	Consumed Connection Path Length	USINT	0
16	Get	Consumed Connection Path		null (no data)
17	Get	Production Inhibit Time	UINT	

Common Services

Service Code	Class	Instance	Service Name
0x05	Yes	Yes	Reset
0x09	Yes	Yes	Delete
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

Instance 2 Attributes (Polled I/O Message Connection)

Attribute ID	Access Rule	Name	DeviceNet Data Type	Data Value
1	Get	State	USINT	1 = configuring 2 = waiting for connection ID 3 = estabilshed 4 = timed out 5 = deferred delete
2	Get	Instance Type	USINT	1 = I/O message
3	Get	Transport Class Trigger	USINT	83 (hex)
4	Get	Produced Connection ID	UINT	01111xxxxxx (binary) xxxxxx=node address
5	Get	Consumed Connection ID	UINT	10xxxxxx101 (binary) xxxxxx=node address
6	Get	Initial Comm. Characteristics	USINT	01 (hex)
7	Get	Produced Connection Size	UINT	20 (hex)
8	Get	Consumed Connection Size	UINT	20 (hex)
9	Get/Set	Expected Packet Rate	UINT	0 (msec)
12	Get	Watchdog Timeout Action	USINT	0 = timeout 1 = auto delete 2 = auto reset 3 = deferred delete
13	Get	Produced Connection Path Length	USINT	6
14	Get/Set	Produced Connection Path	Structure of: USINT USINT USINT USINT USINT USINT Structure of: USINT USINT USINT USINT USINT USINT	single master (default): 20 (hex) 04 (hex) 24 (hex) 64 (hex) 30 (hex) 03 (hex) double master (default): 20 (hex) 04 (hex) 24 (hex) 76 (hex) 30 (hex) 03 (hex)
15	Get	Consumed Connection Path Length	USINT	6

issue date 11.1.2005

Attribute ID	Access Rule	Name	DeviceNet Data Type	Data Value
16	Get	Consumed Connection Path	Structure of: USINT USINT USINT USINT USINT USINT Structure of: USINT USINT USINT USINT USINT USINT	single master (default): 20 (hex) 04 (hex) 24 (hex) 88 (hex) 30 (hex) 03 (hex) double master (default): 20 (hex) 04 (hex) 24 (hex) 9A (hex) 30 (hex) 03 (hex)
17	Get/Set	Production Inhibit Time		

Common Services

Service Code	Class	Instance	Service Name
0x05	Yes	Yes	Reset
0x09	Yes	Yes	Delete
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

Instance 4 Attributes (Cyclic/Change of State)

Attribute ID	Access Rule	Name	DeviceNet Data Type	Data Value
1	Get	State	USINT	1 = configuring 2 = waiting for connection ID 3 = estabilshed 4 = timed out 5 = deferred delete
2	Get	Instance Type	USINT	1 = I/O message
3	Get	Transport Class Trigger	USINT	12 (hex)
4	Get	Produced Connection ID	UINT	01101xxxxxx (binary) xxxxxx=node address
5	Get	Consumed Connection ID	UINT	10xxxxxx010 (binary) xxxxxx=node address
6	Get	Initial Comm. Characteristics	USINT	01 (hex)
7	Get	Produced Connection Size	UINT	20 (hex)
8	Get	Consumed Connection Size	UINT	20 (hex)
9	Get/Set	Expected Packet Rate	UINT	0 (msec)
12	Get	Watchdog Timeout Action	USINT	0 = timeout 1 = auto delete 2 = auto reset 3 = deferred delete
13	Get	Produced Connection Path Length	USINT	6

Attribute ID	Access Rule	Name	DeviceNet Data Type	Data Value
14	Get/Set	Produced Connection Path	Structure of: USINT USINT USINT USINT USINT USINT Structure of: USINT USINT USINT USINT USINT USINT	single master (default): 20 (hex) 04 (hex) 24 (hex) 64 (hex) 30 (hex) 03 (hex) double master (default): 20 (hex) 04 (hex) 24 (hex) 76 (hex) 30 (hex) 03 (hex)
15	Get	Consumed Connection Path Length	USINT	4
16	Get	Consumed Connection Path	Structure of: USINT USINT USINT USINT	single master (default): 20 (hex) 2B (hex) 24 (hex) 01 (hex)
17	Get/Set	Production Inhibit Time		

Common Services

Service Code	Class	Instance	Service Name
0x05	Yes	Yes	Reset
0x09	Yes	Yes	Delete
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

10.2.5 Parameter Object

Class Code: 15

Instances 1: I/O Data

Attribute ID	Access Rule	Name	DeviceNet Data Type	Data Value
1	Get/Set	Parameter Value	UINT	Byte 1: Production Instance, Byte 2: Consume Instance
2	Get	Link Path Size		0x00
3	Get	Link Path		
4	Get	Descriptor	UINT	0x20
5	Get	Data Type	EPATH	0xC7
6	Get	Data Size	USINT	0x02

issue date 11.1.2005

With this parameter the DeviceNet POLL Connection Production/Consume Path and the Cyclic/COS Production Path can be easily modified. The values are the assembly instances of the Production and Consume Path. If the current path values are inconsistent for this parameter the reading is 0.

Byte 1 modifies the Production Path of the POLL and the Cyclic/COS connection. Byte 2 the Consume Path of the POLL Connection.

10.2.6 AS-i Master Object

Class Code: 100

1 Instance for every AS-i circuit

Attribute ID	Access Rule	Name	DeviceNet Data Type	Default Data Value
100	Get	ec-Flags	UINT (16 bit)	
101	Get/Set	hi-Flags	USINT	
102	Get/Set	Operation Mode BOOL	USINT	
103	Get	LDS	UDINT	
104	Get/Set	LPS	UDINT	
105	Get	LAS	UDINT	
106	Get	LPF	UDINT	
107	Get/Set	Store_Actual_Configuration	BOOL	
108	Get/Set	Store_Actual_Parameters	BOOL	
109	Get/Set	Change_Slave_Adress	UINT	
110	Get/Set	Lock Pushbuttons	BOOL	

EC-Flags (16 bit)

EC-Flags (16 bit)								
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Pok	OR	APF	NA	CA	AAv	AAs	S0	Cok

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

Hi-Flags (8 bit):

Hi-Flags		
2^2	2^1	2^0
AAe	OL	DX

AAe Auto_Address_Enable
OL Off-line
DX Data_Exchange_Active

Operation Mode (8 bit):

1: configuration mode
0: protected mode

LDS, LAS, LPS, LPF (64 bit):

LDS, LAS, LPS, LPF								
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

Store actual Parameter/store actual configuration/lock pushbuttons:

True: Do the action

Change slave address (16 bit):

Change slave address								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	–		B	source address				
1	–		B	target address				

Meaning of bit B:

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

issue date 11.1.2005

10.2.7 AS-i Slave Object

Class Code: 101

64 Instances for every AS-i circuit, 1 for every AS-i slave

Instance ID	AS-i Slave
1	slave 0, circuit 1
2	slave 1A, circuit 1
…	…
32	slave 31A circuit 1
33	empty, circuit 1
34	slave 1B, circuit
…	…
64	slave 31B, circuit 1
65	slave 0, circuit 2
…	…
96	slave 31A, circuit 2
97	empty, circuit 2
…	…
98	slave 1B, circuit 2
…	…
128	slave 31B, circuit 2

Attribute ID	Access Rule	Name	DeviceNet Data Type	Remark
100	Get	Actual Configuration	UINT	
101	Get/Set	Permanent Configuration	UINT	slave 0, 32: not read-/ writeable
102	Get/Set	Actual Parameters	USINT	
103	Get/Set	Permanent Parameters	USINT	
104	Get/Set	xID1	USINT	slave 0: writeable only, slave 0 - 32: readable

Actual configuration/permanent configuration (16 bit):

Actual configuration/permanent configuration															
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
ID				IO				xID2				XID1			

issue date 11.1.2005

Parameter xID1 (8 bit):

Parameter xID1							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
–				data			

10.2.8 IO Data Object

Class Code: 102

1 Instance for every AS-i circuit

Attribute ID	Access Rule	Name	DeviceNet Data Type	Default Data Value
100	Get	Input Data Image, Single and A-slaves	ARRAY[16] of USINT	
101	Get	Input Data Image, B-slaves	ARRAY[16] of USINT	
102	Get/Set	Output Data Image Single and A-slaves	ARRAY[16] of USINT	
103	Get/Set	Output Data Image, B-slaves	ARRAY[16] of USINT	
104	Get	Analog Input Data slave 1	ARRAY[4] of INT	
…	…	…	…	…
134	Get	Analog Input Data slave 31	ARRAY[4] of INT	
135	Get/Set	Analog Output Data slave 1	ARRAY[4] of INT	
…	…	…	…	…
165	Get/Set	Analog Output Data slave 31	ARRAY[4] of INT	

Input and Output Data Image:

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			

issue date 11.1.2005

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
Off-Line: 0=OnLine, 1=Off-Line
LOS-master-bit 0=Off-Line by ConfigError deactivated
1=Off-Line by ConfigError activated

Analog values (16 bit):.

Analog value															
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Analog data:

Analog data								
byte	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	slave 31-n/8, channel 1, high byte							
2	slave 31-n/8, channel 1, low byte							
3	slave 31-n/8, channel 2, high byte							
4	slave 31-n/8, channel 2, low byte							
…	…							
n-1	slave 31, channel 4, high byte							
n	slave 31, channel 4, low byte							

10.2.9 Advanced Diagnostics Object

Class Code: 103

1 Instance for every AS-i circuit

Attribute ID	Access Rule	Name	DeviceNet Data Type	Default Data Value
100	Get/Set	LOS (List of Offline slaves)	UDINT	
101	Get	Error Counters A	ARRAY[32] of USINT	
102	Get	Error Counters B	ARRAY[32] of USINT	

issue date 11.1.2005

Error Counters::

Single and A Slaves	
Index	Error Counter
1	slave 1/1A
2	slave 2/2A
3	slave 3/3A
…	…
31	slave 31/31A

B Slaves	
Index	Error Counter
1	slave 1B
2	slave 2B
3	slave 3B
…	…
31	slave 31B

10.2.10 Short Command Interface Object

Class Code: 104

1 Instance

Attribute ID	Access Rule	Name	DeviceNet Data Type	Default Data Value
100	Get/Set	Content Command Toggle-Bit and AS-i Circuit Data	ARRAY[12] of USINT [0] [1] [2 … 11]	

10.2.11 Long Command Interface Object

Class Code: 105

1 Instance

Attribute ID	Access Rule	Name	DeviceNet Data Type	Default Data Value
100	Get/Set	Content Command Toggle-Bit and AS-i Circuit Data	ARRAY[36] of USINT [0] [1] [2 … 35]	

For special details about command interface commands see chapter 11.

issue date 11.1.2005

11 Command Interface

11.1 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							
…	…							
36	response parameter byte 34							

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.

The execution of a command interface is declined, if the command interface is too small.

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

The commands for reading and writinig exist in two variations. At the first variation the bits in the slave lists are arranged as usually with Pepperl+Fuchs 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 Pepperl+Fuchs arrangement is selected, otherwise the Siemens compatible arrangement is selected.For Siemens compatibility bit 2^6 should not be deleted.

issue date 11.1.2005

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							
…	…							

11.1.1 List of all Commands

Values for command				
command	**value**	**meaning**	**Req Len**	**Res Len**
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
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_FLAGS	47_{16}	Get_Flags	2	5
GET_DELTA	57_{16}	Get list of config. diff.	2	10
GET_LCS	60_{16}	Get LCS	2	10
GET_LAS	45_{16}	Get_LAS	2	10
GET_LDS	46_{16}	Get_LDS	2	10
GET_LPF	$3E_{16}$	Get_LPF	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
SET_OP_MODE	$0C_{16}$	Set_Operation_Mode	3	2
STORE_CDI	07_{16}	Store_Actual_Configuration	2	2
READ_CDI	28_{16}	Read_Actual_Configuration	3	4
SET_PCD	25_{16}	Set_Permanent_Config	5	2
GET_PCD	26_{16}	Get_Permanent_Config	3	4
SET_LPS	29_{16}	SET_LPS	11	2

issue date 11.1.2005

Values for command				
command	**value**	**meaning**	**Req Len**	**Res Len**
GET_LPS	44_{16}	Get_LPS	2	10
STORE_PI	04_{16}	Store_Actual_Parameter	2	2
WRITE_P	02_{16}	Write_Parameter	4	3
READ_PI	03_{16}	Read_Parameter	3	3
SET_PP	43_{16}	Set_Permanent_Parameter	4	2
GET_PP	01_{16}	Get_Permanent_Parameter	3	3
SET_AAE	$0B_{16}$	Set_Auto_Address_Enable	3	2
IDLE	00_{16}	No order	2	2
READ_IDI	41_{16}	Read IDI	2	36
WRITE_ODI	42_{16}	Write_ODI	34	2
READ_ODI	56_{16}	Read ODI	2	34
SLAVE_ADDR	$0D_{16}$	Change_Slave_Address	4	2
WRITE_XID1	$3F_{16}$	Write_Extended_ID-Code_1	3	2
SET_OFFLINE	$0A_{16}$	Set_Offline_Mode	3	2
SET_DATA_EX	48_{16}	Set_Data_Exchange_Active	3	2
BUTTONS	75_{16}	Disable pushbuttons	3	2
FP_PARAM	$7D_{16}$	„Functional Profile“ Param.	≥3	≥2
FP_DATA	$7E_{16}$	„Functional Profile“ Data	≥3	≥2
EXT_DIAG	71_{16}	ExtDiag generation	6	2
INVERTER	$7C_{16}$	Configure Inverter Slaves	12	4

issue date 11.1.2005

11.1.2 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 decteced"
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"

11.2 Commands of the Command Interface

11.2.1 Analog Data

11.2.1.1 Overview of the Commands

Values for command				
command	value	meaning	Req Len	Res Len
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
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

issue date 11.1.2005

11.2.1.2 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							

11.2.1.3 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						

issue date 11.1.2005

11.2.1.4 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/DeviceNet gateway.

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							

11.2.1.5 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 succesive 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							

11.2.1.6 Write 4 7.3-Slave out.Data (WR_7X_OUT_X)

With this command the four 16 bit channels of four AS-i output slaves with successive addresses according to slave profile 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						

11.2.1.7 Read 4 7.3-Slave out.Data (RD_7X_OUT_X)

With this command, the four 16 bit channels of four AS-i output slaves with successive addresses according to slave profile 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							

11.2.1.8 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 ≡ 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 btye 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						

11.2.1.9 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 ≡ 0, the string is being read from the slave, otherwise the function responses out of the memory; the data can be read consistently.

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

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

issue date 11.1.2005

11.2.1.10 RD_74_ID

With this function the ID string of a slave according to profile S-7.4 is 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 ≡ 0, the string is being read from the slave, otherwise the function responses out of the memory, the data can be read consistently.

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

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

11.2.1.11 RD_74_DIAG

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

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

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

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							

issue date 11.1.2005

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

11.2.2 Diagnosis Data

11.2.2.1 Overview of the Commands

Values for command				
Command	value	meaning	Req Len	Res Len
GET_LISTS	30_{16}	Get LDS, LAS, LPS, Flags	2	29
GET_FLAGS	47_{16}	Get_Flags	2	5
GET_DELTA	57_{16}	Get list of config. diff.	2	10
GET_LCS	60_{16}	Get LCS	2	10
GET_LAS	45_{16}	Get_LAS	2	10
GET_LDS	46_{16}	Get_LDS	2	10
GET_LPF	$3E_{16}$	Get_LPF	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

11.2.2.2 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/DeviceNet gateway:

- 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					

issue date 11.1.2005

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	6A	5A	4A	3A	2A	1A	0A
…	LDS							
18	31B	30B	29B	28B	27B	26B	25B	24B
19	7A	6A	5A	4A	3A	2A	1A	0A
…	LPS							
26	31B	30B	29B	28B	27B	26B	25B	24B
27	–							Pok
28	OR	APF	NA	CA	AAv	AAs	S0	Cok
29	–					AAe	OL	DX

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

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

issue date 11.1.2005

Cok Config_Ok
AAe Auto_Address_Enable
OL Offline
DX Data_Exchange_Active

11.2.2.3 Get Flags (GET_FLAGS)

With this call, the following entry is read out of the AS-i/DeviceNet gateway: 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	AAs	S0	Cok
5	–					AAe	OL	DX

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

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

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

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

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

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

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

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

Cok Config_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.

11.2.2.4 Get Delta List (GET_DELTA)

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

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

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

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

issue date 11.1.2005

11.2.2.5 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/DeviceNet gateway.

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

11.2.2.6 Get List of Activated Slaves (GET_LAS)

With this call, the following entry is read out of the AS-i/DeviceNet gateway: 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

issue date 11.1.2005

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

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

With this call, the following entry is read out of the AS-i/DeviceNet gateway: 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

issue date 11.1.2005

11.2.2.8 Get list of peripheral faults (GET_LPF)

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

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

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

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

11.2.2.9 Get List of Offline Slaves (GET_LOS)

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

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

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

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

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

11.2.2.10 Set List of Off-line Slaves (SET_LOS)

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

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

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

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

issue date 11.1.2005

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

11.2.2.11 Get transm.err.counters (GET_TECA)

With this call the error counters of all single slaves/A-slaves can be read (see chapter 9).

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

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

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

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

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

11.2.2.12 Get transm.err.counters (GET_TECB)

With this call, the counts of the error counters for B-slaves are being read out (see chapter 9).

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

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

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

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

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

11.2.2.13 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.

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

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

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

11.2.2.14 Functional profiles

Further diagnosis functions for "Safety at Work" and for availability vice versa for warnings of integrated sensors are explained detailed in the chapter "Functional profiles" (chapter 11.2.3).

11.2.3 Functional profiles

11.2.3.1 "Safety at Work" List 1

Function: 00_{16}

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 11.2.4.4: 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 11.2.2.3: "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-Inferface addresses 4 and 10
1 safety slave with unreleased safety function,
AS-Inferface address 5.

Reponse: 7E 00 01 25 10 04 00 00

11.2.3.2 "Safety at Work" Monitor Diagnosis

Function: 02_{16}

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							

issue date 11.1.2005

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							

11.2.3.3 Integrated AS-i Sensors: Warnings

Function: 03_{16}

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

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

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

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

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

11.2.3.4 Integrated AS-i Sensors: Availability

Function: 04_{16}

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

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

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

issue date 11.1.2005

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

11.2.4 Configuration of the AS-i Master

11.2.4.1 Overview of the Commands

Values for command				
comamnd	value	meaning	Req Len	Res Len
SET_OP_MODE	$0C_{16}$	Set_Operation_Mode	3	2
STORE_CDI	07_{16}	Store_Actual_Configuration	2	2
READ_CDI	28_{16}	Read_Actual_Configuration	3	4
SET_PCD	25_{16}	Set_Permanent_Config	5	2
GET_PCD	26_{16}	Get_Permanent_Config	3	4
SET_LPS	29_{16}	SET_LPS	11	2
GET_LPS	44_{16}	Get_LPS	2	10
STORE_PI	04_{16}	Store_Actual_Parameter	2	2
WRITE_P	02_{16}	Write_Parameter	4	3
READ_PI	03_{16}	Read_Parameter	3	3
SET_PP	43_{16}	Set_Permanent_Parameter	4	2
GET_PP	01_{16}	Get_Permanent_Parameter	3	3
SET_AAE	$0B_{16}$	Set_Auto_Adress_Enable	3	2

11.2.4.2 Set Operation Mode (SET_OP_MODE: Set_Operation_Mode)

This call switches between configuration mode and protected mode.

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

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

issue date 11.1.2005

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/DeviceNet gateway 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

11.2.4.3 Store Actual Configuration (STORE_CDI)

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

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

This command can only be executed in the configuration mode.

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

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

11.2.4.4 Read Actual Configuration (READ_CDI)

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

11.2.4.5 Set Permanent Configuration (SET_PCD)

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

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

The configuration data are stored permanently on the EEPROM of the AS-i/DeviceNet gateway 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.

issue date 11.1.2005

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

This command can only be executed in the configuration mode.

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

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

Meaning of bit B:

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

11.2.4.6 Get Extended Permanent Configuration (GET_PCD)

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

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

The configuration data are specified by the manufacturer of the 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				I0			

issue date 11.1.2005

Meaning of bit B:

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

11.2.4.7 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						

11.2.4.8 Get List of Projected Slaves (GET_LPS)

With this call, the following entry is read out of the AS-i/DeviceNet gateway: 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					

issue date 11.1.2005

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

11.2.4.9 Store Actual Parameters (STORE_PI)

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

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

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

issue date 11.1.2005

11.2.4.10 Write Parameter (WRITE_P)

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

The parameter is stored in the AS-i/DeviceNet gateway 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

11.2.4.11 Read Parameter (READ_PI: Read_Parameter)

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

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

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

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

Meaning of bit B:
B = 0 Single AS-i slave or A-slave
B = 1 B-slave

11.2.4.12 Set_Permanent_Parameter (SET_PP)

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

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

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						

11.2.4.13 Get_Permanent_Parameter (GET_PP)

With this call, a slave-specific parameter value stored on the EEPROM of the AS-i/DeviceNet gateway 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

issue date 11.1.2005

11.2.4.14 Set Auto Address Enable (SET_AAE)

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

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

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

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

11.2.5 Other Commands

11.2.5.1 Overview of the Commands

Value for command				
command	value	meaning	Req Len	Res Len
IDLE	00_{16}	Kein Auftrag	2	2
READ_IDI	41_{16}	Read IDI	2	36
WRITE_ODI	42_{16}	Write_ODI	34	2
READ_ODI	56_{16}	Read ODI	2	34
SLAVE_ADDR	$0D_{16}$	Change_Slave_Address	4	2
WRITE_XID1	$3F_{16}$	Write_Extended_ID-Code_1	3	2
SET_OFFLINE	$0A_{16}$	Set_Off-Line_Mode	3	2
SET_DATA_EX	48_{16}	Set_Data_Exchange_Active	3	2
BUTTONS	75_{16}	Disable Pushbuttons	3	2
FP_PARAM	$7D_{16}$	„Functional Profile“ Param.	≥3	≥2
FP_DATA	$7E_{16}$	„Functional Profile“ Data	≥3	≥2
INVERTER	$7C_{16}$	Configure Inverter Slaves	12	4

11.2.5.2 IDLE

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

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

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

11.2.5.3 Read Input Data Image (READ_IDI)

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

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

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

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

11.2.5.4 Write Output Data Image (WRITE_ODI)

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

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

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

11.2.5.5 Read Output Data Image (READ_ODI)

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

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

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

11.2.5.6 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.

Note

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

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

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

Meaning of bit B:
B = 0 Single AS-i slave or A-slave
B = 1 B-slave

11.2.5.7 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						

11.2.5.8 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/DeviceNet gateway 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/DeviceNet gateway 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.

11.2.5.9 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						

11.2.5.10 BUTTONS

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

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

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

11.2.5.11 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 11.2.3).

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							

11.2.5.12 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 11.2.3).

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							

11.2.5.13 EXT_DIAG

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

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

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

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

11.2.5.14 INVERTER

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

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

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

11.3 Command Interface Examples

11.3.1 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 interaface 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 11.1.2 "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.

11.3.2 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 11.1.2005

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 11.1.2 "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.

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

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.

<u>3. Set master to protected mode</u>

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.

issue date 11.1.2005

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. It must being watited until the master is switching in this operation mode.

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

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

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

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

No result because toggle bit = 0.

Setting the toggle bit:

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

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

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

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

Normal operation mode is necessary to run the application safely.

11.3.3 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}

issue date 11.1.2005

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 11.1.2 "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.

<u>Setting the toggle bit:</u>

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.

<u>4. Write permanent parameter (power on parameter) to master</u>

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.

issue date 11.1.2005

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.

issue date 11.1.2005

12 Commissioning Tools and Accessories

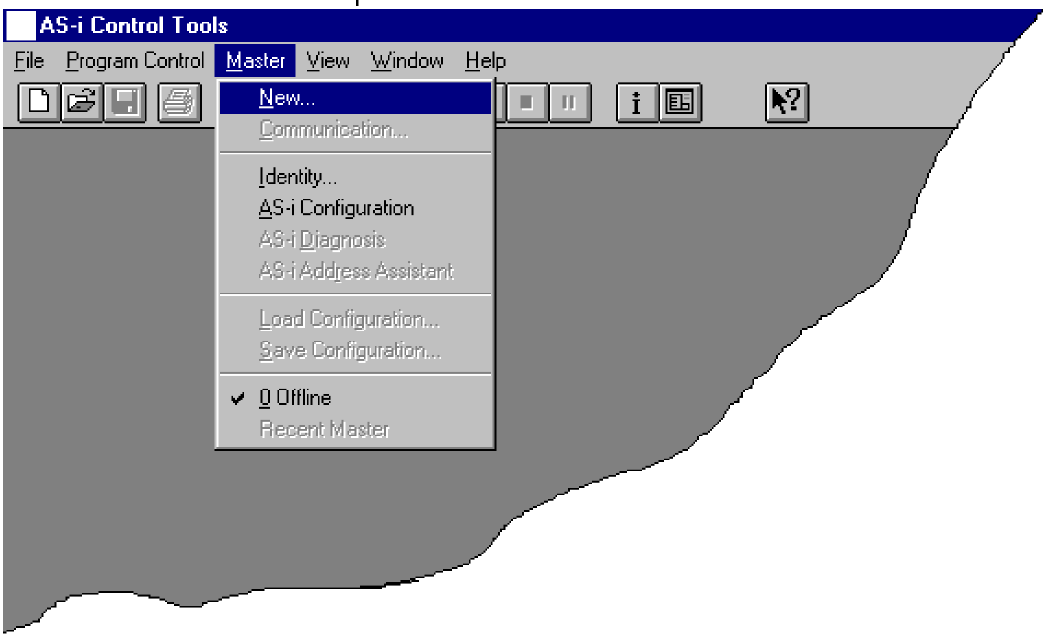
The AS-i circuit on the AS-i master can be put into operation with the comfortable Windows software "AS-i Control Tools".

The software package communicates with the AS-i master via a serial cable.

12.1 Windows Software AS-i Control Tools

The Windows software AS-i Control Tools enables you to configure the AS-i circuit in a very comfortable manner.

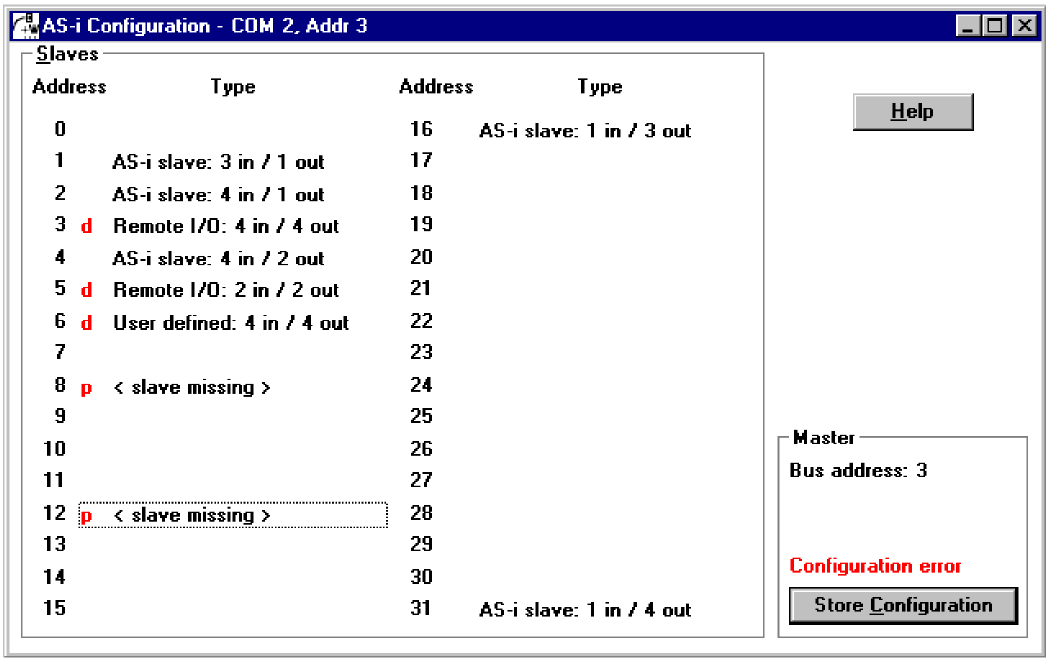
1. Start the AS-i Control Tools.
2. Call the command Master | New.



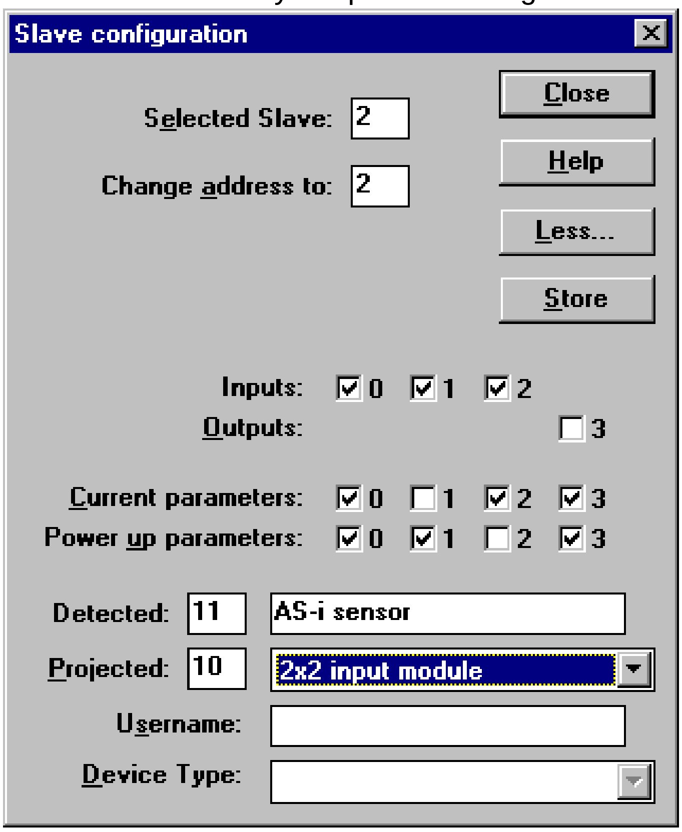
3. Choose DeviceNet as protocol.
4. Do the appropriate settings.

issue date 11.1.2005

5. Call the command Master | AS-i configuration.
   The AS-i configuration editor will be started. All detected and projected AS-i slaves are displayed in this window.



6. Click on a slave entry to open the dialog box slave configuration.



   Changing a slave address, setting AS-i parameters or AS-i configuration data is possible here. Additionally, inputs and outputs can be tested.

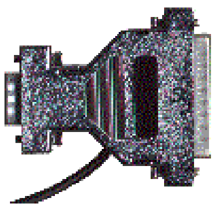
A very easy approach to configure the AS-i circuit is connecting each AS-i slave to the line and setting the AS-i slave address one after the other. After that press the button "Store configuration" to adopt the detected AS-i circuit to the AS-i master as projected data.

Furthermore you can use the **AS-i Address Assistant**. This tool automatically changes the address of an AS-i slave to the desired address after connecting the slave to the AS-i line. The desired AS-i configuration can be created oofline before and then be stored to a file. When building up the plant you only have to connect the AS-i slaves to the AS-i line one after the other.

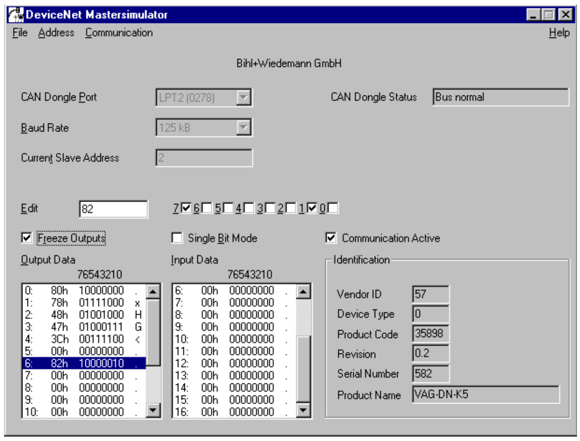
Further descriptions to all features of the software can be obtained from the integrated help.

12.2 DeviceNet Master Simulator

DeviceNet



parallel port



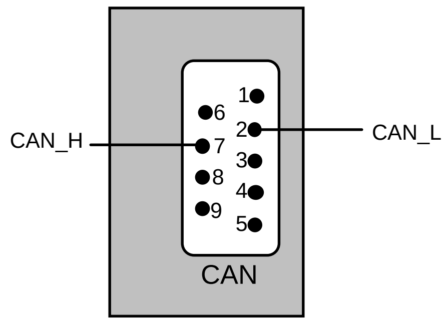
The DeviceNet Master Simulator is an easy to use software for data exchange with DeviceNet slaves of many suppliers. The DeviceNet Master Simulator can exchange data with the slaves even without EDS-file. Input data can be read, output data can be written and the DeviceNet diagnosis can be displayed. Furthermore it is possible to read and write any object independent of the state of communication.

The DeviceNet Master Simulator offers the possibility to scan a DeviceNet network and find all connected slaves. The I/O data is displayed binary and hexadecimal.

In single bit mode it is possible to set an output as long as the mouse button is pressed.

The device identification is read out of the DeviceNet slave and displayed together with the I/O data.

The DeviceNet Master Simulator consists of the software and a DeviceNet dongle. The DeviceNet dongle is the ideal interface converter between the parallel interface of a PC and DeviceNet. The converter needs no extra power supply. Therefore it is also suitable for mobile use with a laptop or a notebook.

Technical data of the DeviceNet Dongle	
Type	DeviceNet Dongle
Dimensions (L, W, H)	63 mm, 54 mm, 17 mm
Interfaces	Standard parallel PC interface with 25-pin D-sub-plug (male) CAN interface with 9-pin D-sub-plug (male)
Power supply	powered by the keyboard interface of the PC
Length of connector cables	max. 2 m
Transfer rates	125, 250 or 500 KBaud
Operating temperature	0°C … +55°C
Sorage temperature	-25°C … +70°C
Connections of D-sub plug	

13 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.

In the basic state of the protected operating mode, the display is either blank or displays the address of a faulty assignment (see chapter 7.4.2).

During manual address programming, the slave address display has a different meaning (see chapter 7.5 and 7.6).

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 occured.
40	The AS-i master is in offline phase.
41	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.
70	Hardware error: The AS-i master's EEPROM cannot be written.
72	Hardware error: The PIC processor does not respond.
73	Hardware error: The PIC processor does not respond.
74	Checksum error in the EEPROM.
75	Error in the internal RAM.
76	Error in the external RAM.
80	Error while attempting to exit the configuration mode: A slave with address zero exists.
81	General error while changing a slave address.
82	The front panel operation is blocked. Until repowering-up the device can only be accessed from the host via the interface.
83	Program reset of the AS-i Control programm: The AS-i Control programm is being read out of EEPROM and copied into the RAM.
88	Display test while starting up the AS-i master
90	Error while changing a slave address in protected operating mode: No slave with address 0 existing.
91	Error while changing slave address: Target address is already used.
92	Error while changing slave address: New address could not be set.
93	Error while changing slave address: New address could only be stored volatilely in the slave.
94	Error while changing the slave address in protected operating mode: Slave has wrong configuration data.
95	Error while changing slave address in protected operating mode: The configuration error was caused by a superfluous slave (instead of a missing slave).

issue date 11.1.2005

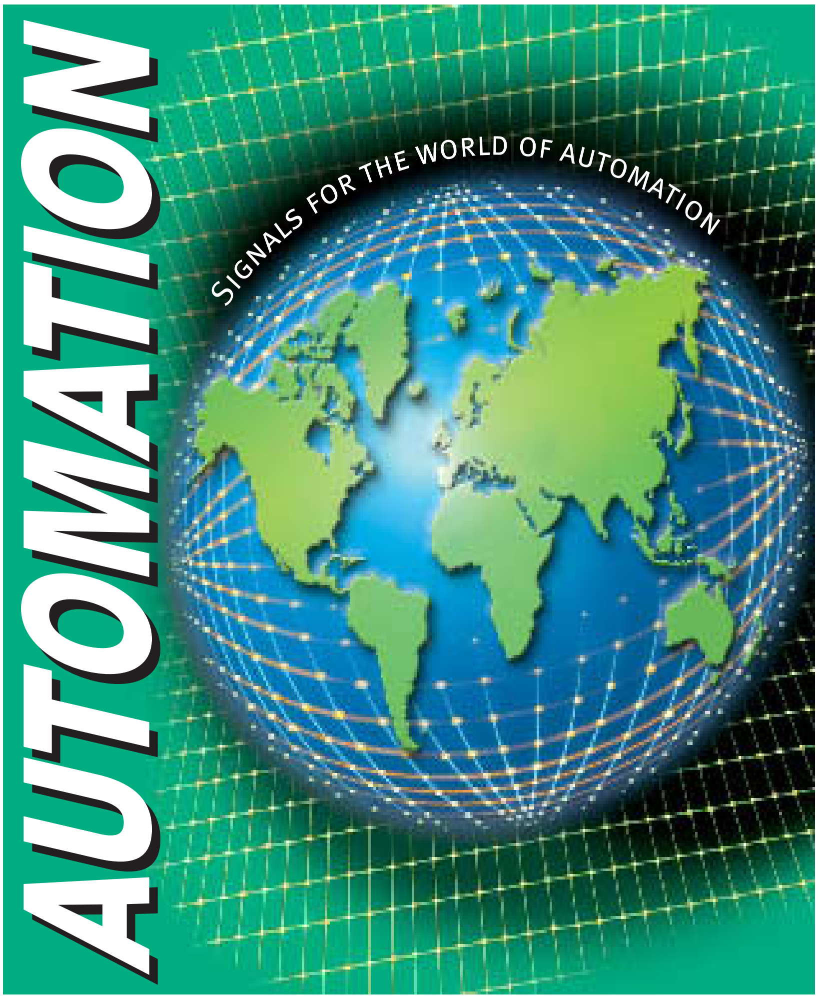
14 Appendix: The First Commissioning of AS-i

Note

In this chapter an example is given, how quickly and easily an AS-i network can be put into operation without the need for external devices. The addressing of the components connected to the AS-i network can be performed directly on the AS-i master. It certainly is more comfortable to do the addressing with a handheld programming device or with the Windows software AS-i Control Tools. However, it is possible to configure even complex networks using only the AS-i master.

What to do ?	How to go about it?
The AS-i master has to be properly supplied with power.	Connect the AS-i power supply unit to the terminals AS-i + and AS-i - of the master, connect the ground terminal. Turn on the power supply.
After the self-test: the LEDs “power”, “config err”, ”U ASI” and “prj mode” are on. The LC display shows “”: the AS-i master is in the offline phase. Shortly after that a “” will be displayed: the AS-i master stays in the detection phase.	
Switch the device to the projecting mode if the yellow LED does not light up.	Press the “mode” button for approx. five seconds.
The yellow LED “prj mode” lights up. The device is now in projecting mode.	
Add a slave with the address 0 to the AS-i line.	Connect the slave's terminals with the terminals AS-i +/- of the master.
The green LED “ASI active” lights up. The LC display shows “0”. This indicates that the AS-i master has detected the slave.	
Change the slave address to address to "1".	Select address "1" by pressing the “set” button shortly, if necessary repeatedly; after each pressing the next free address is displayed. When a “1” appears on the display, press the “set” button for approx. five seconds until the display flashes. Press the “set” button again shortly to assign the new address to the slave.
The AS-i master detects the slave with address "1" and displays “1”.	
Connect another slave with address "0" to the AS-i line and allocate the address "2" to it.	Connect the slave to the AS-i line. The addressing can be carried out the same way as the previous slave.
The addresses of all detected slaves are now displayed sequentially.	
Switch to the protected operating mode and store the AS-i configuration.	Leave the configuration mode by pressing the “mode” button for at least five seconds until the “prj mode” LED goes out.

What to do ?	How to go about it?
The configuration of the AS-i master is now finished. Now the hierarchically higher fieldbus system can be put into operation.	



www.pepperl-fuchs.com

Worldwide Headquarters
Pepperl+Fuchs GmbH · Königsberger Allee 87
68307 Mannheim · Germany
Tel. +49 621 776-0 · Fax +49 621 776-1000
e-mail: info@de.pepperl-fuchs.com

USA Headquarters
Pepperl+Fuchs Inc. · 1600 Enterprise Parkway
Twinsburg, Ohio 44087 · USA
Tel. +1 330 4253555 · Fax +1 330 4254607
e-mail: sales@us.pepperl-fuchs.com

Asia Pacific Headquarters
Pepperl+Fuchs Pte Ltd. · P+F Building
18 Ayer Rajah Crescent · Singapore 139942
Tel. +65 67799091 · Fax +65 68731637
e-mail: sales@sg.pepperl-fuchs.com

PEPPERL+FUCHS
SIGNALS FOR THE WORLD OF AUTOMATION

122970 01/05 02