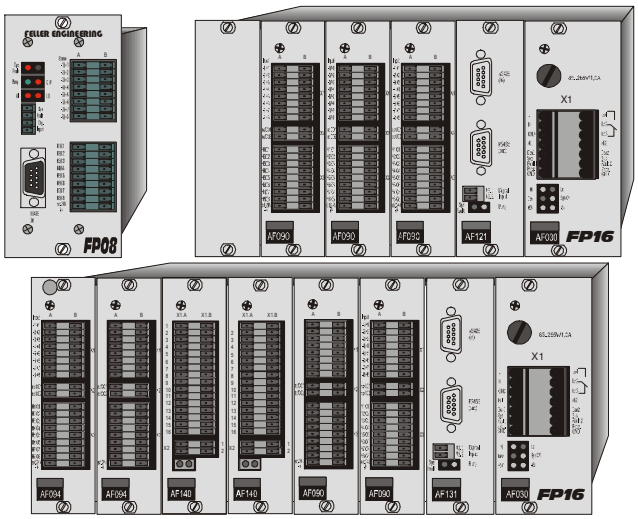
FELLER ENGINEERING GmbH

Profile Description

PROFIBUS - DP

for



FP08, FP16, FP16+

FELLER ENGINEERING GmbH
Carl-Zeiss-Straße 14
63322 Rödermark / Germany
www.fellereng.de

Phone: +49(6074)8949-0
Fax: +49(6074)8949-49
Technical-Hotline: +49(6074)8949-31
info@fellereng.de

Version 2.0

Stand: 07/13-1106

Contents

1 Version of Documentation 4

2 Temperature controller with Profibus 5

2.1 Technical details 5

2.1.1 GSD-file 5
2.1.2 Addressing 5
2.1.3 Bus-parameters 5
2.1.4 Pin assignment 5

3 Transfer of Utility Data 6

3.1 Safety of the consistence 6

3.2 Check after demand of data referring to required data in the input section 6

3.3 Attention to the format of the utility data 7

3.4 Transfer settings only with change 7

4 Definition of the input and output sections 8

4.1 The output section in the bus-master (sent from master to slave) 8

4.2 The input section in the bus-master (sent from slave to master) 9

4.2.1 Examples: 10

4.3 Zone status 11

4.4 Global values 12

5 History 13

1 Version of Documentation

This document is an update of the status 7/02. The technique of the Profibus link to the controllers of FELLER ENGINEERING did not change. The previous version is valid as before.

2 Temperature controller with Profibus

The series FP08, FP16 and FP16+ are designed for stand-alone temperature controllers. The default fitting is a serial interface RS485 for the direct connection of operator units or supervising systems. The Profibus interface is alternatively available in FP08 and additionally available for FP16 or FP16+.
The interface Profibus DP is designed according to DIN 19245 / pr EN 50170, unrestricted for all functions.

The interface Profibus DP:

- has a transmission rate of max. 12 Mbits/s.
- is electrical isolated.
- allows the transmission of all settings and parameters.
- is absolutely compatible to previous versions of the controllers.
- may be added to default FP16 or FP16+ by simple exchange of the processor board. No additional wiring is required except the Profibus cable.
- is wired to a compatible Sub-D plug according to DIN 19245.
- Was certified by the Organisation of Profibus Users (PNO).

2.1 Technical details

2.1.1 GSD-file

The required GSD-file “FEL_00A9.GSD” for various bus-masters is available from FELLER ENGINEERING. Even the file ”FE00A9TD.200“ for operation of different older SIEMENS-bus-masters is available. These files are prepared for download at www.fellereng.de.
The official PNO identification code is 00A9.

2.1.2 Addressing

The selection of the address has to be done exclusively by the DIP-switches on the controller (see manual). The SSA service of Profibus is not executed. The addresses are not able to be changed via bus. Valid addresses are 1..32.

2.1.3 Bus-parameters

The bus-parameters are fixed in the GSD-file. The data are to transfer via 10 datawords for input and 10 datawords for output. Manipulation of the length of data is not allowed. The controllers are limited to a max. baud rate of 12Mbit. This will be recognized automatically.

2.1.4 Pin assignment

The connection for Profibus has to be wired to the labelled. The assignment of the pins refers to the Profibus specification.

A - wire	= PIN 8
B - wire	= PIN 3
RTS	= PIN 4
GND	= PIN 5
+5V	= PIN 6

3 Transfer of Utility Data

All multi channel control systems are fit with several settings like setpoint, alarm limits and different control parameters for each single zone. There is further information about the actual condition of the zone (actual temperature, alarm reports, output rate) as well as global collective parameters for all zones.

The Profibus profile of Feller Engineering enables the handling of all these settings to present the controller as comfortable as possible.

It is not possible and makes no sense to transfer all these utility data together in a single telegram. That is why the bus-master has to demand all required data from the control system.

The data transfer with the controller happens by an input section and an output section, each with a size of 20 bytes.

Each section consists of a 4 bytes header and 16 bytes (8 words) utility data.

The bus-master demands the required data from the controller by the description of the output section. The controller will set these data into the input section of the bus-master.

That is the reason why the link to these controllers has more expense than to “smaller” partners like scales or valves, that may keep all data in a single section.

There are some important items the programmer has to concern about:

3.1 Safety of the consistence

During the first step of the program, before the output section will be created, the consistence-byte must be set to”0”. All transferred telegrams will be declared “invalid” this way.

After the complete description of the output section the consistence-byte has to be set, to declare this set of data “valid”.

This is required as many Profibus-master do not transfer the data synchronous with the utility program. So there will be transferred sets of data, which are not yet completed (as the user is just executing his program).

These failures in transmission appear seldom and accidentally and are very difficult to locate. So it is very important to keep the sequence.

3.2 Check after demand of data referring to required data in the input section

The input section is not fit with the required data just after demand, as the slave has to collect these before sending an answer. That is why the program of the master has to wait for the required data by checking byte 1 and byte 2 of the input section.

3.3 Attention to the format of the utility data

All utility data will be stored in integer format according to the “INTEL-FORMAT”. The LO-byte is always followed by the HI-byte.

Some Profibus-master (e.g. Siemens) use the “MOTOROLA-FORMAT” which sets the HI-byte before the LO-byte.
In this case the programmer has to change the bytes before transfer.

3.4 Transfer settings only with change

To reduce the load of processor and bus the settings should only be transferred with changes. There is no sense to transfer continuously the same identical data to the controller. The controller stores these values durable in its EEPROM independently of the power supply.

4 Definition of the input and output sections

4.1 The output section in the bus-master (sent from master to slave)

Byte No.	Name	Function (contents)	
0	*oAction*	**1** = read values from slave **2** = send values to slave	HEADER
1	*oGroup*	A group includes a series of 8 zones. **1** = zones 1..8 **2** = zones 9..16 **3** = zones 17..24 a.s.o. Special group: **0** = global settings (see below)	
2	*oSignification*	The number of the referring parameter will be set here. **0** = setpoint **1** = parameter 1 (referring to the manual of the controller) **2** = parameter 2 (referring to the manual of the controller) ... a.s.o. ... **252** = heater current (not for all controllers) **253** = output rate **254** = actual value **255** = zone status (description see below)	
3	*oConsistency*	According to the above description the Consistency-byte has to be set to **0** before any further change of the output section. Only after the setting all data of the output section, the Consistence-byte has to be set with the last action. The consistence-byte consists of 8 bit. These sign the validity of the following datawords 1..8. Each bit signs a valid dataword, if it is set to1. (bit0 ref. to dataword 1, bit7 ref. to dataword 8). This allows to enable the orders for single or more zones.	
4 5	*oDataword 1*	Value for the 1st zone of a group. With an order (Byte 0 = 1) the contents is not valid.	UTILITY-DATA
6 7	*oDataword 2*	Value for the 2nd zone of a group With an order (Byte 0 = 1) the contents is not valid.	
8 9	*oDataword 3*	Value for the 3rd zone of a group With an order (Byte 0 = 1) the contents is not valid.	
10 11	*oDataword 4*	Value for the 4th zone of a group With an order (Byte 0 = 1) the contents is not valid.	
12 13	*oDataword 5*	Value for the 5th zone of a group With an order (Byte 0 = 1) the contents is not valid.	
14 15	*oDataword 6*	Value for the 6th zone of a group With an order (Byte 0 = 1) the contents is not valid.	
16 17	*oDataword 7*	Value for the 7th zone of a group With an order (Byte 0 = 1) the contents is not valid.	
18 19	*oDataword 8*	Value for the 8th zone of a group With an order (Byte 0 = 1) the contents is not valid.	

4.2 The input section in the bus-master (sent from slave to master)

Byte No.	Name	Function (contents)	
0	*iAction*	**3** = values accepted by the slave **4** = slave reports exceeded range, one or more values are not set	HEADER
1	*igroup*	The slave sets the number of the group that was demanded by the master. Only when the group-number of the input and output section are identical, the received utility-data should be used.	
2	*iSignification*	The slave sets the signification as it was demanded in the output section of the master. Only when the signification of the input and output section are identical, the received utility-data should be used.	
3	*iConsistency*	Only when Bit0 of the consistence was set, the following utility-data shall be used. Bit 1 toggles in the transmission-cycle of the slave from 0 to 1.	
4 5	*iDataword 1*	Transferred value of the 1st zone within the group	UTILITY-DATA
6 7	*iDataword 2*	Transferred value of the 2nd zone within the group	
8 9	*iDataword 3*	Transferred value of the 3rd zone within the group	
10 11	*iDataword 4*	Transferred value of the 4th zone within the group	
12 13	*iDataword 5*	Transferred value of the 5th zone within the group	
14 15	*iDataword 6*	Transferred value of the 6th zone within the group	
16 17	*iDataword 7*	Transferred value of the 7th zone within the group	
18 19	*iDataword 8*	Transferred value of the 8th zone within the group	

4.2.1 Examples:

The master of the bus demands the actual values of the zones 9..16:

1. ***oConsistence*** set to 0
2. ***oAction*** set to 1 (to read out)
3. ***oGroup*** set to 2 (demand for zones 9..16)
4. ***oSignification*** set to 254 (demand actual values)
5. ***oConsistence*** set to 255 (all 8 bits = 1)
6. wait for ***iGroup*** = ***oGroup*** = 2
7. wait for ***iSignification*** = ***oSignification*** = 254
8. wait for setting bit0 of ***iConsistence***
9. Now the required values may be read from ***iDataword1*** .. ***iDataword8*** for zones 9..16.

The master of the bus sets the setpoint of zone 20 to 300°C. All other zones have to keep the values stable.

1. ***oConsistence*** set to 0
2. ***oAction*** set to 2 (to write)
3. ***oGroup*** set to 3 (zone 20 the 4th zone within group 3)
4. ***oSignification*** set to 0 (parameter 0=setpoint)
5. ***oDataword4*** set to 300 (byte 10= 44, byte 11 = 1. Attention: LO-byte first!)
6. ***oConsistence*** set to binary *00001000* = 8. This way only Dataword 4 is valid.
7. wait for ***iGroup*** = ***oGroup*** = 3
8. wait for ***iSignification*** = ***oSignification*** = 0
9. wait for setting bit0 of ***iConsistence***.
10. If ***iAction*** = 3, the value was accepted.
    If ***iAction***=4, the section has exceeded.
11. The new setpoint can be taken from ***iDataword4*** for check.

4.3 Zone status

The status of a zone is built bit-wise. 16 bit are transferred for each zone with the following signification:

BIT	Signification			
0	1=Zone ok 0=zone failed			
1	0=O.K. 1=LO-Alarm			
2	0=O.K. 1=HI-Alarm			
3	0=O.K. 1=broken sensor / override			
4	0=O.K. 1=shorted sensor			
5 6	0 zone off 0	1 manual mode 0	0 control mode 1	1 drop 1
7	0=O.K. 1=optimising failed			
8	1=optimising demanded			
9	0=O.K. 1= neg. deviation alarm			
10	0=O.K. 1= pos. deviation alarm			
11	0=O.K. 1=alarm after change of setpoint			
12	0=O.K. 1=heater current failure			
13 14 15	always 0 always 0 always 0			

The self-optimising of the zone will be enabled or disabled by setting bit no.8 of the status (sole accepted order). The descriptions referring to the process are to find in the operator manual.

4.4 Global values

Global values for the complete unit will be transferred instead of zone-specific data as soon as the master sets a "0" to the 2nd byte (group). The different parameters are to send in READONLY or READ/WRITE mode (see column R /RW). The 3rd byte (contents of telegram) indicates which global values are transferred.

Byte 2 „Group"	Byte 3 „Signific. contents"	Byte 5..20 „Values"	R / RW
0	0	Firmware-Ident-number (AZ-number)	R
		Firmware-version	R
		Firmware-Date (Day)	R
		Firmware-Date (Month)	R
		Firmware-Date (Year)	R
		Serial number	R
		Position of internal DIP-switch	R
		Number of zones in the controller	R
0	1	Heater current supervision (1=enabled, 0=disabled)	R
		Profile-version	R
		Spare	R
		Spare	R
		Spare	R
		Spare	R
		Spare	R
		Spare	R
0	2	Controller outputs (0=disable, 1=enable)	RW
		Alarm-Delay in seconds (0=no delay)	RW
		Net frequency (0=50cps, 1=60cps)	RW
		Max. temperature setpoint (HI-value)	RW
		Drop mode (0=normal, 1=drop)	RW
		Setpoint program (1 or 2)	RW
		Function of plc-input (from firmware V6.2)	RW
		Spare	RW
0	3	*Status of the outputs heating zone 1..8*	R
		Status of the outputs heating zone 9..16	R
		Status of the outputs heating zone 17..24	R
		Status of the outputs heating zone 25..32	R
		Status of the outputs cooling zone 1..8	R
		Status of the outputs cooling zone 9..16	R
		Status of the outputs cooling zone 17..24	R
		Status of the outputs cooling zone 25..32	R
0	4	0=no reaction 1=load default parameters *) (original settings) 2=Reset of the unit *) (controller warmstart)	W
		Spare	W
		Spare	W
		Spare	W
		Spare	W
		Spare	W
		Spare	W
		Spare	W

*) Different routines may be started in the controller by group 0, signification 4. The activation of these routines may result in transfer pause of several seconds.

5 History

Profile version 2.0
The function of consistence was changed. Each bit describes a zone. This way the setting of single zone is enabled.
The profile version is transferred by the 2nd byte of group 0, signification 1 (here =2).

Profile version 2.02
A toggle-bit was created.

Profile version 2.05
The status of the outputs for heating and cooling was changed for Profibus protocol.

Profile version 2.06
The standby mode of the FP16 has sent this value for setpoint.