# MODBUS PROTOCOL
# SMART3H-FM LITE

## Addendum

## Manual

MT5001 rev.1 08/11/2023©
Sensitron S.r.l., Viale della Repubblica 48, Cornaredo (MI)
sales@sensitron.it
TEL. +390293548155 FAX. +390293548089

# 1.1 Introduction

The MODBUS standard defines an application layer messaging protocol, positioned at level 7 of the OSI model that provides “client/server” communications between devices connected on different types of buses or networks. Ii standardizes also a specific protocol on serial line to exchange MODBUS request between a master and one or several slaves. The objective of this document is to present the MODBUS protocol over serial line, in order to be used by all system designers when they want to implement MODBUS protocol their serial line products. Thus, this document will facilitate interoperability between devices using the MODBUS protocol. MODBUS is a protocol request/response type and offers services specified by function codes. A sub-set of the function codes has been implemented which is provided by the MODBUS. Also some memory locations assume certain functions in if you have to write or read them, allowing you to be able to access the values of registers and variables of the equipment, as well as enable specific commands such as ZERO CALIBRATION and SPAN CALIBRATION commands. Also, it’s possible to access directly to the input/output discrete provided by the detector. All this allows the detector SMART3H-FM LITE to communicate with any equipment that has the MODBUS RTU protocol on board such as gas Detection Control Panel Systems, Computer or PLC. Once understood the mechanism of how each memory location reachable through the protocol, assume a particular meaning, becomes trivial using the standard function code of the MODBUS to interact with the SMART3H-FM LITE detector. MODBUS protocol on serial line, exists in two typologies: MODBUS ASCII and MODBUS RTU. ASCII mode foresees that all characters that carry information between DEVICES, should be converted to ASCII characters so as to leave control characters to establish the beginning and end of a frame. This implies a remarkable increase of bytes that must be transmitted from a device to another. To overcome this has been introduced MODBUS RTU.  MODBUS RTU is a binary protocol in which all 256 values of byte carry information. The beginning and end of the frame take place by detecting the timing of pauses between one frame and another and between one character and the other. If it encounters a pause of 3.5 times the transmission time of a character on the serial line, it means that the frame is terminated and then you can proceed to its analysis. The slave response occurs after the interpretation of the frame received, however always after a break of at least 3.5 characters. If it encounters a pause of 1.5 characters between a character and the other instead, the current message is discarded and start again to receive a new message. In this way all data can be transmitted without undergoing conversion to ASCIII and therefore the numbers of bytes for each frame is considerably reduced and communication is faster. It’s for this reason that MODBUS RTU protocol has been chosen to be implemented in the SMART3H-FM LITE. The ASCII protocol, although it is required by the specification, has not been implemented due to problems with internal resources to the equipment. It is beyond the scope of this manual to explain exhaustively the protocol it self. Those who wish to deepen their knowledge of the said protocol, can download from the MODICON site, specific of PI-MBUS-300 RevJ which are the standard of the protocol. There is also a website www.modbus.org in which are discussed issues related to the MODBUS protocol and there is also a useful list of interesting links to MODBUS resources.

## 1.2 Communication parameters

On SMART3H-FM LITE detectors are implemented two types of communication protocols:

I The first is the “GALILEO HIGH SECURITY COMMUNICATION PROTOCOL”, a proprietary protocol developed by software department to meet the SIL3 requirements when the detector is connected to Central Unit Galileo Multisystem.

II The second one is the MODBUS protocol in RTU data format, described in this document.

SMART3H-FM LITE detector automatically discriminates between MODBUS Protocol and GALILEO HIGH SECURITY Protocol. SMART3H-FM LITE detector communicates through an RS485 serial port, in Half Duplex. For both protocols the communication parameters are:

I Baud Rate: 19200 bps Bit per second.
II Parity: None.
III Start Bit: None
IV Data Bits: 8
V top Bits: 1

SMART3H-FM LITE detector is the slave device in the communication and always responds after a query sent by the master device only. The slave address is settable by a specific MODBUS command, writing the new address on MBAddress (RS485 Address) register. The address number 0 in MODBUS protocol identify a broadcast address and can’t be assigned.

## 1.3 Memory Definition on SMART3H-FM LITE Detector

Four different MODBUS Memory data areas are defined inside SMART3H-FM LITE Detectors. Every Memory data areas has a specific meaning and can be reached by specific function code.

| Areas Nome | Area # | Size Type | Access Type | Comments |
|---|---|---|---|---|
| Coils | 0 | Single Bit | Read-Write | Reserved. |
| Discrete Inputs | 1 | Single Bit | Read | Reserved. |
| Input Registers | 3 | 16-bit-word | Read | These data are read only. |
| Holding Registers | 4 | 16-bit-word | Read | These data are read only. |

The four data areas can be overlapped, the separation is not mandatory. In the SMART3H-FM LITE Detector in these areas are mapped the working registers, the user variables and the control commands.
USING MEMORIES AREAS
- Coils Memory Area are mapped all registers with the operating mode specific of discrete outputs.

These Command has no input parameters and don’t reply any value to the sender.

Note:
- Addressing the specific COIL to start-up.
- It’s not mandatory to set / reset the Coil.

Reading the address specific to these instruction the return code is Zero if detector is in normal state and is NOT Zero if the detector is a different state, for example: WARMUP, ERROR, SETUP MODE, CALIBRATION, and so on.

I Discrete Inputs  Memory Area is not used.

II Input Register  In this Memory Area are mapped all registers with the operating mode specific to the Input Register Commands. The MODBUS standard define 16 bits long registers, instead SMART3H-FM LITE detector has 16 bits and 32 bits registers and variables, then the access in these area must be done always reading 2 (two) adjacent registers; for the 32 bits float variables a 32 bit.

III Holding Register In this Memory Area are mapped all registers with operating mode specific to the configuration and status. Registers in MODBUS standard are 16 bits long, instead SMART3H-FM LITE detector has registers and variables 16 bits or 32 bits long, then the access in this Memory Area must be done reading two adjacent registers. Accessing to variables or to floating point registers, can be done with two 16 bits separated transmission, although we do not recommend the operation. Please take into consideration the following things:

1) In reading sequence the data could be change while the first communication. If the reading sequence low-word high-word is not correctly completed or if the data to be read change between the two communications it is possible to get erroneous from data. To avoid false reading use the function code 03h to read two registers.

2) In writing sequence the first word to write is the low-word of the 32 bits (registers or variables) then write the high-word. If the writing sequence low-word high-word is not correctly completed it is possible to write erroneous data. To avoid false writing use the function code 10h to write two registers.

DATA TYPES (SEE MODBUS DOCUMENTATION)
Serial communication is done with transmission of 16bits binary words.

Date types are the following:

I Floating point  [ float ] (  2 words ).,

II Integer [ int16 ] ( 1 word ).

III String  ( array of n byte ( 8 bits )  ).

The FREESCALE Microprocessor, implemented in SMART3H-FM LITE Detectors, is the MK22FN128VLH10 and has an internal memory organization in BIG ENDIAN format.

Big-Endian and Little-Endian are two different methods used by computers to store in the memory data of larger size than the byte (for example word, dword, qword).

The difference between the two systems is given by the order in which the byte constituting the data to be stored are memorized:

Big-endian is the memorizing that start from the most significant byte to end with the lower significant one; it’s used by Motorola processor and by protocol used in Internet.

Little-endian is the memorizing that start from the lower significant byte to end with the most significant one it's used by Intel processors.

This differentiation does not concern the bit position inside the byte (in which case we speck of bit order)  or the position of the characters in a string. Instead it's important in the interpretation (or decoding)  of the multi-byte encoding of string of characters (such as: encoding UTF-16 of the standard unicode).

The big-endian order, which has been chosen as the standard order in several standard protocols used on the internet, is therefore also called network byte.

In the case of a WORD (16 bits), the hexadecimal number 0x0123 will be stored as:

| | Little Endian | | Big Endian | |
|---|---|---|---|---|
| | 0x23 | 0x01 | 0x01 | 0x23 |
| BYTE: | 0 | 1 | 0 | 1 |

In the case of a DWORD (32 bits), the hexadecimal number 0x01234567 will be stored as:

| | Little Endian | | | | Big Endian | | | |
|---|---|---|---|---|---|---|---|---|
| | 0x67 | 0x45 | 0x23 | 0x01 | 0x01 | 0x23 | 0x45 | 0x67 |
| BYTE: | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |

(In the above examples value 0x01 is the most significant byte)

In case of FLOATING POINT (32 bits)
The values of floating point follow the specific IEEE 754 with 32 bits floating point standard.

| MSB | LSB |
|---|---|
| SEEEEEEEE | MMMMMMMM |
| EMMMMMMM | MMMMMMMM |
| WORD A | WORD B |

S: sign of the number 0 positive 1 negative
E: exponent at 8 bit
M: mantissa of the number 23 bit

Example reading values in floating point:
To read 3 Holding register expressed in float point starting from the register MODBUS 0x4000, 6 word will have to be asked, then the registers 0x4000 – 0x4005.

**Analyzer 1**

**Input Format**

auto  decimal  binary  hexadecimal

Hexadecimal is little-endian (right-to-left)

**Rounding Mode**

Round to the Nearest Value  Round toward Zero
Round toward Positive Infinity  Round toward Negative Infinity

**Value to analyze:**

20.00

**Syntax Entered:** Decimal (Real number)

**Decimal value:** 20.00

**Normalized binary value:** 1.0100b4

**Binary32:** 41A00000

| Status | Sign [1] | Exponent [8] | Significand [23] |
|---|---|---|---|
| Normal | 0 (+) | 10000011 (+4) | 1.01000000000000000000000 (1.25) |

**Binary64:** 4034000000000000

| Status | Sign [1] | Exponent [11] | Significand [52] |
|---|---|---|---|
| Normal | 0 (+) | 10000000011 (+4) | 1.0100000000000000000000000000000000000000000000000( |

**Binary128:** 40034000000000000000000000000000

| Status | Sign [1] | Exponent [15] | Significand [112] |
|---|---|---|---|
| Normal | 0 (+) | 100000000000011 (+4) | 1.0100000000000000000000000000000000000000000000000( |

Close This Analyzer

## 1.4 Read input registers

In the following table are listed the MODBUS INPUT REGISTERS with their address and with their function in SMART3H-FM LITE detector.

| Modbus Address | Data Length | R/W | Register's Name | Function Description | Measuring Unit |
|---|---|---|---|---|---|
| 0x0006<br>0x0007 | Input reg.<br>float 32bit | r | Sensor[0].<br>Temperature | Temperature Sensor #0 | Kelvin |
| 0x0008<br>0x0009 | Input reg.<br>float 32bit | r | Sensor[0].<br>Concentration | Target Gas Concentration Sensor #0. | Current measuring unit ppm, %LEL or %VOL |
| 0x0016<br>0x0017 | Input reg.<br>float 32bit | r | Sensor[0].<br>Warning | Warning Code on Sensor #0 | See Table |
| 0x0018<br>0x0019 | Input reg.<br>float 32bit | r | Sensor[0].<br>Status | Status Code on Sensor #0 | See Table |
| 0x001A<br>0x001B | Input reg.<br>float 32bit | r | Sensor[0].<br>Error | Error Code on Sensor #0 | See Table |
| 0x005A<br>0x005B | Input reg.<br>float 32bit | r | Detector.<br>Error | Detector.Error | See Table |

## 1.5 Warning code list

| | |
|---|---|
| NO_WARNING | All bits in 0 state |
| WARMUP_WARNING | If BIT 0 in 1 state |
| INVALID_TEMPERATURE_WARNING | If BIT 1 in 1 state |
| INVALID_ACTIVE_WARNING | If BIT 2 in 1 state |
| INVALID_REFERENCE_WARNING | If BIT 3 in 1 state |
| INVALID_PEAK_WARNING | If BIT 4 in 1 state |
| INVALID_DATA_WARNING | If BIT 5 in 1 state |
| CALIBRATION_MODE_ACTIVE | If BIT 6 in 1 state |
| CALIBRATION_MODE_FAULT | If BIT 7 in 1 state |
| MISMATCH_HYBRID_VALUE_WARNING | If BIT 8 in 1 state |
| SENSOR_HW_TEST_WARNING | If BIT 9 in 1 state |
| SENSOR_END_OF_LIFE_WARNING | If BIT 10 in 1 state |
| SENSOR_MANTEINANCE_WARNING | If BIT 11 in 1 state |
| SENSOR_HW_TEST_TIMEOUT_WARNING | If BIT 12 in 1 state |
| SENSOR_HW_TEST_CAPA_WARNING | If BIT 13 in 1 state |
| INVALID_ZERO_WARNING | If BIT 16 in 1 state |
| INVALID_SENSITIVITY_WARNING | If BIT 17 in 1 state |

# 1.6 Status code list

| | |
|---|---|
| NO ALARM | No Alarm, Detector OK. All bits in 0 state. |
| FAULTLOW | Bit 0 in 1 state1 Concentration <<br>-10% of Full Scale, or Hardware Error. |
| UNDERSCALE | Bit 1 in 1 state Reading < 0% of Full Scale |
| THRESHOLD1 | Bit 2 in 1 state Threshold #1 of Alarm reached |
| THRESHOLD2 | Bit 3 in 1 state Threshold #2 of Alarm reached |
| THRESHOLD3 | Bit 4 in 1 state Threshold #3 of Alarm reached |
| OVERSCALE | Bit 5 in 1 state Reading > 100% of Full Scale |
| FAULTHIGH | Bit 6 in 1 state Reading > 110% of Full Scale |

# 1.7 Error Code List

| | | |
|---|---|---|
| NO_ERROR | 0 | No Error, Detector OK |
| WARMUP_ERROR | 1 | Detector is in Warm-up phase |
| DETECTOR_FRAM_ERROR | 2 | Fram Memory on Detetcor Base Board Error |
| FLASH_ERROR | 3 | Internal FLASH EPROM Error |
| RAM_ERROR | 4 | Internal RAM Error |
| VIN_ERROR | 5 | External Power Supply Voltage Error.<br>Voltage is under 10V or above 30V |
| I2C_ERROR | 6 | I2C Interface Error |
| WD_ERROR | 7 | External Watch Dog Error |
| ADC_ERROR | 8 | Internal ADC Error |
| VDD_ERROR | 9 | Internal 3.3V Power Supply Voltage Error.<br>Voltage 3.3V +/- 5% Out of range |
| HEAD_FRAM_ERROR | 10 | Fram Memory on Sensor Head Error |
| VREF_ERROR | 11 | Internal Voltage Reference Error |
| ANALOG_420MA_ERROR | 12 | Analog Output 4-20mA Error |
| ANALOG_OUTPUT_TYPE_ERROR | 13 | Number of analog outputs Set-up |
| AVERAGE_VALUE_ERROR | 14 | Analog Input Out of Range |
| DETECTOR_CONFIGURATION_ERROR | 15 | Invalid Configuration Error |
| HEAD_MISMATCH_ERROR | 16 | Invalid Head Type for the Detector |
| HEAD_UNCONFIGURED_ERROR | 17 | Head Unconfigured Error |
| HEAD_MISSING_ERROR | 18 | Head Not Recognized By Detector Base |
| HEAD_PLACEMENT_ERROR | 19 | Head Not Correctly Installed |
| HEAD_PAIRING_ERROR | 20 | Head Not Paired On the Detector |
| UART0_BAUD_RATE_ERROR | 21 | Invalid Baud Rate Setting for UART0 |
| FIRMWARE_ERROR | 22 | Invalid Firmware for the Detector Base |
| HEADS_MAINTENANCE_ERROR | 23 | Maintenance on Heads not allowed |
| HEADS_MAINTENANCE | 24 | Maintenance on Heads allowed |
| RESERVED | 25 | Reserved Error Code |
| SENSOR_END_OF_LIFE_ERROR | 26 | Timeout Expired on Sensor End Of Life |
| SENSOR_MAINTENANCE_ERROR | 27 | Timeout Expired on Periodic Sensor EOL |
| ANALOG_EXT_420MA_ERROR | 42 | Error on 4-20mA feedback |
| KEY_ERROR | 43 | Magnetic Key locked |

# 1.8 Read holding registers

In the following table are listed the MODBUS HOLDING REGISTERS with their address and with their function in SMART3H-FM LITE detector.

| Modbus Address | Data Length | R/W | Register's Name | Function Description | Measuring Unit |
|---|---|---|---|---|---|
| 0x01CE<br>0x01CF | Holding reg. (32bit) | r | FwMajor | Firmware Revision Major Number | Number MSWORD – LSWORD |
| 0x01D0<br>0x01D1 | Holding reg. (32bit) | r | FwMinor | Firmware Revision Minor Number | Number MSWORD – LSWORD |
| 0x01D2<br>0x01D3 | Holding reg. (32bit) | r | FwRev | Firmware Revision | Number MSWORD – LSWORD |
| 0x01D4<br>0x01D5 | Holding reg. (32bit) | r | FwDD | Firmware Revision Date ( DD ) | Number MSWORD – LSWORD |
| 0x01D6<br>0x01D7 | Holding reg. (32bit) | r | FwMM | Firmware Revision Date ( MM ) | Number MSWORD – LSWORD |
| 0x01D8<br>0x01D9 | Holding reg. (32bit) | r | FwYYYY | Firmware Revision Date ( YYYY ) | Number MSWORD – LSWORD |
| 0x01F6<br>0x01FD | Holding reg. String | | HardwareModel | String of ASCII Chars for Detector Model | ASCII String 16char MAX |

# Contatti/Contacts

Viale della Repubblica, 48
20007 Cornaredo (MI) – ITALY

+39 0293548155
sales@sensitron.it
sensitron.it

Sensitron S.r.l.