EvominiSER

Transmitter with Modbus RTU output signal and I²C interface for digital sensors

USER MANUAL

Summary

Read and carefully follow the instructions given in this manual

1 General safety rules

Before performing any operations on the device, read carefully the following recommendations:

¤ Before hooking up the device, read the installation instructions given in this manual.
¤ To wire up the device, a proper wiring size must be chosen according to voltage and current specifications found in technical data.
¤ The device is not equipped with an ON/OFF switch. Therefore, it powers up immediately as soon as power is supplied to it.
¤ The device must be powered with DC voltage greater than 9V and lower than 32V. Higher voltages will cause device failure while lower voltages will not allow the device to work properly.
¤ Make sure that the working environment conditions falls within the range specified in chapter 6 'Technical data'
¤ The device is not designed for operation in hazardous atmospheres (i.e., flammable or explosive): its use in such conditions is therefore **prohibited.**
¤ The device is intended for industrial use only. Do not use the device in situations where compliance with strict safety precautions is required, such as applications directly or indirectly correlated with medical equipment.
¤ The device must not be disassembled or repaired by unauthorised people. Contact your local dealer for any repairs.

1.1 Recommendations for workers

When performing their tasks, workers should observe the following recommendations:
¤ Workers with specific qualifications for assigned tasks are required
¤ Workers must be authorised by the plant's owner/operator
¤ Workers must be familiar with federal/national regulations
¤ Before doing any action, workers must read and understand the instructions given in the manual and the supplementary documentation (if available and depending on the application)
¤ Workers must comply with the instructions and basic conditions

1.2 Typical applications

Typical applications include HVAC systems (heating, ventilation, and air conditioning), agriculture, incubators, cold rooms, standard measurements, seasoning rooms etc.
¤ Only use this device in full compliance with the general conditions listed in this manual and, if applicable, in the additional documentation.

Improper use:
Italcoppie Sensori s.r.l. is not liable in any way for damage or injury caused by tampering, incorrect or improper use of the device.

1.3 Safety rules

Refer to the relevant regulations.

2 Introduction

The EvominiSER is a programmable digital transmitter with Modbus RTU output. It can be connected to different sensors in order to detect different data, such as temperature, humidity, pressure, light etc.
By using a specific cable, several EvominiSER can be connected in series. One device only can be a master.
With its small size, low-cost, IP67 degree of protection, easy-cabling and use quality, it is an extremely interesting equipment for several applications in industrial environment (e.g., HVAC systems). Its M12 connectors allow for a fast and easy cabling feature.
Furthermore, by using this transmitter it is possible to implement modular systems. If one or more devices are to be added to the network, it will not be necessary to design again the measurement system. Thus, no added design costs are to be sustained.
The Modbus protocol is a very popular standard for field bus applications due to its reliably and capability of efficiently handling large data flows. With the Modbus implementation, the Evo series can be hooked up directly to the majority of commercially available PLC's and SCADA packages, with the option to connect these modules together with other devices (PLC's, operator panels, CNC equipment, etc...) over a single network.

2.1 Working principle

The EvominiSER block diagram is shown in Figure 1.

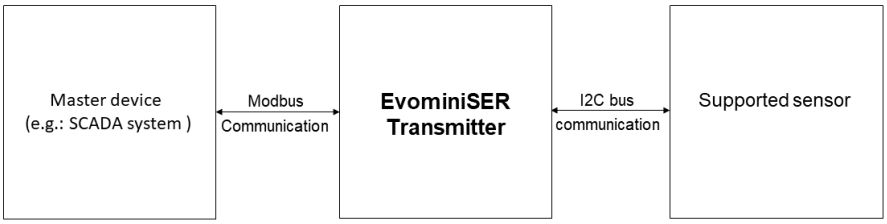


Figure 1: EvominiSER block diagram

The EvominiSER transmitter exchanges data periodically with the connected sensor through the I^2C bus. Data are then processed and output with Modbus RTU protocol form.
A master unit, e.g., a PLC unit or a PC with SCADA (Supervisory Control And Data Acquisition) has to be connected to the network for data monitoring.

3 Installation

WARNING!
This device must be installed by qualified and authorized people.

3.1 Mounting instructions

A picture of the EvominiSER is shown in Figure 2.

Figure 2: EvominiSER

In order to serially connect several transmitters, a specific M12 tee connector can be ordered together with its extension cables (for details refer to chapter 7).
A specific sensor can be connected using an M12 connector by screwing it to the metallic threaded ring. The connection is IP67.

3.2 Operating conditions

Installing ambient temperature: (-40÷ +80) °C

Installing ambient relative humidity: (0 ÷ 100) %

Protection degree: IP67 according to IEC 60529

3.3 Electric connections

The EvominiSER comes with two 4-pole M12 connectors. A series connection of transmitters can be arranged by using an overmolded extension cable with a 4-pole M12 female connector. Tee connectors are available to extend the network. This solution allows for an IP65/IP67 connection. It is possible to use other commercially available female M12 connectors. In such a case, the degree of protection depends upon the cabling system.

WARNING!
The electric connections shall be executed as described in the following chapters. Cabling errors might break either the device or the entire devices' network.

3.3.1 Power supply requirements

¤ Device power supply: typical $24V_{DC}$, or within $(9 \div 32)V_{DC}$ range.
¤ Use a power supply with a $24V_{DC}$ output, according to ELV regulations.
¤ Maximum current consumption: 10mA (at power-on for about 8 ms).
¤ Average current consumption: about 4mA.

On the power line, insert an insulation switch together with an appropriate fuse. The voltage supply must be continuous, isolated and with low ripple.

3.3.2 Pinout and connections

The pinout of the output connector is shown in Figure 3.

Figure 3: M12 output connector pinout

WARNING!
- *The device is protected against polarity inversion.*
- *In case of incorrect electrical wiring, the device is protected only for voltages less than $25V_{DC}$. A misconnection with higher voltages will cause a device failure.*

The EvominiSER comes with half-duplex EIA RS-485 serial interface. The cabling is simplified by the M12 connectors[1].

[1] 4-pin M12x1 male connector (according to IEC 61076-2-101)

WARNING!
The device serial interface is not Galvanically isolated. If the master unit is equipped with RS232 serial interface, an isolated transceiver should be used. If the master unit is equipped with RS485 serial interface, a converter with Galvanic isolator should be used. As an option, a RS485 repeater/isolator can be ordered (cod. EVO005).

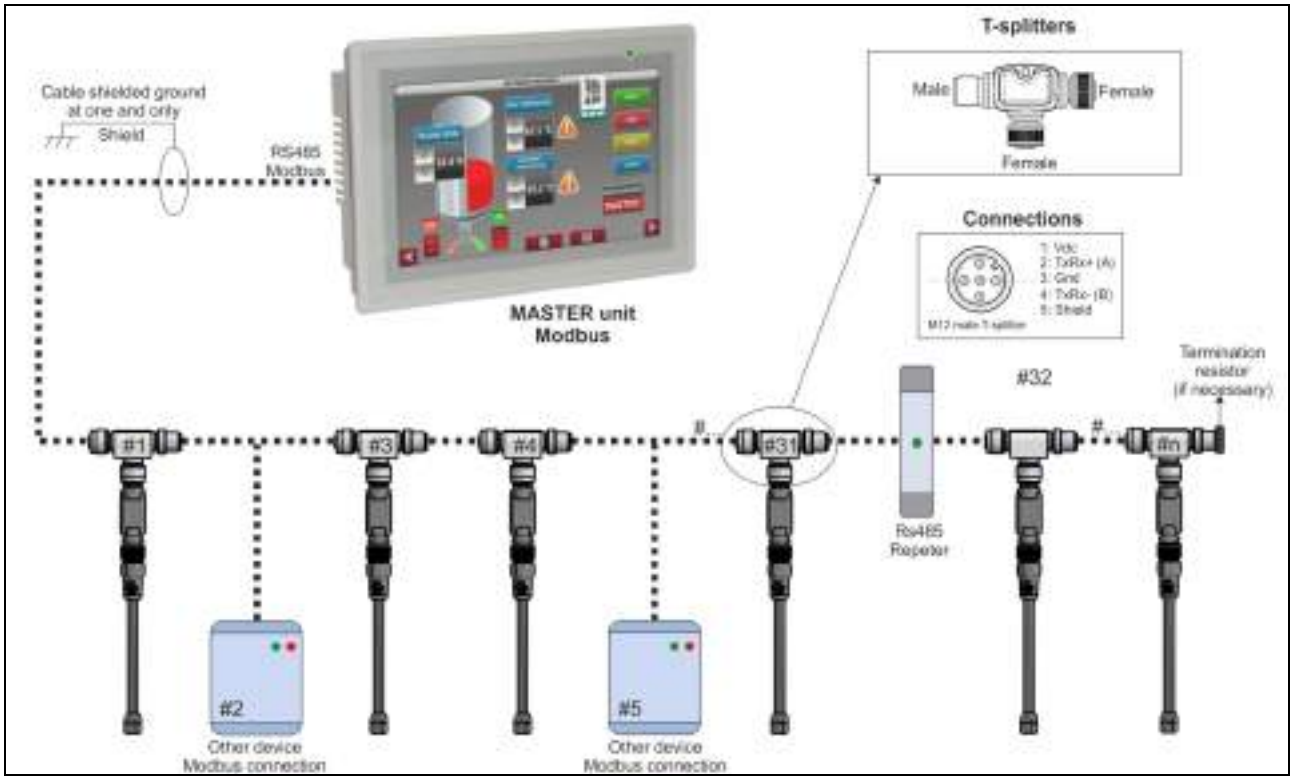


Figure 4: application example

3.3.3 Cabling good practices

The RS485 (EIA-485) is a standard defining the electrical characteristics of drivers and receivers for use in serial communications systems. Electrical signal is balanced, and multipoint systems are supported. Ideally, the two ends of the cable will have a termination resistor connected across the two wires. Without termination resistors, signal reflections off the unterminated end of the cable can cause data corruption. Termination resistor also reduce electrical noise sensitivity due to the lower impedance. The value of each termination resistor should be equal to the cable characteristic impedance (typically, 120Ω for twisted pairs). The master is usually already equipped with the termination resistance. The maximum length is not defined as depends upon the data rate, the signal-to-noise ratio and the cable quality.
There are some good practices that allows for external interference reduction. Cables that carry higher currents to power devices should be kept as far as possible from cables that carry communication signals. Cables reserved for communication signals must be kept as far as possible from power switching devices, such as contactors and relays, as well as from electric motors and generators. Choosing a cable suited for the particular application is also considered a good practice. In particular, the lower its capacitance per meter the longer the network can be. Typically, the cable should have a value within the range (50 ÷ 100)pF/m. The RS485 consists of two conductor cables and twisted pair cabling is suggested to ensure that each conductor is equally exposed to any external magnetic fields that could induce

unwanted noise. A metallic electrostatic shielding is also suggested to provide immunity to RF interference.
A star topology network is not recommended. As explained above, a 120Ω termination impedance is required for the two end line termination devices to avoid reflections. Therefore, a very low impedance will result from a star topology network. Thus, no communication can result at a certain point.

3.3.4 Polarity

According to RS485/422 standard, the signal connections are indicated with A and B. However, it might be possible to find different indications on some products, such as HI/LO or +/-. If a device is unable to connect to the network, try inverting A and B and to connect the device again.

3.3.5 Terminating impedance

Signal reflections occur when a signal is transmitted along a transmission medium, such as a copper cable. Some of the signal power may be reflected back to its origin rather than being carried all the way along the cable to the far end. This happens because imperfections in the cable cause impedance mismatches and non-linear changes in the cable characteristics.
The signal reflections can be canceled by adapting the terminal impedance, whenever it is necessary. If the time required for a single bit transmission is, at least, ten times the signal propagation time on the network, the terminating impedance is not necessary.
The value of cable impedance depends on its dielectric material and mechanical dimensions. The velocity of propagation in PE cable is about 2/3 the speed of signal if the dielectric cable was vacuum. Therefore, the propagation time is:

$$t_p = \frac{1}{0.66 * c_0} * l$$

For example, for 9600baud and 1200m of cabling, t_p=6μs. The time required for a single bit transmission is:

$$t_{bit} = \frac{1}{baud(\frac{bit}{s})} = \frac{1}{9600} = 104\mu s$$

Thus, in this example, the terminating impedance is not strictly necessary. However, whenever the terminating impedance is required, a resistor must be inserted in the last node (from the master) between terminal A and B (pin 2 and pin 4 in the M12 connector). Its value must be equal to the cable impedance. In typical applications, the terminating impedance has only resistive components and its typical values are in the range 120Ω to 560Ω, depending on the cable and number of nodes on the network.

3.3.6 Allowed maximum number for series connected transmitters

The maximum number of transmitters that can be connected in series is dependent upon the connection length and cable parameters.
A calculation worksheet that indicates to the user the maximum number of devices that can be connected to an EvominiSER network given the cable cross-section (i.e., AWG) and cable length is available.

For example, with a $24V_{DC}$ voltage supply and a 300m cabling:

- AWG20: up to 60 devices
- AWG22: up to 45 devices
- AWG24: up to 35 devices

The given indications refer to the given power supply. The serial communication needs a repeater every 32 connected device.
If the power supply cabling is longer than 100 meters, it is strongly suggested to use a MLCC (1000pF, 1kV; for example, RDE5C3A102J2M1H03A) connected between GND of the power supply and GND of the shield.

4 Mechanical dimensions

The transmitter mechanical dimensions are shown in Figure 5.

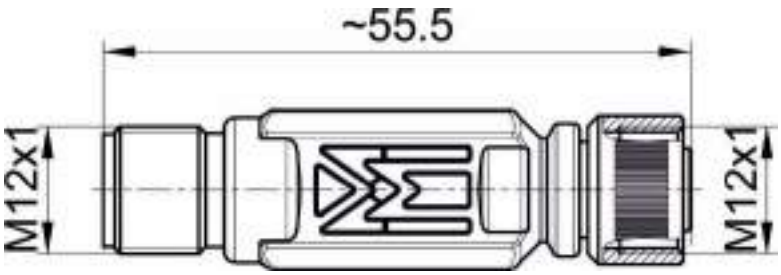


Figure 5: mechanical dimensions

5 Technical data

Power supply:
- (9÷32)V_{DC} (protected agaist polarity inversion)

Current consumption:
- 4mA (working behavior), <10mA (at power-on for about 8ms)

Transmitter operating temperature (with plastic body):
- (-40÷80) °C

Output signal:
- Non-insulated serial interface (EIA RS-485 with RTU Modbus protocol)
- Baud rate settings: 9600bps, 19200bps, 38400bps
- Stop bit: 1 or 2
- Parity bits: none, even or odd
- Network length: up to 1000m[2]
- Max. nr. of devices: 247 (a repeater is necessary every 32 devices)

Connection:
- Sensor: female 4-poles M12 connector (according to IEC 61076-2-101)
- Power supply + serial: 4-poles male M12 connector (according to IEC 61076-2-101)

Body:
- Plastic body
- IP65/IP67 protection degree (according to IEC 60529)

Marked indications:
- PWR blue LED with symbol indication "⏻" for device power status, red LED with **ERR** indication for either memory errors or boot mode operation, white LED with TX and RX indication, serial and batch number.

5.1 LED indicators

	PWR blue LED on: device is working correctly
	PWR blue LED slow blinking (about every half second): no communication between sensor and transmitter.
	PWR blue LED fast blinking (less than half second): sensor short circuit
ERR	ERR LED on: memory error or boot mode activated
Tx	On data transmission white LED on
Rx	On data reception white LED on

[2] The network length is dependent on the characteristics of the cable, on the power supply and on the number of connected devices (refer to chapter 3 for further details).

All LEDs on	Undetected or not compatible sensor at power on (after initialization)

- PWR LED(blue): indicates the EvominiSER power state and working behavior. At power-on, all the LEDs are on for about two seconds. Then, the PWR LED blinks for about 3 seconds. After that, the PWR LED should be permanently on, to indicate that the device power state is on.
  If PWR LED is permanently on, the device has been powered and works properly. When PWR LED blinks with a slow frequency (f~2Hz), a communication error between the transmitter and the sensor has occurred. Please, check that the engagement nut has been properly screwed to the sensor's connector. If the problem has not been solved, contact the technical service.
  When PWR LED blinks rapidly (f>2Hz), a sensor short circuit condition has been detected. The device must be turned off immediately. NOTE: if the device is not turned off, the LED blinking will not stop even if the problem is solved. If by restarting the device the problem has not been solved, contact the technical service.
- ERR LED(red): this LED is normally off.
  When this LED is blinking, a memory error has been detected. Restore the default values (Factory default, refer to the system parameters) to solve the problem.

  If the problem persists, contact technical assistance and remove the device from the network as it could provide incorrect and dangerous data for the functionality of the system
  If it is steady on, the device is in boot mode (ready for firmware update).
- TX & RX LEDs (white): transmission and reception indicators. If active, they blink respectively in case of message transmission and reception over the serial line.
- PWR LED (blue), ERR LED (red) and TX & RX LEDs (white) permanently on after device initialization indicate that the connected sensor is not compatible with the uploaded firmware.
  Please check the indication marked directly on the sensor and download the compatible firmware form www.italcoppie.it. For further information, contact the local dealer.

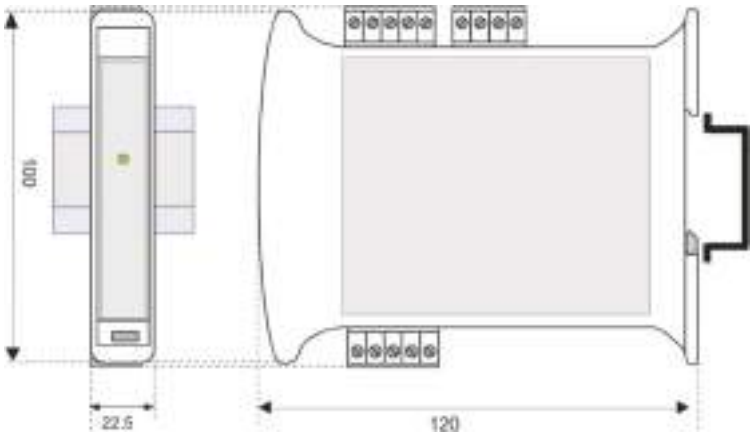
6 Certifications

CE	This device satisfies the UE regulations. The manufacturer certifies that these requirements have been satisfied from CE marking appliance.
EMC	Directive: 2014/30/EU Harmonization laws: EN 61326-1:2013
RoHS2	Directive: 2011/65/EU from 08-06-2011

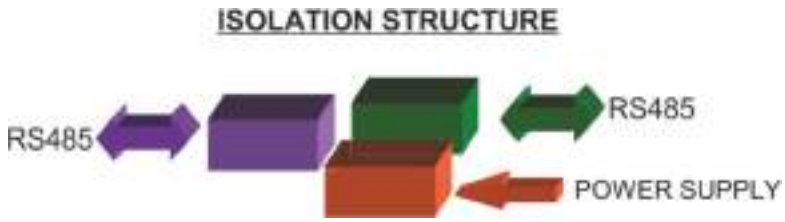
7 Options

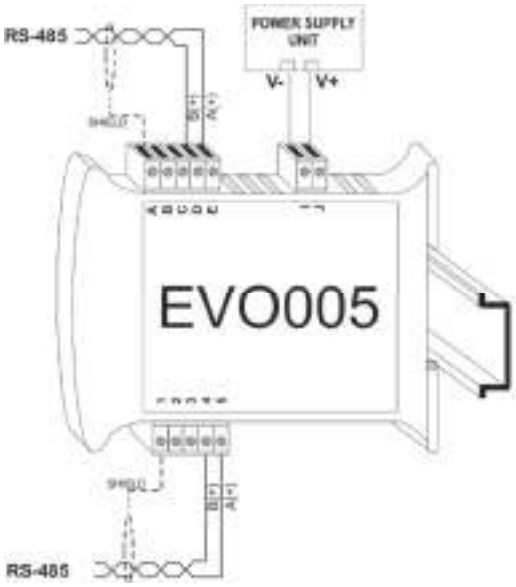
7.1 Isolator / Serial repeater

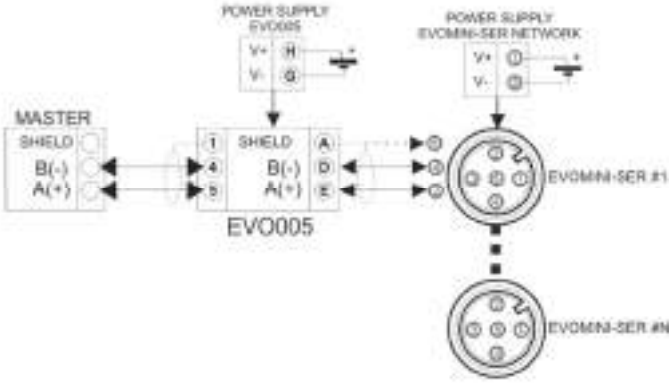
It allows for signal isolation and amplification between the RS485 serial port and the power supply of the EvominiSER. By using this device, more than 32 devices can be connected on the same networks (see Figure 4).



Measurements are in mm







Power supply	(10÷30)V_{DC}
Baud rate	up to 115200 baud
Internal terminator resistance (optional)	120Ω
Isolation: *Power supply/RS485-422* *RS485-422/ RS485-422*	 2000Vac, 50Hz, 1min 2000Vac, 50Hz, 1min
Maximum distance / baud rate ratio (recommended)[3]	1km @ 38400bps
Current consumption	35mA
TX/RX switching time (RS485)	150µs
Operating temperature	(-20 ÷ 60) °C
Storage temperature	(-40 ÷ 85) °C
Relative humidity without condensation	(0 ÷ 90) %
DIN rail compatible	
Automatic baud rate adaptation	
Serial connection mounting and removal contact	
Order code: EVO005	

7.2 Connection cables

A specifically designed optional cable for Modbus serial networks is available as an option. It consists of two twisted cables (used for RS485 serial line) that allows greater noise rejection, and of two power cables with larger diameter to sustain higher currents. The two pairs of cables are shielded using an overall aluminum foil and a drain wire, allowing for an easy connection with M12 connectors. An external PVC jacket is used for cable enclosure.
Cable characteristics are indicated in the following tables.

Conductors	
Conductors nr.	4 + drain wire and shield
Features	1 twisted pair AWG26 [serial signal] (0.14 mm^2) 1 pair AWG22 [power supply] (0.34 mm^2)
Material	Copper
Color	1 twisted pair [serial]: white-blue Pair [power supply]: red-black

Shield foil	
Material	Aluminum with drain wire

Outer jacket	
Material	PUR 92ShA -40÷90°C (static), Halogen free
Color	Matte grey
Diameter	(5.1 ± 0.2)mm

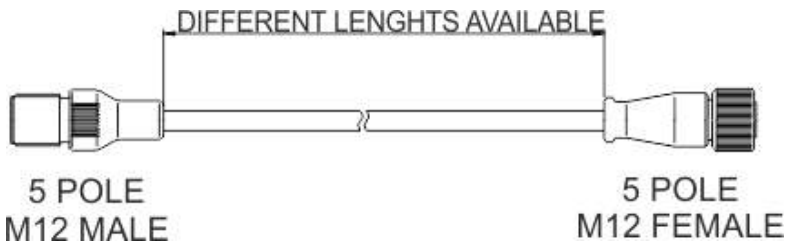
[3] The maximum distance depends on the number of devices connected, type of cabling, noises, etc.

Electrical and mechanical data			
Cables for static application			
Characteristic impedance at 1 MHz		(87 ± 10) Ω	
Maximum capacitance		58pF/m	
Resistance at 20°C		Twisted pair[serial signal]: 154.6Ω/km Pair [Power supply]: 64.2Ω/km	
Minimum bending cable		15 times the cable diameter	
Working temperature		(-40°C÷+90) °C (static)	
Maximum working voltage		300Vrms	
Color code			
V_{IN}	1	red	0.34 mm²
GND	3	black	
-RxTx	2	White	0.14 mm² twisted cable
+RxTx	4	blue	
SHIELD	5	-	-

Order code: CAVV151

7.3 Extension cables with M12 connectors

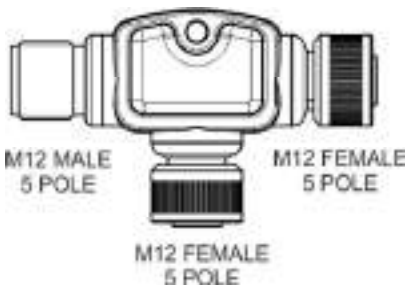
Several extension cables are available with different lengths. The cable is described in chapter 7.2 and has two M12 5-pole connectors overmolded.

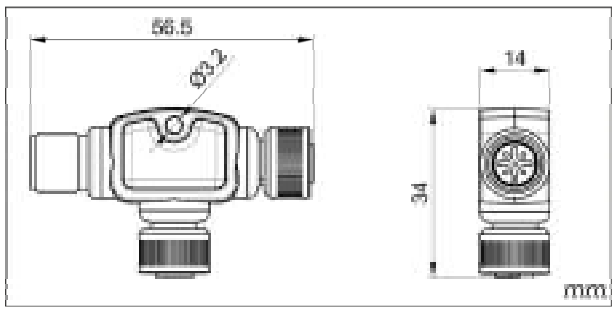


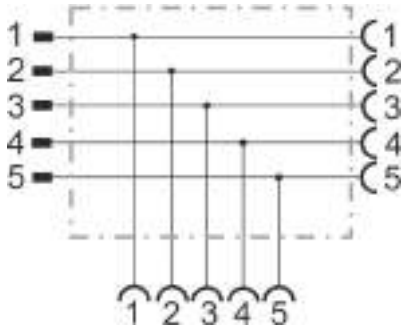
Order code: PRV#...

7.4 Tee connector

Tee connectors are available, with 5-pole M12 connectors. These connectors allow for a simple and fast connection between several devices.





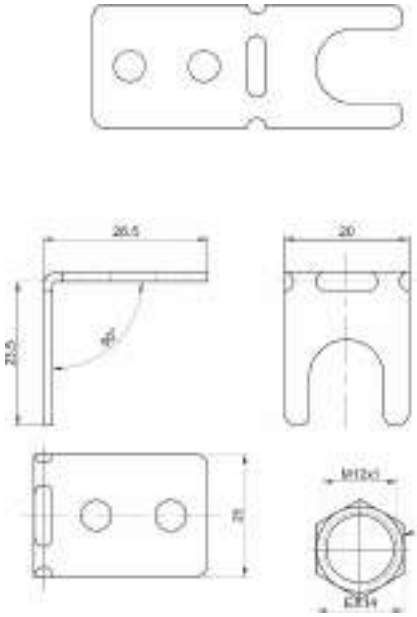


Connections

Ordering code: CONV425

7.5 Wall fixing bracket

The EvominiSER device, together with its probe, can be fixed to the wall with this its bracket.



Ordering code: AMEC028.

7.6 Configuration kit

This kit is useful for programming Evo series devices. The Evoplatform configuring software is downloadable from the website www.italcoppie.it

Order code: EVOPLATFORMSET

8 Order code

EVO032

Digital transmitter with Modbus RTU output signal (RS485 serial interface), digital input for I2C bus sensors, power supply (9÷32)V_{DC}, IP67, operating temperature (40÷80)°C, M12 output male connector and M12 female input connector.

EVO033

Digital temperature sensor (-40÷120)°C and relative humidity sensor (0÷100)%, I2C bus interface, Sensirion SHT31 sensing element, IP67, M12 4-pole male connector, protection filter, stem (6mm diameter and 80mm length).

CONV425

Tee connector with M12 5-pole connector points 1:1 connection, Male / Feale Female

AMEC028

Stainless steel wall fixing bracket for EvominiSER.

EVO005

Isolator and repeater RS485 with baud rate up to 115200bps, galvanically isolated at $200V_{AC}$, power supply $24V_{DC}$ (supports up to 32 devices).

CAVV151

Serial cable, two AWG26 stranded copper conductors (white-blue) Ø1.2 (87Ω impedance) + two Cu-Sn AWG22 stranded tinned copper conductors (red-black) Ø1.3 (64.2Ω/km impedance), twisted, with Al-Mylar shielded drain wire, thermoplastic sheathed jacket PUR 90, gray Øext5.1 ±0.2mm

PRV#...

Extension cables with overmolded M12 5-pole female connectors, CAVV151 cable. Several lengths are available.

EVOPLATFORMSET

Programming kit for the Evo series devices. It contains a USB and serial cables (downloadable from www.Italcoppie.it)

9. Compatible sensors

The EvominiSER transmitter can be connected to different digital sensors. For each sensor, its firmware is downloadable form www.italcoppie.it
The correct firmware has to be installed on the transmitter (see chapter 12 for firmware download user guide).
These sensors are currently available:

- Temperature and relative humidity sensor (FW rel. x.xx.1)
- Temperature sensor (FW rel. x.xx.2)
- Luminosity sensor (FW rel. x.xx.3)

If the firmware is not compatible with the connected sensor, at power on all LEDs will be permanently on after the transmitter initialization.

9.1 Temperature and relative humidity sensor

Compatible firmware release:
- x.xx.1 (where x.xx can be 1.00 o newer)

Connection:
- Male 4-poles M12 connector, according to IEC 61076-2-101

Sensor stem:
- 6mm tube diameter, according to AISI 316L

Sensor working temperature:
- (-40÷125) °C

Connector operating temperature:
- (-40÷80) °C

Humidity sensor[4]:
- Hysteresis @ 25°C: ±0.8%RH
- Long-term drift: <0.25%RH/year
- Typical accuracy @25°C: ±2% (refer to Figure 6)

Temperature sensor[4]:
- Long-term drift: <0.03°C/year
- Typical accuracy: ±0.2°C from 0°C to 90°C (refer to Figure 6)

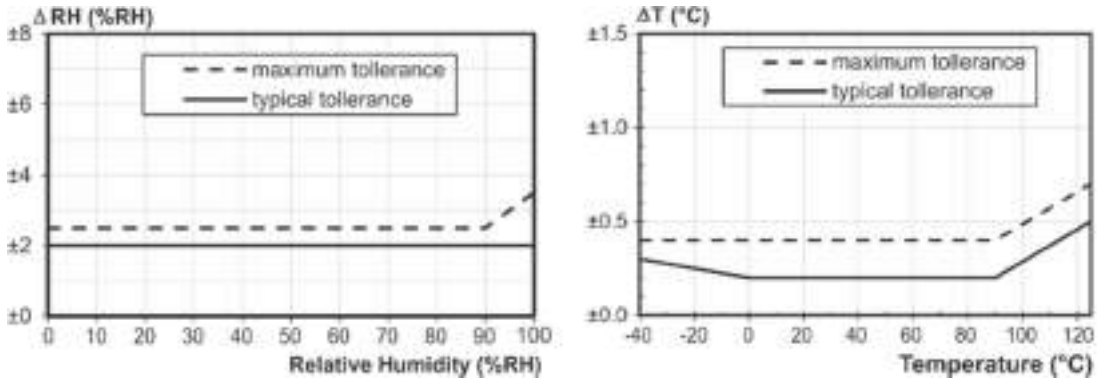


Figure 6: Sensirion SHT31 temperature and relative humidity accuracy

The temperature and relative humidity sensor mechanical dimensions are reported in Figure 7. The mechanical dimensions of the device are reported in Figure 8.

[4] For further information regarding the sensor, please refer to Sensirion SHT3x-DIS datasheet, considering the SHT31-DIS device

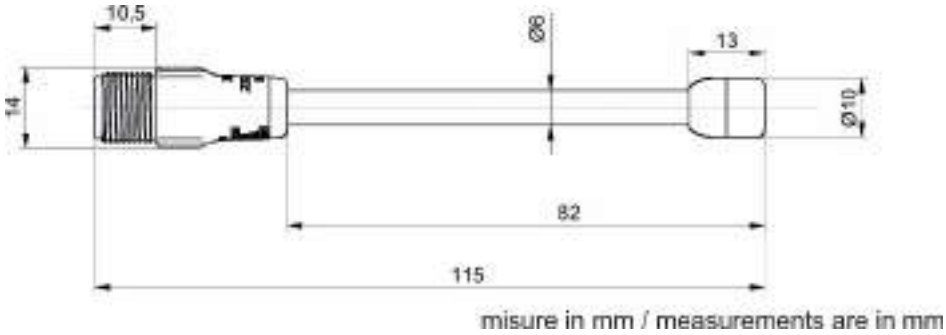


Figure 7: sensor mechanical dimension

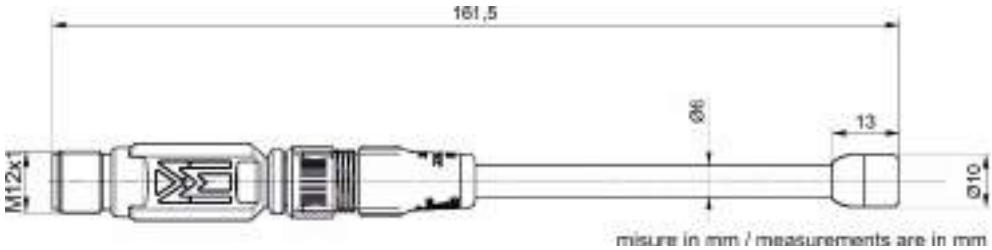


Figure 8: device mechanical dimension

9.2 Temperature sensor with magnetic connector

Compatible firmware release:
- x.xx.2 (where x.xx can be 1.00 o newer)

Connection:
- Male 4-poles M12 connector, according to IEC 61076-2-101

Sensor stem:
- Acciaio AISI 316L

Sensor working temperature:
- (-20÷85) °C

Temperature sensor[5]:
- Long-term drift: <0.03°C/year
- Typical accuracy: ±0.2°C from 0°C to 90°C (refer to Figure 9)

[5] For further information regarding temperature sensor, please refer to STS3x-DIS datasheet

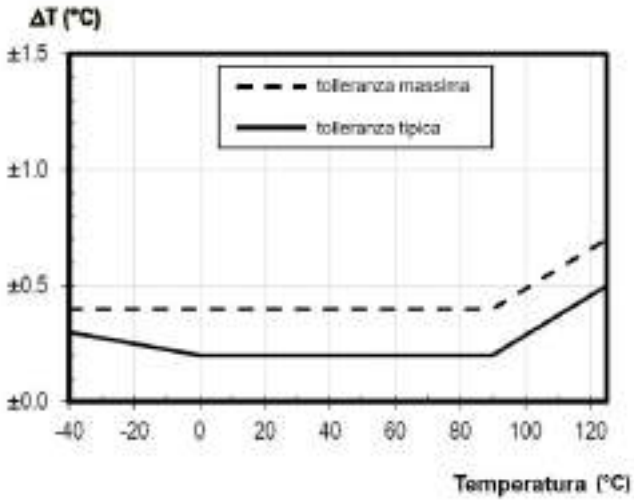


Figure 9: Sensirion STS31 temperature accuracy

The temperature sensor mechanical dimensions are reported in Figure 10.

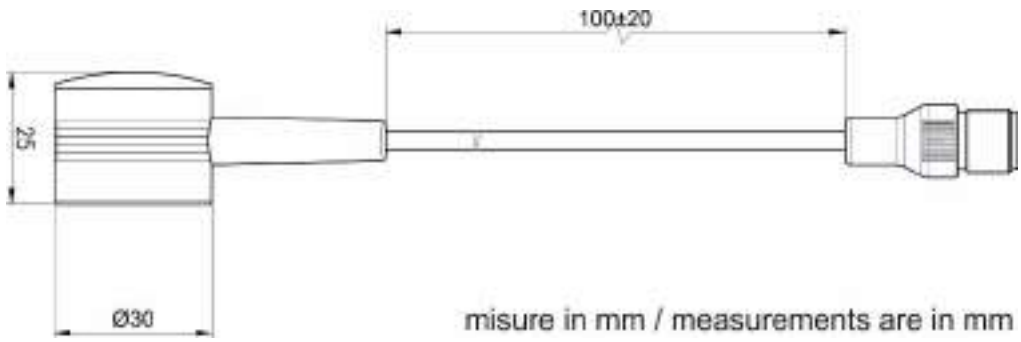


Figure 10: Sensor mechanical dimensions

9.3 Lux sensor

Compatible firmware release:
- x.xx.3 (where x.xx can be 1.00 o newer)

Connection:
- 4 Pole M12 male connector (according to IEC 61076-2-101)

Sensor body:
- ABS black (IP54)

Sensor working temperature:
- (-25÷80) °C

Lux sensor specifications[6]:
- Digital sensor model Vishay VEML7700
- Range: 0÷120KLux

[6] For further information regarding temperature sensor, please refer to Vishay VEML7700 datasheet

- Typical accuracy: see Figure 11
- Temperature compensated
- Sensitivity spectrum close to human eye photopic curve V(λ) (see Figure 12)

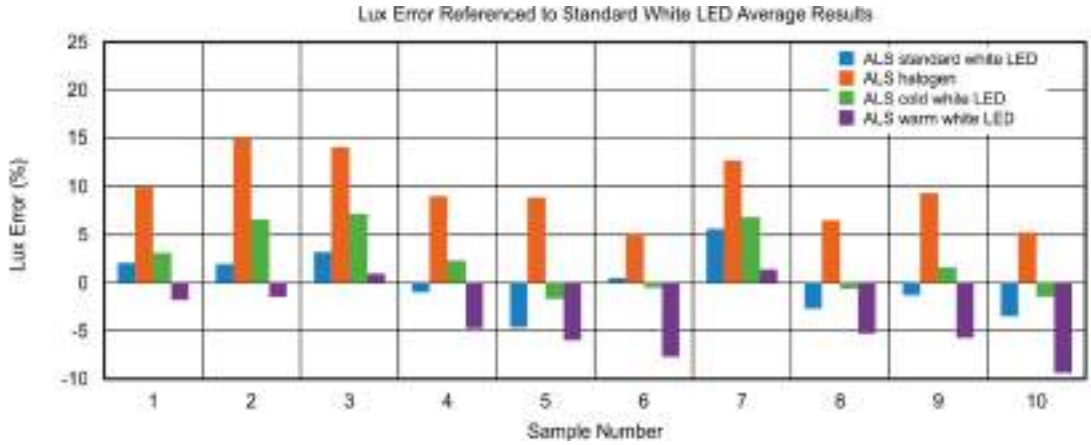


Figure 11: Sensor accuracy

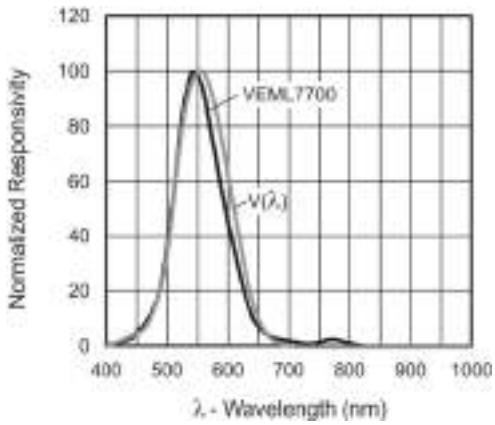


Figure 12: wavelenght

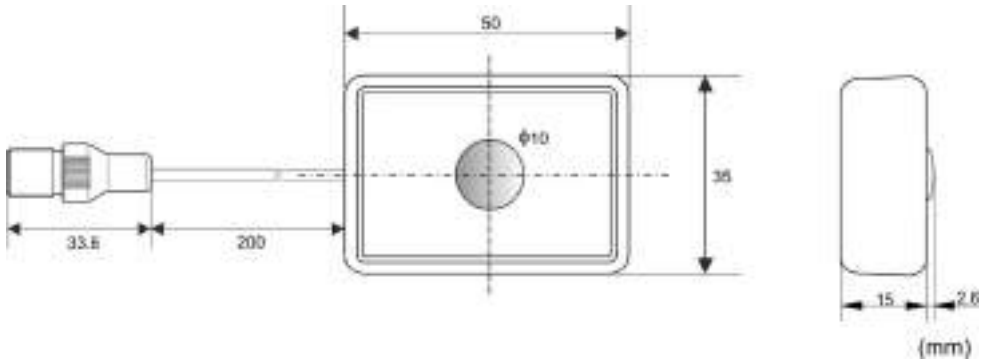


Figure 13: mechanical dimensions

10. Modbus registers table

In this chapter, the implemented registers and coils are reported. While process data change depending on the implemented firmware (i.e., depending on supported sensors), the other registers are device independent. Some of these register values are memorized in the device non-volatile memory, in order to maintain the value when device is turned off. For these registers, the column E^2P is marked in the following tables. Some of the implemented registers are readable and writable while some others are only-readable. Between writable registers, there are some in which the effect of the new inserted value is immediate while others in which a device reset is necessary to enable the new inserted value. In the following tables, the range of acceptable values for each register has been specified.

10.1 Modbus communication parameters

Modbus address	Register name	Range	Notes	E^2P
40051	Modbus address	1÷247	Read/Write*	X
40052	Baud Rate Modbus	2÷4* * 2 => 9600bps 3 => 19200bps 4=> 38400bps	Read/Write*	X
40053	Modbus parity	0÷2* * 0 => no parity bit 1 => even parity bit 2 => odd parity bit	Read/Write*	X
40054	Data bit number	0* * 0 => 8 bit per data	Read/Write*	X
40055	Stop Bits number	0÷1* * 0 => 1 stop bit 1 => 2 stop bit	Read/Write*	X
40056	Modbus delay time	0÷255 (2ms resolution)	Read/Write**	X

* The effect of parameter's modification is valid only after device reset.

** The effect of parameter's modification is immediate.

10.1.1 Modbus communication parameters description

Legenda=> [R]: *read* ***[R/W]:*** *write*

Address 40051: Modbus address[R/W]
Default value:1 Range: 1÷247
This parameter is the specific identification of the device inside the network. This value is saved into non-volatile memory and, in order for its modification to have effect, the device should be restarted.

Warning: the Modbus network cannot have two devices sharing the same address. Otherwise, there will be a communication conflict.

Address 40052: Baud rate[R/W]
Default value:2 (9600 bps); Range: 2÷4
This parameter specifies the set data rate, expressed in bit/s. This value is saved into non-volatile memory and, in order for its modification to have effect, the device should be restarted.
Admissible values: 2 corresponds to 9600bps, 3 to 19200bps, 4 to 38400bps.
Warning: all devices on Modbus network must share the same baud rate.

Address 40053: Parity[R/W]
Default value:0 (none); Range: 0÷2
This parameter specifies if a parity bit has been set, and which parity option has been chosen. This value is saved into non-volatile memory and, in order for its modification to have effect, the device should be restarted.
Admissible values: 0 corresponds to no parity bit selected, 1 corresponds to an even parity bit, 2 to an odd parity bit.
Warning: all devices on Modbus network must share the same parity setting.

Address 40054: Number of data bits[R/W]
Default value:0 (8 data bit)
This parameter specifies the serial communication's data bit number. The only available option is 8-bit data. This value is saved into non-volatile memory.

Address 40055: Number of stop bits[R/W]
Default value:0 (1 Stop bit); Range: 0÷1
This parameter specifies the number of stop bits added to the frame. This value is saved into non-volatile memory and, in order for its modification to have effect, the device should be turned off.
Admissible values: 0 corresponds to one stop bit, 1 corresponds to two stop bits.
Warning: in order to have the same frame size, it is suggested to use one stop bit if one parity bit is used and to use two stop bits if no parity bits are used. As a consequence, the frame size is always 11 bits.

Address 40056: Delay time Rx-Tx[R/W]
Default value:0 (0 ms) Range: 0÷255
This value identifies the delay added between query reception and its response transmission. Each unit represents a delay of 2ms.
For example:
0x0000 = 0 ms
0x0001 = 2 ms
…
0x00FF = 255x10 = 0.51 s.

10.2 Device parameters

Modbus address	Register name	Range	Notes	E²P
40061	Device name (first part)	2 ASCII characters	Read/Write*	X
40062	Device name (second part)	2 ASCII characters	Read/Write*	X
40063	Watch dog time	0÷250 (500ms resolution)	Read/Write*	X
40064	Firmware release	0÷65535	Read only	**
40065	Hardware release	0÷65535	Read only	**
40066	Serial number	0÷65535	Read only	X
40067	Batch number	0÷65535	Read only	X
40068	Working time (MSW)	0÷65535 (15 min resolution)	Read only	X
40069	Working time (LSW)	0÷65535 (15 min resolution)	Read only	X
40070	System errors 0 – Device status OK 1 – Short circuit error detection 2 – Non-volatile memory error 4 – Unconnected sensor 8 – Measurement error	--	Read only Copy of register 40001	

* The effect of parameter's modification is immediate.

** HW and FW release are permanently defined into program memory.

10.2.1 Device parameters description

Legenda=> [R]: *read* ***[R/W]:*** *write*

Addresses 40061 and 40062: Device name[R/W]
Default value: depends on the supported sensor. For temperature and relative humidity sensor is 'EVOH'; for temperature sensor is 'EVOT'.
32-bit ASCII char indicating transmitter name. The user can set a personal name in this register for the transmitter. This value is saved into non-volatile memory.

Address 40063: Watchdog time[R/W]
Default value:1 (0,5 seconds) Range: 0÷250
This parameter specifies the Modbus watchdog timer value. One unit corresponds to 0.5 seconds. If the watchdog coil has been enabled and the device does not receive commands within the time set in this register, the watchdog coil event will be set.
This value is saved into non-volatile memory and its modification has immediate effect.
Examples:
0x0001 = 0.5 seconds
0x00FF = 127.5 seconds.

Address 40064: Firmware release[R]
This parameter specifies the FW release (the value is multiplied by 100). For example, a value of 100 corresponds to a firmware version 1.00.

Address 40065: Hardware release[R]
This parameter specifies the HW release (the value is multiplied by 10). For example, a value of 10 corresponds to a hardware version 1.0.

Address 40066: Serial number[R]
This read-only register contains the serial number. This number is marked onto device as well. This value is saved into non-volatile memory.

Address 40067: Batch number[R]
This read-only register contains the batch number (week number and year of production). This number is marked onto device as well. This value is saved into non-volatile memory.

Addresses 40068[MSW] and 40069[LSW]: Working time[R]
This value indicates the device working time. One unit corresponds to 15 minutes. The value read at address 68 represents the least significant word while the value read at address 67 the most significant word. Therefore, the working time expressed in minutes corresponds to the value read from these two registers multiplied by 15. For example, if a value of 1 is read at address 67 and a value of 3 is read at address 68, the device is working for 16384 hours. This value is saved into non-volatile memory.

Address 40070: System errors[R]
This register indicates device status. Each implemented error is assigned to a specific register bit. If the bit is cleared, the correspondent error has not been detected. On the contrary, if the bit is set, the error has been detected.
Bit 0 is assigned to short circuit monitoring, bit 1 is assigned to non-volatile memory error detection, bit 2 is assigned to sensor connection monitoring and bit 3 is assigned to measurement error.
It is the copy of register 40001

10.3 System parameters

Modbus address	Register name	Range	Notes	E^2P
40101	Reserved address	-	-	
40102	Peak reset	0÷1*** *** 0 => no reset 1 => reset	Read/Write*	
40103	Default parameters	0xAAAA	Write**	
40104	Reserved address	-	-	
40105	Application FW checksum (MSB)	-	Read only	
40106	Application FW checksum (LSB)	-	Read only	
40107	Reserved address	-	-	

* The effect of parameter's modification is immediate.

** After device response (back to master), an automatic reset operation is executed in order to validate this command.

10.3.1 System parameters description

Legenda=> [R]: *read* ***[R/W]:*** *write*

Address 40101: Reserved

Address 40102: Peak reset[W]
Default value:0 Range: 0÷1
When this register is set, peaks of process data will be reset with immediate effect.

Address 40103: Default values[W]
Default value:0 Admissible value: 0xAAAA (dec. 43690)
By writing the hexadecimal value AAAA to this write-only register, default factory values are set for the following parameters.

Register	Default register value	Parameter value
All the offset	0	0
Modbus address	1	1
Modbus baud rate	2	9600bps
Modbus parity	0	None
Number of data bits	0	8 bits
Number of stop bits	0	1 bit
Modbus delay	0	0ms
Device name	**	**
Watchdog time	1	0.5s
Minimum peaks	Actual value	Actual value
Maximum peaks	Actual value	Actual value
TX e RX LED status (coil)	0	ON
Configuration LED status (coil)	0	ON

* If temperature and relative humidity sensor is supported

** EVOH if temperature and relative humidity sensor is supported, EVOT if temperature sensor is supported, ELUX if luminosity sensor is supported

Address 40104: Reserved
Reserved addresses.

Addresses 40105[MSW] and 40106[LSW]: Application FW checksum[R]
This register specifies the checksum value of the FW programmed in the specified device. The value read at address 105 represents the least significant word while the value read at address 104 the most significant word.

Address 40107: Reserved

10.4 Additional information

Modbus address	Register name	Range	Notes	E^2P
40151	Compatibility with EvoSER	0 – Not compatible 1 - Compatible	Read only	*
40152	Reserved address	-	-	

* Definito (in modo permanente) all'interno della memoria programma.

10.4.1 Additional information description

Legenda=> [R]: *read* ***[R/W]:*** *write*

Address 40151: Compatibility with EvoSER[R]
This register specifies whether the FW is compatible with EvoSER (1) or not (0).

Address 40152: Reserved

10.5 Process data

The data read from the connected sensor are called process data. Depending on the connected sensor and the installed FW, these data change.

10.5.1 Process data for temperature and relative humidity sensor

- Process data FW rel. x.xx.1 (temperature and relative humidity sensor)

Modbus address	Register name	Range	Notes	E²P
40001	System errors 0 – Device status OK 1 – Short circuit error detection 2 – Non-volatile memory error 4 – Unconnected sensor 8 – Measurement error	--	Read only	
40002	Temperature	(-40.0÷125.0) °C	Read only	
40003	Relative humidity	(0.0-100.0) %	Read only	
40004	Dew point	--	Read only	
40005	Min. temperature peak	(-52.5÷137.5) °C**	Read only	X
40006	Max. temperature peak	(-52.5÷137.5) °C**	Read only	X
40007	Min. humidity peak	(-12.5÷112.5) %**	Read only	X
40008	Max. humidity peak	(-12.5÷112.5) %**	Read only	X
40009	Temperature offset	(-12.5÷12.5) °C	Read/Write*	X
40010	Humidity offset	(-12.5÷12.5) %	Read/Write*	X

* The effect of parameter's modification is immediate.

** Values that are outside measurement range are only due to the offset setup.

10.5.2 Process data for temperature and relative humidity sensor description

Legend=> [R]: *read* ***[R/W]:*** *read/write*

Address 40001: System errors[R]
This register indicates device status. Each implemented error is assigned to a specific register bit. If the bit is cleared, the correspondent error has not been detected. On the contrary, if the bit is set, the error has been detected.
Bit 0 is assigned to short circuit monitoring, bit 1 is assigned to non-volatile memory error detection, bit 2 is assigned to sensor connection monitoring and bit 3 is assigned to measurement error.

Address 40002: Temperature[R]
Detected temperature values, expressed in tenths of Celsius degree. For example, a value of 100 corresponds to a temperature of 10.0°C. The measurement range is (-40÷125)°C.

Address 40003: Relative humidity[R]
Detected relative humidity, expressed in tenths of percentage. For example, a value of 100 corresponds to 10.0%RH. The measurement range is (0÷100)%.

Address 40004: dew point[R]
Dew point calculated value expressed in tenths of Celsius degree. For example, a value of 100 corresponds to a dew point temperature of 10.0°C.
NOTE: this value is update only when no measurement error is detected and when the sensor is connected.

Address 40005: Temperature min. peak [R]
Temperature minimum detected peak, expressed in tenths of Celsius degree. This value is stored into device non-volatile memory. Therefore, this value is permanently memorized.
To reset this value, a specific action should be taken (acting on address 40102).

Address 40006: Temperature max. peak [R]
Temperature maximum detected peak, expressed in tenths of Celsius degree. This value is stored into device non-volatile memory. Therefore, this value is permanently memorized.
To reset this value, a specific action should be taken (acting on address 40102).

Address 40007: Relative humidity min. peak [R]
Relative humidity minimum detected peak, expressed in tenths of percentage. This value is stored into device non-volatile memory. Therefore, this value is permanently memorized.
To reset this value, a specific action should be taken (acting on address 40102).

Address 40008: Relative humidity max. peak[R]
Relative humidity maximum detected peak, expressed in tenths of percentage. This value is stored into device non-volatile memory. Therefore, this value is permanently memorized.
To reset this value, a specific action should be taken (acting on address 40102).

Address 40009: Temperature offset[R/W]
Default value:0.0°C; Range: ±12.5°C
This value, expressed in tenths of Celsius degree, adds a positive/negative offset to the sensor measured temperature.
For example, a +100 value of the offset adds +10.0°C to the measured temperature. If the value of 255 is the measured temperature, the value read in register 0 will be 355 corresponding to 35.5°C.

Address 40010: Relative humidity offset[R/W]
Default value:0.0%; Range: ±12.5%
This value, expressed in tenths of percentage, adds a positive/negative offset to the sensor measured relative humidity.
For example, a -100 value of the offset subtracts -10.0% to the measured relative humidity. If the value of 505 is the measured relative humidity, the read value in register 1 will be 405 corresponding to 40.5%RH.

10.5.3 Process data for surface temperature sensor

- Process data FW rel. x.xx.2 (temperature sensor)

Modbus address	Register name	Range	Notes	E²P
40001	System errors 0 – Device status OK 1 – Short circuit error detection 2 – Non-volatile memory error 4 – Unconnected sensor 8 – Measurement error	--	Read only	
40002	Temperature	(-40.0÷125.0) °C	Read only	
40003	Min. temperature peak	(-52.5÷137.5) °C**	Read only	X
40004	Max. temperature peak	(-52.5÷137.5) °C**	Read only	X
40005	Temperature offset	(-12.5÷12.5) °C	Read/Write*	X

* The effect of parameter's modification is immediate.

** Values that are outside measurement range are only due to the offset setup.

10.5.4 Process data for surface temperature sensor description

Legend=> [R]: *read* ***[R/W]:*** *read/write*

Address 40001: System errors[R]
This register indicates device status. Each implemented error is assigned to a specific register bit. If the bit is cleared, the correspondent error has not been detected. On the contrary, if the bit is set, the error has been detected.
Bit 0 is assigned to short circuit monitoring, bit 1 is assigned to non-volatile memory error detection, bit 2 is assigned to sensor connection monitoring and bit 3 is assigned to measurement error.

Address 40002: Temperature[R]
Detected temperature values, expressed in tenths of Celsius degree. For example, a value of 100 corresponds to a temperature of 10.0°C. The measurement range is (-40÷125)°C.

Address 40003: Temperature min. peak [R]
Temperature minimum detected peak, expressed in tenths of Celsius degree. This value is stored into device non-volatile memory. Therefore, this value is permanently memorized.
To reset this value, a specific action should be taken (acting on address 40102).

Address 40004: Temperature max. peak [R]
Temperature maximum detected peak, expressed in tenths of Celsius degree. This value is stored into device non-volatile memory. Therefore, this value is permanently memorized.
To reset this value, a specific action should be taken (acting on address 40102).

Address 40005: Temperature offset[R/W]
Default value:0.0°C; Range: ±12.5°C
This value, expressed in tenths of Celsius degree, adds a positive/negative offset to the sensor measured temperature.

For example, a +100 value of the offset adds +10.0°C to the measured temperature. If the value of 255 is the measured temperature, the value read in register 0 will be 355 corresponding to 35.5°C.

10.5.5 Process data for Lux sensor

- Process data for FW vers. x.xx.3 (Lux sensor)

Modbus address	Register name	Range	Notes	E²P
40001	System errors 0 – Device status OK 1 – Short circuit error detection 2 – Non-volatile memory error 4 – Unconnected sensor 8 – Measurement error	--	Read only	
40002	Lux value [MSW] Lux value [LSW]	0÷120000.00 Lux	Read only	
40004	Min. Lux Peak [MSW] Min. Lux Peak [LSW]	-500.00÷120500.00 Lux**	Read only	X
40006	Max Lux Peak [MSW] Max Lux Peak [LSW]	-500.00÷120500.00 Lux**	Read only	X
40008	Lux offset [MSW] Lux offset [LSW]	-500.00÷500.00 Lux	Read/Write*	

* The effect of parameter's modification is immediate.

** Values that are outside measurement range are only due to the offset setup.

10.5.6 Process data for Lux sensor description

Legend=> [R]: *read* ***[R/W]:*** *read/write*

Address 40001: System errors[R]
This register indicates device status. Each implemented error is assigned to a specific register bit. If the bit is cleared, the correspondent error has not been detected. On the contrary, if the bit is set, the error has been detected.
Bit 0 is assigned to short circuit monitoring, bit 1 is assigned to non-volatile memory error detection, bit 2 is assigned to sensor connection monitoring and bit 3 is assigned to measurement error.

Warning: each variable mentioned below is two 16-bit registers. The first register is always the most significant word (MSW)

Address 40002-40003: Luminosity[R]
Detected luminosity values, expressed in cents of Lux. For example, a value of 1000 corresponds to a luminosity 10.00 Lux. The measurement range is (0÷120000.00) Lux.

Address 40004-40005: Minimum luminosity peak[R]
Luminosity minimum detected peak, expressed in cents of Lux. This value is stored into device non-volatile memory. Therefore, this value is permanently memorized.
To reset this value, a specific action should be taken (acting on address 40102).

Address 40006-40007: Maximum luminosity peak [R]
Luminosity maximum detected peak, expressed in cents of Lux. This value is stored into device non-volatile memory. Therefore, this value is permanently memorized.
To reset this value, a specific action should be taken (acting on address 40102).

Address 40008-40009: Luminosity Offset [R/W]
Default value:0.00Lux; Range: ±500.00°C
This value, expressed in cents of Lux, adds a positive/negative offset to the sensor measured luminosity.
For example, a +1000 value of the offset adds +10.00Lux to the measured luminosity. If the value of 120.00 is the measured Luminosity, the value read in register 0 will be 13000 corresponding to 130.00 Lux.
Note: This double register must be written using the multiple write command (function code 16). Writing of the single register (function code 6) is not accepted.

11. Modbus coils table

Modbus address	Register name	Range	Notes	E^2P
00001	Watch Dog event enable	ON/OFF	Read/Write*	
00002	Watch Dog event	ON/OFF	Read/Write*	
00003	Power Up event	ON/OFF	Read/Write*	
00004	TX and RX LEDs status	ON/OFF	Read/Write*	X
00005	PWR LED status	ON/OFF	Read/Write*	X

* The effect of parameter's modification is immediate.

11.1 Coil description

All the following coils are found at zero address.

Address 00001: Watchdog enable
Default value: OFF (Disabled watchdog)
Set this coil to enable Modbus watchdog.

Address 00002: Watchdog event
If coil 0 has been activated, a set value indicates that the device has not received commands within the set watchdog time value (register 40063). This coil can be cleared to reset the event. If the coil is forced high, a watchdog event is simulated together with its alarm condition.
1 = Alarm
0 = No event
For example, this coil can be used by the SCADA software to monitor the communication.

Address 00003: Power-UP event
This coil is set after power-on event. It must be manually cleared to monitor a reset event.
1 = Reset
0 = No reset

Address 00004: Transmission and reception LED status[R/W]
Default value: OFF (LED OFF)
Clear this register for these LEDs activation; set this register to deactivate these LEDs. This value is saved into non-volatile memory.

Address 00005: PWR LED status[R/W]
Default value: OFF (LED OFF)
Clear this register for this LED activation; set this register to deactivate this LED. This value is saved into non-volatile memory.

Attention: if both coils 4 and 5 are set, it would be possible to save energy. This can be particularly useful at power on, when several devices are connected and the current supplied is limited.

12 Modbus RTU

12.1 Introduction to Modbus protocol

Modbus is a serial communication protocol originally published by Modicon (Schneider Electric) in 1979 for use with its PLCs. It has become a widespread communication protocol, commonly used in industrial electronic device interconnections. When the specifications became public and open, the Modbus protocol was adopted in numerous automation applications and subsequently in all sectors.
For many years it has been a *de facto* standard, and the Modbus protocol can be found on any "intelligent" equipment (IFD - Intelligent Field Device): programmable controllers, NC, drivers, man/machine terminals, measurement equipment, etc.

12.2 Master-slave protocol model

The connection is made over an RS-485 multipoint network, typically by means of a twisted two-wire cable with shielding.
In the master-slave architecture, only one device (the master) can initiate transmission actions (called "queries"). The other devices (the slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. The master can address individual slaves, or can initiate a broadcast messages to all the slaves. Slaves return a message (called a "response") to queries that are addressed to them individually. Response messages are not returned to broadcast queries from the master.
The EvominiSER implements a master-slave, multipoint, half-duplex communication system, in which only the Master (typically a Host PC) can initiate communications with a request ("Query"), while the Slaves respond with a message ("Response") only to the queries addressed directly to themselves. The Modbus protocol establishes the format for the master's query by placing into it the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error-checking field. The slave's response message is also constructed using Modbus protocol. It contains fields confirming the action taken, any data to be returned, and an error-checking field.
One master can address up to 247 slaves on a single line (protocol limit). However, the RS-485 standard interface supports a maximum of 31 slaves on a single line. By replacing the last device on the RS485 line with a bridge or repeater, other 31 slaves can be added to the network. This operation can be repeated until the number of desired slaves are connected on the network (up to 247 slaves as stated before).

12.3 RTU transmission mode

The EvominiSER device communicates on a Modbus serial line using the RTU (remote terminal unit) mode. Each byte in a message contains two 4-bit hexadecimal characters. The main advantage of this mode is that its greater character density allows better data throughput than ASCII mode for the same baud rate. Each message must be transmitted in a continuous stream of characters. The format for each RTU mode byte is:
Coding system: 8 binary bits, from 00 to FF
Two hexadecimal characters are contained in each 8-bit field of the message. Each 8-bit character of the message corresponds to one data byte.
RTU character framing:
1 start bit;
8 data bits, least significant bit sent first;

1 parity bit (even or odd) or no parity bit,
1 stop bit or 2 parity bits (required in case of no parity bit).
Address field: valid slave device addresses are in the range of 0÷247 decimal. The individual slave devices are assigned addresses in the range of 1÷247. A master addresses a slave by placing the slave address in the address field of the message. When the slave sends its response, it places its own address in the address field of the response to let the master known which slave is responding. Address 0 is used for broadcast query (not implemented).
Function field: valid function field codes are in the range of 1÷255 decimal.
When a message is sent from a master to a slave device, the function code field tells the slave what kind of action to perform. When the slave responds to the master, it uses the function code field to indicate either a normal (error-free) response or that some kind of error occurred (called an exception response). For a normal response, the slave simply echoes the original function code. For an exception response, the slave returns a code that is equivalent to the original function code with its most-significant bit set to a logic 1. This tells the master what kind of error occurred, or the reason for the exception.
The EvominiSER system, implements the following Modbus functions:

Modbus function code	***Function***
01	Read Coil Status
02	Read Input Status
03	Read Holding Registers
04	Read Input Registers
05	Force Single coil
06	Preset Single Register
07	Read Exception Status
16	Preset Multiple Registers

In the EvominiSER system, functions 01 and 02 are identical and interchangeable, as well as functions 03 and 04.

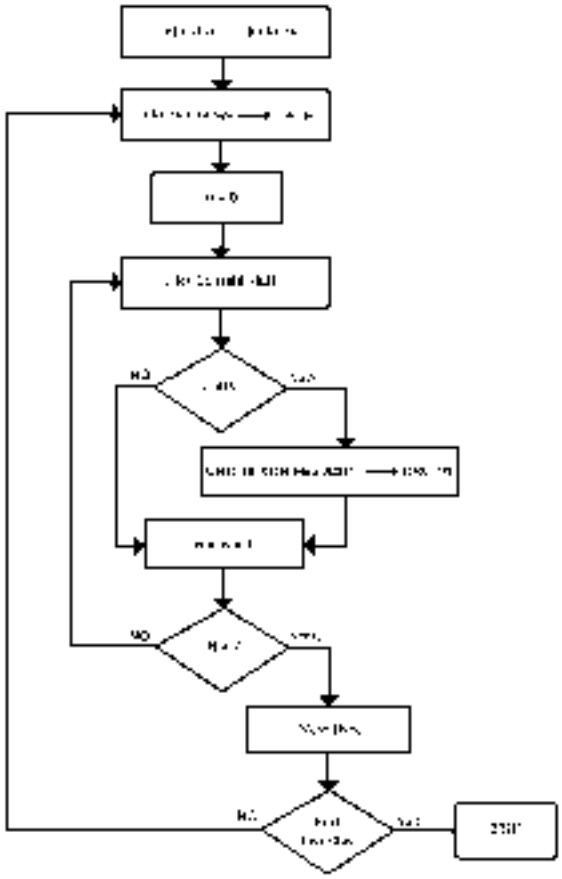
10.3.1 CRC-16

The last two characters of the Modbus RTU contains a 16-bit value error checking field, implemented as two 8-bit bytes. The error check value is the result of CRC (Cyclical Redundancy Check) calculation performed on the message contents. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error has been detected. The CRC-16 algorithm is used. In order to calculate these two characters, the message (address, function code and data, excluding the start, stop and parity bit) is considered as a single binary number of which the most significant bit (MSB) is the first to be transmitted. The message is first multiplied by 2^{16} (moved to the left by 16 bits) and then divided by $2^{16} + 2^{15} + 2^2 + 1$ expressed as a binary number (1100000000000101). The entire quotient is then discarded and the 16-bit remainder (initialized to FFFFh at the beginning to avoid the case of messages consisting of only zeroes) is added to the transmitted message. The resulting message, when it is divided by the receiving device in the same way ($2^{16} + 2^{15} + 2^2 + 1$) will give zero as remainder if no errors have occurred (the receiving device recalculates the CRC). Since the UART unit transmits first the least significant bit (LSB) rather than the MSB, the generator polynomial is inverted. Furthermore, the MSB only influences the quotient and not the remainder. Therefore, it is deleted and the generator polynomial becomes $2^{15} + 2^2 + 1$. The procedure for CRC16 calculation consists of the following steps:

1) preload a 16-bit register to all 1's;
2) exclusive-OR the first character with the most significant byte of the register, and store the result into the 16-bit register;

3) shift the 16-bit register to the right by one bit;
4) if the shifted-out bit is 1, exclusive-OR the generator polynomial with the 16-bit register;
5) repeat steps 3) and 4) for 8 times;
6) exclusive-OR the next character with the most significant byte of the register, and store the result to the 16-bit register;
7) repeat steps 3 to 6 for the entire message;
8) the content of the 16-bit register is the CRC code which must be added to the message.

10.3.2 CRC16 flow chart



12.3.3 Message synchronization

In RTU mode, message frames are separated by a silent interval of at least 3.5 characters. This allows for a baud rate independent synchronization between master and slaves.
If the receiving device does not receive for a period of at least 3.5 characters, it assumes that the previous message was complete. The subsequent received information will be considered a new message. Each device on the network monitors the bus, including during the silent pauses. The first transmitted character is the address of the device. When the address has been received, each device on the network decodes it to determine to which device is addressed the message.
A silent interval of at least 3.5 characters following the message indicates its end. The entire message frame must be transmitted as continuous stream of characters. If a new message starts before the silent interval, the receiver will consider it as being a continuation of the

previous message. This will generate an error, since the value of the last field (CRC) will not be valid for the combined messages.

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	1 CHAR (8 bits)	1 CHAR (8 bits)	N CHARS (n x 8 bits)	2 CHARS (16 bits)	T1-T2-T3-T4

12.4 Modbus function codes

In this chapter, a detailed description of Modbus function codes implemented on EvominiSER is reported.

12.4.1 Read coil/input status (01 or 02)

This function allows the user to read the logical value (ON/OFF) of the bits of the addressed device. The returned data are packaged in bytes, so that the first requested bit occupies the least significant bit of the first byte of data. The others follow in such a way that if the number of bits requested is not a multiple of 8, the last byte in the response will be completed with zeroes.

Master – device frame:

Device address	**Function code (01 or 02)**	**Address of first bit**		**Number of bits to read (max 255)**		**CRC**	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Device – master frame:

Device address	**Function code**	**Number of bytes read**	**First byte of data**	**.....**	**Last byte of data**	**CRC**	
1 byte	1 byte	1 byte	1 byte	.....	1 byte	MSB	LSB

Example: Read 3 bits starting from the bit 0 of device addressed with 1.

Master – device frame:

Device address	**Function code**	**Address of first bit**		**Number of bits to read**		**CRC**	
01	01	00	00	00	03	7C	0B

Device – master frame:

Device address	**Function code**	**Number of bytes read**	**Number of bytes of data**	**CRC**	
01	01	01	04	50	4B

The response tells us that bits 0 and 1 (Enable Watchdog Event and Watchdog Event) are cleared while bit 2 (power-up coil) is set.
NOTE: The response assigns zeroes to addresses which are not requested by the master; this does not mean that their real values are zero.

12.4.2 Read holding/input registers (03 or 04)

This function allows the user to read the values of the registers of the addressed device.

Master – device frame:

Device address	**Function code (03 or 04)**	**Address of register number**		**Number of registers to read (max 16)**		**CRC**	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Device – master frame:

Device address	**Function code**	**Number of bytes read**	**Value of first register**		**.....**	**Value of last register**		**CRC**	
1 byte	1 byte	1 byte	MSB	LSB	.....	MSB	LSB	MSB	LSB

Example: Read 2 registers starting from register 20 (device name) of device addressed with 1.

Master – device frame:

Device address	**Function code**	**Address of register number**		**Number of registers to read**		**CRC**	
01	03	00	14	00	02	84	0F

Device – master frame:

Device address	**Function code**	**Number of bytes read**	**Value of first register**		**Value of last register**		**CRC**	
01	03	04	45	56	4F	55	FB	20

The response tells us that registers 20 and 21 respectively have values 4556h and 4F55h, corresponding to 'EV' 'OU' in ASCII.
Up to 16 registers can be read with one request. A request for reading more than 16 registers, generates an error frame and will be discarded.

12.4.3 Force single coil (05)

This function allows the user to assign the logical values (ON/OFF) of the bits of the addressed device. To deactivate a bit, two hexadecimal 0s must be sent while to activate a bit 01h or FFh must be sent. Such values must be written in the **most significant byte**.

Master – device frame:

Device address	Function code (05)	Address of bit		Value of bit		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Device – master frame:

Device address	Function code (05)	Address of bit		Value of bit		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Example: Activate bit 0 of device addressed with 1.

Master – device frame:

Device address	Function code	Address of bit		Value of bit		CRC	
01	05	00	00	FF	00	8C	3A

Device – master frame:

Device address	Function code	Address of bit		Value of bit		CRC	
01	05	00	00	FF	00	8C	3A

The response tells us that bit 0 (Enable Watchdog Event) has been activated.

12.4.4 Preset single register (06)

This function allows the user to set a specified register of the addressed device.

Master – device frame:

Device address	Function code (06)	Address of register		Value of register		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Device – master frame:

Device address	Function code (06)	Address of register		Value of register		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Example: Assign 10 to register 3 to the device addressed with 1.

Master – device frame:

Device address	Function code	Address of register		Value of register		CRC	
01	06	00	03	00	0A	F9	CD

Device – master frame:

Device address	Function code	Address of register		Value of register		CRC	
01	06	00	03	00	0A	F9	CD

The response tells us that register 3 (temperature bias) has been set to 10 (1.0°C).

12.4.5 Preset multiple registers (16)

This function allows the user to set values for a block of consecutive 16-bit registers.
No more than 16 registers can be set with a single command.

Master – device frame:

Device address	Function code (16)	First addressed register		Number of words		Number of bytes	...DATA...		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	1 byte	MSB	LSB	MSB	LSB

Device – master frame:

Device address	Function code (16)	First addressed register		Number of words		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Example: Write two registers at the same time. Assign the value 4565h to the register 20, and 6D6Fh to the register 21 [Device name 'Demo'] to device addressed with 1.

Master – device frame:

Device address	Function code (16)	First addressed register		Number of words		Number of bytes	...DATA...				CRC	
01	10	00	14	00	02	04	44	65	6D	6F	9B	03

Device – master frame:

Device address	Function code (16)	First addressed register		Number of words		CRC	
01	10	00	14	00	02	01	CC

12.4.6 Modbus exception responses

Except for broadcast messages (not implemented in EvominiSER), when a master device sends a query to a slave device it expects a normal response. One of four possible events can occur from the master's query:

- If the slave device receives the query without a communication error, and can handle the query normally, it returns a normal response.
- If the slave does not receive the query due to communication error, no response is returned. The master program will eventually process a timeout condition for the query.
- If the slave receives the query, but detects a communication error (parity or CRC), no response is returned. The master program will eventually process a timeout condition for the query.
- If the slave receives the query without a communication error, but cannot handle it (for example, if the request is to read a non-existent coil or register), the slave will return an exception response informing the master of the nature of the error.

The exception response message has two fields that differentiate it from a normal response:

1. Function code field: in a normal response, the slave echoes the function code of the original query in the function code field of the response. All function codes have a most-significant bit (MSB) of 0 (their values are all below 80 hexadecimal). In an exception response, the slave sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hexadecimal higher than the value would be for a normal response. With the function code's MSB set, the master's application program can recognize the exception response and can examine the data field for the exception code.
2. Data field: in a normal response, the slave may return data or statistics in the data field (any information that was required in the query). In an exception response, the slave returns an exception code in the data field. This defines the slave condition that caused the exception.

Modbus protocol defines a list of 8 exception codes. However, EvominiSER implements only the first three of them:

Error code	Name	Description
01	ILLEGAL FUNCTION	Invalid function
02	ILLEGAL DATA ADDRESS	Invalid address
03	ILLEGAL DATA VALUE	Invalid data

13 Device setup with EvoPlatform

The EvoPlatform software it has been developed by Italcoppie Sensori in order to configure the Evo series. The EvoPlatform is free downloadable from Italcoppie Sensori website http://www.italcoppie.it/.
After having properly installed the software, connect the EvominiSER to the USB port and wait for drivers to be installed.
The EvoPlatform software automatically recognizes on which COM port the interface has been installed; otherwise, from the Windows Control Panel, in "System-> Device Management-> Ports (COM and LPT) check on which COM the operating system has been assigned to the connected device.

Double-click the icon  to open EvoPlatform.

Figure 14: EvoPlatform device selection

Select the EvominiSER device, and its configuration window opens automatically, as shown in Figure 15.

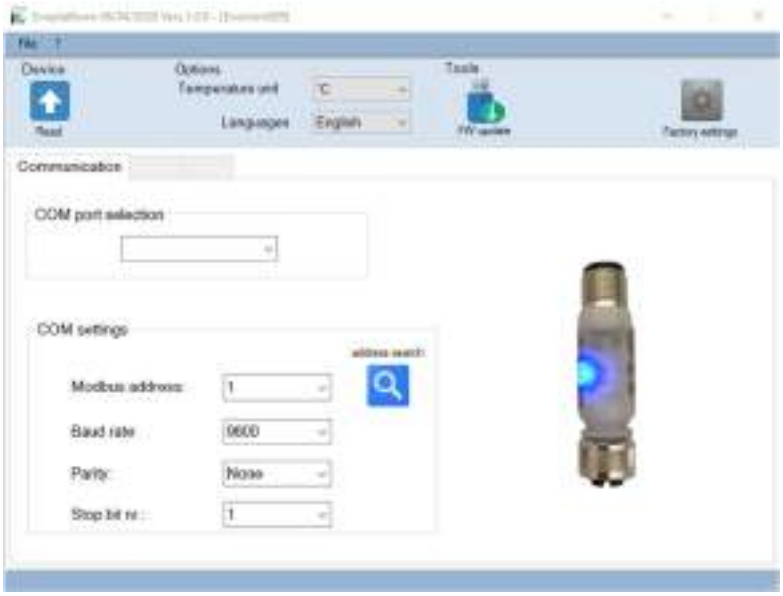


Figure 15: EvominiSER configuration window

Either if the device has not been properly connected to the USB port or the COM has not been automatically detected by the EvoPlatform, the message shown in Figure 16 is displayed. Select OK to proceed, and manually select the COM port from COM port selection menu.

Figure 16: COM port not found

13.1 Configure EvominiSER

13.1.1 Set desired COM parameters and Modbus address

Factory settings of COM port are:

- Modbus address: 1
- Baud rate: 9600bps
- Parity: none
- Stop bit number: 1

It is possible to restore these factory settings from "Factory settings" (refer to chapter 13.1.5).
After having set COM parameters, click on "Read" to read saved configuration from connected device. If the reading operation has been successfully, the following message is displayed:

If the Modbus address is unknown, it is possible to use the "address search" function. The Modbus address is then automatically updated if device has been found.

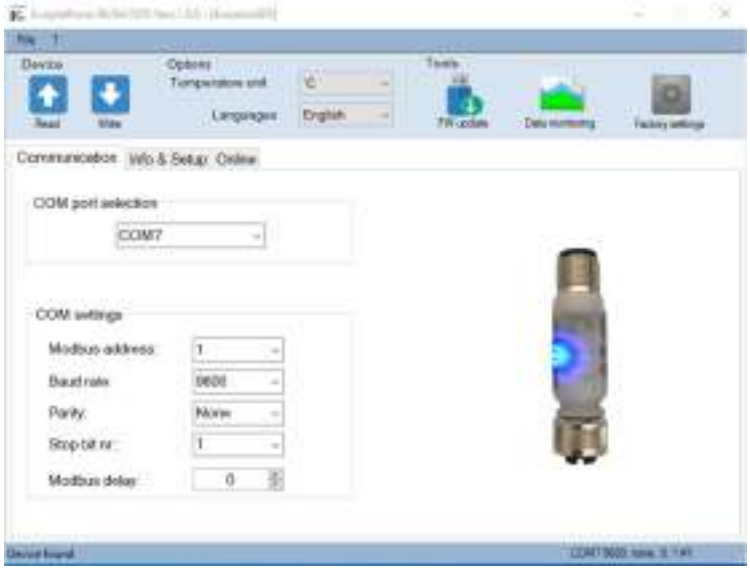


Figure 17: device found and read

In the "Communication" tab select the desired COM settings. To transfer these new COM parameters to the connected device, click on "Write" and wait for the following message to confirm the correct operation.

13.1.2 EvominiSER information and setup

The "Info & Setup" menu is shown in Figure 18: Info & setup menu. For parameters meaning and functionality, refer to chapter 10.2 (device parameters). After having set parameters to desired configuration, transfer them to the connected device by using "Write" command. Such command will transfer the "Device setup", the "Device info" and the offsets (set on the "Online menu") from Online tab.

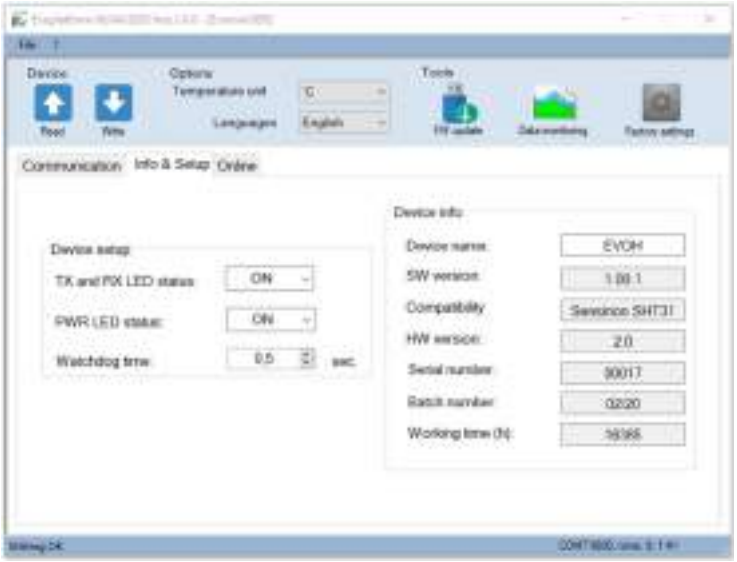


Figure 18: Info & setup menu

13.1.3 Online features

The online functions depend on the type of sensor connected. In particular, the process data varies depending on the sensor. For example, Figure 19 shows the example of the temperature and humidity sensor. Starting the reading, the process parameters are displayed.

From this window it is also possible to set the watchdog and power up event coils, a temperature and relative humidity offset and delete the peaks recorded by the device using the "Reset temperature and humidity peaks" button.

Figure 19: online menu[7]

13.1.4 Data monitoring

It is also possible to plot measured sensor variables and save them on .csv file. Click "Data monitoring" and start the operation for plotting data, as shown in Figure 20. In order to save the acquisition, just mark the "Data logging ON" option before starting the acquisition.

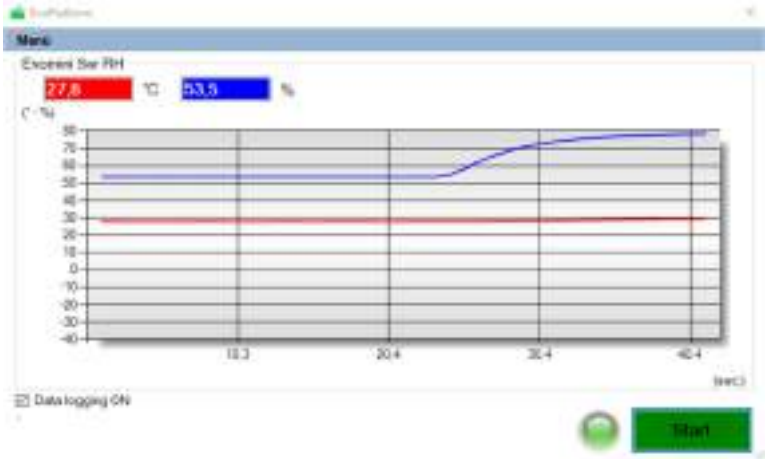


Figure 20: data monitoring[7]

The .csv file is placed in Documents\Italcoppie\EvoPlatform\DataEvoPlatform\EvominiSER and file is named ddmmyyyy_h_min_s.csv where "dd" indicates the day, "mm" the month, "yyyy" the year, "h" the hour, "min" the minute and "s" the second at the moment of file creation.

[7] This image represents a temperature and relative humidity sensor

13.1.5 Factory settings

By clicking on "Factory settings", a password is required as shown below:

ATTENTION: launch factory settings command only if one device is connected to the configuration kit.

If one factory setting command is executed when more devices are connected to the network, all the connected devices will be restored to factory defaults. Therefore, the network of devices will not work after the factory restore. Thus, a new setup for every single device will be necessary.

The password **EVO#SER!** has to be inserted before pressing OK.

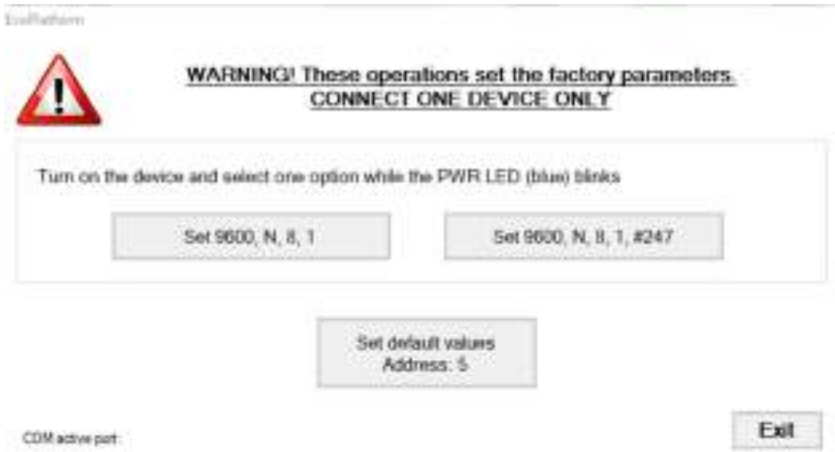


Figure 21: factory settings menu

"Set default values" is available only if a reading operation has been already performed. In order to restore factory settings, power the device on and wait for blue LED blinking. By clicking on "Set 9600, N, 8, 1", the following COM parameters will be restored:

- Modbus address: unchanged;
- Baud rate: 9600bps;
- Parity: none;
- Stop bit nr.: 1;
- Modbus delay: unchanged.

The following window will be displayed.

By clicking OK, the main communication page is displayed and an automatic device search is started. By clicking on “Set 9600, N, 8, 1, #247”, the following COM parameters are set:

- Modbus address: 247;
- Baud rate: 9600bps;
- Parity: none;
- Stop bit nr.: 1;
- Modbus delay: unchanged.

The following window will be displayed.

By clicking OK, the main communication page is displayed and an automatic device search is started.

13.1.6 Configuration file

Click on menu "File-> save as" to save the configuration. The default path for the file is: "Documents \ Italcoppie \ EvoPlatform \ ConfigEvoPlatform \ EvominiSER". The default name for the file corresponds to the device serial number with .ini extension. However, it is possible to select a desired path and file name. The saved file can be opened with a text editor.
In Figure 22 is reported an example of a .ini file saved for a device with serial number 100. If more devices are connected to the network, more files can be saved with different names.

Figure 22: EvominiSER configuration file

14 Firmware update

EvoPlatform allows the user to update its EvominiSER device firmware. The tool is "FW update".

Before executing this operation, read the connected device to import the parameters of its configuration (see chapter 13.1).

An *.evp file must be downloaded from Italcoppie Sensori website and saved to a known location. Select "FW update" from tools and load .evp file containing the FW to be updated, as shown in Figure 23.

Figure 23: FW update

Once the file has been selected, the about session shows some information regarding the new FW, as shown for example in Figure 24.

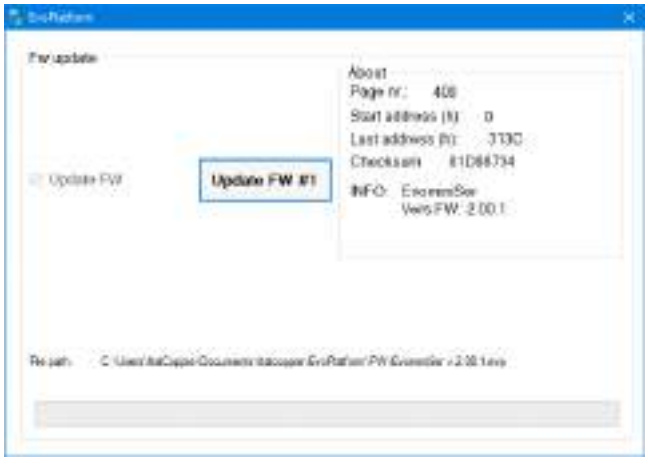
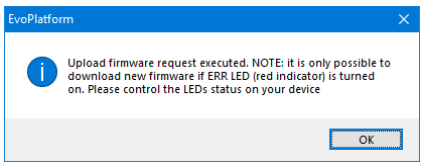


Figure 24: example of loaded .srec file

The new firmware can be updated by using two different procedures.

14.1.1 Firmware update during application

By default, the option "Update FW during application addressed device: x" is marked, where x is the addressed device (Modbus address). By changing the Modbus address in the COM parameters, it is possible to update each single device on the network.
By clicking on "Update FW #x", the following pop-up window is displayed.



If the ERR LED (red) is now permanently on and PWR LED (blue) is off, the device is in boot mode. Click OK to confirm the pop-up window and click "Download" to update the firmware, as shown in Figure 25.

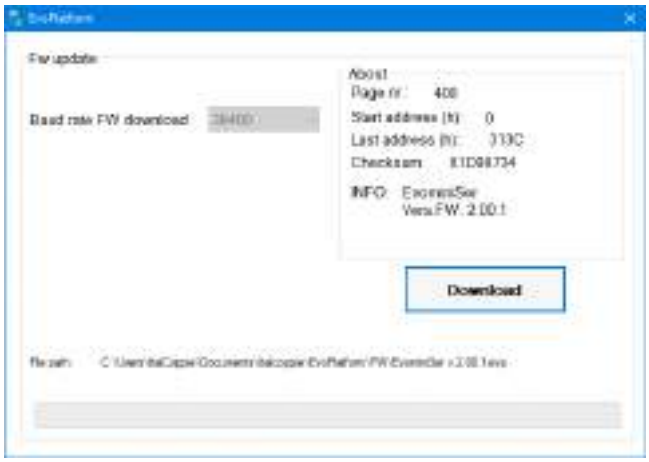


Figure 25: FW update download

Wait for the end of the download and confirm the pop-up window at the end of the operation. The device will automatically restart with the new FW updated.

ATTENTION: In some cases this procedure may involve restoring the factory default configuration.

14.1.2 Firmware at Power on (PWR)

Power off the connected device, mark the option "Update FW @ PWR" and click on "Update FW". The following pop-up windows is then displayed.

NOTE: this option of FW update must be used with only one device connected. Thus, the device must not be connected to the Modbus network with other devices.
Confirm with OK. Then, a password is required, as shown in Figure 26.

Figure 26: password request

WARNING: Once again, please check that is really necessary to perform this operation and that only one device is connected.
Insert the following password: **EVO#SER!**
Before confirming the password insertion with OK, power up the connected device and wait for PWR LED (blue) blinking. During PWR LED toggling, confirm with OK the password insertion.

If the operation has succeeded, the ERR LED (red) is now permanently on and the PWR LED (blue) is off. It is possible to download FW by pressing "Download", as shown in Figure 25. Wait for the end of the download and confirm the pop-up window at the end of the operation. The device will automatically restart with the new FW updated.

ATTENTION: This procedure ALWAYS involve restoring the factory configuration.

15 Troubleshooting

In this chapter, the most common problem and their possible solutions are described. LEDs on EvominiSER can be used to determine device status.
NOTE: if at power up LEDs are off, they have probably been disabled. Enable them by reading the device with EvoPlatform and turn to ON both led status in "Info & Setup" tab.

15.1 Communication errors

In case of missing communication, the following list of operations should be checked.

1) Selected COM port: the selected COM port must be the device's assigned COM. Check device's COM port in Windows "Device Manager".
2) Check that selected address on EvoPlatform and its set address matches.
3) Check that selected communication parameters match with device's communication parameters.

If the listed above has been executed but there is still a missing communication, please restore COM factory settings as described in chapter 13.1.5.

15.2 LEDs are off after device power up

In case LEDs are off after power-on, please execute the following operations:

1) Check that the USB port is properly connected.
2) Wait about 10 seconds from power on (even if all LEDs are off).
3) Try reading the device with EvoPlatform (after having set the correct communication parameters on EvoPlatform).
4) If there is a missing communication, refer to chapter 13.1.5.
5) After having read the device, check whether PWR LED and TX and RX LEDs status are ON or OFF (Info & Setup). If they are in OFF status, please select ON status for all and write it down to the device. It should be possible now to see all LEDs ON.
6) Turn the device off, wait few seconds and turn on the device on again.

15.3 ERR LED

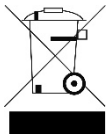
If ERR LED is permanently on, the device is in boot mode. This happens either if the "update FW" command from tools has been sent or if device firmware is corrupted. In the latter case, it is necessary to upload the device with a new firmware. See chapter 14 for more information about this operation.
If ERR LED blinks, the non-volatile memory is corrupted. Try use "Set default values" command (refer to chapter 13.1.5). If, after device reset, ERR LED is still blinking, please contact our technical service: carefully read the instructions given in chapter 5.1.

15.4 PWR LED

If the device is working properly after have being turned on, the PWR LED should be permanently on. However, if PWR LED is still blinking after device initialization, an error has occurred.

If PWR LED blinks rapidly (with period less than one second), a short circuit has been detected. Immediately turn off the device. Wait few seconds and turn the device on again. If the same error is given, turn off the device and contact our technical service.
If PWR LED blinks slowly (with period of about one second), either a missing communication between the transmitter and the sensor probe or a measurement error has occurred.
In both cases, disconnect and connect again the probe for few times. If this will not work, restart the device. If the error is still occurring, the sensor probe might be broken and should be replaced with a new one.

This product must be disposed of according to the European WEEE (Waste Electrical & Electronic equipment)



Engineered and manufactured in Italy

Made in Italy

Pensato, progettato e prodotto in Italia