Controlling several fans (ebm-papst) via Modbus RTU

SIMATIC S7-1200

https://support.industry.siemens.com/cs/ww/en/view/109476801

Siemens Industry Online Support

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG (“Siemens”). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens’ products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.

Siemens’ products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer’s exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: http://www.siemens.com/industrialsecurity.

Table of Contents

1 Task

1.1 Overview

Introduction

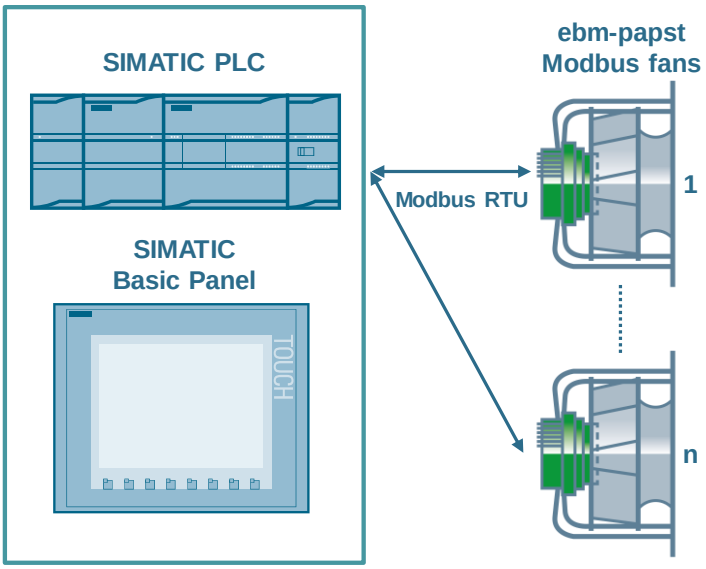
ebm-papst offers fans with an integrated RS485 interface that can be controlled via the Modbus RTU protocol.

These fans are to be connected to a SIMATIC PLC, monitored and controlled.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



Description of the automation task

The application has to satisfy the following requirements:

- Communication between the SIMATIC PLC and the ebm-papst fan (reading and writing the parameters) via Modbus RTU
- Operation and monitoring of the motor parameters via HMI

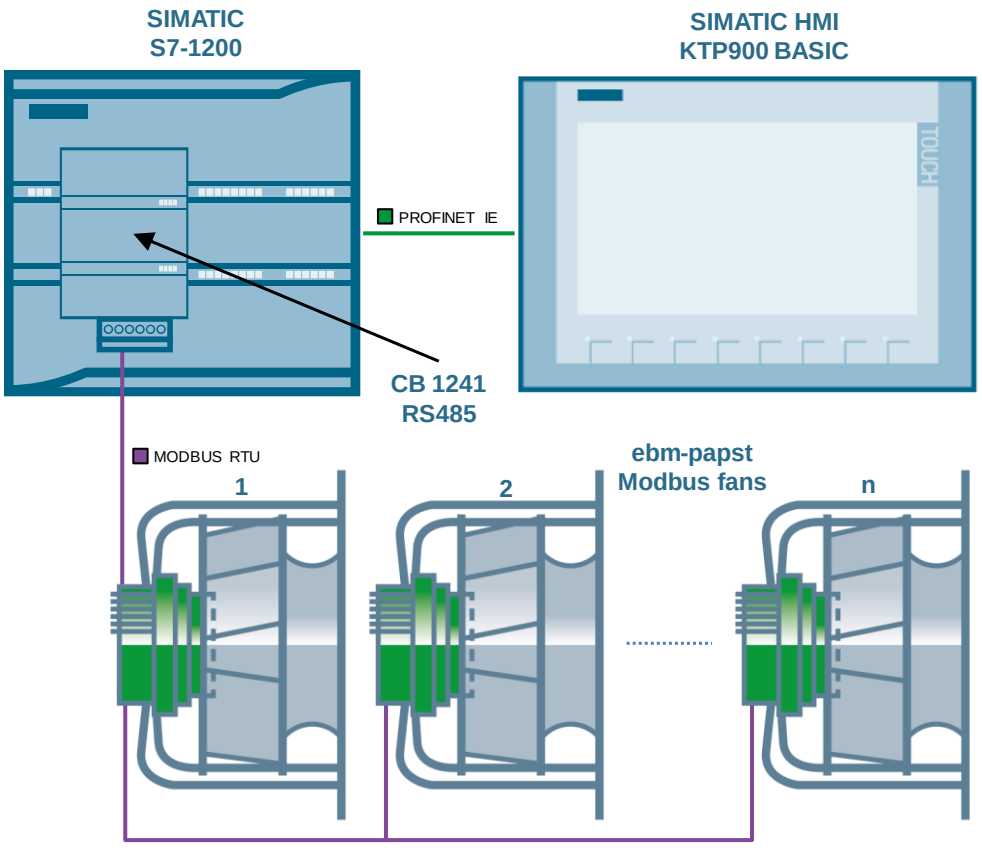
2 Solution

2.1 Overview

Schematic layout

The figure below shows a schematic overview of the most important components of the solution:

Figure 2-1



Configuration

A SIMATIC S7-1200 CPU with firmware V4.2.1 is used as the controller.
It exchanges the motor data with up to 32 ebm-papst fans sequentially using the serial communication board CB 1241 RS485 via the Modbus RTU data protocol.

To operate and monitor the motor data, a SIMATIC HMI KTP900 BASIC (or its simulation in the TIA Portal) is used.

Configuration is performed with STEP 7 V15.

Note Instead of the CB 1241 RS485, the CM 1241 RS422/485 with 9-pin D-sub connector (6ES7241-1CH32-0XB0) may be used alternatively (\3\).

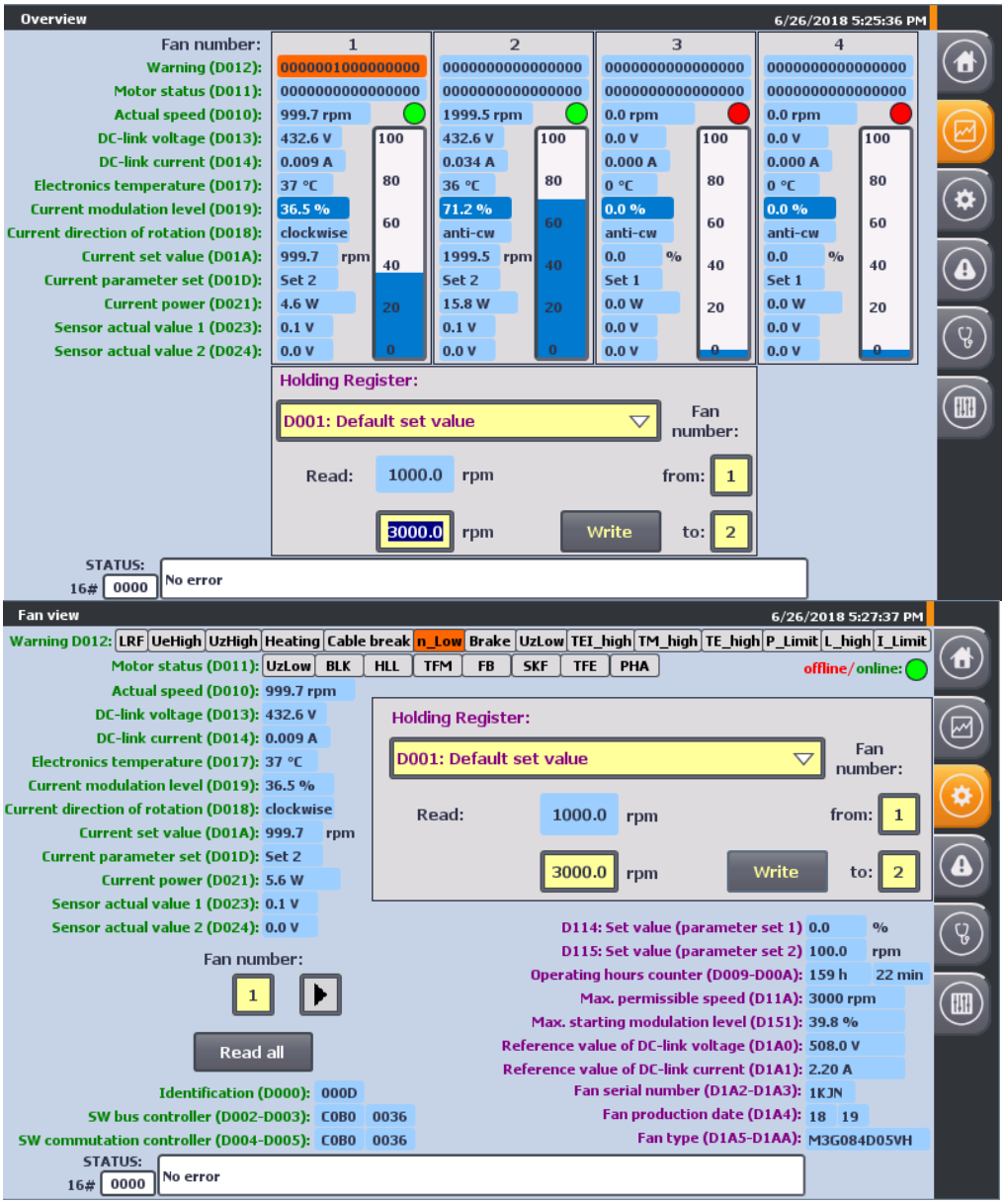
However, the CB 1241 RS485 is better suited for direct wiring of the fan (without D-sub connector).

2.2 Description of the core functionality

The application example's core functionality is the operation of the ebm-papst fan using the HMI.

Overview and description of the user interface

Figure 2-2



The application example can be operated using the following 2 displays:

- Fan overview with editing of the most important parameters
- Fan view with monitoring and editing the most common parameters

Chapter 5 gives a more detailed description of how to operate the user interface.

Advantages of this solution

In addition to the cyclic reading of the most important input registers of the ebm-papst fan, the application example allows you to edit the most common holding registers and to read fan-specific (non-writable) parameters upon request.

This application example offers the following advantages:

- Input and output of the parameters with the correct unit and meaning through the KTP900 Basic PN operator panel
- Sequential cyclic reading of the key input registers of all fans
- Overview of all (4) fans with bar graph display of the current modulation level
- Automatic retrieval of the most common holding registers after selection and subsequent editing of all fans
- Collective retrieval of device-specific input registers and non-writable holding registers of all fans upon request
- Access to the register of each fan via the control program
- Communication can be expanded to up to 32 fans

Topics not covered by this application

This application example establishes the sequential communication between the S7-1200 and the ebm-papst Modbus fans by means of Modbus RTU.

The selected registers are converted to the correct unit according to their meaning in line with the MODBUS parameter specification for “ebm-papst series 84 / 112 / 150 / 200” V5.00 and output.

To safely operate your ebm-papst fans, you have to familiarize yourself with this specification.

This specification does not provide a detailed description of the parameters.

Using this application example you can also access other registers from the control program (in addition to the ones selected in chapter 3.2). However, the conversion into the correct unit is not implemented for these registers.

Note The MODBUS parameter specification for “ebm-papst series 84 / 112 / 150 / 200” V5.00 is provided by ebm-papst when purchasing a fan/motor of this series and upon request (see chapter 6.3).

Assumed knowledge

Basic knowledge and experience with the following subjects are required for this application example:

- SIMATIC S7-1200 controller family
- TIA Portal engineering interface
- MODBUS-RTU communication protocol

2.3 Hardware and software components

2.3.1 Validity

This application example is valid for

- STEP 7 V15 or higher (TIA Portal)
- SIMATIC S7-1200 with firmware V4.2.1 or higher

2.3.2 Components used

The application example has been created with the following components:

Hardware and software components

Table 2-1

Component	Qty.	Article number	Note
S7-1200 PM1207 POWER SUPPLY	1	6EP1332-1SH71	
COMPACT SWITCH MODULE CSM 1277	1	6GK7277-1AA10-0AA0	
CPU 1214C, DC/DC/RELAY, 14DI/10DO/2AI	1	6ES7214-1HG40-0XB0	Firmware V4.2.1 (\6\)
COMMUNICATION BOARD CB 1241, RS485	1	6ES7241-1CH30-1XB0	Alternatively, you can use the COMMUNICATION MODULE CM 1241, RS422/485 with 9-PIN D-SUB CONNECTOR (6ES7241-1CH32-0XB0) (\3\)
SIMATIC HMI KTP900 BASIC	1	6AV2123-2JB03-0AX0	Optional (can also be simulated in WinCC Basic V15)
Ethernet cable TP CORD RJ45/RJ45 2M	3	6XV1870-3QH20	
Standard sectional rail	1	6ES5 710-8MA11	35 mm
Ebm-papst Modbus fan	2	Type: VarioDrive C M3G084-DF05-VH	Manufacturer: ebm-papst (\10\ ad \11\)
SIMATIC STEP 7 BASIC V15	1	6ES7822-0AA05-0YA5	• Includes WINCC Basic V15; • With update 2 (\5\)
SIMATIC STEP 7 BASIC UPGRADE V15	1	6ES7822-0AA05-0YE5	(As an option) Only for upgrade from STEP 7 Basic V11, V12, V13 or V14

Example files and projects

The following list includes all files and projects that are used in this example.

Table 2-2

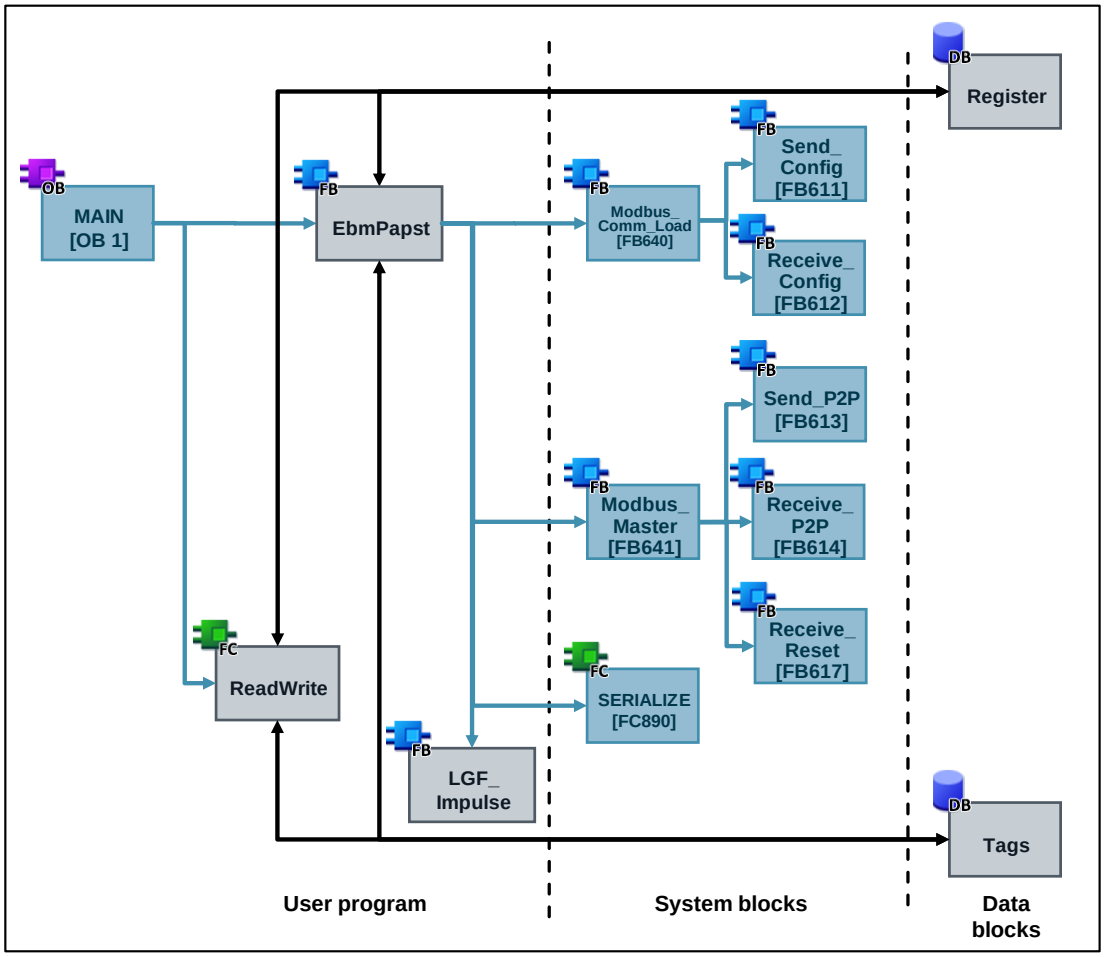
Component	Note
109476801_S7-1200_ebmpapst_TiaV15_PROJ_V20.zip	This zip file contains the STEP 7 project.
109476801_S7-1200_ebmpapst_DOC_V20_en.pdf	This document.

3 Mode of Operation

3.1 General overview

Figure 3-1 shows the chronological sequence of the block calls in the control part of the TIA Portal project.

Figure 3-1



The core functionality includes the function block “EbmPapst”.
This block is called in the cyclic organization block “Main”.
The interfaces of FB “EbmPapst” are fed from the data blocks

- “Register” (this DB contains the register structure and the order of the data to be read) and
- “Tags” (this DB contains the tags to parameterize and operator the FB “EbmPapst”).

The following system blocks are called in the FB “EbmPapst”:

- “Modbus_Comm_Load” [FB640] to initialize the Modbus RTU communication
- “Modbus_Master” [FB641] for data exchange of the S7-1200 as the master with the ebm-papst fans as the slave

- “SERIALIZE” [FB890] to convert words into an array of bytes and subsequent output as a string

The FB "LGF_Impulse" is from the general function library (\4\) and updates the register entries of the fan to be displayed in the “fan view" HMI display in the specified interval.

The actual values of the function blocks are stored as a multi-instance in the instance data block of the FB “EbmPapst”.

The FC “ReadWrite” serves to access additional registers which are not listed in the following chapter. It is called in the cyclic organization block “Main”. It has an interface to the FB “EbmPapst” and is fed by the data blocks “Register” and “Tags”.

3.2 Tab

3.2.1 Holding register

This application example allows you to access the following holding registers through the HMI:

Table 3-1

Address		Length	Designation	Write	Reset required	Password required
hex.	dec.					
D000	53248	1	Reset	X	-	-
D001	53249	1	Default set value	X	-	-
D002	53250	3	Password	X	-	-
D005	53253	1	Factory default for control	X	-	X
D009	53257	1	Operating hours counter	-	-	-
D00A	53258	1	Operating minutes counter	-	-	-
D100	53504	1	Fan address	X	X	-
D101	53505	1	Set value source	X	X	-
D102	53506	1	Preferred running direction	X	-	-
D103	53507	1	Save set value	X	X	-
D104	53508	1	Source parameter set	X	X	-
D105	53509	1	Internal parameter set	X	-	-
D106	53510	1	Control mode (parameter set 1)	X	X	-
D107	53511	1	Control mode (parameter set 2)	X	X	-
D112	53522	1	Motor stop enable (parameter set 1)	X	X	-
D113	53523	1	Motor stop enable (parameter set 2)	X	X	-
D114	53524	1	Set value (parameter 1)	-	-	-
D115	53525	1	Set value (parameter 2)	-	-	-
D119	53529	1	Maximum speed	X	X	X
D11A	53530	1	Max. permissible speed	-	-	-
D11F	53535	1	Ramp-up time	X	X	-
D120	53536	1	Ramp-down time	X	X	-
D145	53573	1	Ceiling speed for speed monitoring	X	X	X
D148	53576	1	Source running direction	X	X	-
D149	53577	1	Transfer rate	X	X	X
D14A	53578	1	Parity configuration	X	X	X

Address		Length	Designation	Write	Reset required	Password required
hex.	dec.					
D150	53584	1	Shedding function	X	X	X
D151	53585	1	Max. starting modulation level	-	-	-
D152	53586	1	Number of start attempts	X	X	X
D153	53587	1	Relay drop-out delay	X	X	-
D15B	53595	1	Emergency operation running direction	X	X	X
D15C	53596	1	Emergency operation function on/off	X	X	X
D15D	53597	1	Set value for emergency operation	X	X	X
D15E	53598	1	Emergency operation time delay	X	X	X
D1A0	53664	1	Reference value of DC-link voltage	-	-	-
D1A1	53665	1	Reference value of DC-link current	-	-	-
D1A2	53666	2	Fan serial number	-	-	-
D1A4	53668	1	Fan production date	-	-	-
D1A5	53669	6	Fan type	-	-	-

3.2.2 Input register

This application example allows you to read the following input registers through the HMI:

Table 3-2

Address		Designation
hex.	dec.	
D000	53248	Identification
D002	53250	Software name bus controller
D003	53251	Software version bus controller
D004	53252	Software name commutation controller
D005	53253	Software version commutation controller
D010	53264	Actual speed
D011	53265	Motor status
D012	53266	Warning
D013	53267	Link voltage
D014	53268	Link current
D017	53271	Electronics temperature
D018	53272	Current direction of rotation
D019	53273	Current modulation level
D01A	53274	Current set value
D01D	53277	Current parameter set
D021	53281	Current power
D023	53283	Sensor actual value 1
D024	53284	Sensor actual value 2

3.3 PLC data types

3.3.1 "typeCyclic"

The PLC data type “typeCyclic” includes the order of the data to be exchanged.

Table 3-3

Name	Data type	Description
last	Int	Last field element to be edited
data	Array[0..9] of "typeData"	Order of the data to exchange

3.3.2 "typeData"

The PLC data type “typeData” includes the parameters for reading and writing to the registers.

Table 3-4

Name	Data type	Description
mbAddr	UInt	Modbus address of the slaves to be addressed (1…247)
mode	USInt	Mode selection: Defines the type of request [read (0 for holding registers; 104 for input registers) or write(1)]
addr	UDInt	Start address in the slave: Defines the start address of the data to access: • 453249 to 453259 (16#D000 to 16#D00A) and 453505 to 453675 (16#D100 to 16#D1AA) for reading (mode = 0) and writing holding registers (mode = 1) • 53248 to 53286 (16#D000 to 16#D026) for reading input registers (mode = 104)
len	UInt	Data length: Defines the number of words which this request shall access (a maximum of 9 words is possible due to the slave)

3.3.3 "typeFan"

The PLC data type "typeFan" contains the data of the respective fan.

Table 3-5

Name	Data type	Description
mbAddr	UInt	Modbus address of the slaves to be addressed (1…247)
errorStatus	Word	Status evaluation (error code)
holding0	Array[53248..53258] of "typeRegister"	Holding register: D000 (reset) to D00A (operating minutes counter)
holding1	Array[53504..53674] of "typeRegister"	Holding register: D100 (fan address) to D1AA (fan type)
input	Array[53248..53286] of "typeRegister"	Input register: D000 (identification) to D026 (reserved)
productionData	"typeProductionData"	Fan production data

3.3.4 "typeHMI"

The PLC data type “typeHMI” contains information about the visibility display in the HMI.

Table 3-6

Name	Data type	Description
unit	"typeUnit"	Includes information about the correct representation of the setpoint unit (“1/min” or “%”)
buttonInvisible	Bool	Hides the write and “read all” buttons
readBusy	Bool	Is TRUE as long as a holding register is read
writeBusy	Bool	Is TRUE for as long as the “Write holding register” job is executed
readAllBusy	Bool	Is TRUE for as long as the “Read all registers” job is executed
readAllProgress	Real	Progress of the “Read all registers” job in percent
readFan	USInt	Fan data to be displayed in the “Fan view” display
readFanNotIncreasable	Bool	If TRUE, the button for increasing “readFan” is hidden
errorFan	USInt	Variables for the display of the faulty fan
multiplex	Bool	Request for copying the data record from "readFan" to the display data in the “Fan view” display

3.3.5 "typeManual"

The PLC data type “typeManual” contains the information for exchanging data between the FB “EbmPapst” and the FC “ReadWrite”.

Table 3-7

Name	Data type	Description
step	USInt	Step counter for FC “ReadWrite”
req	Bool	Request of manual access via FC “ReadWrite”
fanNumber	USInt	Fan to be addressed
data	"typeData"	Contains the information which data to read
dataPtr	Array[0..8] of UInt	Data buffer with the values to read or write
busy	Bool	FC “ReadWrite” is being edited
done	Bool	FC “ReadWrite” was completed without errors (active for one cycle)
error	Bool	FC “ReadWrite” was completed with errors (active for one cycle)
errorStatus	Word	Status evaluation (error code) of FC “ReadWrite”

3.3.6 "typeProductionData"

The PLC data type “typeProductionData” includes the fan's manufacturing data.

Table 3-8

Name	Data type	Description
serialNumber	String[4]	Fan serial number (consecutive number)
productionDate	"typeProductionDate"	PLC data type with the fan's production date (year/week)
fanType	String[12]	Fan type (designation)

3.3.7 "typeRead"

The PLC data type “typeRead” includes the placeholder to read registers.

Table 3-9

Name	Data type	Description
req	Bool	Request to read a register
data	"typeData"	PLC data type with the read parameters
uint	UInt	Read data value
real	Real	Calculated floating point value

3.3.8 "typeRegister"

The PLC data type "typeRegister" contains the read and calculated value of the respective register as well as the calculation instruction.

Table 3-10

Name	Data type	Description
uint	UInt	Read data value
calc	USInt	Algorithm
real	Real	Calculated floating point value

3.3.9 "typeWrite"

The PLC data type “typeWrite” includes the placeholder to read registers.

Table 3-11

Name	Data type	Description
req	Bool	Request to write to registers
fanRange	"typeFanRange"	Contains the number of the first and last fan to be written (the number of the first fan to be written is at the same time the number of the fan which is read)
data	"typeData"	PLC data type with the write parameters
uint	Array[0..2] of UInt	Data values to write in the “UInt” data format
minUint	UInt	Lower input limit value for the HMI input field for the “UInt” data format
maxUint	UInt	Upper input limit value for the HMI input field for the “UInt” data format
real	Real	Data value to write in the “Real” data format
maxReal	Real	Upper input limit value for the HMI input field for the “Real” data format

3.4 FB “EbmPapst”

This function block was especially developed for serial communication with several fans from ebm-papst via the Modbus RTU data protocol (based on the MODBUS parameter specification for “ebm-papst series 84 / 112 / 150 / 200” V5.00).
It is called in the cyclic organization block “Main”.

Figure 3-2: FB “EbmPapst”

Table 3-12: Parameter of the FB “EbmPapst”

Name	P type	Data type	Comment
initialCall	IN	Bool	First cycle after restarting the CPU (parameter “Initial_Call” from the calling OB1)
port	IN	PORT	Hardware ID of the RS485 CB or CM used
baud	IN	UDInt	Baud rate (default: 19200 bit/s)
parity	IN	UInt	Parity (default: 2 = even)
readAll	IN	Bool	Request to read all predefined registers of all fans
errorAck	IN	Bool	Acknowledgment of the pending error message (“status”)
reset	IN	Bool	Reset, restart block
errorStatus	OUT	Word	Status evaluation (error code)

Name	P type	Data type	Comment
lastFan	IN_OUT	USInt	Last fan to be addressed (is limited by the FB: 1 ≤ lastFan ≤ FAN_MAX)
read	IN_OUT	"typeRead"	PLC data type to read a register
write	IN_OUT	"typeWrite"	PLC data type to write holding registers
cyclic	IN_OUT	"typeCyclic"	PLC data type with the order of the data to be exchanged cyclically
acyclic	IN_OUT	"typeCyclic"	PLC data type with the order of the data to be exchanged acyclically
manual	IN_OUT	"typeManual"	PLC data type to exchange data with the FC “ReadWrite”
hmi	IN_OUT	"typeHMI"	PLC data type with display information for the HMI
fan	IN_OUT	Array[0.."FAN_MAX"] of "typeFan"	Array with the data of the individual fans (1 to user constant "FAN_MAX"; 0 = multiplex placeholder for the HMI display “Fan view")

The FB “EbmPapst” includes the calls of the system blocks “Modbus_Comm_Load” and “Modbus_Master”.

The “Modbus_Master” only allows blocks to be handed to the parameter “DATA_PTR” in which the attribute property “optimized block access” is not enabled. No block of this application example (FB or DB) has the property “optimized block access”. This prevents cycle time increases due to recopying optimized DBs in non-optimized DBs.

Note For a more detailed description of the communication instructions “Modbus_Comm_Load” and “Modbus_Master”, see the STEP 7 V15 online help or the chapter "Modbus RTU" in the S7-1200 system manual (\8\).

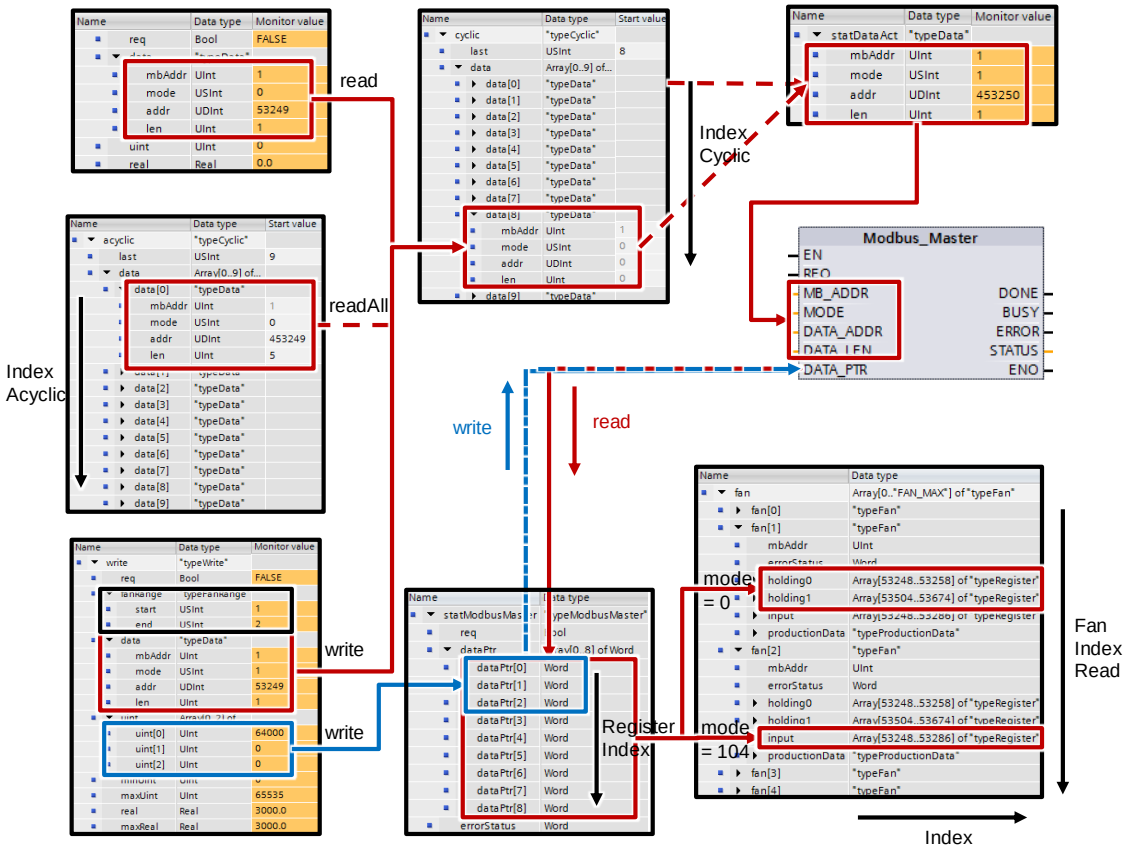
The FB “EbmPapst” reads the most important input registers of the fans in cyclic intervals of approximately 400 ms per fan. Additionally, you can retrieve and write to predefined holding registers of a fan with the configured HMI and to write a command to one or several fans (see chapter 3.2).

The FC “ReadWrite” enables you to access further registers in the control program part (see chapter 3.5).

Program workflow

Figure 3-3 shows the schematic structure of the FB “EbmPapst”.

Figure 3-3



The sequence for reading the input registers of each fan is processed cyclically as a function of the running index “IndexCyclic”. The information (access mode “mode”, register address “addr” and register length “len”) are provided as start values in the interface parameter “cyclic” in the field 0 to 7.

The following is written into the last field (“cyclic.last”) via the HMI depending on the request (“read”, “readAll” or “write”):

- the holding register to read (provided in the interface parameter “read”)
- the sequence for reading the acyclic registers as a function of the running index “IndexAcyclic” (provided in the interface parameter “acyclic”)
- the holding register to write to (provided in the interface parameter “write”)

The field (“cyclic.last”) is only processed if the data length “len” is greater than zero.

Based on the running index “IndexCyclic”, the information is transferred to the static placeholder “statDataAct” and handed to the communication block “Modbus_Master”.

The system block “Modbus_Master” provides the interface for exchanging data with the ebm-papst fans via Modbus RTU.
The 9-word field “dataPtr” in the static parameter “statModbusMaster” functions as a data buffer for the read values/the values to write. It is limited to 9 words due to the permitted maximum telegram length of the fan.

With read access, the read data are successively (“RegisterIndex”) written into the corresponding register file (“Index”) of the addressed fan depending on the access mode:

- Interface parameter “holding” for the holding registers (“mode” = 0)
- Interface parameter “input” for the input registers (“mode” = 104)

If necessary, the read value (“uint”) is subsequently converted into the corresponding floating point value (“real”) depending on the calculation instruction (“calc”).

With write access (“write.req”), the entered floating point value (“real”) is converted into the corresponding data value in the format “UInt” if necessary and subsequently written into the data buffer “dataPtr” of the Modbus_Master and transmitted to the fans ("write.fanRange.start" bis "write.fanRange.end") based on the calculation instruction (“calc”).

As the only exception, the holding register “Password” to write to has a length of 3 words (see Table 3-1). Here, the first and additionally the two following field elements are transmitted (“dataPtr [0..2] = uint [0..2]”).

3.5 FC “ReadWrite”

This function gives individual access to all registers of an ebm-papst fan. To operate the block, you have to be familiar with the MODBUS parameter specification for “ebm-papst series 84 / 112 / 150 / 200” V5.00. The FC “ReadWrite is only functional in connection with the FB “EbmPapst”.
It is called in the cyclic organization block “Main”.

Figure 3-4: FC “ReadWrite”

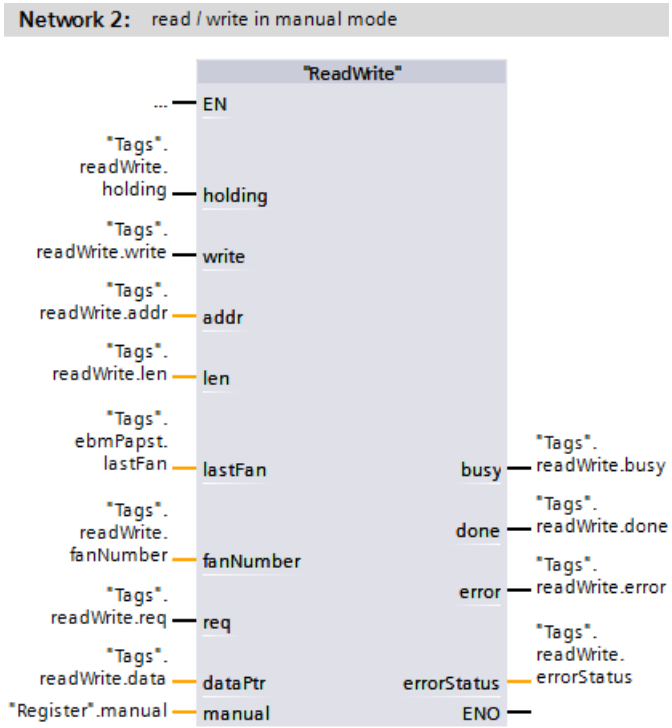


Table 3-13: Parameters of FC “ReadWrite”

Name	P type	Data type	Comment
holding	IN	Bool	Register type (false = input register; true = holding register)
write	IN	Bool	Only relevant for holding registers (false = read; true = write)
addr	IN	UInt	Start address in the slave: Defines the start address of the data to access: • 53248 through 53286 (16#D000 through 16#D026) for input registers (holding = false) • 53248 to 53258 (16#D000 to 16#D00A) / 53504 to 53674 (16#D100 to 16#D1AA) for holding registers (holding = true)
len	IN	USInt	Data length: Defines the number of words which this request shall access (a maximum of 9 words is possible due to the slave)
lastFan	IN	USInt	Last fan to be addressed (same parameter as FB “EbmPapst”); serves for limiting the “fanNumber”
busy	OUT	Bool	FC “ReadWrite” is being edited
done	OUT	Bool	FC “ReadWrite” was completed without errors (active for one cycle)

Name	P type	Data type	Comment
error	OUT	Bool	FC “ReadWrite” was completed with errors (active for one cycle)
errorStatus	OUT	Word	Status evaluation (error code) of FC “ReadWrite”
fanNumber	IN_OUT	USInt	Fan to be addressed (is limited by the FC: 1 ≤ fanNumber ≤ lastFan)
req	IN_OUT	Bool	Request of manual access via FC “ReadWrite” (is reset after completion)
dataPtr	IN_OUT	Array[0..8] of UInt	Data buffer with the values to read or write
manual	IN_OUT	"typeManual"	PLC data type to exchange data with the FB “EbmPapst”

The FC “ReadWrite” was made to resemble the instruction “Modbus_Master” and places the manual register access to the last position of the data transfer of FB “EbmPapst” that is to be processed cyclically.

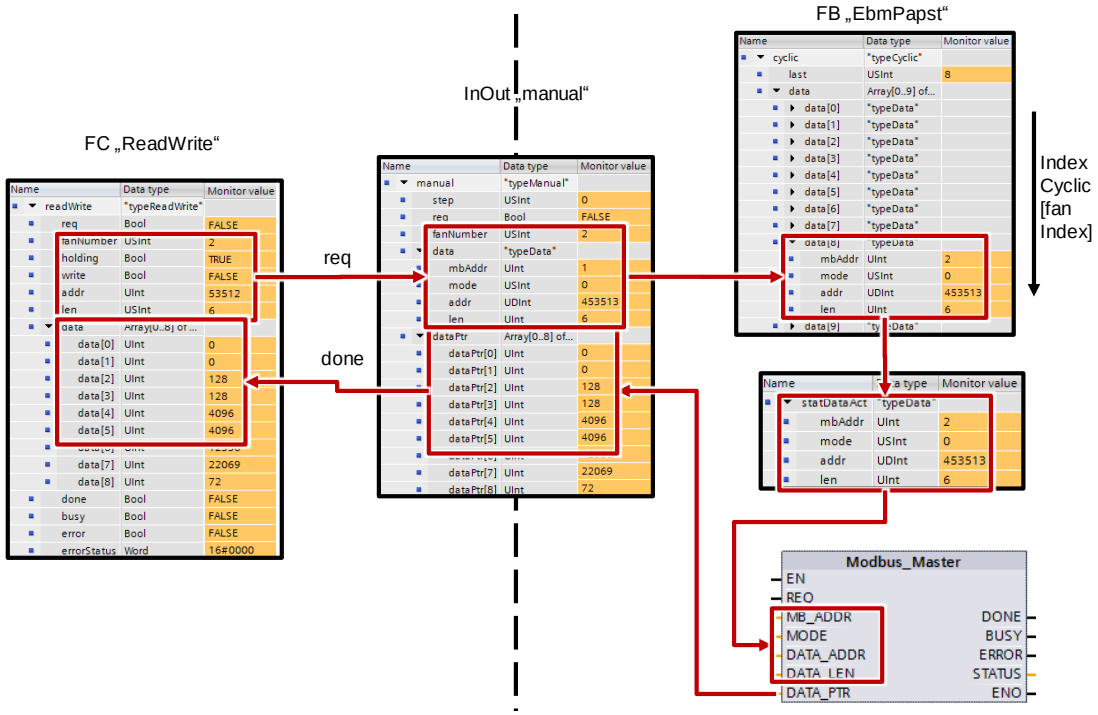
Execution of the FC “ReadWrite” may hence be delayed by max. 400 ms * number of fans to be addressed (“lastFan”).

Note For a more detailed description of the communication instructions “Modbus_Comm_Load” and “Modbus_Master”, see the STEP 7 V15 online help or the chapter "Modbus RTU" in the S7-1200 system manual (\8\).

3.5.1 Mode of operation

The function mechanism is demonstrated at the example of “reading the 6 holding registers starting from address 16#D108 of fan no. 2” in Figure 3-5.

Figure 3-5



The task is transmitted to the FC “ReadWrite” via the following input parameters:

- "fanNumber" = 2
- “holding” = “TRUE” (register type: holding register)

- “write” = “FALSE” (access type: read)
- “addr” = 53512 (corresponds to 16#D108)
- “len” = 6 (6 words are to be read)

The FC “ReadWrite” converts this information for the request “req” into the parameter in the PLC data type “typeData” which the instruction “Modbus_Master” understands:

- "mbAddr" = 2 (is read from "Register".fan[2].mbAddr)
- “mode” = 0 (corresponds to: “read holding register”)
- “addr” = 453513 (= holding register offset 400001 + x)
- “len” = 6 (6 words are to be read)

This data type is transferred “manually” to the FB “EbmPapst” by means of the input/output data and written into the last field (“cyclic.last”) of the data transfer jobs (“cyclic”) to be processed cyclically.

If the conditions “IndexCyclic” and “cyclic.last” are fulfilled, the information is transmitted to the static placeholder “statDataAct” and handed to the communication block “Modbus_Master”.

The system block “Modbus_Master” receives the read data from the selected fan and writes it into the data buffer “dataPtr” in the input/output data “manual” and transmits it to the FC “ReadWrite” in case of successful transmission “done”.

The output “done” is only active for one cycle and resets the request bit “req”.

Only the data to be transmitted is valid respectively (defined by the data length “len”). Before the FC “ReadWrite” is called again, the data has to be processed or it will be overwritten.

The read registers (“holding” or “input”) are also written to the correct field in the DB “Register”.

Besides the example “Reading holding registers” already described you can also read input registers or write to holding registers.

If errors occur, the output bit “error” is set for one cycle.
The status “errorStatus” delivers the corresponding error code (remains pending).

Note For a detailed list of the error codes for the “Modbus_Master” instruction, refer to the online help of the TIA Portal using F1.

You can check the function of the FC “ReadWrite” using the watch table “ReadWrite”:

Figure 3-6

	i	Name	...	Display format	Monitor value	Modify value	
1		"Tags".readWrite.req	...	Bool	FALSE	TRUE	☑ ⚠
2		"Tags".readWrite.fanNumber	...	DEC	2	2	☑ ⚠
3		"Tags".readWrite.holding	...	Bool	TRUE	TRUE	☑ ⚠
4		"Tags".readWrite.write	...	Bool	FALSE		☐
5		"Tags".readWrite.addr	...	Hex	16#D108	16#D108	☑ ⚠
6		"Tags".readWrite.len	...	DEC	6	6	☑ ⚠
7	// dataPtr						
8		"Tags".readWrite.data[0]	...	DEC	0		☐
9		"Tags".readWrite.data[1]	...	DEC	0		☐
10		"Tags".readWrite.data[2]	...	DEC	128		☐
11		"Tags".readWrite.data[3]	...	DEC	128		☐
12		"Tags".readWrite.data[4]	...	DEC	4096		☐
13		"Tags".readWrite.data[5]	...	DEC	4096		☐
14		"Tags".readWrite.data[6]	...	DEC	12356		☐
15		"Tags".readWrite.data[7]	...	DEC	22069		☐
16		"Tags".readWrite.data[8]	...	DEC	72		☐
17	// output						
18		"Tags".readWrite.busy	...	Bool	FALSE		☐
19		"Tags".readWrite.done	...	Bool	FALSE		☐
20		"Tags".readWrite.error	...	Bool	FALSE		☐
21		"Tags".readWrite.errorStatus	...	Hex	16#0000		☐

3.5.2 Sequential call of FC “ReadWrite”

To execute several register accesses successively, the parameters have to be handed to the FC “ReadWrite” in the form of a sequencer.

In the OB “Main” this is presented in networks 4 through 7.

Task

The following actions are to be performed successively:

1. Reading input register 16#D015 of fan 1
2. Reading holding registers 16#D108 through 16#D10D of fan 2 (see chapter 3.5.1)
3. Writing the value 1 to holding register 16#D105 of fan 1 (“internal parameter set”) (corresponds to: parameter set 2)

Figure 3-8 shows the program sequence.

The sequencer is started in network 4. The step is displayed in white font. The jobs are processed in the networks 5 through 7. A job consists of the parameter handover and the subsequent action in case of the ready message (“done”). Subsequently, the program continues with the next step.

The steps are called in the reverse order of the networks.
This is necessary to prevent “sliding through” the steps.

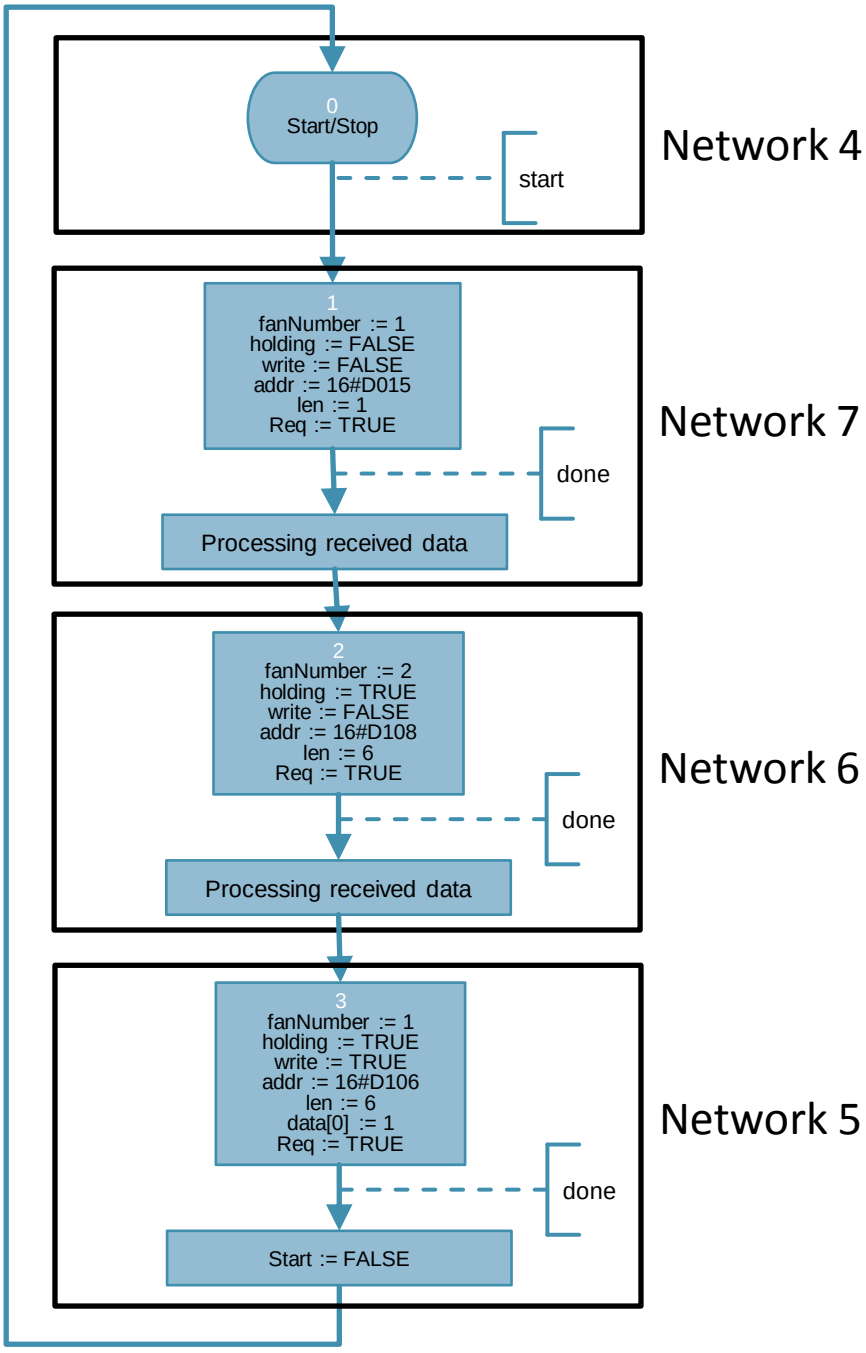
You can check the function of the sequencer using the watch table “ReadWrite”:

Figure 3-7

	i	Name	...	Display format	Monitor value	Modify value	
22		// "Register"					
23		"Tags".start	...	Bool	FALSE	TRUE	
24		"Tags".readInput	...	DEC	31		
25		"Register".fan[1].input[53269].uint	...	DEC	31		
26		"Tags".readholding[0]	...	DEC	0		
27		"Tags".readholding[1]	...	DEC	0		
28		"Tags".readholding[2]	...	DEC	128		
29		"Tags".readholding[3]	...	DEC	128		
30		"Tags".readholding[4]	...	DEC	4096		
31		"Tags".readholding[5]	...	DEC	4096		
32		"Register".fan[2].holding1[53512].uint	...	DEC	0		
33		"Register".fan[2].holding1[53513].uint	...	DEC	0		
34		"Register".fan[2].holding1[53514].uint	...	DEC	128		
35		"Register".fan[2].holding1[53515].uint	...	DEC	128		
36		"Register".fan[2].holding1[53516].uint	...	DEC	4096		
37		"Register".fan[2].holding1[53517].uint	...	DEC	4096		
38		"Register".fan[1].input[53277].uint	...	DEC	1		

Successful writing to holding register 16#D105 is accomplished by reading the register again or using Cyclically retrieved values (input registers): Current parameter set (D01D): Set 2 or Figure 3-7 in line 38.

Figure 3-8



4 Installation and Commissioning

The application example was set up using a CPU 1214C and a 1241 RS485 communication board. If you are using a different CPU, please follow the instructions in chapter 4.3.1.

Instead of the CB 1241 RS485 communication board you can also use the CM 1241 RS422/485 communication module to communicate with the ebm-papst fan (here: "VarioDrive C") (\3\).

4.1 Installing the hardware

The figure below shows the hardware setup of the application.

Figure 4-1

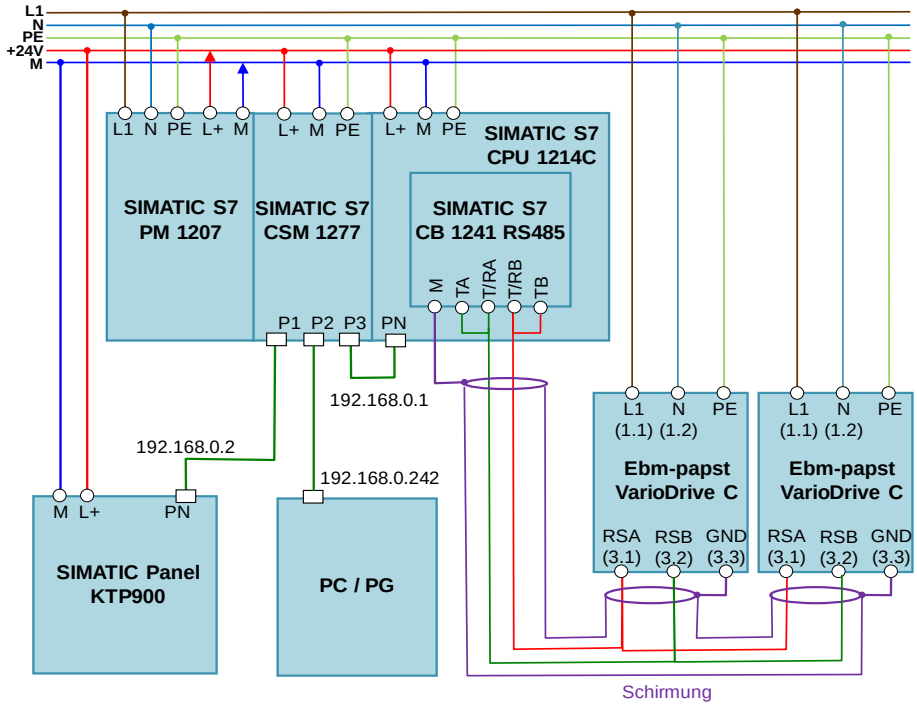


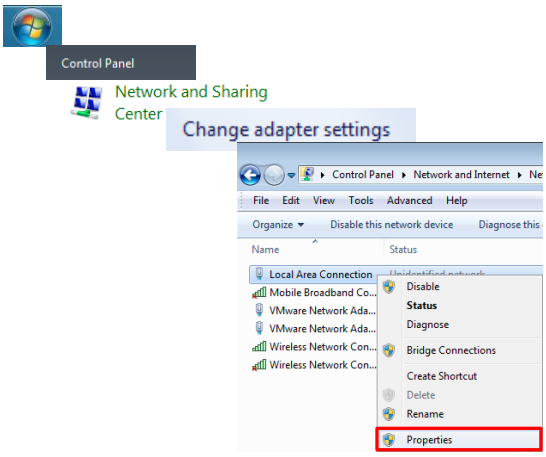
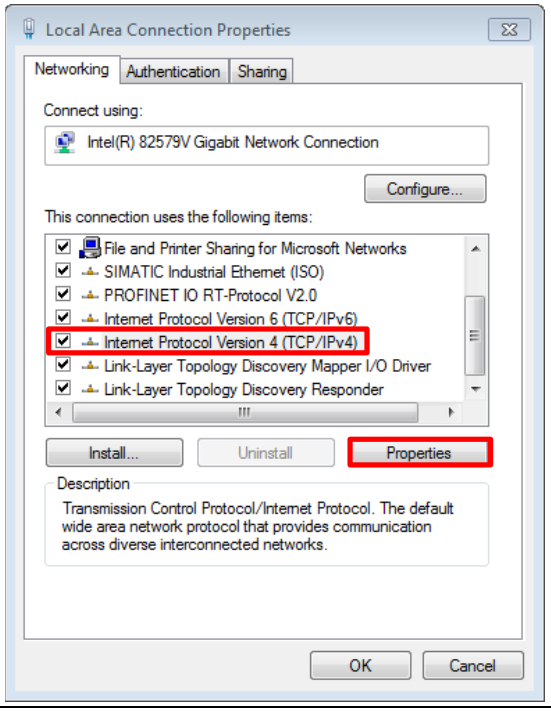
Table 4-1

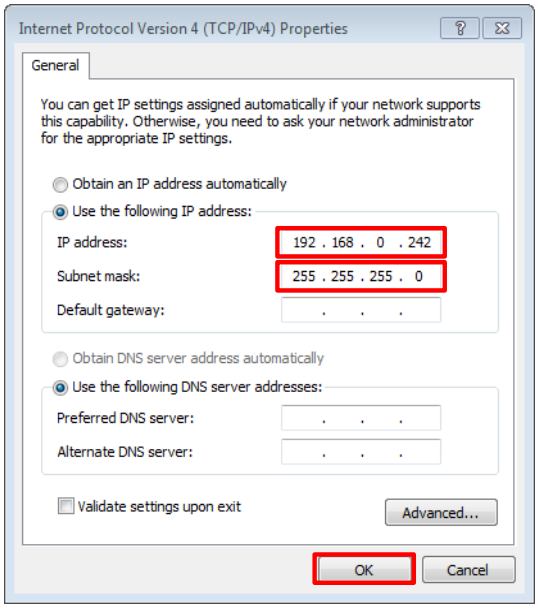
No.	Action	Remarks
1.	Install all required S7-1200 components on a top hat rail.	See Table 2-1
2.	Wire and connect all required components as described. Pay special attention to the crossed communication link between CB 1241 RS485 and fan (T/R**A** – RS**B** and T/R**B** – RS**A**).	See S7-1200 manual (\8\) Chapter A "Technical Data" and operating instructions for ebm-papst motor (\11\)
3.	Finally, connect the fan (here: "VarioDrive C") and the SIMATIC PM 1207 power supply to the AC power system (230 V).	

4.2 Network connections

To configure the controller and the HMI, the LAN network card of the programming unit needs a static IP address.
The configuration of the LAN connection is described in the following.

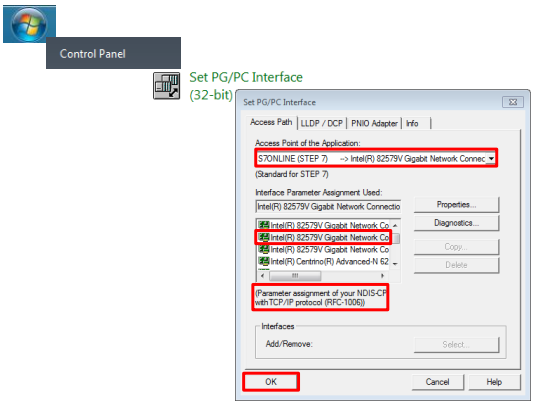
Table 4-2

No.	Action	Remark
1.	Click “Start > Control Panel > Network and Sharing Center > Change adapter settings” to open the network connections. • Select your network connection. • Right-click to open the properties.	
2.	Select the “Internet Protocol Version 4 (TCP/IPv4)” element in “Networking” and open its properties.	

No.	Action	Remark
3.	• Choose “Use the following IP address”. • Select an IP address in the CPU's subnet mask. • Confirm the settings with “OK” and “Close”.	

4.2.1 Setting the PG/PC interface

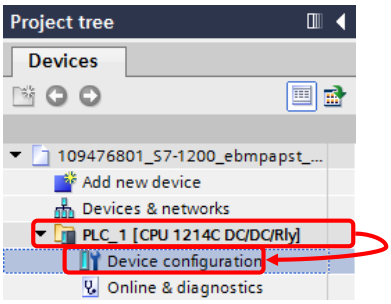
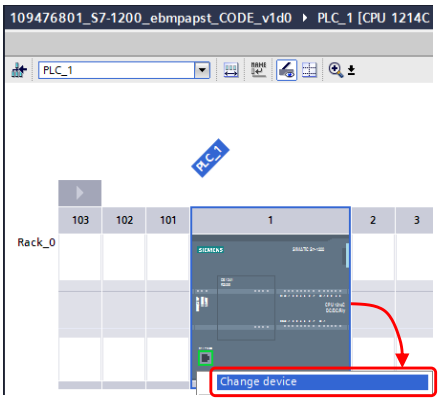
Table 4-3

No.	Action	Remarks
1.	Open the PG/PC interface settings via “Start > Control Panel” to set the correct access path for STEP 7 V15. • Select “S7ONLINE (STEP 7)” as the application's access point. • Select your network card with “(Parameter assignment of your NDIS CP with TCP/IP protocol (RFC-1006))” as the interface configuration used. Confirm the settings with “OK”.	

4.3 Configuration instructions

4.3.1 Adjusting the device configuration

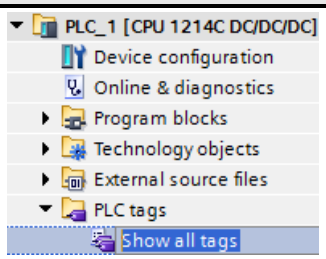
Table 4-4

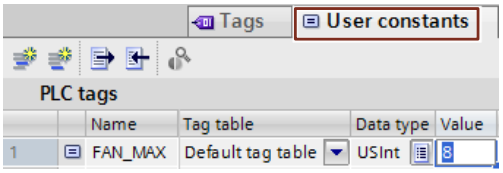
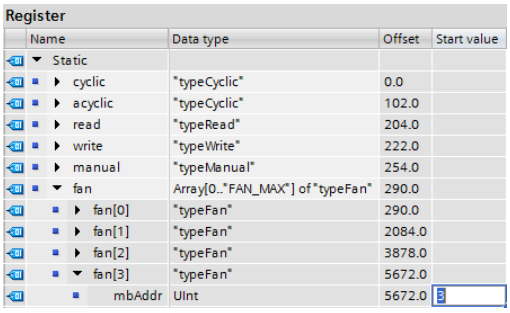
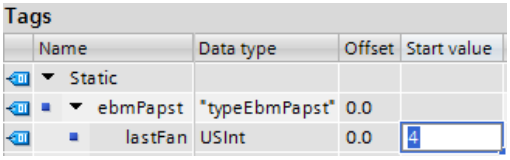
No.	Action	Remarks
1.	Network the S7-1200 controller with your programming unit. Assign the IP addresses specified in Figure 4-1 .	Configuring the CPU for communication: S7-1200 manual (\8\)→ Chapter 6.9
2.	Open the project file (ap15) with STEP 7 V15.	see Table 2-2
3.	Open the device configuration of the controller “PLC_1”.	
4.	If necessary, adjust the CPU configured in the project to the real CPU by selecting the configured CPU and clicking the right mouse button to choose “Change device…” (S7-1200 manual (\8\)→ Chapter 6.5).	

4.3.2 Parametrization

The application example is designed for up to 4 fans and was tested with 2 fans. Here you will find out how to expand the application example to up to 32 fans and how to assign the Modbus addresses of the fans.

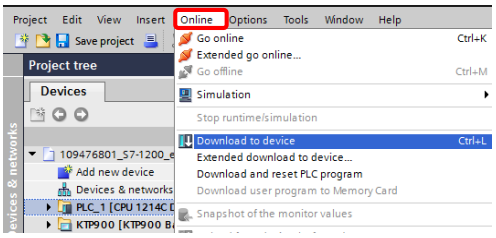
Table 4-5

No.	Action	Remarks
1.	Display all PLC tags of the controller in the TIA Portal project.	

No.	Action	Remarks
2.	Open the “User constants” register and increase “FAN_MAX” to the maximum number of fans to be operated. This expands the array "fan" and therefore also the RAM usage of the CPU project. Note: When decreasing “FAN_MAX” (< 4), the tag references of the HMI are also deleted which leads to errors during compilation!	
3.	Open the DB “Register”. Under “fan[x].mbAddr”, enter the Modbus address of the respective fan as a start value (1 ≤ x ≤ "FAN_MAX"; 0 serves as a placeholder for the HMI display “Fan view”).	
4.	Open the DB “Tags”. Enter the actual number of fans under *"ebmPapst".lastFan*. The actual value is limited by the FB "EbmPapst" to 1 ≤ *lastFan* ≤ "FAN_MAX".	

4.3.3 Downloading the controller project part

Table 4-6

No.	Action	Remarks
1.	If changes have been made, save and compile the project. Select the program folder of the S7-1200 and transfer the program into the controller by clicking the “Download to device” option. Select the option “Start module” in the “Load results” under “Start modules” and complete the loading process.	 see S7-1200 manual (\8\)→ Chapter 15.2

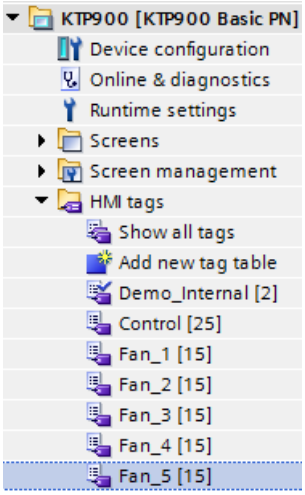
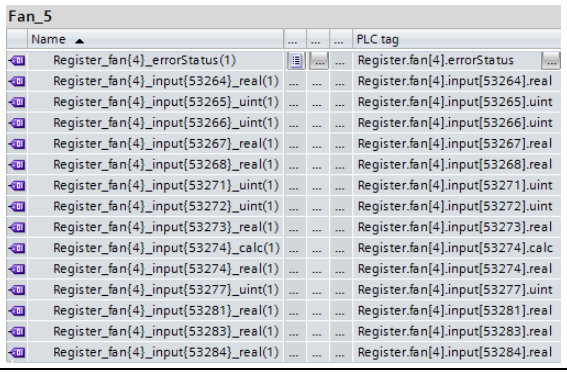
4.4 HMI project part

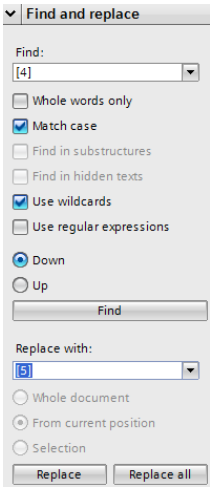
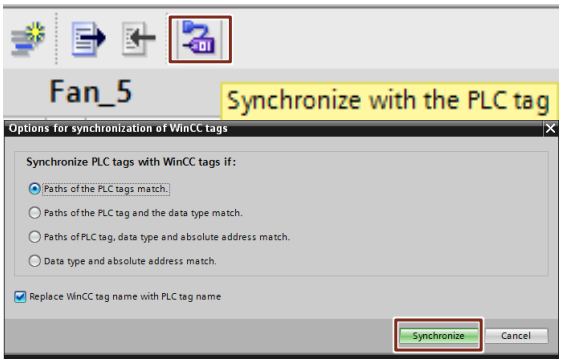
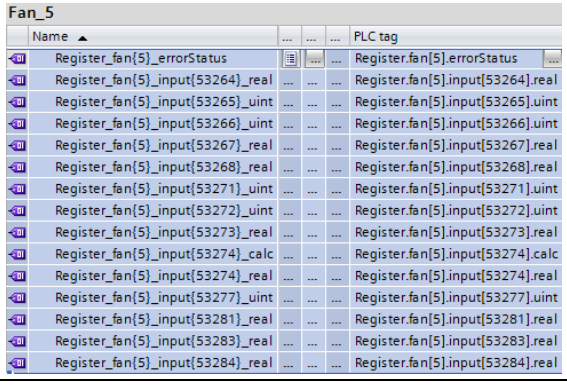
4.4.1 Expanding the “Overview” display

In the “Fan view” display, the data of the fan to be displayed are copied to the *Register".fan[0]* placeholder by selecting the respective fan. In doing so, you can visualize the data of each fan.

The “Overview” display shows the input registers of several fans side by side. The following chapter shows how to expand the overview screen.

Table 4-7

No.	Action	Remarks
1.	• Open the HMI tags of the KTP900 operator panel. • Copy a "Fan_x" tag table and paste it. A new tag table with the next available number is created (here: “Fan_5”).	
2.	• Open the new tag table "Fan_5". The column “PLC tag" still contains the copied reference to "fan[4]". • Select all entries and confirm “Ctrl" + "F".	

No.	Action	Remarks
3.	The “Find and replace” option of the “Jobs” task card opens. • Enter the following: - Find: [4] (existing fan reference) - Replace: [5] (new fan reference) • Confirm with “Replace all”.	
4.	The entries will now reference the PLC tags of the new fan (prerequisite: The user constant "FAN_MAX" has been modified accordingly -> Table 4-4 step 2). • Click the “Synchronize with the PLC tag” button. • Take over the default settings and click the “Synchronize” button.	
5.	Now the HMI tags for the new fan 5 have been created.	

Then, you have to create a display field for the input registers of the new fan and assign the display elements with the new HMI tags.
For example, you could create a copy of the display “10_Overview” and connect the two displays ("10_Overview" <-> "11_Overview").
You then have to adapt the following references to HMI tags in the display field. To do so, replace x with the new fan number (here: 5):

Figure 4-2

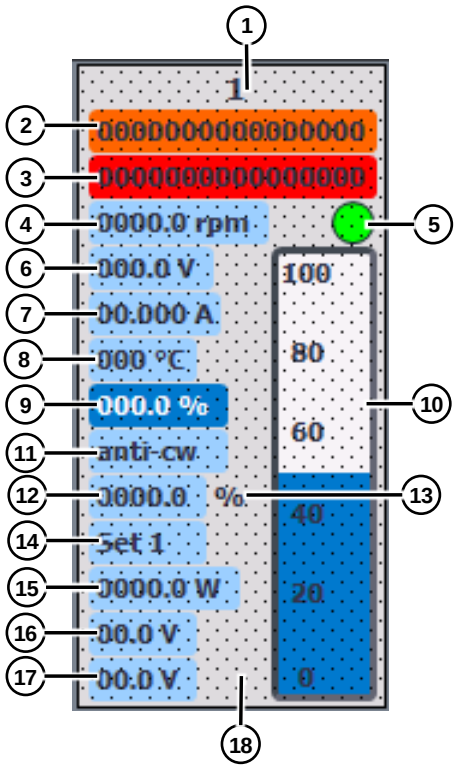


Table 4-8

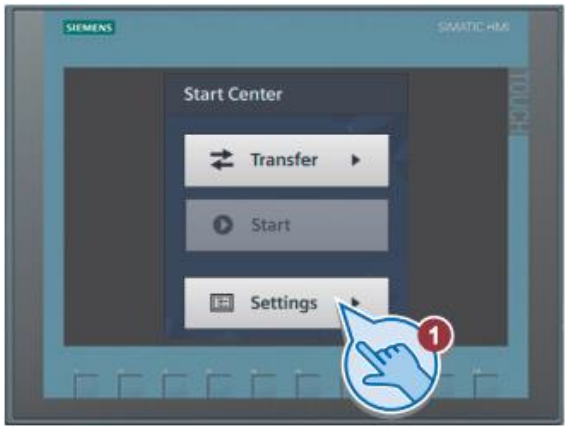
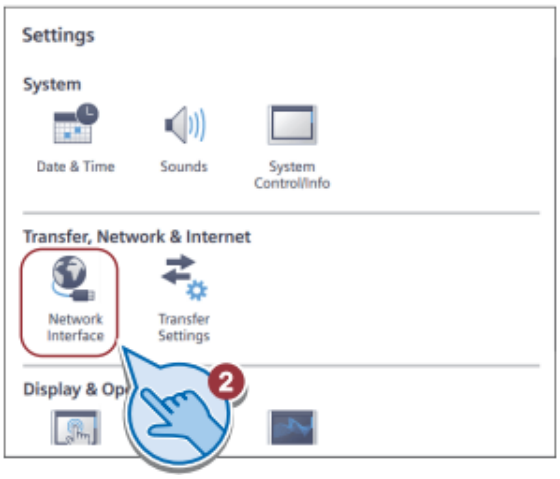
No.	Object	Reference
1.	TextField_FanNumber_x	Properties > General > Tag: x
2.	I/Ofield_Warning_x	Properties > General > Process > Tag: Register_fan{x}_input{53266}_uint
		Properties > Animations > Display > Layout > Tag: Register_fan{x}_input{53266}_uint
3.	I/Ofield_MotorStatus_x	Properties > General > Process > Tag: Register_fan{x}_input{53265}_uint
		Properties > Animations > Display > Layout > Tag: Register_fan{x}_input{53265}_uint
4.	I/Ofield_Speed_x	Properties > General > Process > Tag: Register_fan{x}_input{53264}_real
5.	Circle_Online_x	Properties > Animations > Display > Layout > Tag: Register_fan{x}_errorStatus
6.	I/Ofield_DClinkVoltage_x	Properties > General > Process > Tag: Register_fan{x}_input{53267}_real
7.	I/Ofield_DClinkCurrent_x	Properties > General > Process > Tag: Register_fan{x}_input{53268}_real
8.	I/Ofield_ElectronicsTemp_x	Properties > General > Process > Tag: Register_fan{x}_input{53271}_uint
9.	I/Ofield_ModulationLevel_x	Properties > General > Process > Tag: Register_fan{x}_input{53273}_real
10.	Bar_ModulationLevel_x	Properties > General > Process > Process tag: Register_fan{x}_input{53273}_real
11.	SymbolicI/Ofield_Direction_x	Properties > General > Process > Tag: Register_fan{x}_input{53272}_uint

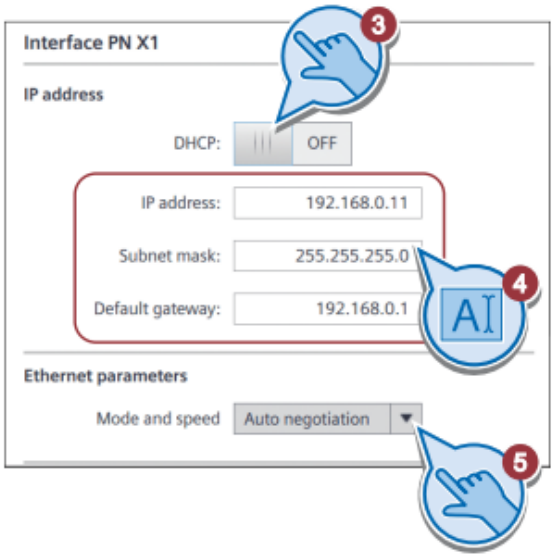
No.	Object	Reference
12.	I/Ofield_SetValue_x	Properties > General > Process > Tag: Register_fan{x}_input{53274}_real
13.	SymbolicI/Ofield_Unit_x	Properties > General > Process > Tag: Register_fan{x}_input{53274}_calc
14.	SymbolicI/Ofield_ParamSet_x	Properties > General > Process > Tag: Register_fan{x}_input{53277}_uint
15.	I/Ofield_Power_x	Properties > General > Process > Tag: Register_fan{x}_input{53281}_real
16.	I/Ofield_SensorValue1_x	Properties > General > Process > Tag: Register_fan{x}_input{53283}_real
17.	I/Ofield_SensorValue2_x	Properties > General > Process > Tag: Register_fan{x}_input{53284}_real
18.	Button_SetReadFan_x	Event: Set tag "Tags_EbmPapst_HMI_readFan" = x

4.4.2 Configuring the HMI

If the KTP900 Basic is used as an operator panel, the project-specific IP address (see Figure 4-1) must be set.

Table 4-9

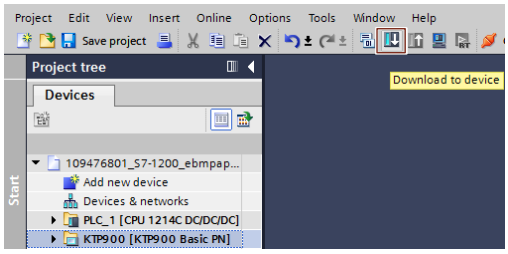
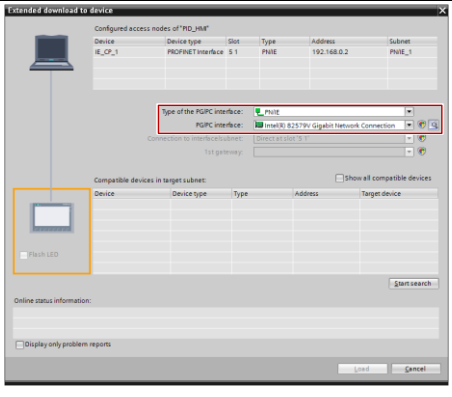
No.	Action	Remarks
1.	• Connect the KTP900 Basic to the power supply voltage. • Access the settings by clicking the “Settings” button in the “Start Center”.	
2.	• Next open the settings of the network interface by clicking the “Network interface” button.	

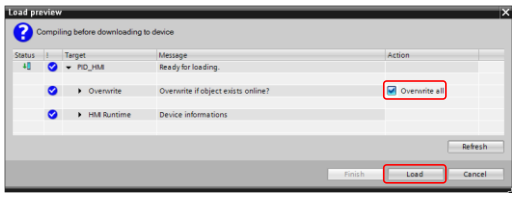
No.	Action	Remarks
3.	• Disable the option “DHCP”. • Enter the network setting from Figure 4-1 for the HMI: - IP address: 192.168.0.2 - Subnet mask = 255.255.255.0 - “Default gateway” is irrelevant. • Select “Auto negotiation” in the Ethernet parameters to automatically determine the speed transmission.	

Downloading the HMI project part into the KTP900 Basic

Connect your PG/PC to the HMI either directly or using the CSM1277 switch.

Table 4-10

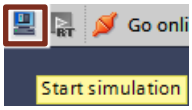
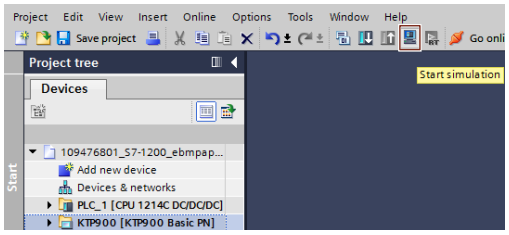
No.	Action	Remarks
1.	• Select the operator panel folder “HMI_1 [KTP900 Basic PN]”. • Click the “Download to device” button to download the HMI project part into the KTP900 Basic.	
2.	• When the “Extended download to device” window opens, select “PN/IE” as the type of the PG/PC interface. “PN/IE”. • Select the network card you are using as the PG/PC interface.	

No.	Action	Remarks
3.	• If necessary, activate the option “Overwrite all”. • Click the “Load” button.	
4.	• Depending on the operator panel settings, you may have to trigger the transfer by clicking the “Transfer” button in the Start Center of the KTP900 Basic.	

Starting the PC runtime

If you want to use the PG/PC as the operator panel, start the PC runtime as follows:

Table 4-11

No.	Action	Remarks
1.	• Select the operator panel folder “PID_HMI [KTP900 Basic PN]”. • Click the “Start simulation” button. 	

5 Operating the Application Examples

Overview and description of the user interface

Figure 5-1

The user interface consists of 6 menus:

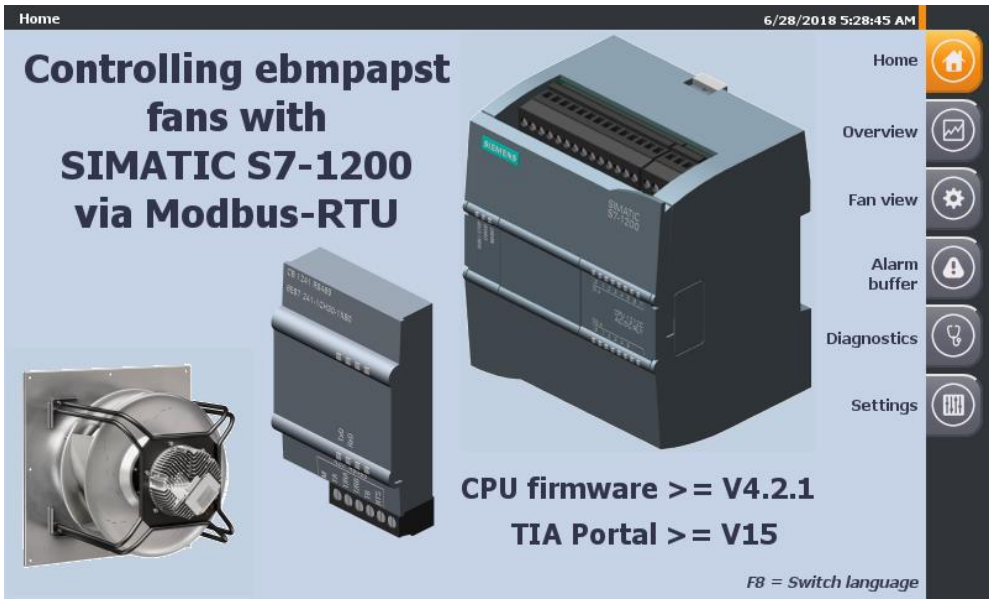
- Home screen (intro)
- Overview screen of the fans
- Fan view
- Alarm buffer
- Diagnostics
- System functions

5.1 Intro (home screen)

The home screen provides information about the current topic.
It presents the control of several fans from ebm-papst with the SIMATIC S7-1200 product family using Modbus RTU. The used system functions “Modbus_Comm_Load” and “Modbus_Master” V3.1 are available in the SIMATIC S7-1200 controller from CPU firmware V4.2 in connection with STEP 7 (TIA Portal) from version V15 (\3\).

Additionally, operation of the right menu bar is explained.
This bar is available on each display.

Figure 5-2



-  takes you to the home screen (this display).
-  takes you to the overview of fans screen (Figure 5-6).
-  takes you to the fan view (Figure 5-7).
-  takes you to the alarm buffer (Figure 5-9).
-  takes you to the diagnostics (Figure 5-10).
-  takes you to the system functions (Figure 5-11).
- The F8 key can be used to switch between German and English.

The currently selected menu is highlighted by the orange background of the icon, e.g.  (for the home screen) and has the respective title in the header (left):

The current date and time is displayed in the header on the right side:

The header is also visible in all displays.

5.2 Recurrent display elements

The following subchapters describe the recurrent display elements available both in the “Overview” and in the “Fan view” display:

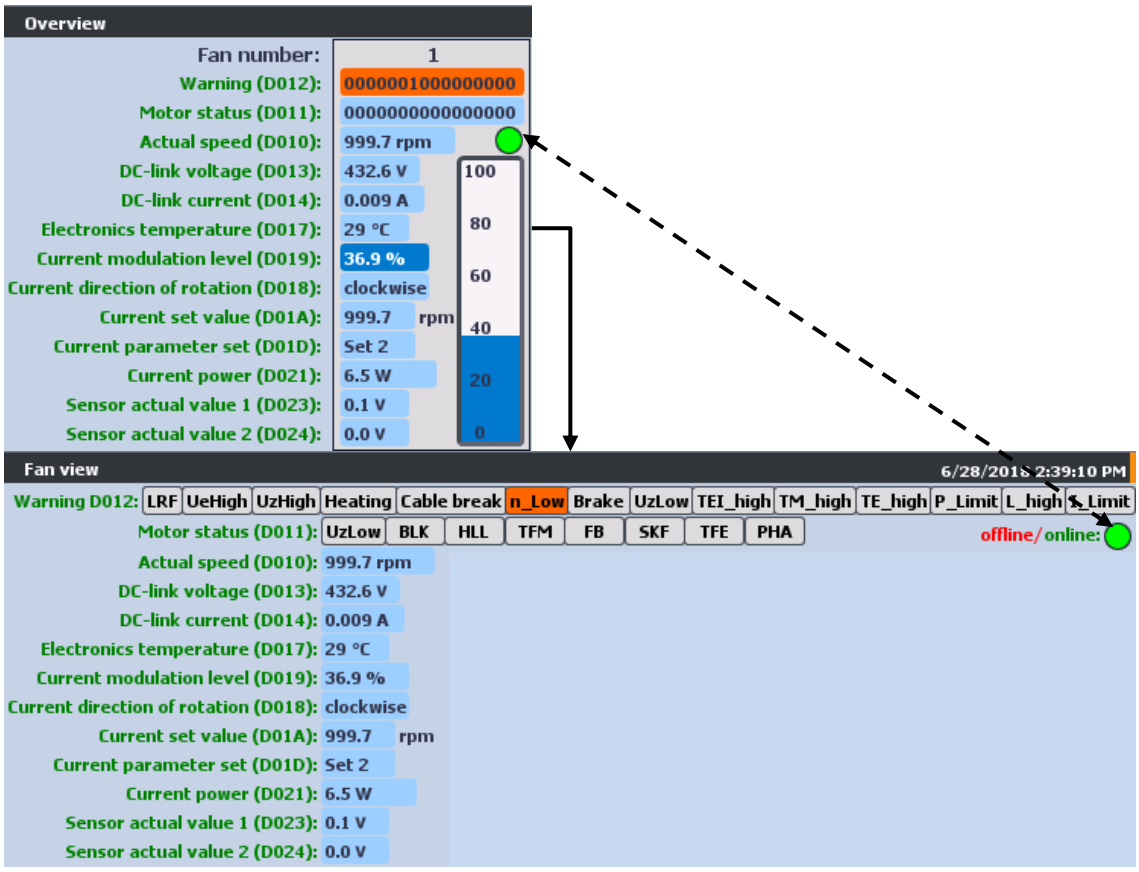
5.2.1 Cyclically retrieved values (input registers)

The upper part of the display shows the most important input registers (in green) which are read cyclically.

These values are available in the “Overview” as well as in the “Fan view" display (with the difference that the “current modulation level" is displayed as a bar graph in the “Overview” display and the “Warning" and “Motor status” screens in the “Fan view" display is shown in more detail).

Clicking the field with the fan data in the “Overview” display takes you to the “Fan view” of the selected fan.

Figure 5-3



All registers are displayed numerically or as text.

The statuses for “Motor status (D011)” and “Warnings” (D012) are displayed in color:

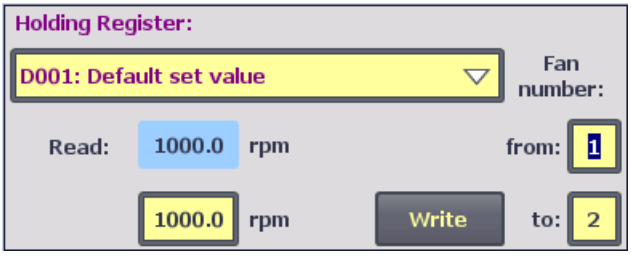
These messages are also listed in the alarm buffer in text form (chapter 5.5).

5.2.2 Writing to the holding registers

The holding registers (purple font) are written according to the selection of the corresponding register in the drop-down menu.
Depending on the selected display, you can access all writable registers from Table 3-1 in the “Fan view” display and only the following registers in the “Overview” display:

- D001: Default set value
- D102: Preferred running direction
- D105: Internal parameter set

Figure 5-4



If the start fan (from:) to be addressed or the register to be written changes, it will be read automatically (blue background) and the read value is taken over from the input/output field (yellow background) to be written to. You can now edit the value within the specified limits and transfer the value to all fans in the from/to area using the “Write” button.
Subsequently, the selected start fan register (from:) is automatically read again.
If the transfer has been successful, the read value corresponds to the written value.

Certain holding registers (see Table 3-1) are first written to the fan's EEPROM and in order to be transferred to the RAM, they require the “Param.” function to be executed in the “Reset” (D000) holding register. These parameters have the edit value field displayed with an orange background: 

If the register access requires authorization with a password (see Table 3-1) that has not been transferred yet, error code 16#8388 (“Error in the response of the slave to write request”) occurs in the status evaluation.

Note

When writing the default set value (D001), make sure that all fans to be addressed (from to) must be in the same operating mode (Control [%] or speed control [rpm]).
The default set value is otherwise interpreted incorrectly and the fans will be controlled differently!

Default settings to operate the application example

Starting from the factory settings of the interface parameter settings,

Table 5-1

Address	Designation	Value
D100	Fan address	1
D149	Transfer rate	19200 bit/sec
D14A	Parity configuration	8E1

the following default settings of the holding registers have to be made for the application example to run smoothly:

When operating several fans, make sure that only one fan with Modbus address 1 can be reached by the S7-1200 at once. For fan 1, change its Modbus address to an address that is entered as a start value *"Register".fan[x].mbAddr* of another fan. After "Reset" = "Param." for confirming the address change, the fan will be reached under the number x. Now expand the bus with the next fan with default settings and repeat the procedure until all fans can be reached.

Table 5-2

Address	Designation	Value
D101	Set value source	RS485
D104	Source parameter set	Internal parameter set
D148	Source running direction	Preferred running direction

All 3 parameter defaults are written to the fan's EEPROM. To be applied, the values have to be copied into the RAM. This is accomplished by executing (write) the “Param.” function in the “Reset” (D000) holding register.

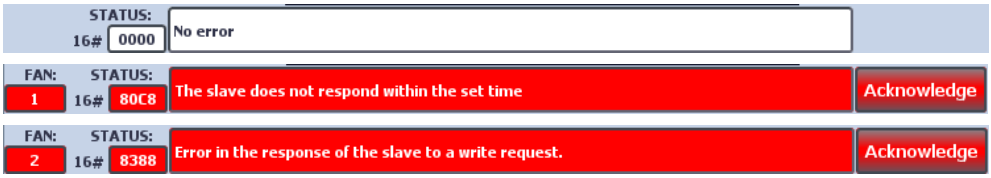
Note

You can restore the factory settings using the “Control default setting” (D005) holding register by executing the function “W -> D”.
For this purpose, you need a 12-digit hexadecimal password which you have to write into the “Password” (D002) holding register first.
This password is available from ebm-papst.

5.2.3 Status evaluation

At the bottom edge of the display, you see a status display of the data exchange with the fans.

Figure 5-5



During error-free operation, the status is marked with hexadecimal code 16#0000 (“No error”).
When communication errors with the respective fan occur, the error code in connection with the plain text output provides information about the error that has occurred.

The following table lists the most frequent errors:

Table 5-3

Code	Text	Cause
16#80C8	The slave does not respond within the set time	Disturbed connection to the fan (e.g. missing voltage supply of fan)
16#8388	Error in the response of the slave to a write request	Missing password authorization to write the selected holding register

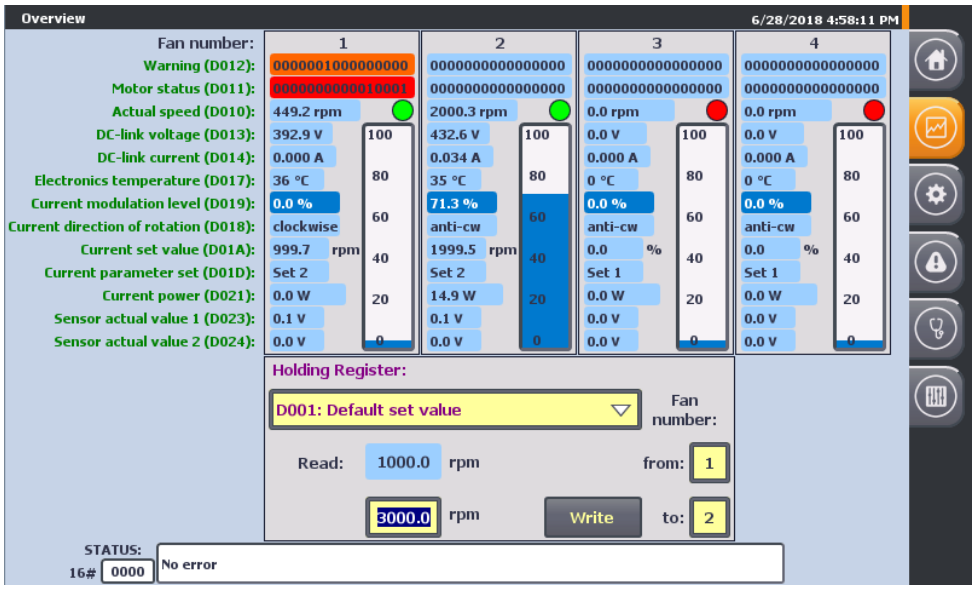
Note Further conditional codes are provided in the chapter "Modus RTU Instructions" in the S7-1200 system manual (\8\).

Errors that occur remain pending. The error display is only reset after the error has been acknowledged using the Acknowledge button (provided that the error no longer applies).

5.3 Overview

The "Overview" display shows the sequentially read input registers of fans 1 to 4 side by side. In addition, the availability of the fans is signaled with the offline/online: screen and the "Current modulation level (D019)" is shown as a bar graph. Pending "Warnings (D012)" and "Motor status (D011)" are highlighted in color. Clicking the respective fan data field takes you to the detailed "Fan view".

Figure 5-6

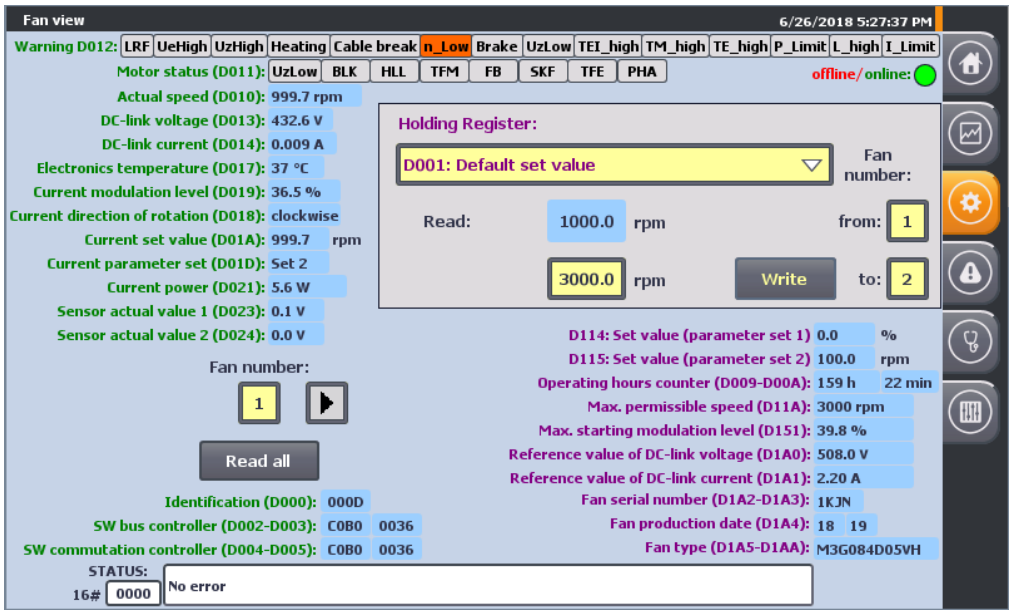


Depending on which operating mode is set (holding register D106/D107), the current set value (D01A) is displayed in the unit "rpm" for speed control and in "%" for control.

5.4 Fan view

The "Fan view" display gives access to the recurrent display elements (chapter 5.2) and also to all registers (see chapter 3.2).

Figure 5-7

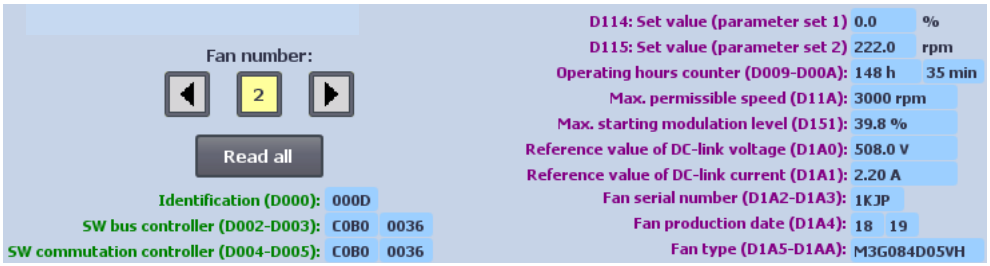


The drop-down menu in the dialog for writing to the holding registers provides access to all writable holding registers (see Table 3-1).

Additionally, you can read all other non-writable registers (blue background) of all fans to be addressed (1 to "Tags".ebmPapst.lastFan) using the Read all button:

- Device-specific input registers (green font)
- Non-writable holding registers (purple font)

Figure 5-8



While all other parameters are being read, the Read all button disappears and a progress bar display 44.4 % appears.

Reading of the other parameters is a snapshot.
For an update, the action has to be repeated (for example for the operating hours counter: holding register D009 / D00A).

Use 

to select the fan for data display.

Holding registers D114/D115 are only shown, when "Save set value" (D103) was specified.

Note

In order to control the fan via Modbus RTU without any problems, certain holding registers need to have the corresponding default settings (see chapter 5.2.2)

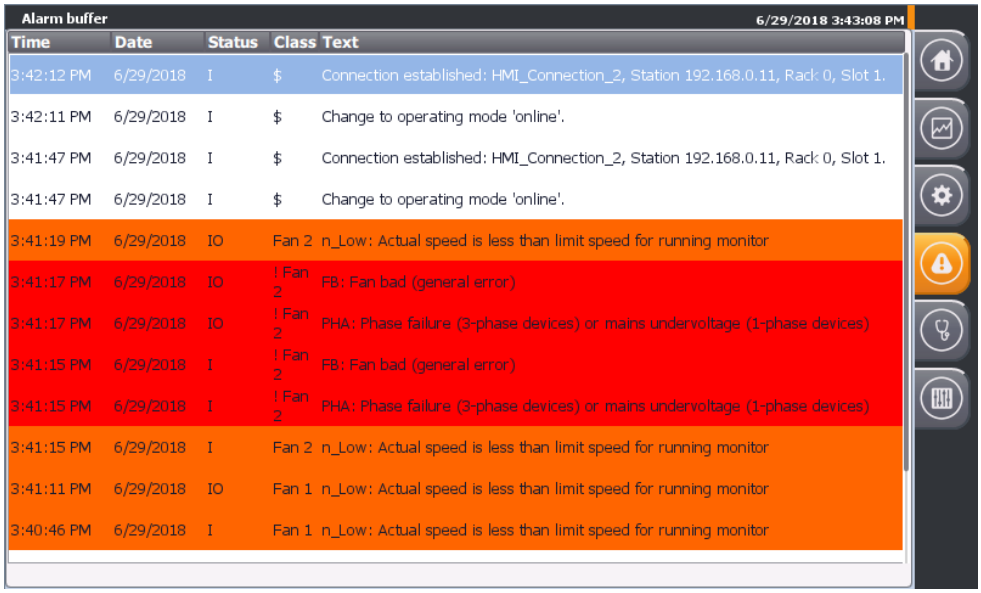
5.5 Alarm buffer

This display lists the following alarm messages including time stamp and status in text form as alarm buffer:

- System alarms (marked by a prefixed “$”)
- Motor status of the individual fans (input register D011 – marked by a prefixed “!” – red background)
- Warnings of the individual fans
  (input register D012 – orange background)

Both the incoming event (status “I”) and the outgoing event (status “IO”) are recorded.

Figure 5-9

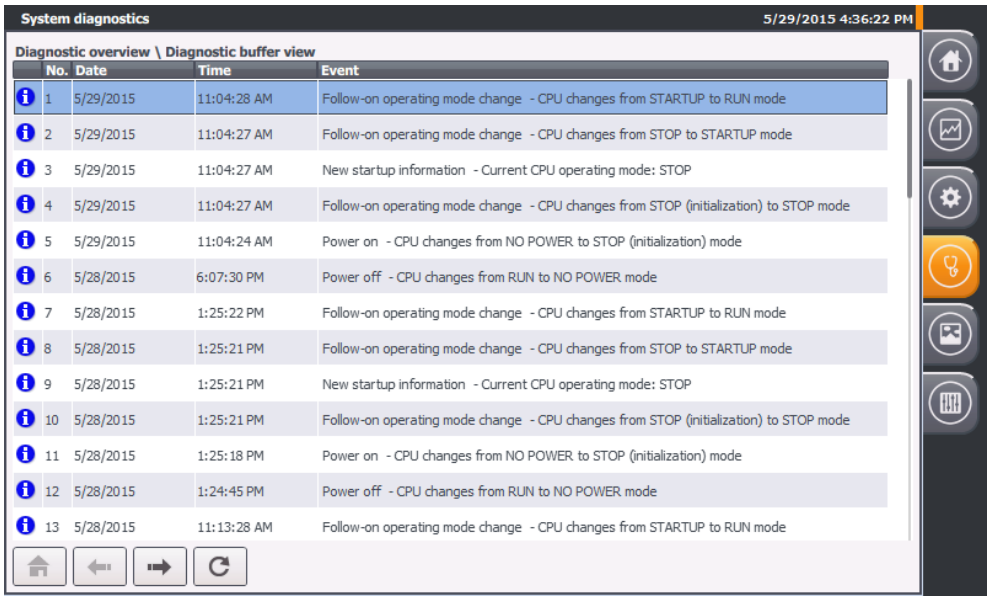


Note For more information about the input registers D011 (motor status) and D012 (warning), see chapter 5.2.1 and the MODBUS parameter specification for “ebm-papst series 84 / 112 / 150 / 200” V5.00 (see chapter 6.3).

5.6 Diagnostics

The “System diagnostics” display shows the system diagnostics buffer of the S7-1200 controller.

Figure 5-10

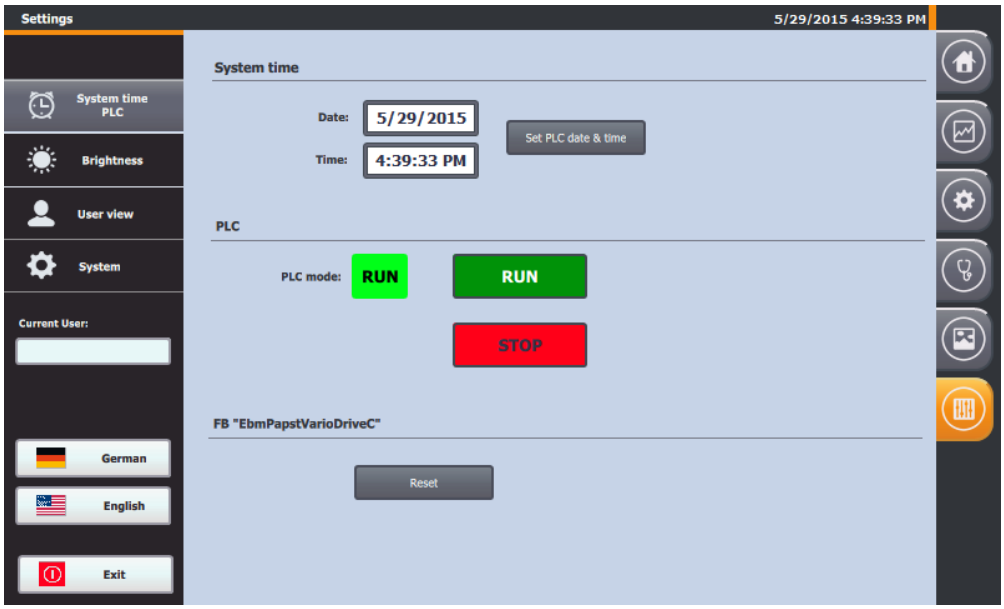


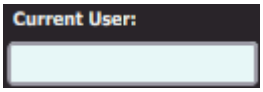
5.7 Settings

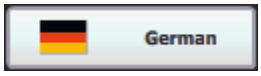
The settings menu contains the dialogs

- System time / PLC
- Brightness
- User
- System

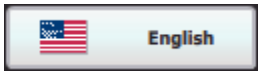
Figure 5-11



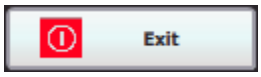
The user currently logged-on is displayed: 

Click 

to select “German” as the display language.

Click 

to select “English” as the display language.

Click 

to exit the HMI runtime.

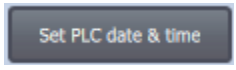
5.7.1 System time/PLC

System time

The application example has a time synchronization between PLC and HMI.

allows you to edit the date and 

to edit the time.

Click 

to apply the settings and adjust the PLC system time.

PLC operating state

shows the current PLC mode.

Click 

to switch the PLC to “RUN” mode.

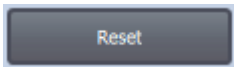
Click 

to switch the PLC to “STOP” mode.

In the PLC mode 

, the header and side bar alternately flash orange:

FB “EbmPapst”

Click the 

button to reset the sequencer of the function block “EbmPapst”.

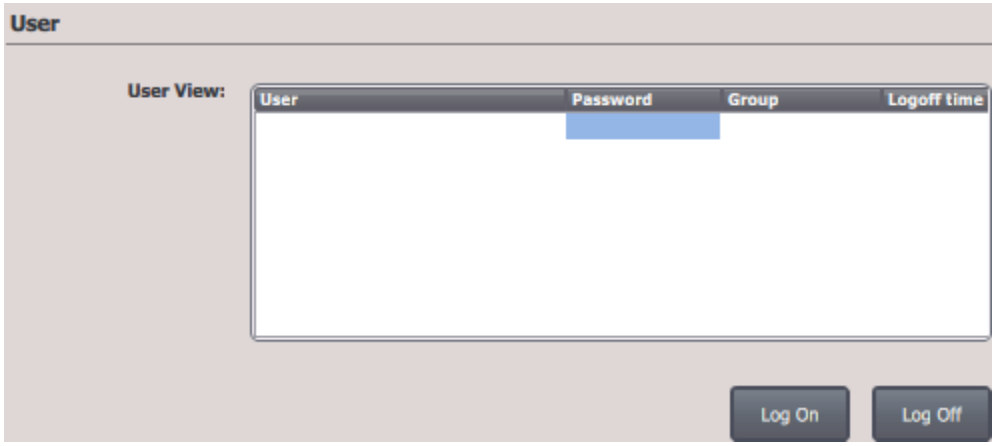
5.7.2 Brightness

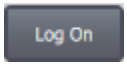
Allows you to regulate the brightness in percent (setting range: 30 to 100%) of the operator panel (only possible when using a real HMI).

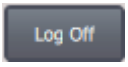
5.7.3 User display

When a user logs on, the details (user, password, group and logoff time) are displayed here. No authorization was assigned in the application example. Therefore, no user logon is necessary.

Figure 5-12

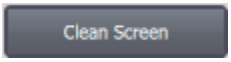


Click 

to log on and 

to log off.

5.7.4 System

Click 

to temporarily disable the touch panel in order to clean it (only possible when using a real HMI).

6 Appendix

6.1 Service and Support

Industry Online Support

Do you have any questions or need support?

Siemens Industry Online Support offers access to our entire service and support know-how as well as to our services.

Siemens Industry Online Support is the central address for information on our products, solutions and services.

Product information, manuals, downloads, FAQs and application examples – all information is accessible with just a few mouse clicks at:
https://support.industry.siemens.com

Technical Support

Siemens Industry's Technical Support offers quick and competent support regarding all technical queries with numerous tailor-made offers
– from basic support right up to individual support contracts.

Please address your requests to the Technical Support via the web form:
www.siemens.en/industry/supportrequest

SITRAIN – Training for Industry

With our globally available training courses for our products and solutions and using innovative teaching methods, we help you achieve your goals.

More information on the training courses offered as well as on locations and dates is available at:
www.siemens.de/sitrain

Service offer

Our service offer includes the following:

- Plant Data Services
- Spare Parts Services
- Repair Services
- On Site and Maintenance Services
- Retrofit and Modernization Services
- Service Programs and Agreements

Detailed information on our service offer is available in the Service Catalog:
https://support.industry.siemens.com/cs/sc

Industry Online Support app

Thanks to the "Siemens Industry Online Support" app, you will get optimum support even when you are on the move. The app is available for Apple iOS, Android and Windows Phone:
https://support.industry.siemens.com/cs/ww/en/sc/2067

6.2 Links and literature

Table 6-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the entry page of the application example https://support.industry.siemens.com/cs/ww/en/view/109476801
\3\	How can you read input words in the address range from 9999 to 65535 with the SIMATIC S7-1200 via Modbus RTU? https://support.industry.siemens.com/cs/ww/en/view/109474481
\4\	Library with general functions for (LGFP) for STEP 7 (TIA Portal) and S7-1200 / S7-1500 https://support.industry.siemens.com/cs/ww/en/view/109479728
\5\	Updates for STEP 7 V15 and WinCC V15 https://support.industry.siemens.com/cs/ww/en/view/109755826
\6\	Firmware update for CPU 1214C, DC/DC/RELAIS, 14DI/10DO/2AI https://support.industry.siemens.com/cs/ww/en/view/107539979
\7\	Operating system update V2.1.0 for CM 1241 https://support.industry.siemens.com/cs/ww/en/view/108819199
\8\	SIMATIC S7-1200 Programmable controller manual https://support.industry.siemens.com/cs/ww/en/view/109741593
\9\	SIMATIC STEP 7 Basic/Professional V15 and SIMATIC WinCC V15 system manual https://support.industry.siemens.com/cs/ww/en/view/109755202
\10\	Ebm-papst website https://www.ebmpapst.com/en/
\11\	Operating Instructions M3G112GA5271 http://img.ebmpapst.com/products/manuals/M3G112GA5271-BA-ENG.pdf

6.3 Contacts

If you have questions regarding compatible fans (e.g.

- The MODBUS parameter specification for “ebm-papst series 84 / 112 / 150 / 200" V5.00 or
- passwords required to write protected holding registers)

please contact:

ebm-papst Mulfingen GmbH & Co. KG
Bachmühle 2
D - 74673 Mulfingen
Telephone: +49 7938 81-0

mailto: info1@de.ebmpapst.com

6.4 Change documentation

Table 6-2

Version	Date	Modification
V1.0	08/2015	First version
V2.0	09/2018	Expansion to control several fans