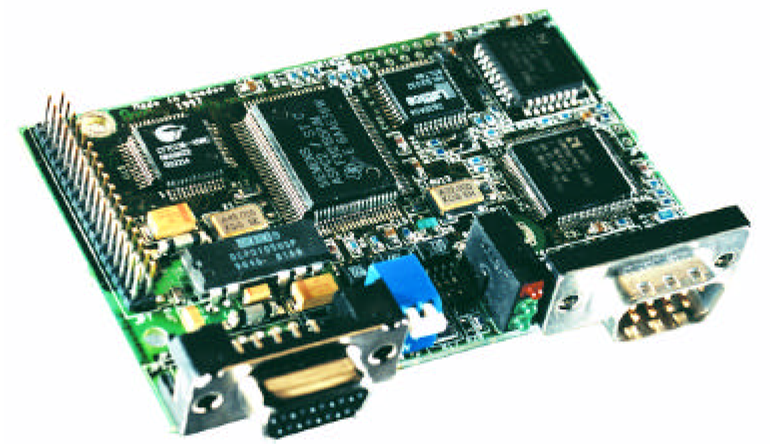
ANYBUS®
HMS FIELDBUS SYSTEMS AB

MANUAL

PROFIBUS-DP MASTER

Doc. No. PDP-M-0.993



Lighthouse PLCs, Inc.
2549 Rosebay Street | Suite 100
Eugene, OR 97402-6148 USA
Ph 541.463.1496 Fx 541.463.1497
Em info@lighthouseplcs.com
URL http://www.lighthouseplcs.com

HMS FIELDBUS SYSTEMS AB
Pilefeltsgatan 93-95
S - 302 50 Halmstad
SWEDEN

Phone: +46 - 35 – 172 900
FAX: +46 - 35 – 172 909
e-mail: info@hms.se
www: www.hms.se

1. Introduction

The AnyBus® modules constitute HMS Fieldbus Systems AB's adaptable system which follows the official standard for each AnyBus interface.

This specification is intended to provide information necessary to implement a Master Module using PROFIBUS specific information.

Refer to the following documents for additional information on the AnyBus modules.

- ANYBUS I/O module specification (ABIO-DG-1.1)
- ANYBUS DATATRANSFER design guide (ABDT-DGP-1.1)
- ANYBUS–S design guide (ABS-DGP-1.0)
- Profibus-DP Master implementation guide (PDP-M-0.99)
- Profibus-DP Configuration Tool (PDP-M-CON-1.0)

Throughout this document the term "user" refers to the person or persons who have developed the Profibus Master module. The end-user of the Profibus master module is referred to as the "customer".

Throughout this document the term "host" refers to the application, for example a PLC carrier board. The term "device" refers to the Profibus Master module. The term "Remote node" refers to an external node connected on the network, for Master-Slave network the term "Remote Node" is similar with the term "Slave".

As in all communication systems the terms "read" and "write" can be ambiguous, since their meaning depend upon which end of the link is being referenced. The convention in this document is that "read" reads input data from the connected slaves of the fieldbus. "Write" writes output data to the connected slaves of the fieldbus

1.1 General Description of the Profibus-DP Master module

The Profibus Master module operates as a Master to a fieldbus network. The connection to the host system is made via a Dual Port Memory (DPM) area. Data written to the DPM from the host will be written to the remote nodes on the fieldbus. Data read to the DPM from the host will be read from the remote nodes on the fieldbus.

For different fieldbus system there will be different firmware for the AnyBus slave- and master modules, but the electrical and mechanical data are always the same for all modules. All modules are using the same memory structure but minor difference can occur regarding to differences in the fieldbus systems.

The modules will support at least 8192 I/O points (4096 Inputs and 4096 Outputs) of process data and also have a structure for telegram services. On bus systems limited to a smaller amount of data or limited telegram function, the module will be limited to the same extent.

Two hardware versions for Profibus DP are available:

- Standard version with 8 bit data bus and maximum of 4kB address bus.
- Enhanced version with 16 bit data bus and maximum of 64kB address bus

The Profibus-DP module can be configured either by using the external Profibus-DP Configurator (please refer to separate documents regarding external Configuration Software), or the master can be configured internally via the dual port RAM (for an example, please refer to the implementation guide).

1.2 Technical Details

- All baud rates from 9,6 kbit/s up to 12 Mbit/s
- Up to 124 stations
- Max. 512 byte in of process data (standard version)
- Max. 3.5 k byte in of process data (16 bit version)
- Max. 512 byte out of process data (standard version)
- Max. 3.5 k byte out of process data (16 bit version)

1.3 Supported Profibus-DP Services

All functions of a class 1 Master are supported. In addition, the 'Set_Slave_Address' Telegram can be issued via the configuration software.

The following class 1 services are available to the user:

- Get_Slave_Diagnostics
- Global_Control_Command (sync, freeze)
- Read / Clear Statistic Counter

The other services are embedded within the device and automatically handled by the module according to the Profibus-DP state machine.

2. Mechanical Design Specification

The Profibus Master module is mechanically compatible with the AnyBus DataTransfer slave modules.

2.1 Mechanical Drawing

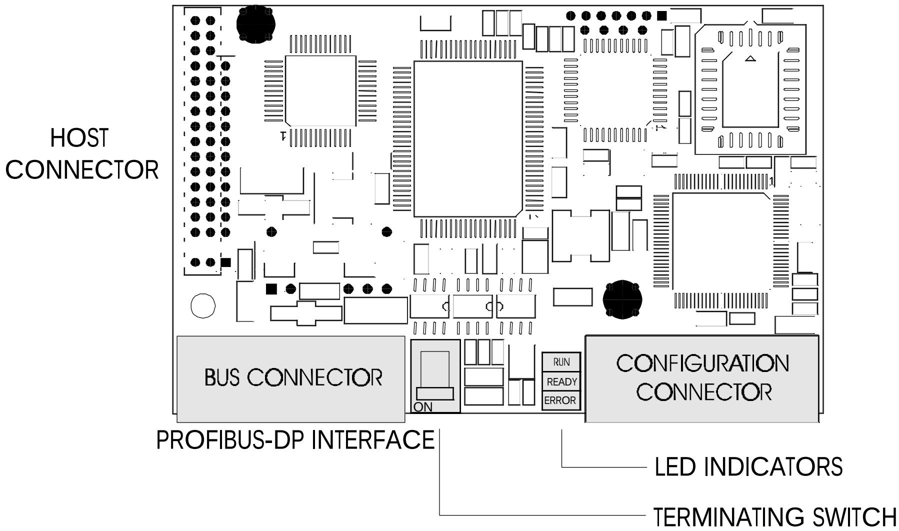


Figure 1. Overview Profibus Master module

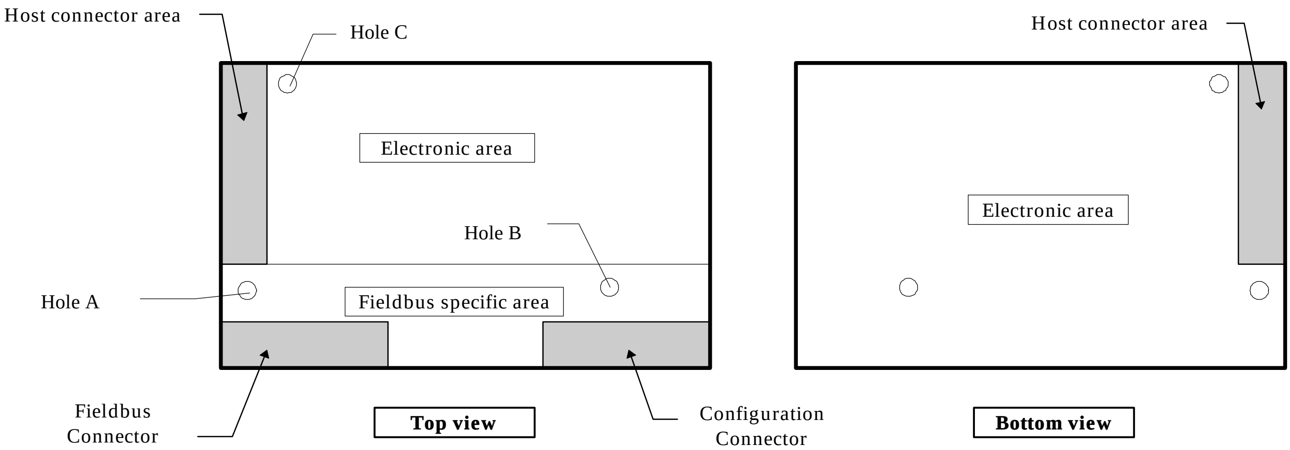


Figure 2. Overview of Profibus Master module PCB

1. Electronic Area
   This is the area where the Profibus Master module hardware like DPRAM, CPU, EPROM etc. is mounted.
2. Fieldbus Specific Area
   This is the area where the fieldbus specific components that do not fulfil the mechanical specification of the electronics area are placed. This includes components such as the fieldbus connector, transformers, protection capacitors etc. Since different hardware is required for different fieldbus systems the worst case measurements is specified for this area.
3. Host Connector Area
   Two Profibus Master versions are available; Standard version with 8-bit data bus and enhanced version

with 16-bit data bus. Standard version uses a 34 pin Host Connector and the enhanced version uses a 34+14 pin Host Connector.

The Profibus Master module has the following mechanical specification.

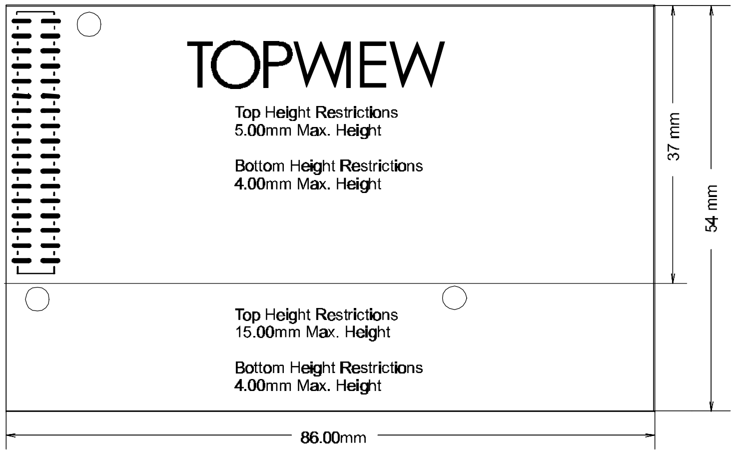


Figure 3. Profibus Master module mechanical drawing

2.1.1 Fieldbus Specific Area

The following type of components can be placed in the fieldbus specific area of the PCB.

- Fieldbus and configuration connectors
- Indication LED (Light Emitting Diodes).
- DC / DC converter
- Node address settings (optional) and / or other configuration switches.
- Components with mechanical measurements that prevents them from being placed at another location (transformers, oscillators, etc.).
- Special components used for the communication that has to be mounted near the fieldbus connector, like a termination switch used on some systems.

The only component that has to be placed at the same position for all MASTER modules is the fieldbus connector.

1. Fieldbus connector.

The fieldbus connector should be placed in the left corner of the module, viewed from top. See the exhibit below for the exact position. The placement of the other fieldbus specific components are unique for each module.

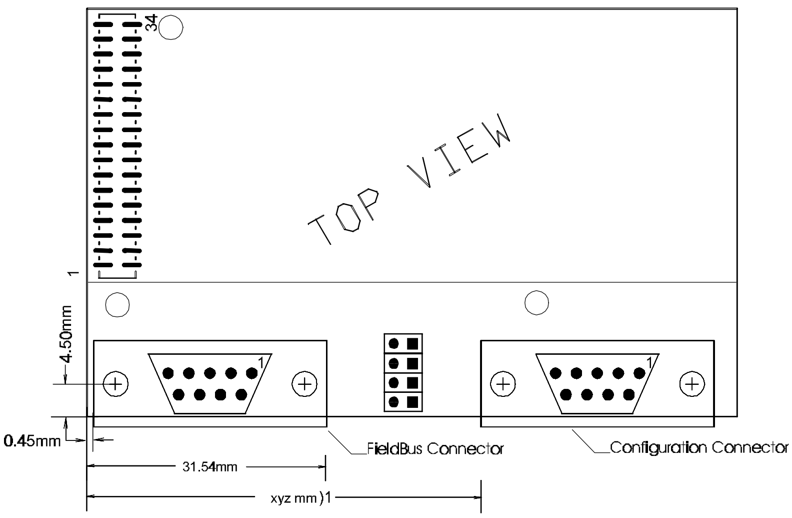


Figure 4. Fieldbus connector

2.1.2 Electronic Area

The only restrictions of the electronics area are the maximum heights given in Figure 3.

2.1.3 Host Connector Area

In order to achieve interchangeability between the Profibus Master and the different AnyBus modules, the application connector is positioned according to the following specification on all modules. Please note the extended Host Connector in the Enhanced version:

In the standard 8-bit version a 34 pin male strip connector is used (2 rows x 17 pins in 2,0 mm spacing). These connectors are available in several standard versions:

- Straight connector for top or bottom mounting, several different heights.
- Angled connector for top or bottom mounting.

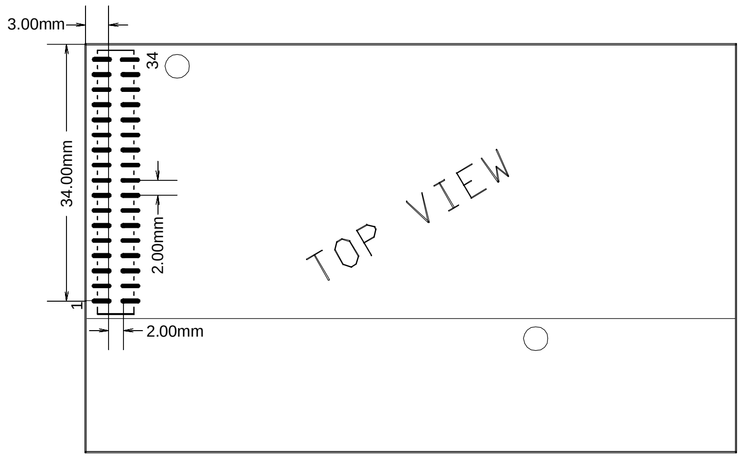


Figure 5. Standard Application Connector (8-bit version)

In the 16 bit version the standard Host Connector is replaced with a 48 pin male strip connector (2 rows x 17 pins + 1 row x 14 pins). This connector is available in several standard versions:

- Straight connector for top or bottom mounting, several different heights.

No other components are allowed in this area.

For more information regarding the connectors, please consult HMS.

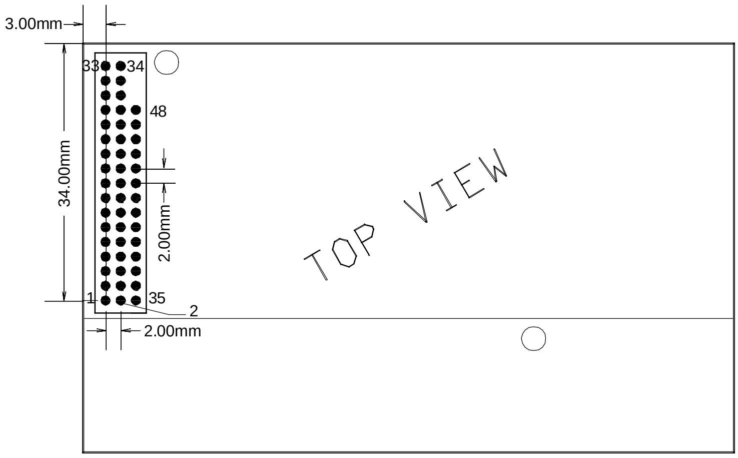


Figure 6. Enhanced application connector (16 bit version)

2.2 Mounting the Device

For mechanical fastening of the module, three mounting holes are provided. Two of these holes are metal plated and must be connected via conductive holders to the carrier board. One of these holes is used to connect to GND, the other hole is used for a connection to protected earth (PE) connection to the module.

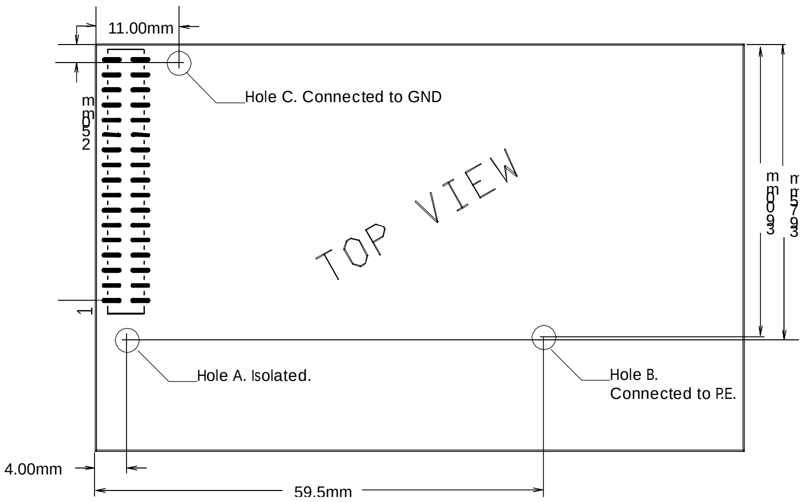


Figure 7. Mounting holes

A) Isolated hole. [1]
For use with non conductive screws or similar.

B) Connected to PE[2]
For use with conductive screws or similar.

C) Connected to GND
For use with conductive screws or similar.

---

[1] Hole A might be used for PE instead (this is under investigation; details are available: latest 15th January 1998

[2] Hole B might be unconnected (see previous footnote).

The following dimensions are used on the conductive holes (B and C). The isolated hole (A) have the same dimensions but the marked area is non conductive.

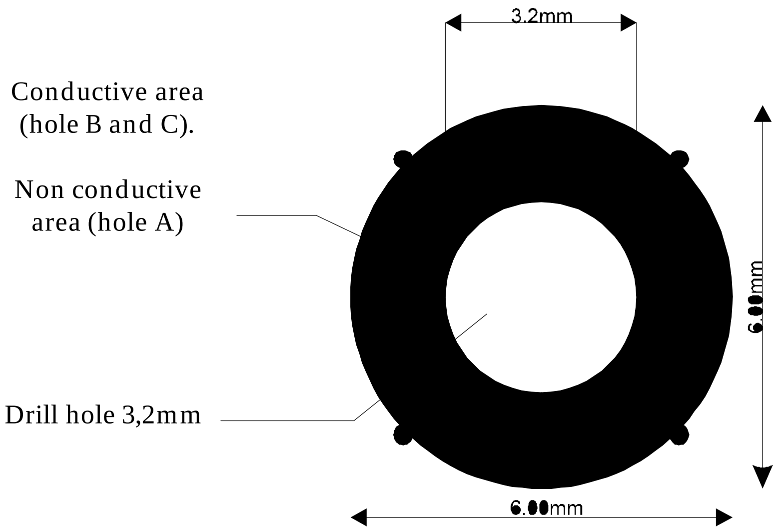


Figure 8. Mounting holes - mechanics

2.3 Connector in the Connector Area

2.3.1 Configuration Connector

The configuration port is always a non isolated RS232 communication port. Three different types of Configuration connectors shall be available. All three types are available in angled or straight versions.

- D-sub (9 pin). Pin 1 always at fixed position.
- Crimp terminal 2,54 mm spacing. Pin 1 always at fixed position.
- Plugable terminal 3,81 mm spacing. Pin 1 always at fixed position.

Pin in D-sub	Pin in other connector types	I/O	Signal	Description
1	-	-	-	-
2	1	Input	RXD	Receive Data
3	2	Output	TXD	Transmit Data
4	3	Output	DTR	Data Terminal Ready
5	4	Ref.	GND	Ground
6	-	-	-	-
7	5	Output	RTS	Request To Send
8	6	Input	CTS	Clear To Send
9				

Table 1. Configuration connector

2.3.2 Profibus-DP Connector

Three different types of Fieldbus connectors shall be available. All three types are available in angled or straight versions. The pin assignment is fieldbus dependent.

- D-sub (9 pin). Pin 1 always at fixed position.
- Crimp terminal 2,54 mm spacing.
- Plugable terminal 3,81 mm spacing.

Pin in D-sub	Pin in other connector types	Signal
1	5	Shield
2	-	-
3	4	B-Line
4	6	RTS (TTL)
5	2	GND Bus
6	1	+5V Bus
7	-	-
8	3	A-Line
9	-	-

Table 2. Profibus-DP connector

3. Hardware Design

Since the hardware on the Profibus Master modules are dependent on the integrated fieldbus technology, only the module interface is specified in this section. This chapter is divided into three parts:

1. Hardware interface in standard version
2. Hardware interface in enhanced version
3. Status (LED) indicator

3.1 Hardware interface in standard version (8 bit version)

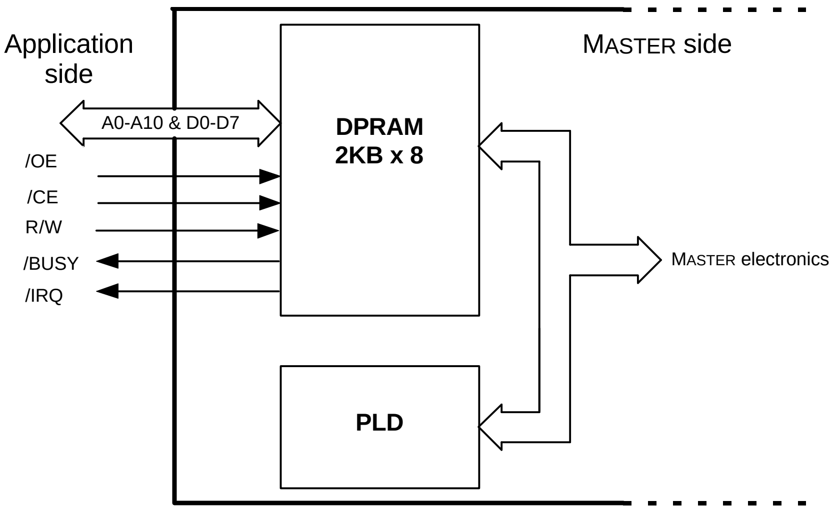


Figure 9. Application interface in standard version

3.1.1 Host Connector

The Host connector is a 34 pin strip connector in dual row. Several pin lengths are available. The Host connector can be mounted on the bottom side or the top side of the board. An angled version is also available.

Pin No.	Name	Function	Type
1,5	VCC	+5V to electronics	Power
2,6	GND	0V to electronics	Power
3	-	Not Connected.	NC
4	-	Not Connected.	NC
7	-	Not Used. Internally pulled up to VCC with 100k	Input
8	-	Not Used.	NC
9 – 18	A0 - A9	Address bus to DPRAM. A0 at pin 9, A9 at pin 18	Input
19 – 26	D0 - D7	Data bus to DPRAM. D0 at pin 19, D7 at pin 26	Bi-directional
27	/BUSY	DPRAM busy signal - Open Collector Type	Output
28	/IRQ	DPRAM interrupt signal - Open Collector Type	Output
29	/OE	DPRAM output enable signal	Input
30	R/W	DPRAM read / write signal	Input
31	/CE	DPRAM chip enable signal	Input
32	/RESET	MASTER module RESET signal	Input
33	A10	Address bus to DPRAM	Input
34	A11	Not connected	NC

Table 3. Application connector pin description, standard version

3.1.2 Signal Description

Pin 1,5	**Power supply to Electronics**
+5V power supply to module electronics	

Pin 2,6	**GND to Electronics**
0V power supply to module electronics	

Pin 3,4	**Not Connected**
Do not connect these pins	

Pin 7,8	**Not Used**
Transmit and receive signals for asynchronous serial interface on slave boards. Pin 7,8 are not used by the Profibus Master modules	

Pin 9 - 18	**Address bus A0 - A9**
Address bus to DPRAM. The Profibus Master modules are normally equipped with a 55 ns or faster	

DPRAM. A_0 is LSB.

Pin 19 – 26	**Data bus**
Data bus to DPRAM. The Profibus Master modules are normally equipped with a 55 ns or faster DPRAM. D_0 is LSB and D_7 is MSB.	

Pin 27	**/BUSY**
Indicates simultaneously access of the same memory location from both sides of the DPRAM. Active low signal. Open collector output. This signal has to be pulled-up via a resistor on the Host side. When using the IRQ handshaking system the BUSY signal does not have to be implemented. If the host side shall access the Dual Port Memory regardless of the bus update cycle it is necessary to use the BUSY signal to ensure that there will not be any simultaneously read / write on the same byte from both sides of the DPM.	

Pin 28	**/IRQ**
DPRAM interrupt signal. Indicates when new data are available in the Dual Port Ram. When this interrupt occurs the Host can read address 7FEh. This address contain information that can be used for determine which data are updated. This signal is active low open collector output. This signal shall be pulled-up via a resistor on the Host side.	

Pin 29	**/OE**
DPRAM Output enable signal. From Host to DPRAM. Active low.	

Pin 30	**R/W**
DPRAM Read Write signal. From Host to DPRAM. Write is active low.	

Pin 31	**/CE**
DPRAM Chip Enable signal. From Host to DPRAM. Active low.	

Pin 32	**/RESET**
MASTER module Reset signal. This signal resets and disables the modules functions. Active low.	

Pin 33	**Address bus A10**
Address bus to DPRAM. The Profibus Master modules are normally equipped with a 55 ns or faster DPRAM.	

Pin 34	**Not Connected**
This line is internally not connected	

3.1.3 Dual Port Memory

Please see Appendix B for electrical and timing characteristics of the DPRAM Cypress CY7C136.

3.1.4 Accessing the DPM

The DPM is accessed using the Intel addressing mode. This results in a few considerations which are necessary if a chosen Host processor is not using the same conventions. The following table should give you an overview about how to access different areas.

Please note that unless otherwise stated, word and long values are stored in Intel format.

The data transfer between the Host and the Device is performed via a Dual Port Memory located on the Profibus Master module.

In addition to the general specification, information regarding the following is added:

- Profibus-DP system information
- Firmware information
- Mailboxes

	Example offset	
Byte - even address	0x000	0x000 = byte – data at D0-D7, A0=0
Byte - odd address	0x001	0x001 = byte – data at D0-D7, A0=1
Word	0x010	0x010 = low byte – data at D0-D7, A0=0, 0x011 = high byte – data at D0-D7, A0=1
Long	0x020	0x020 = Least significant byte – data at D0-D7, A0=0, … 0x023 = Most significant byte – data at D0-D7, A0=1

Table 4. 8-bit DPM access

3.2 Hardware interface in enhanced version (16 bit version)

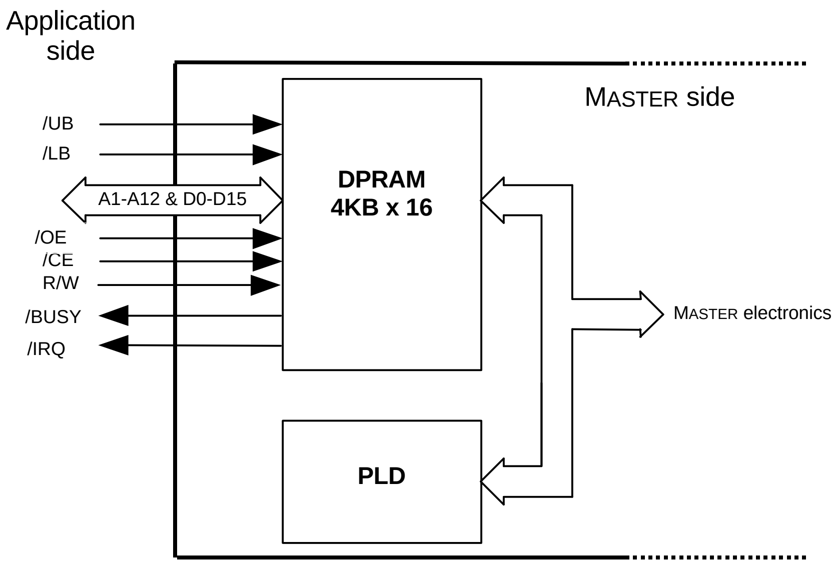


Figure 10. Application interface in standard version

3.2.1 Host Connector

The Host connector is a 48 pin strip connector in three rows (2 rows x 17 pins + 1 row x 14 pins). Several pin lengths are available. The host connector can be mounted on the bottom side or the top side of the board.

Pin No.	Name	Function	Type
1,5	VCC	+5V to electronics	Power
2,6	GND	0V to electronics	Power
3	-	Not Connected	NC
4	-	Not Connected	NC
7	/UB	Higher byte Select of DPM	Input
8	/LB	Lower byte Select of DPM	Input
9	A0	Not Connected	NC
10 - 18	A1 - A9	Address bus to DPRAM. A1 at pin 10, A9 at pin 18	Input
19 - 26	D0 - D7	Data bus to DPRAM. D0 at pin 19, D7 at pin 26	Bi-directional
27	/BUSY	DPRAM busy signal	Output
28	/IRQ	DPRAM interrupt signal	Output
29	/OE	DPRAM output enable signal	Input
30	R/W	DPRAM read / write signal	Input
31	/CE	DPRAM Low byte chip enable signal.	Input
32	/RESET	MASTER module RESET signal	Input
33	A10	Address bus to DPRAM.	Input
34	A11	Address bus to DPRAM.	Input
35	/SENSE	Connected to pin 2. Can be used to detect enhanced master.[3]	
36	-	Not Connected	NC
37	A12	Address bus to DPRAM	Input
38-40	A13-A15	Reserved for extending address bus.	NC (Input)
41-48	D8-D15	Data bus to DPRAM. D8 at pin 41, D15 at pin 48	Bi-directional

Table 5. Application connector pin description, enhanced version

3.2.2 Signal Description

Pin 1,5	**Power supply to Electronics**
+5V power supply to module electronics.	

Pin 2,6	**GND to Electronics**

[3] Connect pin 35 with a pull up resistor to VCC on the Host side. When using the enhanced Profibus Master this signal is put down to 0V via the /SENSE pin. In standard version pin 35 is not used and the signal is then pulled up to VCC on the host side.

0V power supply to module electronics.

Pin 3,4 Not Connected

Do not connect these pins

Pin 7,8 /UB and /LB

/LB enables the low byte (D0-D7) of the Dual Port Memory. /UB enables the high byte (D8-D15) of the Dual Port Memory. For byte access to the Dual Port Memory $A0_{HOST}$ can be connected to /LB and $A0_{HOST}$ can be connected via an inverter to /UB. For Word access to the Dual Port Memory /LB and /UB lines have to be enabled at the same time.

Pin 9 A0 (Not Used)

Not used by enhanced Profibus Master. To keep a compatible design to the standard DT modules connect this pin to A0.

Pin 10 - 18 Address bus A1-A9

Address bus to DPRAM. The Profibus Master modules are normally equipped with a 55 ns or faster DPRAM. A_1 is LSB.

Pin 19 - 26 Data bus D0-D7

Data bus to DPRAM. The Profibus Master modules are normally equipped with a 55 ns or faster DPRAM. D_0 is LSB and D_7 is MSB.

Pin 27 /BUSY

Indicates simultaneously access of the same memory location from both sides of the DPRAM. Active low signal (CMOS/TTL output).

When using the IRQ handshaking system the BUSY signal does not have to be implemented. If the Host side shall access the Dual Port Memory regardless of the bus update cycle it is necessary to use the BUSY signal to ensure that there will not be any simultaneously read/write on the same byte from both sides of the DPM.

Pin 28 /IRQ

DPRAM interrupt signal. Indicates when new data are available in the Dual Port Ram. When this interrupt occurs the Host can read address 7FEh. This address contain information that can be used for determine which data are updated.
This signal is an active low output.

Pin 29 /OE

DPRAM Output enable signal. From Host to DPRAM. Active low.

Pin 30 R/W

DPRAM Read / Write signal. From Host to DPRAM. Write is active low.

Pin 31 /CE

DPRAM Chip Enable signal. From Host to DPRAM. Active low. This signal only enables the low byte on data lines D0-D7

Pin 32	**/RESET**
The Profibus Master module reset signal. This signal resets and disables the modules functions. Active low.	

Pin 33,34	**Address bus A10, A11**
Address bus to DPRAM. The Profibus Master modules are normally equipped with a 55 ns or faster DPRAM. A_{11} is MSB and located on pin 34.	

Pin 35	**/SENSE**
Connected to pin 2 (GND). Using a pull up to this pin on the Host side, a sense of the enhanced board is possible – in comparison to the standard board.	

Pin 36	**Not Connected**
Do not connect this pin	

Pin 37	**Address bus A12**
MSB bit of Address bus to DPRAM. The Profibus Master modules are normally equipped with a 55 ns or faster DPRAM. Total address range 8 kbyte	

Pin 38-40	**Address bus A13-A15**
Expansion of Address bus. A_{15} is MSB. Total address range expandable to 64kB bit	

Pin 41 - 48	**Data bus D8-D15.**
Data bus to DPRAM. The Profibus Master modules are normally equipped with a 55 ns or faster DPRAM. D_8 is LSB and D_{15} is MSB.	

3.2.3 Dual Port Memory

Please see Appendix C for electrical and timing characteristics of the DPRAM Cypress CY7C024.

3.2.4 Accessing the DPM

Address	Example offset	
Byte - even address	0x000	0x000 = byte – data at D0-D7, /LB=0
Byte - odd address	0x001	0x001 = byte – data at D8-D15, /UB=0
Word	0x010	0x010 = low byte – data at D0-D7, /LB=0, 0x011 = high byte – data at D8-D15, /UB=0
Long	0x020	0x020 = Least significant byte – data at D0-D7, /LB=0, … 0x023 = Most sig. byte – data at D8-D15, /UB=0

Table 6. 16-bit DPM access

3.3 LED - Indicators

Three LED indicators are located on the Profibus Master module. These LED's indicate the module status and the fieldbus status.

The Host must take care of LED indications outside the Profibus Master module. Most fieldbus systems are using 3 (three) LED indicators. However, it is recommended to prepare the Host for maximum 8 (eight) LED indicators for future systems. The information for LED indicators are available in the DPM.

3.3.1 Ready

This LED corresponds to D7 (Ready) in Control to Host.

Colour Green	Signal
ON	Module OK 'Device Error' = 0
cyclic flash (approx. 1 Hz.; 50% ON, 50% OFF)	Flash only contain boot loader, no valid firmware stored in flash 'Device Error' <> 0
cyclic flash (approx. 4 Hz.; 50% ON, 50% OFF)	Hardware or system error or firmware / configuration database download in progress 'Device Error' <> 0
OFF	Hardware error

Table 7. LED ready

If a hardware or system error was detected, then the error code is placed in address offset 7F2 hex. 'Device Error'.

3.3.2 RUN

This LED corresponds to the Run / Ready / Com bits in 'Control to Host' register (see chapter 4.3.8).

Colour Green	Signal
cyclic flash (approx. 4 Hz.; 50% ON, 50% OFF)	ready for communication Ready bit = 1 Run bit = 1 Com bit = 0
acyclic flash (see following note)	configuration error or fatal error Ready bit = 0 / 1 Run bit = 0 Com bit = 0
On	communication running Ready bit = 1 Run bit = 1 Com bit = 1

Table 8. Led - Run

- Acyclic flash

The acyclic flashing indicates a configuration error or fatal error. – e.g. no valid configuration loaded. If this error does not resolve after a repeated download of a new configuration, please contact HMS.

The following picture is indicating the general look. – The period of the indication is approx. 10 seconds.

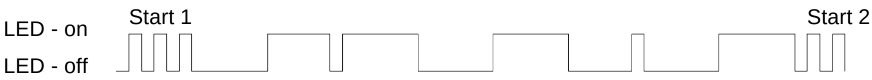


Figure 11. Indication of the general look

3.3.3 Error

This LED is lit if a bus error occurred (e.g. a remote node was not found).

Colour Red	Signal
Off	No Errors detected
On	Error on communication line

Table 9. LED - Error

The error indicated here is fieldbus dependent. For Profibus-DP the address offset 740 hex. 'global bits' can be used to derive the LED state.

4. Dual Port Memory (DPM)

The data transfer between Host and Device are performed via a Dual Port Memory located on the Profibus Master module.

The following description is based on a general master implementation. For a more detailed description regarding the functionality of the mailboxes, system information and firmware information, please refer to the corresponding fieldbus specific specification.

4.1 Memory Map in Standard Version (8 bit version)

An 8-bit dual port memory of 2k x 8 size is used as the common interface between Host and Device.

Address:	Name	Function:
000h - 1FFh	Write data (512 bytes)	Process data written from host to the DPM. Data is transmitted via the device to the connected remote nodes on the network. Linear data area which is connected to remote nodes via the configuration tool. Used for fast cyclic data.
200h - 3FFh	Read data (512 bytes)	Process data read from DPM to the Host. Data is received via the device from the connected remote nodes on the network. Linear data area which is connected to the remote nodes via the configuration tool. This is used for fast cyclic data.
400h - 51Fh	Device Mailbox (288 bytes)	Mailbox for telegrams from Host to Device.
520h - 53Fh	reserved	Do not use!
540h - 65Fh	Host Mailbox (288 bytes)	Mailbox for telegrams from Device to Host
660h - 7EFh	system information	Runtime information about the firmware and the fieldbus system used
7F0h - 7FBh	firmware information	Information about the firmware used
7FCh - 7FDh	reserved	Do not use!
7FEh	Control to Host (1 byte)	Control of data exchange. Written by Device, which sets Interrupt to Host. Read from Device, which clears Reset to Host.
7FFh	Control to Device (1 byte)	Control of data exchange. Written by Host, which sets Interrupt to Device. Read from Host, which clears Reset to Device.

Table 10. DPM memory map in enhanced version

4.2 Memory Map in Enhanced Version (16 bit)

A 16-bit dual port memory of 4k x 16 size is used as the common interface between the Host and the Device. The mapping of the highest 1 kbyte of memory is similar to the 8 bit version.

One 16-bit dual port memory of 4k x 16 bit size is used as common interface between the Host and the Device. The mapping is similar to the 8 bit version.

When using a 16 bit access, please make sure that the microprocessor is reading the memory map according to the following conventions:

Address	Name	
even addresses	A0 = '0', /LB='0' (active)	D0-D7 valid e.g. address 7FE hex
odd addresses	A0 = '1', /UB='0' (active)	D8-D15 valid e.g. address 701 hex.
word addressing	A0 not used, /LB = /UB = '0' (active)	D0-D7 lower byte (even address), D8-D15 upper byte (odd address)

Table 11. DPM memory map

Address:	Name	Function:
0000h - 0DFFh	Write data (3584 bytes)	Process data written from host to the DPM. Data are transmitted via the device to the connected remote nodes on the network. Linear data area which is connected to remote nodes via the configuration tool. Used for fast cyclic data.
0E00h - 1BFFh	Read data (3584 bytes)	Process data read from DPM to the Host. Data are received via the device from the connected remote nodes on the network. Linear data area which is connected to the remote nodes via the configuration tool. Used for fast cyclic data.
1C00h - 1D1Fh	Device Mailbox (288 bytes)	Mailbox for telegrams from Host to Device.
1D20h - 1D3Fh	reserved	Do not use!
1D40h - 1E5Fh	Host Mailbox (288 bytes)	Mailbox for telegrams from Device to Host
1E60h - 1FEFh	system information	Runtime information about the firmware and the fieldbus system used
1FF0h - 1FFBh	firmware information	Information about the firmware used
1FFCh-1FFDh	Control to Host (MSB byte)	Control of data exchange. Written by Device, which sets Interrupt to Host. Read from Device, which clears Reset to Host.
1FFEh-1FFFh	Control to Device (MSB byte)	Control of data exchange. Written by Host, which sets Interrupt to Device. Read from Host, which clears Reset to Device.

Table 12. DPM memory map in enhanced version

4.3 Detailed Description of Dual Port Memory

The addresses indicated in this section are based on the 2kbyte standard version. For the enhanced version, an offset of +6k (+1800h) must be added to all addresses indexed 0400h or higher. This means for example, that the Device Mailbox starts at 1C00h in enhanced version instead of 0400h in standard version. An exception to this offset are the flags 'Control to Host' and 'Control to Device' which are starting at 1FFCh respectively 1FFE in the enhanced version.

4.3.1 Write Data (000 hex - 1FF hex)

The process data is written from the host to the DPM. Data is transmitted via the device to the connected remote nodes on the network. Linear data area which is connected to remote nodes via the configuration tool.

The size of this area is 4096 output points (512 bytes) in standard version and the number of output points are (3584 bytes) in the enhanced version.

Using one of the controlled handshaking procedures, presented later, consistency is guaranteed over the entire area. In the uncontrolled mode there will be only byte consistency (word consistency in the enhanced version only).

In order to find out where the write data for an actual remote node is stored, it is necessary to check with the configuration performed using the configuration software.

4.3.2 Read Data (200 hex - 3FF hex)

The process data is read from DPM to the Host. The data is received via the device from the connected remote nodes on the network. The linear data area, which is connected to the remote nodes via the configuration tool. Used for fast cyclic data.

Size of this area is 4096 input points(512 bytes) in standard version and 28672 input points (3584 bytes) in enhanced version.

Using one of the controlled handshaking procedures, presented later, consistency is guaranteed over the entire area. In the uncontrolled mode there will be only byte consistency (word consistency in the enhanced version).

In order to find out where the read data for an actual remote node is stored, you would have to check with the configuration you performed using the configuration software.

4.3.3 Device Mailbox (400 hex - 51F hex)

The mailbox handles message telegrams from the Host to the Device. The Host puts the message into the mailbox and indicates new data via control flags. The Device reads new data and gives acknowledge via control flags. See 4.5 for message structure and how to use them.

4.3.4 Host Mailbox (540 hex - 65F hex)

The mailbox handles message telegrams from the Device to the Host. The device puts message into the mailbox and indicates new data via control flags. Host reads new data and give acknowledge via control flags. See 4.5 for message structure and how to use them.

4.3.5 System Information (660 hex - 7EF hex)

Area	type	address	description
Firmware Name	16 byte	660 h - 66F h	Text String (readable char)
Firmware Version	16 byte	670 h - 67F h	Text String (readable char)
Reserved	-	680 h - 6BF h	do not use!
Fieldbus_Init	64 byte	6C0 h - 6FF h	initialisation information for initialisation routine
Reserved	-	700 h - 73F h	do not use!
Fieldbus_State	64 byte	740 h - 77F h	state information about the connected fieldbus
Reserved	-	780 h - 7EF h	do not use!

Table 13. Overview System Information

The following table shows an example of how Firmware name and Firmware version could look like.

Variable	address	hex dump	character
Firmware Name	660 h - 66F h	44 50 4D 20 20 20 20 20 48 4D 53 2D 44 50 4D 20	DPM HMS-DPM
Firmware Version	670 h - 67F h	56 30 31 2E 30 30 30 20 30 34 2E 30 36 2E 39 37	V01.000 04.06.97

Table 14. Example System Information

4.3.6 Profibus-DP Init (6C0 hex. - 6FF hex.)

The following values are taken over after an initialisation sequence triggered by the Init bit in 'Control to Device'. This means that a Data Exchange Mode can be overwritten using the Init. sequence.

Variable	type	address	description
Data Exchange Mode	byte	6C0 h	0: bus synchronous, device controlled 1: buffered, device controlled 2: uncontrolled 3: buffered, host controlled 4: bus synchronous, host controlled
Cycle Time	byte	6C1 h	Bus cycle time in [ms] – allows changing the bus cycle time
Reserved	-	6C2 h - 6FF h	do not use!

Table 15. Overview Profibus-DP Init Parameter

4.3.7 Profibus-DP State (740 hex. - 77F hex.)

Variable	Type	address	description
Global bits	1 byte	740 h	global error bits
Master state	1 byte	741 h	main state of the master system
Err_Rem_Addr	1 byte	742 h	Error remote address
Err_Event	1 byte	743 h	error number
Bus_Error_Count	2 byte	744 h – 745 h	Heavy bus error count
Time_out_count	2 byte	746 h – 747 h	Number of rejected Profibus telegrams
Reserved	8 byte	748 h - 74F h	do not use
Slave Config	16 byte	750 h - 75F h	bit field to classify every remote node as configured ‘1’ or not configured ‘0’
Slave state	16 byte	760 h - 76F h	bit field to classify every remote node as active ‘1’ or inactive ‘0’
Slave diagnosis	16 byte	770 h - 77F h	bit field to show the diagnostic bit of every remote node

Table 16. Overview Profibus-DP state

4.3.7.1 Global Bits (740 hex.)

Bit no.	value	description
7 - 3	0	reserved
2	no data	1 = at least one remote node is not in the data exchange mode or reports fatal error 0 = OK
1	auto clear	1 = device branched into mode auto clear, because of a remote node error 0 = OK
0	control	1 = a parameter error occurred 0 = OK

Table 17. Global bits

Auto clear:

If the device is configured in auto clear mode – see 'configuration software' or 'set master parameter' telegram – and at least one remote node is not in the data exchange mode, then the outputs of all remote nodes will be cleared and this bit will be set.

4.3.7.2 Master / Network State (741 hex.)

Value	description
00 h	OFFLINE
40 h	STOP
80 h	CLEAR
C0 h	OPERATE

Table 18. Profibus-DP Master / network state

4.3.7.3 Error Remote Address (742 hex)

This flag indicates the node number of a faulty node. The error code is specified in error event, presented in the next section.

Value	description error remote address
00 h - FE h	lowest no. of node, which has an error. See also section 4.2.4
FF h	device has an internal error. See also section 4.2.5

Table 19. Error address

4.3.7.4 Error Event – External Error (743 hex.)

Depending on the value in 742 hex, this value either represents an internal or an node error. If 742 hex. not equals FF Hex., a node (external) error according the following table is noted.

Value	description error event	error source	help
0	Remote nodeOK		
3	function in remote node is not activated	remote node	check if remote node is Profibus-DP norm conform or the correct GSD files are used
9	no answer data	remote node	check bus cable
17	no response of the slave	remote node	check bus cable, bus address of remote node
18	The device is not into the logical token ring	device	check FDL / node address of master or highest station address of other master systems

Table 20. External error event

4.3.7.5 Error Event – Internal Error (743 hex.)

Depending on the value in 742 hex, this value either represents an internal error or a node error. If 742 hex. equals FF Hex., an internal error according the following table is recognised.

Value	description error event	error source	help
0	no error		
50 – 53	internal error	device	contact HMS
54	no master parameter	device	execute configuration download again
55	faulty parameter-value in the master parameter	project planing	contact HMS
56	no remote node parameter	project planing	execute configuration download again
57	faulty parameter-value in the remote node parameter	project planing	contact HMS
58	double remote node address	project planing	check remote node addresses
59	projected send process data offset address of a node outside the allowable border	project planing	check projected send offset addresses
60	projected receive process data offset address of a node outside the allowable border	project planing	check projected receive offset addresses
61	data areas of remote nodes are overlapping in the receive process area	project planing	check projected receive offset addresses
62	data areas of remote nodes are overlapping in the send process area	project planing	check projected send offset addresses
202	no segment free	device	contact HMS
212	faulty reading of configuration data	device	execute configuration download again
213	system fault	device	contact HMS
others	not allowed	-	contact HMS

Table 21. Internal error event

4.3.7.6 Time Out Count (746 hex – 747 hex..)

Number of rejected Profibus Telegrams. One reason for an increment here might be, that there is a bus short circuit.

4.3.7.7 Slave Config (750 hex. – 75F hex.)

Remote node state	16 byte	750 h – 75F h	
Remote node no. 0	bit	bit 750.0	0 = Remote node is not configured 1 = Remote node is configured
Remote node no. 1	bit	bit 750.1	
…	…	…	
Remote node no. 7	bit	bit 750.7	
Remote node no. 8	bit	bit 751.0	
…	…	…	
Remote node no 126	bit	bit 75F.6	

Table 22. Overview Remote node Config

4.3.7.8 Slave State (760 hex. – 76F hex.)

Remote node state	16 byte	760 h - 76F h	
Remote node no. 0	Bit	bit 760.0	0 = Remote node inactive 1 = Remote node is active
Remote node no. 1	Bit	bit 760.1	
…	…	…	
Remote node no. 7	Bit	bit 760.7	
Remote node no. 8	Bit	bit 761.0	
…	…	…	
Remote node no 126	Bit	bit 76F.6	

Table 23. Overview Remote node state

4.3.7.9 Slave Diagnostic (770 hex. – 77F hex.)

Remote node state	16 byte	770 h - 77F h	
Remote node no. 0	Bit	bit 770.0	0 = no new diagnostic available 1 = new diagnostic available[4]
Remote node no. 1	Bit	bit 770.1	
…	…	…	
Remote node no. 7	bit	bit 770.7	
Remote node no. 8	bit	bit 771.0	
…	…	…	
Remote node no 126	bit	bit 77F.6	

Table 24. Overview remote node diagnostics

[4] Cleared after the diagnostics are received by the host

4.3.8 Firmware Information (7F0 hex - 7FB hex)

Variable	type	address	short signification
Reserved	byte	7F0h - 7F1h	do not use!
Device Error	byte	7F2h	Device Error - see table 9 "Device Error"
Reserved	byte	7F3h - 7F5h	do not use!
Master recognition	byte	7F6h	=0 on Master Modules, <> 0 on slave modules
Data Exchange Mode (upper nibble)	byte	7F7h	1xh: bus synchronous, device controlled 2xh: bus asynchronous, device controlled 3xh: standard, uncontrolled 4xh: bus asynchronous, host controlled 5xh: bus synchronous, host controlled
DPM Size	byte	7F8h	valid values are 1 dec. to 64 dec.; e.g.: 02h : 2k byte DPRAM installed, standard version 08h : 8k byte DPRAM installed, enhanced version
Device Type	byte	7F9h	Device code 34h: Profibus Master
FB Type	byte	7Fah	Type of Fieldbus 31h: 8-bit Profibus-DP Master 32h: 16-bit Profibus-DP Master
Reserved	byte	7FBh	fixed to 48h = "H" - HMS Fieldbus Systems

Table 25. Firmware information

Error No.	Symbol	Description
0	-	no error
14		OS module, Firmware download
50	RAM_TEST	RAM check not OK
53	FLASH_TEST	FLASH PROM checksum not OK
100 – 107	SYSTEM	Internal System Error
200	Unknown_IRQ	Unknown interrupt received – e.g. through system crash
201	Watchdog	internal watchdog expired
202	TX_IRQ	Unexpected transmit interrupt from serial channel
203	RX_IRQ	Unexpected receive interrupt from serial channel
252	Download active	Firmware Download or Database Download active
253	Bootloader active	Bootstrap loader active, firmware not running

Table 26. Device Error – Error Codes

4.3.9 Reserved (7FC hex - 7FD hex)

Variable	type	address	short signification
reserved	byte	7FCh - 7FDh	reserved in 8 bit version. Used in 16 bit version for the handshaking flags.

Table 27. Reserved

4.3.10 Handshake Flag: 'Control to Host'(7FE hex)

Interrupt generates a byte for indication that the Host must make a response. This byte can use up to eight flags (8 bits). These flags refer to both linear area and mailbox area. This byte is READ ONLY for the Host. The Host can use this as an interrupt function if using the /IRQ DPM or poll this register to check for new data. New data written in this byte from Device will always set the /IRQ DPM signal to low level. A read on this byte from the Host will Reset the /IRQ DPM signal to high level.

Bit	Signal	Description
D7	Ready	device running; corresponds to LED Ready 1: device is ready 0: device is not ready
D6	Run	communication; corresponds to LED Run 1: communication is running[5] 0: communication is not running
D5	Com	process data exchange active 1: data is getting exchanged 0: data is not getting exchanged
D4 - D3	reserved	
D2	PdAck	process data synchronisation bit (see 4.4)
D1	DevAck	acknowledge bit for Device Mailbox (see 4.5)
D0	HostCom	command bit for Host mailbox (see 4.5)

Table 28. Host flags

[5] D6 is indication that the communication is running. D6 will be set even if one or several nodes are disconnected

4.3.11 Handshake Flag: Control to Device (7FF hex)

Interrupt generates a byte for indication that the Host must make a response. This byte can use up to eight flags (8 bits). These flags refer to both linear area and mailbox area. The byte is WRITE ONLY for the Host.

Bit	Signal	Description
D7	Reset	reset of device with initialisation in accordance to default parameter 1: reset device 0: device is not getting a reset
D6	Init	init. of device with parameters from DPM 1: initialise device 0: device is not getting initialised
D5	NotRdy	Host is not ready 1: host is not ready 0: host is ready
D4 - D3	reserved	
D2	PdCom	process data synchronisation bit (see 4.4)
D1	DevCom	command bit for Device Mailbox (see 4.5)
D0	HostAck	acknowledge bit for Host mailbox (see 4.5)

Table 29. Device flag

- Reset (D7)
 This flag can be used to reset the device by software, instead of driving the reset pin low on the connector. See also 5.1 "Module Start-up".
- Init (D6)
 Setting the Init flag will start initialising the device with the parameters written to the dual port memory. See also 5.1.3 "Software Init"
- NotRdy (D5)
 A ‘1’ in this flag will indicate that the Host is not ready to communicate. This will stop all action from the device to communicate through the dual port memory with the host and via the fieldbus. Meaning that there will be no process data update and no handshaking through the mailbox.

5. Access of Process Data

The following flags of the handshake register are used for the delivery of process data (see chapter 4.3.6 / 7):

- PdAck in 'Control to Host'
- PdCom in 'Control to Device'

There are five different handshaking algorithms for exchanging data in the dual port memory. Each method is intended for specific applications and are chosen during configuring of the system.

Synchronisation	controlling	consistence
bus synchronous	device controlled	consistency
buffered	device controlled	consistency
none	uncontrolled	no consistency[6]
buffered	user controlled	consistency
bus synchronous	user controlled	consistency

Table 30. Access methods

5.1.1 Delivery Procedure: Uncontrolled

The uncontrolled access method can be used where no data consistency is necessary, and where the host does not want or is not able to do any handshaking.

The Com bit in the Control to Host' flag is set as soon as the data exchange over the bus happens. To each valid bus cycle the device makes an update with the DPRAM, ignoring the bits PdAck, PdCom.

The Host side must use the BUSY signal to avoid data collision in the memory. Standard version (8 bit version) gives byte consistency only. Enhanced version (16 bit version) gives word consistency when using even address.

Accesses which require consistency for larger areas should not use this access method.

5.1.2 Delivery procedure: bus synchronous, Device controlled

This data exchange method can be used, if fast data delivery is important, and the host is able to update the DPRAM data within the bus cycle time (approx. a few ms).

- The Device is initiating data exchange through the dual port RAM.
- The Host has to respond on the initiative.
- If the Host fails to respond, the master stops communication.

[6] Word consistency in enhanced mode

Device internal action	PdAck	Device	Host	PdCom	Host internal action
device owns DPRAM, waits until fieldbus cycle finished	0			0	
Device updates DPRAM data	0			0	
pass access rights to Host	1	toggle PdAck →		0	
	1			0	Host updates DPM data
	1		← toggle PdCom	1	pass access rights to Device
wait for bus cycle finished	1			1	
Device updates DPRAM data	1			1	
pass access rights to Host	0	toggle PdAck →		1	
	0			1	Host updates DPRAM data
	0		← toggle PdCom	0	pass access rights to Device
wait for bus cycle finished	0			0	

Figure 12. Delivery procedure: Bus synchronous, device controlled

To each valid bus cycle, the device makes an updating data exchange with the process data buffers in the dual port RAM. The end of the exchange is shown to the Host by inverting the PdAck bit of the cell 'Control to Host'. The bits PdAck and PdCom become unequal and the Host program gets the access on both data buffers.

The Host program must receipt its access within the following bus cycle by inverting the bit PdCom of the cell 'Control to Device'. The states of the bits PdAck and PdCom get equal and the device gets back the access on the process data of the dual port memory.

If the Host program does not succeed to adjust the bit until the next bus cycle is done, the bus communication stops. A communication stop will lead to a system halt. The only way to resolve this situation would be a warm start (Init) or a cold start (reset).

5.1.3 Delivery procedure: buffered, Device controlled

This data exchange method can be used if fast data delivery is important, but where the host can not guarantee to update the DPRAM data within the bus cycle time (approx. a few ms). While the host owns the DPRAM, the DPRAM process data will not be updated.

- The Device is initiating data exchange through the dual port RAM.
- The Host has to respond on the initiative.

Device internal action	PdAck	Device	Host	PdCom	Host internal action
device owns DPRAM and waits until bus is cycle finished	0			0	
Device updates DPRAM data	0			0	
pass access rights to Host	1	toggle PdAck		0	
	1			0	Host updates DPM data
	1		toggle PdCom	1	pass access rights to Device
wait for bus cycle finished	1			1	
Device updates DPRAM data	1			1	
pass access rights to Host	0	toggle PdAck		1	
	0			1	Host updates DPRAM data
	0		toggle PdCom	0	pass access rights to Device
wait for bus cycle finished	0			0	

Figure 13. Delivery Procedure: Buffered, Device Controlled

To each valid bus cycle the device does an update with the two internal buffers. If the device has access rights on the process data area of the DPRAM (PdAck == PdCom) an exchange between the internal buffers and the DPRAM is executed. The end of the exchange is shown to the Host by inverting PdAck in Host Flag (PdAck != PdCom) and the Host gets access to the process data buffers.

If the Host program has finished its work on the process data areas, it has to return the access rights by inverting PdCom (PdAck == PdCom). There will be no update of data in the DPRAM until accessed is passed back.

5.1.4 Delivery procedure: buffered, Host controlled

This data exchange method can be used when the host wants to keep access to the DPRAM as long as possible. After updating the DPRAM information, it passes the access to the device for updating the device internal data buffer.

- The Host is initiating data exchange through the dual port RAM.
- The Device responds on the initiative.

Device internal action	PdAck	Device	Host	PdCom	Host internal action
	0			0	Host updates DPRAM data
	0		toggle PdCom	1	pass access rights to Device
Device updates DPM data and internal buffer	0			1	
pass access rights to Host	1	toggle PdAck		1	
	1			1	Host updates DPRAM data
	1		toggle PdCom	0	pass access rights to Device
Device updates DPM data and internal buffer	1			0	
pass access rights to Host	0	toggle PdAck		0	

Figure 14. Delivery procedure: buffered, Host controlled

The Com bit in the cell host flags is set as soon as the data exchange over the bus happens. To each valid bus cycle the device does an update with the two internal buffers. By inverting the PdCom bit in the cell 'Control to Device' the access to the DPRAM is passed over to the device (PdCom != PdAck).

After the Device has exchanged the data, PdAck is assigned the value of PdCom (PdCom == PdAck) passing back the access rights to the Host.

5.1.5 Delivery procedure: bus synchronous, Host controlled

This data exchange method can be used, where data should be transferred as fast as possible to the Fieldbus. Whenever the Access rights are passed over to the device, one bus cycle is issued.

- The Host is initiating data exchange through the dual port RAM.
- The Device responds on the initiative.
- Returns access after bus cycle is finished

Device internal action	PdAck	Device	Host	PdCom	Host internal action
	0			0	Host updates DPRAM data
	0		toggle PdCom	1	pass access rights to Device
Device reads DPRAM data	0			1	
start bus cycle	0			1	
Device writes DPRAM data	0			1	
pass access rights to Host	1	toggle PdAck		1	
	1			1	Host updates DPM data
	1		toggle PdCom	0	pass access rights to Device
Device reads DPRAM data	1			0	
start bus cycle	1			0	
Device writes DPRAM data	1			0	
pass access rights to Host	0	toggle PdAck		0	
	0			0	start bus cycle
	0			0	Host updates DPRAM data

Figure 15. Delivery procedure: bus synchronous, Host controlled

The start point of a bus cycle is fixed by the Host program. After actualisation of both process data buffers the Host starts exactly one bus cycle by inverting bit PdCom (PdCom != PdAck). After the bus cycle is finished the device inverts PdAck (PdCom == PdAck), which equals PdAck and PdCom.

6. Mailbox Interface

6.1 Introduction

A message is a data structure which is transferred via the mailbox system in the dual port memory. This interface is used to exchange information between the host and the device. The contents of this information is either to send or to get information from the device or the connected fieldbus.

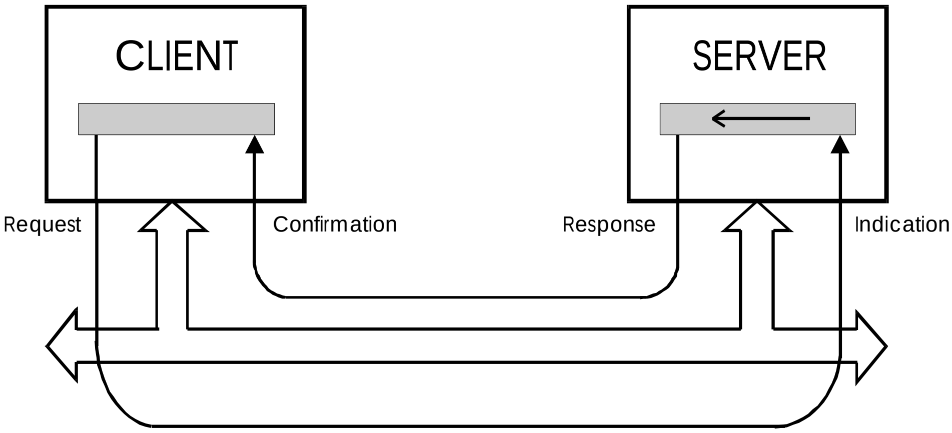


Figure 16. Client - Server Relationship

Communication relationships between a client and a server can be classified in the following manner:

- Request: send from a client to the server
- Indication: request telegram received from client
- Response: answer from server on request telegram
- Confirmation: response telegram received by client

Comparing the above picture with the Profibus Master system, the terms Client and Server would be used in the following way:

If the Host wants to take action; e.g. obtain some slave (slave=remote node) diagnostics, it would act as a client towards the Device (server), requesting the information. If the information is not available within the device, the device itself would act as a client towards a node in the Profibus-DP network.

In this chapter we are referring to request telegrams and response telegrams; implying indication and confirmation.

A message is built up out of a *Mailbox header*, *Telegram header* and *Telegram data.*

The mailbox header is eight bytes long and defines the sender and the receiver of the task – using this structure, the device is able to distinguish by the mailbox header where the telegram is going to (e.g. device internal or fieldbus).

The Telegram header is eight bytes long (if used). This header defines the code or command being send. Depending on the type of command, all necessary data might already be stored in the telegram header. Otherwise additional information will follow within the telegram data.

The telegram data contains the necessary information being send. It is either additional information to the telegram header or, if standing without telegram header, it will contain all the necessary information – e.g. if a fieldbus specific telegram is issued. – Telegram data and telegram header can have a maximum total length of 255 bytes.

Type:	Size	Function:
Mailbox header	8 byte	Defines the sender and receiver of the telegram. Total length of the telegram. Control data from the operation system which defines the transfer of the message
Telegram header	0 or 8 byte	Defines the function code / command. Defines data type, where data are located and number of data.
Telegram data	0 - 255 bytes	Message data

Table 31. *General message structure 5 mailbox interface*

Type	Size	Parameter	Function:
Mailbox header	byte	Msg.Rx	Receiving Task no. within device 0: Operating System 1 … 7: other destination within device 16: Host rest: not used
	byte	Msg.Tx	sending task no.– coding as Msg.Rx
	byte	Msg.Len	data length no. (max. 255; telegram header + telegram data)
	byte	Msg.Nr	unique message no.
	byte	Msg.Ans	response no.
	byte	Msg.Err	Error code
	byte	Msg.Com	command no.
	byte	Msg.Seq	Sequence Control byte
Telegram header	byte	Msg.DeviceAdr	Remote Node address - Device address corresponds to a remote node on the fieldbus
	byte	Msg.DataArea	data area – area in a device which is going to be addressed
	word	Msg.DataAdr	data address – defines an object within the data area addressed
	byte	Msg.DataIdx	Index within an object
	byte	Msg.DataCnt	no. of byte being accessed
	byte	Msg.DataType	Data type – e.g. octet string / the coding is implementation dependent
	byte	Msg.Fnc	Function to be performed on data[7]; e.g.: 1: read 2: write
Telegram data	byte	Msg.D[247/255]	Utilisation data

Table 32. Message structure for Profibus-DP

- Msg.Rx
  The receiver of the telegram is placed in this byte. It can either be 16 for the host system or a value less than 8 for a specific task in the device. Which tasks are used in the device can be fond in the fieldbus specific design guide.
- Msg.Tx
  The sender of the telegram is placed in this byte, otherwise the handling is the same as in Msg.Rx
- Msg.Len
  The length of the Message (Telegram Header plus Telegram Data) is stored in this byte. Since this byte is only one byte long, the max. size for the telegram (excluding Mailbox header) is 255 byte.

[7] additional information can be found in the telegram description

- Msg.Nr
  In order to be able to send more then one request to the device, a count has to be used to identify the different telegrams. As long as there is a message buffer free, a new request can be sent to the device.

Note that if more then one message is used, unique message no. are necessary.

- Msg.Ans
  If the telegram is an answer to a previous telegram, this byte is always set to the value placed in Msg.Com of the requesting telegram. Msg.Com is set to 0.
- Msg.Err
  If the telegram is an error message to a previous telegram, this byte is set to the corresponding error code. The command telegram corresponding to the error telegram can be found using the byte Msg.Nr.
- Msg.Com
  If the telegram is a command message, this byte is giving the identifier of the command issued.. Msg.Err and Msg.Ans are set to 0.
- Msg.Seq

D7 - D4	0	reserved - do not use
D3, D2	Seq1, Seq0	00: standard 01: first telegram of a sequence 10: following telegram of a sequence 11: last telegram of a sequence
D1	Nak	0: standard handshaking mode 1: response is suppressed if no error occurred
D0	0	reserved - do not use

Table 33. Sequence control byte

Which values are allowed for the Handshaking byte will be shown where the different telegrams are described. By default only ‘0’ is allowed, which results into the standard handshaking sequence being described later.

- Msg.DeviceAdr
  This byte gives the address of a connected remote node
- Msg.DataAdr
  This bytes identifies the data address within a connected remote node
- Msg.DataArea
  This bytes identifies the data area within a connected remote node
- Msg.DataIdx
  This word identifies the object being addressed by the telegram
- Msg.DataCnt
  This byte gives the no. of bytes being accessed.
- Msg.DataType
  This byte indicates the type of data being transmitted
- Msg.Fnc
  This byte defines the access function being applied on the data
- Msg.D[255]
  Telegram Data being send

6.1.1 Structure of a 'Command Message'

For this example it is assumed, that the receiving task is task no. '0' and the transmitting task comes from the host '16'.

Type	Size	Parameter	Function:
Mailbox header	byte	Msg.Rx	receiving task no. '0'
	byte	Msg.Tx	sending task no. '16' (Host)
	byte	Msg.Len	message length = 8 byte for telegram header + size of telegram data
	byte	Msg.Nr	unique message number - x
	byte	Msg.Ans	no answer message = '0'
	byte	Msg.Err	no error message = '0'
	byte	Msg.Com	command message = command id
	byte	Msg.Seq	Sequence byte
Telegram header	byte	Msg.DeviceAdr	Remote node Address, e.g. node address in Profibus-DP
	byte	Msg.DataArea	data area
	word	Msg.DataAdr	data address
	byte	Msg.DataIdx	data index
	byte	Msg.DataCnt	data count
	byte	Msg.DataType	data type
	byte	Msg.Fnc	function
Telegram data	byte	Msg.D(m)	data to be transferred

Table 34. General command message structure

6.1.2 Structure of an 'Answer Message'

For this example it is assumed that the response is got from the previous command.

Type	Size	Parameter	Function:
Mailbox header	byte	Msg.Rx	receiving task no. '16'
	byte	Msg.Tx	sending task no. '0'
	byte	Msg.Len	message length = 8 byte for telegram header + size of telegram data
	byte	Msg.Nr	unique message number x - corresponds to command message
	byte	Msg.Ans	if answer message = command id; else '0'
	byte	Msg.Err	if error message = error code; else '0'
	byte	Msg.Com	no command message = '0'
	byte	Msg.Seq	Sequence byte
Telegram header	byte	Msg.DeviceAdr	Remote node Address, e.g. node address in Profibus-DP
	byte	Msg.DataArea	data area
	word	Msg.DataAdr	data address
	byte	Msg.DataIdx	data index
	byte	Msg.DataCnt	data count
	byte	Msg.DataType	data type
	byte	Msg.Fnc	function
Telegram data	byte	Msg.D(n)	if error message, no data if answer message, data

Table 35. Message structure for Profibus-DP

6.2 How to operate with the Mailbox

Data exchange through the parameter channel is done via four bits. DevAck, HostCom in 'Control to Host' and DevCom, HostAck in 'Control to Device'. The implemented handshaking is similar to the handshaking used for the process data.

If the corresponding bits are not equal (DevCom != DevAck, HostCom != HostAck) a telegram is activated. If those bits are becoming equal, this telegram is getting consumed.

The following example shows how a handshaking is performed using the interrupts from the DPRAM.

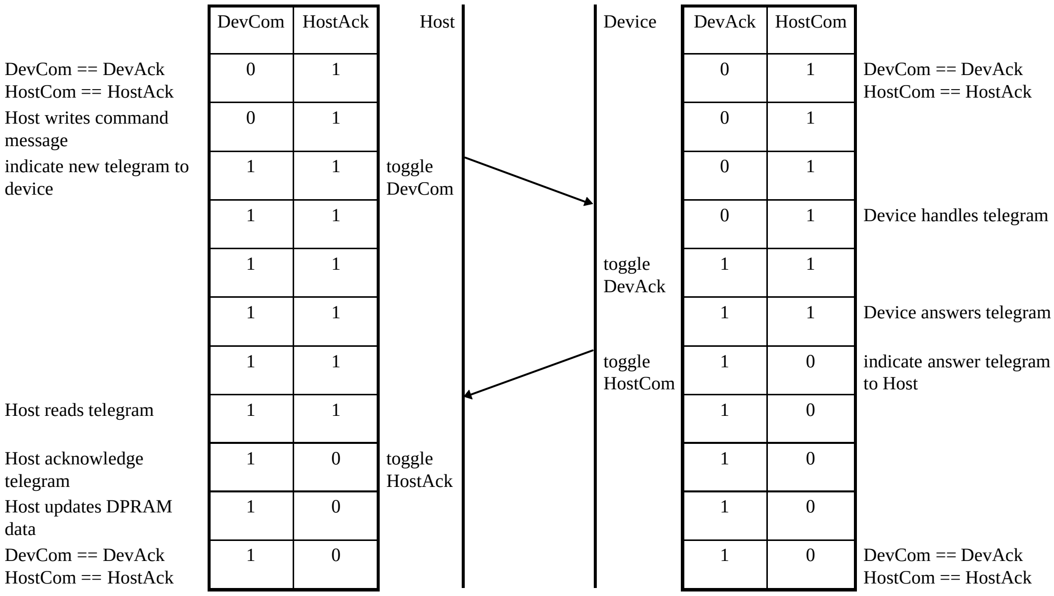


Figure 17. Handshaking using the interrupts from the DPRAM

6.3 Profibus-DP system commands

The following telegrams are of interest if there is need to control certain system properties in the Profibus-DP network.

6.3.1 Global control command

This command initiates the Profibus-DP global control command.

For a more detailed description refer to PROFIBUS norm DIN 19245 Part 3.

Variable	type	value	
Msg.Rx	byte	3	receiver = User Interface
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	3	telegram data=3; telegram header=0 -> overall telegram length=3
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	no error
Msg.Com	byte	70	command, DDLM_Global_Control
Msg.Seq	byte	0	-
Msg.D(0)	byte	0-127	Remote node address
Msg.D(1)	byte	ctrl	control command
Msg.D(2)	byte	group	group select

Table 36. Global Control Command – Command message

- Number of the message
  these flags assign an identification number to the message. The same number will be found on the answering message. It is only necessary to use this number if more then one message will be issued at any time.
- Remote node address
  0-127 are valid values. 127 is sending a global control command to all selected remote nodes (selected by 'group select'). 0-126 only initiate action to the corresponding remote node
- Control command:

D7 - D6	0	reserved
D5	Sync	freeze output data, until sync command is neutralised
D4	Unsync	neutralise the sync command
D3	Freeze	freeze input data
D2	Unfreeze	unfreeze input data
D1	Clear_Data	clear output data
D0	0	reserved

Table 37. Global Control Command – coding

Bit 2 or 4	Bit 3 or 5	
0	0	no function
0	1	function will be activated
1	0	function will be inactive
1	1	function will be inactive

Table 38. Global control Command -– Coding

- group select
  The "group select" function is used to select a grope of remote nodes and put the same tag reference to these remote nodes. The command will be activated on the corresponding remote nodes, if Group_Ident (parameter of remote node) of the slave and Group select are identical.

Answer message:

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	3	transmitter = User Interface
Msg.Len	byte	1	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	70	Global_Control_Command
Msg.Err	byte	0	no error
Msg.Com	byte	0	no command / it was an answer
Msg.Seq	byte	0	Sequence byte
Msg.D(0)Hs	byte	x	Req. remote node address (from command mesage)

Table 39. Global control Command -– Answer message

The global control command will be sent with a multicast command. Therefore this command is always successful and no error will be placed in Msg.Err of the answer message.

- number of message:
  it will be the same number as on the requesting telegram

6.3.2 Slave diagnostic

Get extended slave diagnostic from each individual remote node.

Variable	type	value	
Msg.Rx	byte	3	receiver = User Interface
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	8	telegram length (only telegram header)
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	error, status
Msg.Com	byte	66	command, DDLM_Slave_Diag
Msg.Seq	byte	0	standard handshaking
Msg.DeviceAdr	byte	k	Remote node Address (Logical bus address for Profibus-DP node)
-	byte	0	-
-	word	0	-
-	byte	0	-
Msg.DataCnt	byte	32 / 106	data count = length of the diagnosis structure to be read. This value is fixed to 32 in standard version and 106 in enhanced version.
Msg.DataType	byte	5	data type = octet-string
Msg.DataFnc	byte	1	read

Table 40. Slave diagnostics – Command message

The command serves to read the status of a remote node. The requested remote node address has to be placed in Msg.DeviceAdr corresponding to the real address of the remote node on the bus. The calling of the command results in no bus access, since the master saved the last diagnostic message of any remote node in an internal buffer. The status structure can always be requested from the device. The corresponding status bit of the remote node in the bus status field in the dual-port memory shows, if a diagnosis of the module is available, see section 4.2.

- number of the message
  these flags assign an identification number to the message. The same number will be found on the answering message. It is only necessary to use this number, if more then one message will be issued at any time.
- node address:
  0-126 initiates action to the corresponding remote node

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	3	transmitter = User Interface
Msg.Len	byte	114	length of message
Msg.Nr	byte	j	message no.
Msg.Ans	byte	66	answer, DDLM_Slave_Diag
Msg.Err	byte	0	no error
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking
Msg.DeviceAdr	byte	k	Remote node address
-	byte	0	-
-	word	0	-
-	byte	0	-
Msg.DataCnt	byte	32 / 106	data count = length of the diagnosis structure to be read. Fixed to 32 in standard version and 106 in enhanced version.
Msg.DataType	byte	5	data type = octet string
Msg.DataFnc	byte	1	read
Msg.D(0)	byte	xx	Stationstatus_1
Msg.D(1)	byte	xx	Stationstatus_2
Msg.D(2)	byte	xx	Stationstatus_3
Msg.D(3)	byte	xx	Master address
Msg.D(4-5)	word	0x1005	ident. Number (0x1006 16-bit version)
Msg.D(6-31)	octet string	xx	extended diagnosis

Table 41. Slave diagnostics – Answer message

- number of the message
  these flags assign an identification number to the message. The same number will be found on the answering message. It will only be necessary to use this number if more than one message will be issued at any time. Otherwise setting this value to '0' will also be appropriate.
- node address:
  0-126 is the respond to action on the corresponding remote node
- data count:
  data count is a fixed value. It is set to 32 in standard version and set to 106 in enhanced version.,

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	3	transmitter = User Interface
Msg.Len	byte	8	length of message
Msg.Nr	byte	j	message no.
Msg.Ans	byte	66	answer, DDLM_Slave_Diag
Msg.Err	byte	161	error, DeviceAdr out of range
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking
Msg.DeviceAdr	byte	k	Remote node address
-	byte	0	-
-	word	0	-
-	byte	0	-
Msg.DataCnt	byte	32	data count = length of the diagnosis structure to be read
Msg.DataType	byte	5	data type = octet string
Msg.DataFnc	byte	1	read

Table 42. Slave diagnostics – Error message

6.3.3 Statistic counter

With the following functions it is possible to read and clear the statistic counter of the master module.

6.3.3.1 Coding of statistic counter

The statistic counter to the different remote nodes are mapped according to the following table. Accessing via the functions DDLM_Upload is done with Add_Offset and Data_Len, where:

Add_Offset = n*6; n=[0-126]

Data_Len = n*6; n=[1-39]

Variable	Size
Frame_sent_count_0	unsigned32
Error_count_0	unsigned16
Frame_sent_count_1	unsigned32
Error_count_1	unsigned16
…	…
Frame_sent_count_126	unsigned32
Error_count_126	unsigned16

Table 43. Mapping of statistic counter

see as well DIN 19245 Part 3 Section 10.9 ‘Coding of Statistic Counter)

6.3.3.2 Read statistic counter

The statistic counter can be read from 1 to a maximum of 40 remote nodes with one message. Using DeviceAdr it is possible to select the first remote node whose statistic counter are to be read.

Variable	type	value	
Msg.Rx	byte	7	receiver = FDL Interface
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	8	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	error, status
Msg.Com	byte	0xFE (hex)	command = statistic counter
Msg.Seq	byte	0	standard handshaking
Msg.DeviceAdr	byte	0, …, 126	Remote node address
-	byte	0	-
-	word	0	-
-	byte	0	-
Msg.DataCnt	byte	1 - 40	Length = remote node count
Msg.DataType	byte	5	data type = octet string
Msg.DataFnc	byte	1	read

Table 44. Read statistic counter – Command telegram

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	7	transmitter = FDL Interface
Msg.Len	byte	8 + 6 x n	length of message (n = remote node count)
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0xFE (hex)	answer = statistic counter
Msg.Err	byte	0	error, status
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking
Msg.DeviceAdr	byte	0, …, 126	Remote node address
-	byte	0	-
-	word	0	-
-	byte	0	-
Msg.DataCnt	byte	1 - 40	Length = remote node count
Msg.DataType	byte	5	data type = octet string
Msg.DataFnc	byte	1	read
Msg.D[0]	uint32		frame send count of first remote node in the list
Msg.D[4]	uint16		error count, of first remote node in the list
…			

Table 45. Read statistic counter – Answer telegram

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	7	transmitter = FDL Interface
Msg.Len	byte	0	length of message (n = remote node count)
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0xFE (hex)	answer = statistic counter
Msg.Err	byte	f	error code
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking

Table 46. Read statistic counter – Error telegram

Error no. f	
0	no error
48	CON_TO time out
52	CON_NI area code unknown
53	CON_EA buffer length to big
55	CON_IP faulty parameter detected
57	CON_SI sequence error
59	CON_DI data incomplete or faulty

Table 47. Read statistic counter – Error codes

6.3.3.3 Clear statistic counter

With this telegram all statistic counters will be cleared.

Variable	type	value	
Msg.Rx	byte	7	receiver = FDL Interface
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	8	length of message (n = remote node count)
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	no error
Msg.Com	byte	0xFE (hex)	command = statistic counter
Msg.Seq	byte	0	standard handshaking
Msg.DeviceAdr	byte	0	-
-	byte	0	-
-	word	0	-
-	byte	0	-
Msg.DataCnt	byte	0	-
Msg.DataType	byte	5	data type = octet string
Msg.DataFnc	byte	4	clear counter values

Table 48. Clear statistic counter – Command

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	7	transmitter = FDL Interface
Msg.Len	byte	8	length of message (n = remote node count)
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0xFE (hex)	answer = statistic counter
Msg.Err	byte	0	error, status
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking
Msg.DeviceAdr	byte	0, …, 126	Remote node address
-	byte	0	-
-	word	0	-
-	byte	0	-
Msg.DataCnt	byte	1 - 40	Length = remote node count
Msg.DataType	byte	5	data type = octet string
Msg.DataFnc	byte	1	read

Table 49. Clear statistic counter – Answer telegram

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	7	transmitter = FDL Interface
Msg.Len	byte	0	length of message (n = remote node count)
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0xFE (hex)	answer = statistic counter
Msg.Err	byte	f	error code
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking

Table 50. Clear statistic counter – Error telegram

Error no. f	
0	no error
48	CON_TO time out
52	CON_NI area code unknown
53	CON_EA buffer length to big
55	CON_IP faulty parameter detected
57	CON_SI sequence error
59	CON_DI data incomplete or faulty

Table 51. Clear statistic counter – Error codes

6.4 Download Profibus-DP parameter

This is for enhanced users only. Knowledge of the Profibus standard is essential !

Instead of using the configuration tool for configuring the master, it is possible that the host is downloading the Profibus-DP parameter via the Mailbox in the dual port RAM. After downloading the parameter set, the device has to be initialised – set the Init flag in 'Control to Device'. The commands, which are necessary, are the following:

- DDLM_Download
- DDLM_Start_Seq
- DDLM_End_Seq

DDLM_Download is used to download the master and slave parameter into the device. If the length of the parameter to be set is to big, the telegrams DDLM_Start_Seq and DDLM_End_Seq are framing the necessary DDLM_Download telegrams.

e.g.:

1. DDLM_Start_Seq
2. DDLM_Download (1st)
3. …
4. DDLM_Download (last)
5. DDLM_End_Seq

The telegrams described here are derived from the corresponding Profibus-DP telegrams referred to in the DIN norm 19245 Part 3 chapter 10.8.1 "Coding of the Bus Parameter Set".

See also chapter 6.2 for Download Profibus-DP configuration

6.4.1 DDLM_download

Variable	type	value	
Msg.Rx	byte	3	receiver = User Interface
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	m+4	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	error, status
Msg.Com	byte	68	command = DDLM_Download
Msg.Seq	byte	0	standard handshaking
Msg.D(0)	byte	0	not used
Msg.D(1)	byte	0-125, 127	DP-remote node no. ID for Master Parameter
Msg.D(2)	word	0-760	Add_Offset
Msg.D(4-244)	m byte	0-255	Data

Table 52. DDLM_Download – Command telegram

- Msg.Len:
  Length of the Profibus-DP specific telegram plus four bytes overhead.
- Msg.Com
  DDLM_Download; refer to PROFIBUS DP norm
- Msg.D(1)
  0 - 125: Profibus-DP remote node to be configured,127:This master is to parameterised
- Msg.D(2)
  0-760: Offset for the telegram; This parameter is only used, if the overall telegram length does not fit within one telegram (see DDLM_Start_Seq, DDLM_End_Seq).
- Msg.D(4-244)
  all the other values are described in the previous section or are described in the general specification

variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	3	transmitter = User Interface
Msg.Len	byte	0	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	68	answer = DDLM_Download
Msg.Err	byte	0	no error
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking

Table 53. DDLM_Download – Answer telegram

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	3	transmitter = User Interface
Msg.Len	byte	0	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	68	answer = DDLM_Download
Msg.Err	byte	f	error code
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking

Table 54. DDLM_Download – Error telegram

Error no. f	
0	no error
48	CON_TO time out
52	CON_NI area code unknown
53	CON_EA buffer length to big
55	CON_IP faulty parameter detected
57	CON_SI sequence error
59	CON_DI data incomplete or faulty

Table 55. DDLM_Download – Error codes

6.4.2 DDLM_download – Master parameter:

Regarding the values in this section, refer to the description in DIN 19245-3.

Variable	type	value	
Msg.Rx	byte	3	receiver = User Interface
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	71	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	error, status
Msg.Com	byte	68	command, DDLM_Download
Msg.Seq	byte	0	standard handshaking
Msg.D(0)	byte	0	not used
Msg.D(1)	byte	127	ID for Master Parameter
Msg.D(2)	word	0	not used for master parameter
Msg.D(4)	word	67	Bus_Para_Len
Msg.D(6)	byte	1-125	FDL_Add; own node address
Msg.D(7)	byte	0-9	baud rate
Msg.D(8)	word	37-16383	TSL
Msg.D(10)	word	1-1023	min TSDR
Msg.D(12)	word	1-1023	max. TSDR
Msg.D(14)	byte	0-127	Tqui
Msg.D(15)	byte	1-255	Tset
Msg.D(16)	double word	255-65535	Ttr
Msg.D(20)	byte	1-255	G
Msg.D(21)	byte	1-126	Highest Station Address
Msg.D(22)	byte	0-7	Max_Retry_Limit
Msg.D(23)	byte	0-255	Bp_flag
Msg.D(24)	word	0-65535	Min_Slave_Intervall
Msg.D(26)	word	0-65535	Poll_Timeout
Msg.D(28)	word	0-65535	Data_Control_Time
Msg.D(30)	6 byte	0-255	reserved
Msg.D(36)	word	2	Master_User_Data_Len
Msg.D(38)	32 byte	ASCII String	Master_Class2_Name
Msg.D(70)	byte	0	Master_User_Data

Table 56. DDLM_download – Master Parameter

For answer message and error codes, refer to the previous section describing the DDLM_Parameter telegram.

6.4.3 DDLM_download – Slave parameter:

Variable	type	value	
Msg.Rx	byte	3	receiver = User Interface
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	0-240	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	error, status
Msg.Com	byte	68	command, DDLM_Download
Msg.Seq	byte	0	standard handshaking
Msg.D[0]	byte	0	not used
Msg.D[1]	byte	0-125	ID for Slave Parameter
Msg.D[2]	word	0	Add Offset: 0 no offset – first telegram n: offset if telegram is splitted
Msg.D[4]	word	0-590	Slave_Para_Len
Msg.D[6]	byte	0-255	Sl_Flag
Msg.D[7]	byte	0-255	Slave_Type
Msg.D[8]	12 byte	0	reserved
Msg.D[20]	word	2-136	Prm_Data_Len
Msg.D[…]	octet string		Prm_Data
Msg.D[…]	word	2-146	Cfg_Data_Len
Msg.D[…]	octet string		Cfg_Data
Msg.D[…]	word	2-248	Add_Tab_Len
Msg.D[…]	octet string		Add_Tab
Msg.D[…]	word	2-50	Slave_User_Data_Len
Msg.D[..]	octet string		Slave_User_Data

Table 57. DDLM_download – Slave parameter

For answer message and error codes, refer to the previous section describing the DDLM_Parameter telegram.

6.4.4 DDLM_Start_Seq:

In case that the Slave Parameter can not be downloaded with one telegram, it has to be subdivided into several telegrams. This has to be performed with a start sequence telegram, followed by the data telegrams and finally the end sequence telegrams.

Please observe the use of the offset word in the command telegram.

Variable	type	value	
Msg.Rx	byte	3	receiver = User Interface
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	4	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	error, status
Msg.Com	byte	67	command, DDLM_Start_Seq
Msg.Seq	byte	0	standard handshaking
Msg.D(0)	byte	0	not used
Msg.D(1)	byte	0-125, 127	remote node no. ID for Master Parameter
Msg.D(2)	word	0-65535	Time-out in (ms)

Table 58. DDLM_Start_Seq – Command message

variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	3	transmitter = User Interface
Msg.Len	byte	1	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	67	answer = DDLM_Start_Seq
Msg.Err	byte	0	no error
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking
Msg.D(0)	byte	240	Max_Len_Data_Unit

Table 59. DDLM_Start_Seq – Answer message

variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	3	transmitter = User Interface
Msg.Len	byte	0	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	67	answer = DDLM_Start_Seq
Msg.Err	byte	f	error code
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking

Table 60. DDLM_Start_Seq – Error message

Error no. f	
0	no error
52	CON_NI area code unknown

Table 61. DDLM_Start_Seq – Error code

6.4.5 DDLM_End_Seq:

Variable	type	value	
Msg.Rx	byte	3	receiver = User Interface
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	1	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	error, status
Msg.Com	byte	69	command, DDLM_End_Seq
Msg.Seq	byte	0	standard handshaking
Msg.D[0]	byte	0	not used

Table 62. DDLM_End_Seq – Command message

Variable	type	value	
Msg.Rx	byte	16	transmitter = Host
Msg.Tx	byte	3	receiver = User Interface
Msg.Len	byte	0	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	69	answer = DDLM_End_Seq
Msg.Err	byte	0	no error
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking

Table 63. DDLM_End_Seq – Answer message

Variable	type	value	
Msg.Rx	byte	16	transmitter = Host
Msg.Tx	byte	3	receiver = User Interface
Msg.Len	byte	0	length of message
Msg.Nr	byte	j	number of the message
Msg.Ans	byte	69	answer = DDLM_End_Seq
Msg.Err	byte	f	error, status
Msg.Com	byte	0	no command
Msg.Seq	byte	0	standard handshaking

Table 64. DDLM_End_Seq – Error message

Error no. f	
0	no error
59	CON_DI data incomplete or faulty

Table 65. DDLM_End_Seq – Error code

7. Configuring the Profibus-DP Master

There are two methods available to configure the Profibus-DP Master. The first and advisable solution is using the Profibus-DP configuration Software. The second possibility is to use the telegram channel in the dual port RAM interface.

7.1 Profibus-DP Configuration Software

This is the easiest way to configure the Profibus-DP Master. Based on the GSD files, the system can be set up within an easy to use graphical interface.

Please contact HMS for a detailed description of the available configuration software.

8. Firmware download

A firmware download can be performed either externally via the external configuration tool or internally via the Dual Port memory.

The internal firmware download function uses a telegram structure to transfer data blocks of firmware down to the microprocessor via the Dual Port Memory.

It will start with a telegram 'Start of Download' followed by telegrams 'Download Data' finalised with an 'End of Download' telegram. Before the firmware download is started, a reset should be issued in order to reach a defined state of the module. After the transmission of the last module and a following reset the module is up and running for the download of the configuration data.

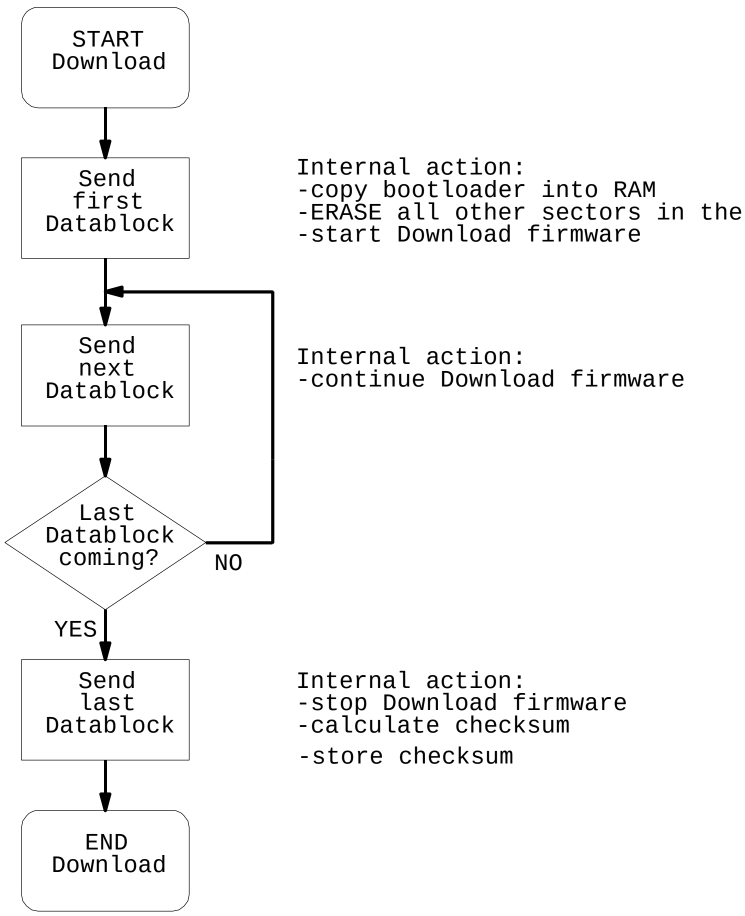


Figure 18. Firmware download sequence

Please note that for the successful download of the configuration file and firmware file, it is important to send the next telegram with an acknowledgement of the previous telegram. If this is not done, the download will be aborted.

8.1 General Message Structure

8.1.1 Command message

Variable	type	value	
Msg.Rx	byte	0	receiver = Operating System
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	1-240	length of message
Msg.Nr	byte	j	message no.
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	no error
Msg.Com	byte	6	command = Firmware Download
Msg.Seq	byte	4, 8, 12	4 = first data telegram 8 = following data telegram 12 = last data telegram
Msg.D[]	byte	-	user data

Table 66. Firmware download - Command message

8.1.2 Answer Message

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	0	transmitter = Operating System
Msg.Len	byte	0	length of message
Msg.Nr	byte	j	message no.
Msg.Ans	byte	6	command = Firmware Download
Msg.Err	byte	0	no error
Msg.Com	byte	0	no command
Msg.Seq	byte	0	-

Table 67. Download - Answer message

8.1.3 Error message

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	0	transmitter = Operating System
Msg.Len	byte	0	length of message
Msg.Nr	byte	j	message no.
Msg.Ans	byte	6	command = Firmware Download
Msg.Err	byte	1-255	error no.
Msg.Com	byte	6	no command
Msg.Seq	byte	0	-

Table 68. Download - Error message

Code	
0	no error
21	database segment not configured - check the segment name in the first message
22	error in Msg.Nr - the difference in the message number between two messages is not one
23	Try to transfer more data then specified in the first download message
24	error in sequence – check Msg.Seq
25	Checksum error
41	command unknown
42	command mode (d[0]) unknown
51	error during flash erase cycle
52	error during flash write cycle

Table 69.Download - Error code

8.2 Telegrams

With this command the Host will take a firmware file and store it in a few segments of the flash. The number of segments used to store the firmware is automatically calculated by the Device.

Note: Each telegram has to be written over the entire length. Do not rely on previously written data!

8.2.1 First data telegram

Variable	type	value	
Msg.Rx	byte	0	receiver = Operating System
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	65	length of message
Msg.Nr	byte	j	message no.
Msg.Ans	byte	0	no answer
Msg.Err	byte	0	no error
Msg.Com	byte	6	command = Firmware Download
Msg.Seq	byte	4	first telegram
Msg.D[0]	byte	6	Mode: firmware download
Msg.D[1, 2]	word	xx xx	Checksum of the firmware. Can be found in byte 0 and 1 of the firmware file
Msg.D[3..10]	String	RCSCODE	Null terminated name of the configuration file. Do not change
Msg.D[11..12]	Byte	0	reserved
Msg.D[13, 14]	word	xx xx	Length of the firmware file divided by 16. If there is a firmware file with a length of 256 kBytes =262144 bytes = 0x40000, this word has to be set to 0x4000 in the order low byte high byte. (INTEL mode). Length can be found in byte 2 to 5 of the firmware file
Msg.D[15]	Byte	6	Data type firmware.
Msg.D[16]	Byte	0	reserved
Msg.D[17-26]	String	RCSCODE	Fix String, do not change
Msg.D[27]	Byte	255	fixed, do not change
Msg.D[28..48]	Byte	0	reserved
Msg.D[49]	Byte	255	fixed, do not change
Msg.D[50..64]	Byte	0	reserved

Table 70.Firmware download -first data telegram

8.2.2 Continue telegram

In the first continue message the first 64 bytes have to be the same as in the first message transmitted in the bytes Msg.D[1-64]. The following bytes up to Msg.D[239] are copied out of the firmware file beginning at offset 64. This means that the first 'Continue Telegram' should start at offset 64 on the Firmware File.

Variable	type	value	
Msg.Rx	byte	0	receiver = Operating System
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	240	Length of the following data beginning at d[0]
Msg.Nr	byte	j	running number of the message is always incremented by one
Msg.Ans	byte	0	response
Msg.Err	byte	0	no error
Msg.Com	byte	6	command = Firmware Download
Msg.Seq	byte	8	continue message
Msg.D[0..63]	byte	xx	copy of first telegram Msg.D[1..64]
Msg. D[64..239]	byte	xx	data out of firmware file starting at offset 64

Table 71.Firmware download - First continue telegram

8.2.3 Other continue telegram

Variable	type	value	
Msg.Rx	byte	0	receiver = Operating System
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	240	Length of the following data beginning at d[0]
Msg.Nr	byte	j	running number of the message is always incremented by one
Msg.Ans	byte	0	response
Msg.Err	byte	0	no error
Msg.Com	byte	6	command = Firmware Download
Msg.Seq	byte	8	continue message
Msg.D[0..239]	byte	xx	Data.

Table 72. Firmware download - Other continue telegram

8.2.4 Last telegram

The last message is transmitted if the remaining length of the data in the firmware file is less or equal 240 bytes.

Variable	type	value	
Msg.Rx	byte	0	receiver = Operating System
Msg.Tx	byte	16	transmitter = Host
Msg.Len	byte	1-240	Length of the following data
Msg.Nr	byte	j	running number of the message is always incremented by one
Msg.Ans	byte	0	response
Msg.Err	byte	0	no error
Msg.Com	byte	6	command = Firmware Download
Msg.Seq	byte	12	last message
Msg.D[0..xx]	byte	xx	Data.

Table 73. Firmware download - Last telegram

8.2.5 Answer telegram

In that case you get a response message which contains data from the Operating System. For all continue messages and the last message you get a response message without any data.

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	0	transmitter = Operating System
Msg.Len	byte	4	4: first answer message
Msg.Nr	byte	j	message no. (same number as in the command message)
Msg.Ans	byte	6	response = Firmware Download
Msg.Err	byte	0	no error
Msg.Com	byte	0	no command
Msg.Seq	byte	0	-
Msg.D[0..3]	long	xx xx xx xx	Length (in bytes) of data from the firmware file the DEVICEexpect.

Table 74. Firmware download - First answer telegram

With the first response the device tells the application how much necessary data from the firmware file are to transmit. The application has to transmit the data from offset 64 in the firmware file up to the length the Operating System tells you in the first answer telegram.

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	0	transmitter = Operating System
Msg.Len	byte	0	0: following answer messages
Msg.Nr	byte	j	message no. (same number as in the command message)
Msg.Ans	byte	6	response = Firmware Download
Msg.Err	byte	0	no error
Msg.Com	byte	0	no command
Msg.Seq	byte	0	-

Table 75. Firmware download - Following answer telegrams

8.2.6 Error telegram

Variable	type	value	
Msg.Rx	byte	16	receiver = Host
Msg.Tx	byte	0	transmitter = Operating System
Msg.Len	byte	0	-
Msg.Nr	byte	j	message no. (same number as in the command message)
Msg.Ans	byte	6	response = Firmware Download
Msg.Err	byte	1 - 255	error number
Msg.Com	byte	0	no command
Msg.Seq	byte	0	-

Table 76. Firmware download - Error telegram

8.2.7 File Description

File offset	type	value	
0 .. 1	word	xx xx	checksum
2 .. 5	long	xx xx xx xx	file length
6 .. 63	byte	0xFF	reserved for future use
64 …	byte	xx	rest of firmware file

Table 77. Firmware download - File description

9. Appendix list

Appendix A. AnyBus Connector specification

Appendix B. Datasheets DPRAM CY7C136

Appendix C. Datasheets DPRAM CY7C136

Appendix D. Environmental specification

10. Tables

10.1 Table of Tables

10.2 Table of Figures