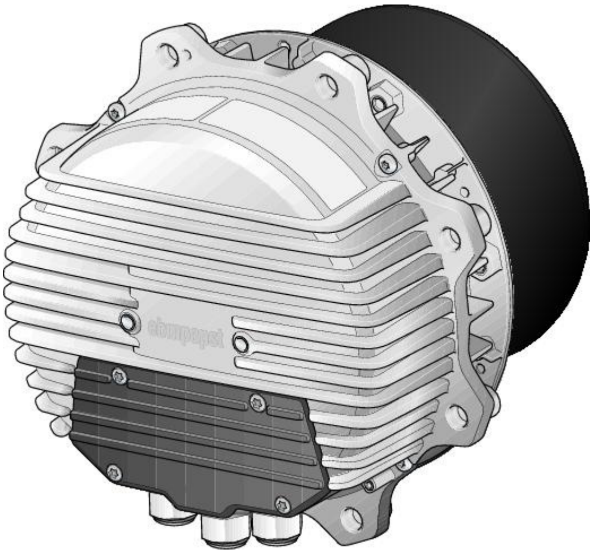
Fantech EC Fans – Type 1

Direct Connectivity Guide

MODBUS Programming and Operation Guide



Committed to

Service and Innovation

Australia

Adelaide	(08) 8294 0530
Brisbane	(07) 3299 9888
Canberra	(02) 6280 5511
Darwin	(08) 8947 0447
Launceston	(03) 6344 6888
Melbourne H.O.	(03) 9554 7845
Newcastle	(02) 4961 6088
Perth	(08) 9209 4999
Sydney	(02) 8811 0400
Townsville	(07) 4775 5222

New Zealand

Auckland H.O.	(09) 444 6266
Christchurch	(03) 379 8622
Wellington	(04) 566 0532

Asia

For agents in Asian region call Eltafantech on (603) 7846 0340 or visit www.eltafantechasia.com

As part of our on-going program of product development and improvement, Fantech reserves the right to make changes to design or specification to products without notice or obligation and to change or discontinue models.

Contents

1. Introduction

The products within the Fantech range of EC products optionally come with RS485 serial interfaces supporting the MODBUS communication protocol for the control of fan speed and feedback of status information. This document covers the applications of MODBUS communication to achieve a level of functionality suitable for HVAC Applications. It is assumed that the reader is applying the document with the intent of configuring a control system interface to operate with a Fantech EC Fan Type 1 and has a working knowledge of MODBUS and high-level control in general.

Knowledge of general Modbus specifications is also required:
- MODBUS over Serial Line Specification & Implementation guide V1.0
- MODBUS Application Protocol Specification V1.1

These documents are available in the internet at modbus.org

The general Modbus specifications form the basis for this document and are valid in full with the exception of the restrictions described in this document. Specific information relating to the data encoding and commands supported by the fan over MODBUS are outlined in Section 5 ‘MODBUS Communications General and Protocol Information’.

Specifics on minimum required registers to achieve functionality between an external MODBUS controller and the fan unit are described in Section 3 ‘ Operation registers and Commands’ which denotes the register addresses and a description of the data encoding for these registers. Information for additional registers for functions above basic operation is noted in Section 4.

2. Wiring configuration

Fantech wiring diagram W028 denotes the overall wiring connection for all Fantech EC fans described in this document. Of all the wiring terminals available, only a number of wire connections are necessary.

This guide assumes that the following control wiring terminals are physically connected as follows:

Wiring Terminal	Wire Colour	Description
RSA / RSA1	Black	RS485 Interface Terminal A, D+
RSB / RSB1	White	RS485 Interface Terminal B, D-
GND1	Blue	Common ground for external controller and fan control circuit
NO	Red + Blue Spots	Forms alarm circuit with 'COM' and opens when fan enters a fault condition.
COM	Green + Red Spots	Forms alarm circuit with 'NO' (optionally NC) and opens when fan enters a fault condition.

Wiring Notes:

- Wires for the RS485 interface are to be of a screened twisted pair construction, 18AWG conductor size. Similar to Belden 8760.
- Maximum wiring loop length for the RS485 connections on site is 1.5km with the use of the correct cabling type.
- The ground terminals for the external controller and fan must be inkind through the screening of the RS485 cable interface or a suitable low resistance connection. Failure to do so may result in damage to the communications interface of the external control or fan control circuit through overvoltage. FANTECH WILL NOT WARRANT ANY PRODUCTS WHICH HAVE FAILED DUE TO OVERVOLTAGE APPLIED TO THE ONBOARD CONTROLS ELECTRONICS DUE TO MISAPPLICATION.
- The NO and COM terminals of a number of fan units may be wired together to form a fault alarm 'loop' that will go to open circuit upon the failure of any of the fans within the loop. However, this may not be sufficient for Run/Fault indication as the alarm does not indicate whether or not the fan is actually operating.

3. Operation registers and commands

Fantech suggest a minimum of three operations that the external controller should be set to perform to achieve a suitable level of functionality with the Fantech EC Fan unit:

- Setting Fan Speed
- Reading of fan fault byte periodically to check the fault status of the fan
- Reading of fan speed to determine run/stop status of the fan

Requested Fan Speed

Register Type: Holding Register
Address : D001
Action Required: Write requested speed in % of maximum speed to the holding register address

Encoding:
The requested fan speed is in the form of a percentage value where 64000 represents 100% speed.

100% speed corresponds to the maximum speed (variable nMax rpm noted in the 'optional parameters' section). To correlate requested speeds to an absolute speed in rpm units, follow the calculation below:

$$\mathit{set\ value}\,[1/\min] \;=\; \frac{\mathit{Data\ bytes}}{64000}\cdot nMax\,[1/\min]$$

nMax [rpm] can be read as per its description in section 5.1 – Maximum speed.

Motor status

Register Type: Input Register (read only)
Address : D011
Action Required: Read appropriate bits in the register address to determine fan fault status

The motor status register specifies errors currently detected in the fan.

Encoding:

MSB	0	UeHigh	UeLow	UzLow	UzHigh	0	TFEI	0
LSB	BLK	HLL	TFM	FB	SKF	TFE	0	PHA

Error detection for the External Modbus controller may be set to either register general errors (the FB bit notes general error status) or can provide more specific error codes if other bits are decoded in the controller software. The errors corresponding to the bits noted in the table above are as follows:

UeHigh	Mains overvoltage
UeLow	Mains undervoltage
UzLow	DC-link undervoltage
UzHigh	DC-link overvoltage
TFEI	Electronics interior overheated
BLK	Locked motor
HLL	Hall sensor error
TFM	Motor overheated
FB	Fan bad (general error) * See note below.
SKF	Communication error between master controller and slave controller
TFE	Power mod overheated
PHA	Phase failure

Note: Fan Bad is a general error flag and is active whenever any other error bit is set.

Actual Fan speed

Register Type: Input Register (read only)
Address : D010
Action Required: Read register address to determine current fan speed in %

Encoding:
The actual speed data is in the form of a percentage value where 64000 represents 100% speed. 100% speed corresponds to the maximum speed (variable nMax rpm noted in the ‘optional parameters’ section). Therefore, if percentage speed for a data readout is not sufficient, actual operating speeds in rpm can be displayed as per the calculation below:

$$\mathit{Actual\ speed}\ [1/\min] \quad = \quad \frac{\mathit{Data\ bytes}}{64000} \cdot \mathit{nMax}\ [1/\min]$$

nMax [rpm] can be read as per its description in section 5.1 – Maximum speed.

Note that if the actual speed of the fan unit exceeds the value of " nMax ", the display will be limited to a value of "1.02 * nMax.

4. Registers for additional functionality

Maximal speed

Register Type: Holding Register
Address : D119
Actions Possible : Read to determine actual speed values for the fan or limit maximum fan speed

NOTE: This variable is stored in the fan controller's EPROM, and should not be written to continuously! It should only be written during initial commissioning.

All parameters with speed specifications (set values, actual values) are related to this value.
The value 64,000 in these speed specifications is equivalent to the maximum speed value specified here.

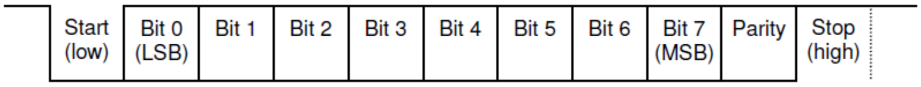
5. MODBUS Communications General and Protocol Information

The MODBUS RTU protocol is used for data transmission between the external controller and fan. Data flow should be controlled by the external controller as the master, with all fan units responding as slave devices. It is important that no slave devices share the same address for this reason. MODBUS-ASCII cannot be supported.

IMPORTANT NOTE: MODBUS-RTU communication with these fan units has a limitation of all communication responses being a maximum of 17 bytes in length. See the item ‘Read Holding Register’ for more information.

Structure of a byte

According to the MODBUS over Serial Line Specification & Implementation guide V1.0. a byte has the following structure:



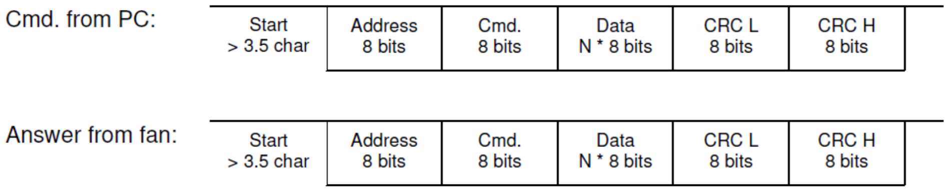
Start (low)	Bit 0 (LSB)	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7 (MSB)	Parity	Stop (high)

The definition of the parity bit ("Even", "Odd", "None") can be set with the parameter "Parity configuration"

The transmission rate is variable and by default is 19,200 bps, with parity settings of 8 Data Bits, Even Parity and 1 stop bit. (8E1)

Communications process

The MODBUS over Serial Line Specification & Implementation guide V1.0 defines the following framework for the transmission protocol:



	Start	Address	Cmd.	Data	CRC L	CRC H
Cmd. from PC:	> 3.5 char	8 bits	8 bits	N * 8 bits	8 bits	8 bits
Answer from fan:	> 3.5 char	8 bits	8 bits	N * 8 bits	8 bits	8 bits

In contrast to the general specifications, the maximum telegram length is 17 bytes!

Commands from External Controller

Initial synchronisation:

A transmission pause of at least 3.5 bytes is used for initial synchronisation.
The following byte is then interpreted as the first byte of a frame (i.e. address).
The pause between the individual bytes of a frame may be a maximum of 1.5 bytes.

Address:

The address field has a size of 8 bits.
The address values 1..247 are permissible
The address 0 is reserved for broadcast commands (i.e. commands to all fans in the network).

Command:

The following commands from the "MODBUS Application Protocol Specification V1.1" general specifications are supported:

Code	Command
0x03	Read holding register
0x04	Read input register
0x06	Write single register
0x08	Diagnostics
0x10	Write multiple register

Other commands are not supported.

Depending on the command concerned, the length of the device’s response and the meaning of the data bits will differ. Please refer to the sections documented in ‘command parameters’.

CRC L / CRC H

A CRC checksum is defined via the complete telegram.
The polynomial for defining the checksum is 1 + x2 + x15 + x16 (i.e. XOR link to 0xA001).
The initial value is 0xFFFF.

The low byte of checksum is transmitted first, then the high byte.

More detailed information about calculating the checksum can be found in the "MODBUS over Serial Line Specification & Implementation guide V1.0".

Responses from the fan unit

A fan will only answer if

- it receives a message through its own address. No answer will be sent to a broadcast address.
- the telegram length is at most 17 bytes.
- the correct number of data bytes have been sent so that the telegram can be interpreted.
- the checksum has been correctly recognised.

Initial synchronisation:
After the command from the PC has been completed, the fan will wait for at least one transmission pause of 3.5 bytes. Depending on the command and on the processing time, the pause may be much longer (until the fan has received all the data it has requested)

Address:
The address is repeated by the command from the PC (i.e. its own fan address)

Command:
If the command can be processed, the command code will be repeated.
If the command cannot be processed, the fan will answer with an exception.

Depending on the command concerned, the length of the device's response and the meaning of the data bits will differ. Please refer to the sections documented in 'command parameters'.

CRC L / CRC H
A CRC checksum is defined via the complete telegram.
The way the checksum is defined is no different from the procedure described above for the command from the external controller.

Read holding register

Command code: 0x03
This command is used to write the content of a number of holding registers. Holding registers are parameters that can be both read- and write-accessed

Command from PC:
4 data bytes are transmitted:

- 1st holding register MSB address
- 1st holding register LSB address
- Number of MSB addresses to be read
- Number of LSB addresses to be read

The description of the holding registers can be found at a later point.

Answer from fan:

The following data bytes are transmitted:

- Byte count (number of addresses to be read * 2)
- Data in 1st holding register MSB
- Data in 1st holding register LSB

Optional:

- Data from the following holding registers (0..n)

Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

0x02: Permissible range of the holding registers 0xD000 to 0xD37F exceeded

0x03: Maximum telegram length for answer (17 bytes) exceeded i.e. either more than 6 holding registers or 0 holding registers were requested.

0x04: A holding register cannot be read due to a defect in the electronics

Read input register

Command code: 0x04

This command is used to write the content of a number of input registers.

Input registers are parameters that only have read access

Command from PC:

4 data bytes are transmitted:

- 1st input register MSB address
- 1st input register LSB address
- Number of MSB addresses to be read
- Number of LSB addresses to be read

The description of the input registers can be found in the preceding sections.

Answer from fan:

The following data bytes are transmitted:

- Byte count (number of addresses to be read * 2)
- Data in 1st holding register MSB
- Data in 1st holding register LSB

Optional:

- Data from the following input registers (0..n)

Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

0x02: Permissible range of the input registers 0xD000 to 0xD01F exceeded

0x03: Maximum telegram length for answer (17 bytes) exceeded i.e. either more than 6 input registers or 0 input registers were requested.

Write single register

Command code: 0x06

This command is used to describe the content of one holding register.

Command from PC:

4 data bytes are transmitted:

- Holding register MSB address
- Holding register LSB address
- MSB data to be written
- LSB data to be written

The description of the holding registers can be found at a later point.

Answer from fan:

4 data bytes are transmitted:

- Holding register MSB address
- Holding register LSB address
- MSB data to be written
- LSB data to be written

Exception codes:

In case of error, only one data byte (the exception code) will be transmitted
Exception codes:
0x02: Permissible range of the holding registers 0xD000 to 0xD37F exceeded
0x04: - The holding register cannot be written due to a defect in the electronics
- There is no write protection in this authorisation level (password).

Diagnostics

Command code: 0x08

This command is used to check the Modbus function

Command from PC:

The following data bytes are transmitted:

- MSB subfunction code
- LSB subfunction code
- 1 - 11 data bytes

Only subfunction code 0000 is supported!

Answer from fan:

The following data bytes are transmitted:

- MSB subfunction code
- LSB subfunction code
- 1 - 11 data bytes

Exception codes:
In case of error, only one data byte (the exception code) will be transmitted
Exception codes:
0x01: Subfunction code not supported (? 0000)

Write multiple register

Command code: 0x10
This command is used to write the content of several holding registers.

Command from PC:
The following data bytes are transmitted:
- Holding register MSB address
- Holding register LSB address
- Number of MSB addresses to be written
- Number of LSB addresses to be written
- Byte count (number of addresses to be written * 2)
- Data to be written in 1st MSB holding register
- Data to be written in 1st LSB holding register

Optional:
- Data to be written to the following holding registers (0..n)

The description of the holding registers can be found at a later point.

Answer from fan:
4 data bytes are transmitted:
- Holding register MSB address
- Holding register LSB address
- Number of MSB addresses to be written
- Number of LSB addresses to be written

Exception codes:
In case of error, only one data byte (the exception code) will be transmitted
Exception codes:
0x02: Permissible range of the holding registers 0xD000 to 0xD37F exceeded
0x03: One of the following three is possible
  - The number of maximum possible registers has been exceeded i.e. either more than 123 holding register data or 0 holding register data were defined.
  - Byte count ? 2 * number of registers
  - Number of data bytes ? byte count

0x04: One of the following two possibilities:
  - The holding register cannot be written due to a defect in the electronics
  - There is no write protection in this authorisation level (password).

Other commands

All other commands are not supported.
An invalid command is always answered with exception code 0x01.