

ISAW Modbus Adapter

User Guide



Document version: v1.2

IAV Technologies SARL

ISAW Products Division
Chemin des Coulevres 4A, 1295 Tannay, Switzerland

Tel: +41 (0)22 960 11 04
E-mail: isaw@iav.ch

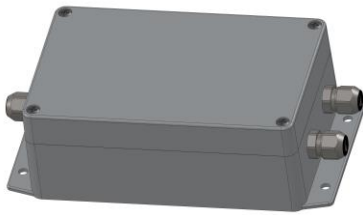
www.isaw-products.com



Table of contents

MOBUS – MODBUS ADAPTER.....	4
Description	5
Modbus function / Register definition	7
INPUT Registers	7
HOLDING Registers	10
DATA TYPE Format.....	10
Modbus function PASS-THROUGH (0x64)	11
Modbus frame examples.....	12
TEST PROCEDURE.....	14
Hardware setup.....	14
Communicating with the sensor	20
UPDATE THE MODBUS ADAPTER’S FIRMWARE	26
Hardware setup.....	26
Update procedure	28

MOBUS – MODBUS ADAPTER

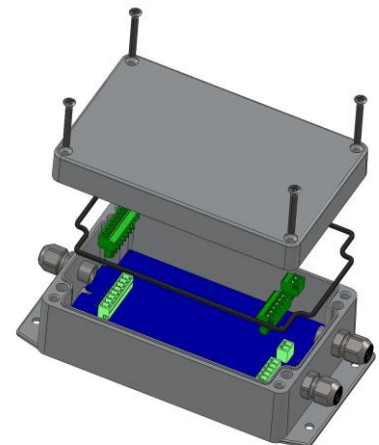


The ISAW Modbus RTU RS485 Adapter (MOBUS) enables the power supply and connection of any ISAW sensor to a Modbus network. MOBUS is the recommended accessory to interconnect the sensor through the open serial Modbus RTU (RS-485) protocol based on a master/slave or client/server architecture. The fieldbus environment is the base level group of digital networks in the hierarchy of plant networks.

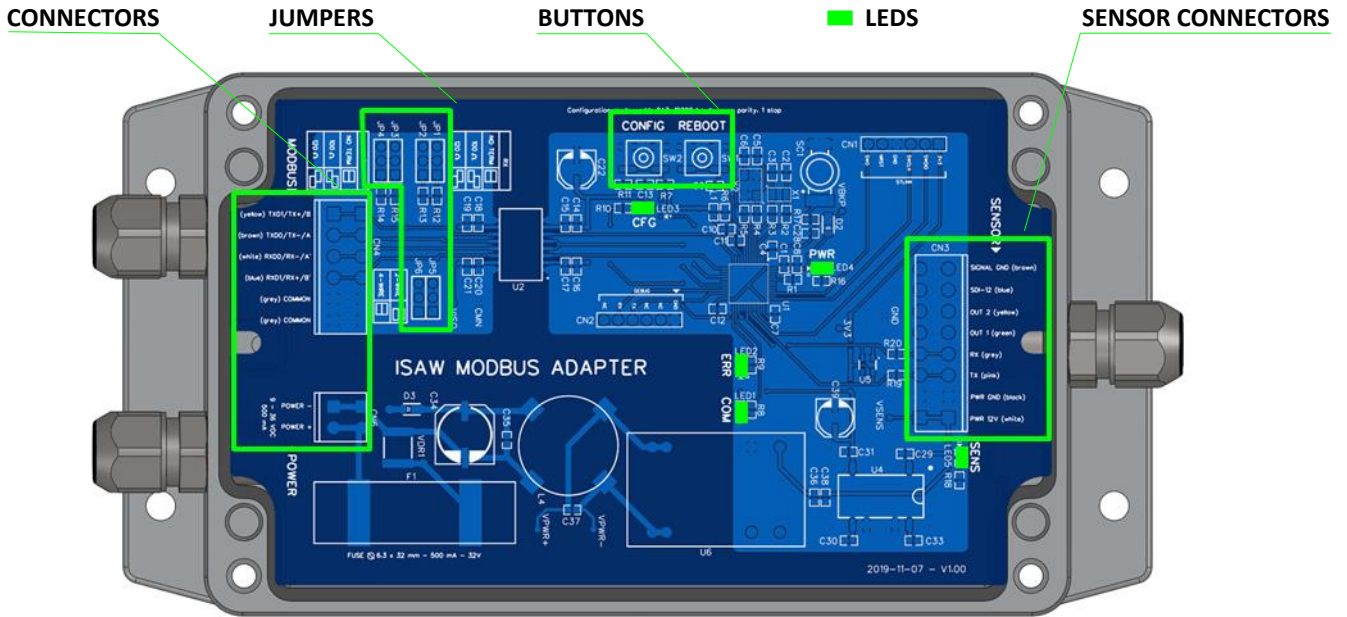
CHARACTERISTICS	
Protocol	MODBUS RTU (V1.1b3)
Physical Layer	EIA/TIA-485 (RS485) 2-wire and 4-wire
Galvanic isolation	Power 3kV RMS, Bus 5kV RMS
Unit load	1/8 unit load, up to 256 nodes on the bus
Startup time	1 s
Power supply	9 to 36 VDC (Typ. 100 mA, Max. 500 mA)
Implemented function codes	0x04 Read Input Registers 0x03 Read Holding Registers 0x06 Write Single Register 0x10 Write Multiple Registers 0x64 Pass-through 0x08 Diagnostic 0x17 Report Server ID
Configurable Baud rate	9600, 19200, 38400, 57600, 115200, 128000, 256000
Configurable Parity	No, Odd, Even
Configurable Stop Bit	1 or 2
Configurable Address	1 to 247

DEFAULT COMMUNICATION PARAMETERS	
Address	247
Baudrate	19200 bauds
Parity	Even
Stop bits	1 bit
Response timeout	1000 ms

MECHANICAL DATA	
Material	Aluminum box Nickel-plated brass cable glands
Protection	IP 68 (up to 10 bar) IP 69 for the cable glands
Dimensions (L×W×H)	160 mm × 100 mm × 60 mm Box thickness: 2.5 mm
Operating temperature	-40°C / +85°C (most sensitive electronic component)
Manufacturer references	Box: Bud Industries, product number AN-2866-AB Cable glands: AGRO, product number 1160.12.065



Description



■ BUTTONS

BUTTONS	
REBOOT	Restarts the Modbus adapter with the Holding register's parameters. Note: Switching the power OFF/ON also restarts the Modbus adapter.
CONFIG	Holding the CONFIG button pressed while starting (or restarting) the Modbus adapter starts (or restarts) the adapter with the default communication parameters (cf. previous page). Note: This operation does not change the parameters stored in the Holding register.

■ LEDS

LEDS	
COM	Flashing during a Modbus communication.
ERR	Flashing when a Modbus communication error occurs. Steady when a critical error occurs requiring a restart.
CFG	Flashing when the sensor is in CONFIG mode (started with the CONFIG button pressed).
PWR	Steady when input power OK.
SENS	Steady when output 12 V sensor power OK.

■ CONNECTORS

SENSOR			
#	Name	Color	Description
1	PWR 12V	White	Power output 12VDC – 300 mA
2	PWR GND	Black	Power ground
3	TX	Pink	Serial input 3V3
4	RX	Grey	Serial output 3V3
5	OUT 1	Green	Not connected
6	OUT 2	Yellow	Not connected
7	SDI-12	Blue	Not connected
8	Signal GND	Brown	Not connected

POWER			
#	Name	Color	Description
1	PWR -	Black	Power ground
2	PWR +	Red	Power input 9...36VDC (500 mA)

MODBUS 4-WIRE			
#	Name	Color	Description
1	TXD1/TX+/B	Yellow	Output terminal 1, Vb voltage (Vb > Va for binary 1)
2	TXD0/TX-/A	Brown	Output terminal 0, Va voltage (Va > Vb for binary 0)
3	RXD0/RX-/A'	White	Input terminal 0, Va' voltage (Va' > Vb' for binary 0)
4	RXD1/RX+/B'	Blue	Input terminal 1, Vb' voltage (Vb' > Va' for binary 1)
5	COMMON	Grey	Signal ground
6	COMMON	Grey	Signal ground

MODBUS 2-WIRE			
#	Name	Color	Description
1	TXD1/TX+/B	Yellow	Transceiver terminal 1, Vb voltage (Vb > Va for binary 1)
2	TXD0/TX-/A	Brown	Transceiver terminal 0, Va voltage (Va > Vb for binary 0)
3	RXD0/RX-/A'	White	Not connected
4	RXD1/RX+/B'	Blue	Not connected
5	COMMON	Grey	Signal ground
6	COMMON	Grey	Signal ground

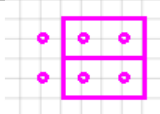
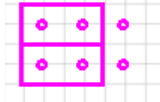
Note: TXD0-RXD0 and TXD1-RXD1 are connected.

The polarity of the "A" and "B" wires can be reversed. Please check in the datasheet of your RS485/RS422 converter the polarity "+" or "-" ("1" or "0") affected to the "A" and "B" labels.

If you don't find this information in your converter documentation, try to plug "A" and "B" wires and if you get no communication, just invert the wiring.

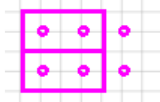
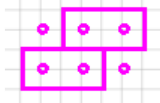
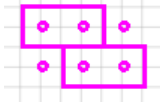
(To avoid confusion, this designation will be removed in the future versions of the Modbus adapter).

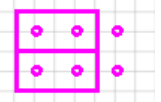
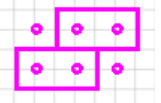
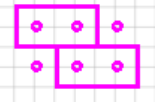
JUMPERS

MODE	
Jumper position	Description
	2-WIRE MODE [default] (TXD0-RXD0 et TXD1-RXD1 are connected)
	4-WIRE MODE

Notes:

1. If the ISAW Modbus adapter node is the last one of the bus, the jumper RX must be set to "100 Ω" or "120 Ω".
2. In 4 wire mode only, if the ISAW Modbus adapter node is the last one of the bus, the jumper TX must be set to "100 Ω" or "120 Ω".

TX TERMINATION	
Jumper position	Description
	NO TERM. [default] No termination resistor on TX pair
	120 Ω Standard 120 Ω termination resistor wired on TX pair
	100 Ω 100 Ω termination resistor wired on TX pair

RX TERMINATION	
Jumper position	Description
	NO TERM. [default] No termination resistor on RX pair
	120 Ω Standard 120 Ω termination resistor wired on RX pair
	100 Ω 100 Ω termination resistor wired on RX pair

Modbus function / Register definition

INPUT Registers

The Input registers contain measurements. The content of these registers is updated each time the sensor sends new measurements to the Modbus adapter.

Modbus function:

- READ INPUT REGISTER (0x04)

FlowCapt FC4, SandFlow SF4 and WindFlow WF4						
Address	Type*	Bytes	Offset	Alias	FC4/SF4	WF4
0x0000	UINT32	4	0	Counter	Flux measurement frame counter	N.A.
0x0002	STRING8	8	2	Unit	Flux measurement unit: "g/m ² /s"	N.A.
0x0006	FLOAT	4	6	Min	Flux measurement minimum	N.A.
0x0008	FLOAT	4	8	Avg	Flux measurement average	N.A.
0x000A	FLOAT	4	10	Max	Flux measurement maximum	N.A.
0x000C	FLOAT	4	12	Std	Flux measurement standard deviation	N.A.
0x000E	STRING8	8	14	Unit	Cumulative flux measurement unit: "g/m ² "	N.A.
0x0012	FLOAT	4	18	Sum	Cumulative flux measurement	N.A.
0x0014	UINT32	4	20	Counter	Wind measurement frame counter	
0x0016	STRING8	8	22	Unit	Wind measurement unit: "km/h"	
0x001A	FLOAT	4	26	Min	Wind measurement minimum	
0x001C	FLOAT	4	28	Avg	Wind measurement average	
0x001E	FLOAT	4	30	Max	Wind measurement maximum	
0x0057	UINT16	2	87	UINT16 Test	Fixed value: 54321 (0xD431)	
0x0058	UINT32	4	88	UNIT32 Test	Fixed value: 1234567890 (0x499602D2)	
0x005A	FLOAT	4	90	FLOAT Test	Fixed value: 3,14159265 (0x40490FDB)	
0x005C	UINT16	2	92	VERmaj	Major version of Modbus adapter firmware (since V1.19)**	
0x005D	UINT16	2	93	VERmin	Minor version of Modbus adapter firmware (since V1.19)**	

* Note: String are zero-padded.

** If not present, Modbus adapter firmware is V1.18

RainFlow RF4						
Address	Type*	Bytes	Offset	Alias	RF4	
0x0000	UINT32	4	0	Counter	Rain measurement frame count	
0x0002	STRING8	8	2	Unit	Rain measurement unit: "mm/h"	
0x0006	FLOAT	4	6	Min	Rain measurement minimum	
0x0008	FLOAT	4	8	Avg	Rain measurement average	
0x000A	FLOAT	4	10	Max	Rain measurement maximum	
0x000C	FLOAT	4	12	Std	Rain measurement stand. deviation	
0x000E	STRING8	8	14	Unit	Cum. rain measurement unit: "mm"	
0x0012	FLOAT	4	18	Sum	Cumulative rain measurement	
0x0020	UINT32	4	32	Counter	Drop measurement frame count	
0x0022	STRING8	8	34	Unit	Drop count unit: "hit"	

RainFlow RF4					
Address	Type*	Bytes	Offset	Alias	RF4
0x0026	UINT32	4	38	Hit count	Drop count
0x0028	STRING8	8	40	Unit	Drop distribution unit: "%"
0x002C	UINT16	2	44	Classe 1	Drop distribution of the class 1
0x002D	UINT16	2	45	Classe 2	Drop distribution of the class 2
0x002E	UINT16	2	46	Classe 3	Drop distribution of the class 3
0x002F	UINT16	2	47	Classe 4	Drop distribution of the class 4
0x0030	UINT16	2	48	Classe 5	Drop distribution of the class 5
0x0031	UINT16	2	49	Classe 6	Drop distribution of the class 6
0x0032	UINT16	2	50	Classe 7	Drop distribution of the class 7
0x0033	UINT16	2	51	Classe 8	Drop distribution of the class 8
0x0034	UINT16	2	52	Classe 9	Drop distribution of the class 9
0x0035	UINT16	2	53	Classe 10	Drop distribution of the class 10
0x0036	UINT16	2	54	Classe 11	Drop distribution of the class 11
0x0037	UINT16	2	55	Classe 12	Drop distribution of the class 12
0x0038	UINT16	2	56	Classe 13	Drop distribution of the class 13
0x0039	UINT16	2	57	Classe 14	Drop distribution of the class 14
0x003A	UINT16	2	58	Classe 15	Drop distribution of the class 15
0x003B	UINT16	2	59	Classe 16	Drop distribution of the class 16
0x003C	UINT16	2	60	Classe 17	Drop distribution of the class 17
0x003D	UINT16	2	61	Classe 18	Drop distribution of the class 18
0x003E	UINT16	2	62	Classe 19	Drop distribution of the class 19
0x003F	UINT16	2	63	Classe 20	Drop distribution of the class 20
0x0040	UINT16	2	64	Classe 21	Drop distribution of the class 21
0x0041	UINT16	2	65	Classe 22	Drop distribution of the class 22
0x0042	UINT16	2	66	Classe 23	Drop distribution of the class 23
0x0043	UINT16	2	67	Classe 24	Drop distribution of the class 24
0x0044	UINT16	2	68	Classe 25	Drop distribution of the class 25
0x0045	UINT16	2	69	Classe 26	Drop distribution of the class 26
0x0046	UINT16	2	70	Classe 27	Drop distribution of the class 27
0x0047	UINT32	4	71	Counter	Hail measurement frame counter
0x0049	STRING8	8	73	Unit	Hail measurement unit: "hit"
0x004D	UINT32	4	77	Hit count	Hail measurement
0x004F	STRING8	8	79	Unit	Hail measurement rate unit: "hit/s"
0x0053	FLOAT	4	83	Mean Rate	Hail measurements mean rate
0x0055	FLOAT	4	85	Max Rate	Hail measurement max rate
0x0057	UINT16	2	87	UINT16 Test	Fixed Value: 54321 (0xD431)
0x0058	UINT32	4	88	UNIT32 Test	Fixed value: 1234567890 (0x499602D2)
0x005A	FLOAT	4	90	FLOAT Test	Fixed value: 3,14159265 (0x40490FDB)
0x005C	UINT16	2	92	VERmaj	Major version of Modbus adapter firmware (since V1.19)**
0x005D	UINT16	2	93	VERmin	Minor version of Modbus adapter firmware (since V1.19)**

* Note: String are zero-padded.

** If not present, Modbus adapter firmware is V1.18

HailFlow HF4					
Address	Type*	Bytes	Offset	Alias	HF4
0x0000	UINT32	4	0	Counter	Hail measurement frame count
0x0002	STRING8	8	2	Unit	Hail measurement unit: "hit/h"
0x0006	FLOAT	4	6	Min	Hail measurement minimum
0x0008	FLOAT	4	8	Avg	Hail measurement average
0x000A	FLOAT	4	10	Max	Hail measurement maximum
0x000C	FLOAT	4	12	Std	Hail measurement stand. deviation
0x000E	STRING8	8	14	Unit	Cum. hail measurement unit: "hit"
0x0012	FLOAT	4	18	Sum	Cumulative hail measurement
0x0020	UINT32	4	32	Counter	Hailstone measurement frame count
0x0022	STRING8	8	34	Unit	Hailstone count unit: "hit"
0x0026	UINT32	4	38	Hit count	Hailstone count
0x0028	STRING8	8	40	Unit	Hailstone distribution unit: "%"
0x002C	UINT16	2	44	Classe 1	Hailstone distribution of the class 1
0x002D	UINT16	2	45	Classe 2	Hailstone distribution of the class 2
0x002E	UINT16	2	46	Classe 3	Hailstone distribution of the class 3
0x002F	UINT16	2	47	Classe 4	Hailstone distribution of the class 4
0x0030	UINT16	2	48	Classe 5	Hailstone distribution of the class 5
0x0031	UINT16	2	49	Classe 6	Hailstone distribution of the class 6
0x0032	UINT16	2	50	Classe 7	Hailstone distribution of the class 7
0x0033	UINT16	2	51	Classe 8	Hailstone distribution of the class 8
0x0034	UINT16	2	52	Classe 9	Hailstone distribution of the class 9
0x0035	UINT16	2	53	Classe 10	Hailstone distribution of the class 10
0x0036	UINT16	2	54	Classe 11	Hailstone distribution of the class 11
0x0037	UINT16	2	55	Classe 12	Hailstone distribution of the class 12
0x0038	UINT16	2	56	Classe 13	Hailstone distribution of the class 13
0x0039	UINT16	2	57	Classe 14	Hailstone distribution of the class 14
0x003A	UINT16	2	58	Classe 15	Hailstone distribution of the class 15
0x0057	UINT16	2	87	UINT16 Test	Fixed Value: 54321 (0xD431)
0x0058	UINT32	4	88	UNIT32 Test	Fixed value: 1234567890 (0x499602D2)
0x005A	FLOAT	4	90	FLOAT Test	Fixed value: 3,14159265 (0x40490FDB)
0x005C	UINT16	2	92	VERmaj	Major version of Modbus adapter firmware (since V1.19)**
0x005D	UINT16	2	93	VERmin	Minor version of Modbus adapter firmware (since V1.19)**

* Note: String are zero-padded.

** If not present, Modbus adapter firmware is V1.18

HOLDING Registers

Holding registers are mainly used to configure the Modbus adapter communication.

Note: Restart the Modbus adapter after changing the configuration.

Modbus functions:

- READ HOLDING REGISTERS (0x03)
- WRITE SINGLE REGISTER (0x06)
- WRITE MULTIPLE REGISTERS (0x10)

Address	Type	Bytes	Offset	Name	Values
0x0000	UINT32	4	0	Serial speed	9600, 19200 [default], 38400, 57600, 115200, 128000, 256000
0x0002	UINT16	2	2	Parity	0: No parity, 1: Even [default], 2: Odd
0x0003	UINT16	2	3	Stop Bit	1 [default] or 2 (if no parity)
0x0004	UINT16	2	4	Device address	1 to 247 [default]
0x0005	UINT16	2	5	Response timeout (ms)	Default: 1000
Total bytes:		12			
Nb. REG:		6			

DATA TYPE Format

- **UINT16 (Big Endian)**

High Byte	Low Byte
-----------	----------
- **UINT32 (Big Endian)**

High Byte			Low Byte
-----------	--	--	----------
- **FLOAT (IEEEE-754)**

SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM
----------	----------	----------	----------

(S: Sign, E: Exponent, M: Mantissa)
- **RAW**

Char 1	Char 2	Char 3	...
--------	--------	--------	-----

EXAMPLES:

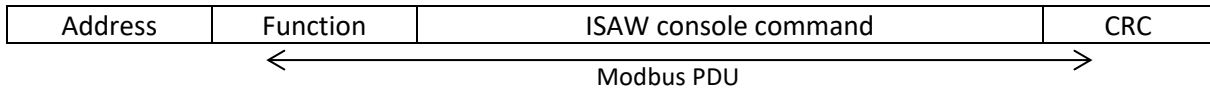
	Decimal	Hexadecimal	Register N	Register N+1
■ UINT16 (Big Endian)	54321	0xD431	0xD431	
■ UINT32 (Big Endian)	1234567890	0x499602D2	0x02D2	0x4996
■ FLOAT (IEEEE-754)	3.14159265	0x40490FDB	0x0FDB	0x4049
■ RAW	"hit"	0x68697400	0x6869	0x7400

Modbus function PASS-THROUGH (0x64)

This user-defined Modbus function, available for Modbus firmware version V1.19 and higher, allows to send an ISAW command through the Modbus (see User Guide > Serial communication for more information about the ISAW console commands). For example, use this function with the “set” or “get” ISAW command to access the sensor’s configuration.

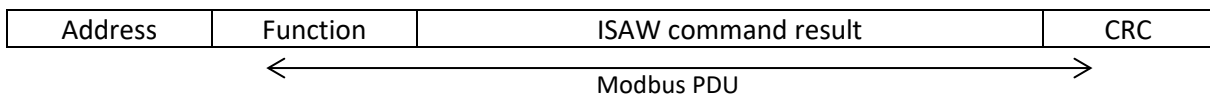
Note: The use of this function is limited by the request and answer lengths.

Request



Field	Size (bytes)	Description
Address	1	Device address (target)
Function	1	0x64
ISAW Command	N	ISAW command as ASCII string
CRC	2	CRC16

Response



Field	Size (bytes)	Description
Address	1	Device address (same as request)
Function	1	0x64
Result	N	ISAW command result as ASCII string
CRC	2	CRC16

Note: The response timeout of this command must be > 2 seconds to allow sensor wake-up.

Modbus frame examples

Raw examples of Modbus communication.

■ READ HOLDING REGISTER (all registers)

TX > 0xF7 0x03 0x00 0x00 0x00 0x06 0xD1 0x5E

RX > 0xF7 0x03 0x0C 0x4B 0x00 0x00 0x00 0x00 0x01 0x00 0x01 0x00 0xF7 0x03 0xE8 0x9D

RX > 0x9E

■ READ INPUT REGISTERS (All registers)

TX > 0xF7 0x04 0x00 0x00 0x00 0x5E 0x65 0x64

RX > 0xF7 0x04 0xBC 0x00 0x01 0x00 0x00 0x68 0x69 0x74 0x2F 0x73 0x00 0x00 0x00 0x00

RX > 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x68

RX > 0x69 0x74 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

RX > 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

RX > 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x68 0x69 0x74 0x00 0x00 0x00 0x00 0x00

RX > 0x00 0x00 0x00 0x25 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

RX > 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

RX > 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

RX > 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

RX > 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

RX > 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

RX > 0x00 0xD4 0x31 0x02 0xD2 0x49 0x96 0x0F 0xDB 0x40 0x49 0x00 0x01 0x00 0x13 0x14

RX > 0xC6

■ READ INPUT REGISTERS (read adapter version only)

Available on firmware version V1.19 or higher. If you receive a Modbus exception, firmware version is V1.18.

TX > 0xF7 0x04 0x00 0x5C 0x00 0x02 0xA5 0x4F

RX > 0xF7 0x04 0x04 0x00 0x01 0x00 0x13 0x7D 0x86

■ PASSTHROUGH (get avg-a)

TX > 0xF7 0x64 0x67 0x65 0x74 0x20 0x61 0x76 0x67 0x2D 0x61 0xEC 0x5D

RX > 0xF7 0x64 0x4F 0x4B 0x3D 0x31 0xD3 0x12

■ PASSTHROUGH (get avg-c)

TX > 0xF7 0x64 0x67 0x65 0x74 0x20 0x61 0x76 0x67 0x2D 0x63 0x6D 0x9C

RX > 0xF7 0x64 0x4F 0x4B 0x3D 0x32 0x93 0x13

■ PASSTHROUGH (get avg-m)

TX > 0xF7 0x64 0x67 0x65 0x74 0x20 0x61 0x76 0x67 0x2D 0x6D 0xEC 0x58

RX > 0xF7 0x64 0x4F 0x4B 0x3D 0x38 0x13 0x14

■ PASSTHROUGH (get hw-sn)

TX > 0xF7 0x64 0x67 0x65 0x74 0x20 0x68 0x77 0x2D 0x73 0x6E 0x68 0x12

RX > 0xF7 0x64 0x4F 0x4B 0x3D 0x30 0x30 0x32 0x45 0x30 0x30 0x34 0x30 0x33 0x36 0x33

RX > 0x32 0x33 0x30 0x33 0x36 0x30 0x43 0x34 0x37 0x33 0x34 0x33 0x31 0x1E 0xDF

■ PASSTHROUGH (get fw-build)

TX > 0xF7 0x64 0x67 0x65 0x74 0x20 0x66 0x77 0x2D 0x62 0x75 0x69 0x6C 0x64 0x0A 0xE0

RX > 0xF7 0x64 0x4F 0x4B 0x3D 0x4A 0x75 0x6C 0x20 0x32 0x32 0x20 0x32 0x30 0x32 0x30

RX > 0x20 0x61 0x74 0x20 0x31 0x36 0x3A 0x31 0x37 0x3A 0x30 0x36 0x20 0x62 0x79 0x20

RX > 0x47 0x43 0x43 0x20 0x37 0x2E 0x32 0x2E 0x31 0xF8 0x2C

■ PASSTHROUGH (set avg-m 8)

TX > 0xF7 0x64 0x73 0x65 0x74 0x20 0x61 0x76 0x67 0x2D 0x6D 0x20 0x38 0x14 0xD8

RX > 0xF7 0x64 0x4F 0x4B 0x07 0x88

■ PASSTHROUGH (reboot)

TX > 0xF7 0x64 0x72 0x65 0x62 0x6F 0x6F 0x74 0x6F 0xC7

RX > 0xF7 0x64 0x4F 0x4B 0x07 0x88

NOTES: Some useful tools to manually forge and decode your Modbus frames:

- Compute CRC16 for Modbus online: <https://crccalc.com/?method=CRC-16/MODBUS>
 - Convert ASCII to HEX online: <https://www.rapidtables.com/convert/number/ascii-to-hex.html>
 - Convert HEX to ASCII online: <https://www.rapidtables.com/convert/number/hex-to-ascii.html>
 - Parse and decode MODBUS frame online (remove 0x prefix on the previous examples): <https://rapidscada.net/modbus/>
-

TEST PROCEDURE

Example of communication between an ISAW sensor and a PC using the Modbus Adapter.

Hardware setup

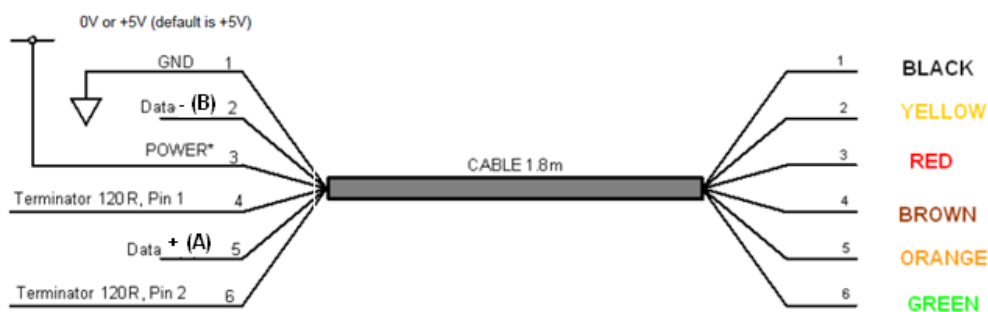
In this example we have an ISAW sensor (model HailFlow HF4), and we want to communicate with it through a PC and via the Modbus adapter.

The sensor is connected directly to the “SENSOR” connector of the Modbus adapter as per Picture 2.

To power both the Modbus adapter and the sensor we simply use the white and black wires of the UDONG accessory, connected to “POWER” connector of the Modbus adapter. We connect UDONG to a USB hub that is connected to the PC (see pictures 1, 3 and 8). (It can also be connected directly to the PC without a hub).

To communicate between the MODBUS ADAPTER and the PC we use any RS485/422 adapter (ex: FTDI USB-RS485-WE-1800-BT) connected to “MODBUS” connector of the MODBUS ADAPTER. The RS485/422 adapter is also connected to the USB hub. (It can be connected directly to the PC without a hub).

USB to RS485 Serial Converter Cable

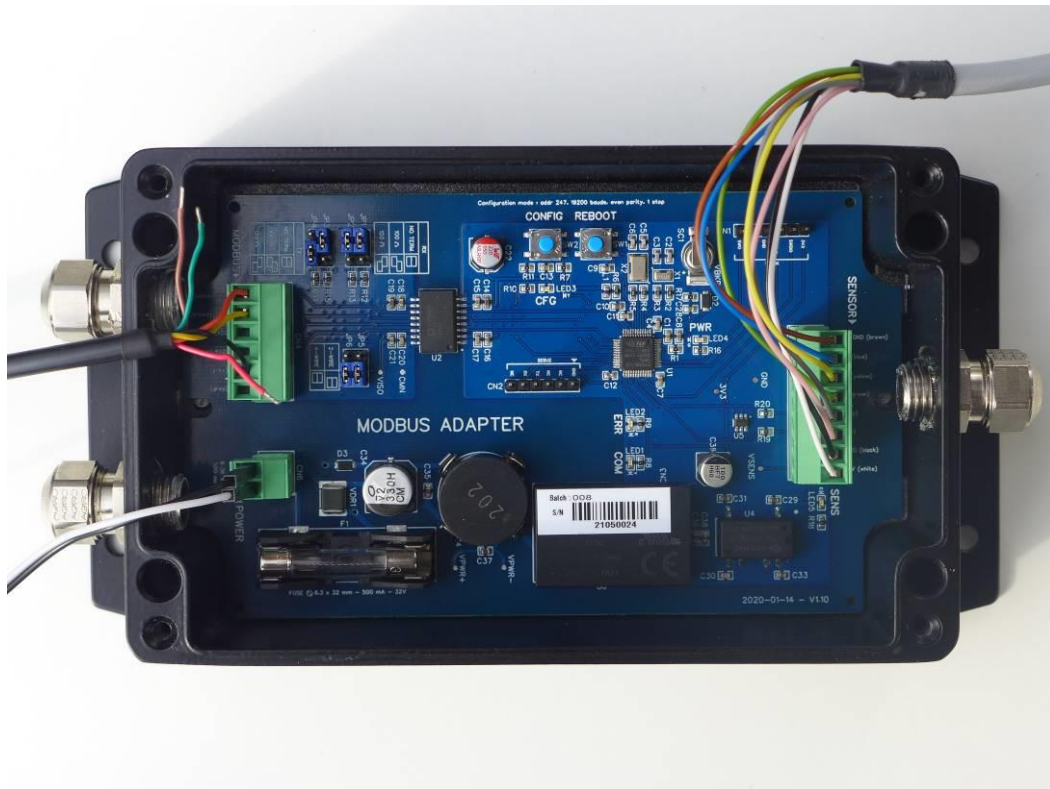


https://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_USB_RS485_CABLES.pdf

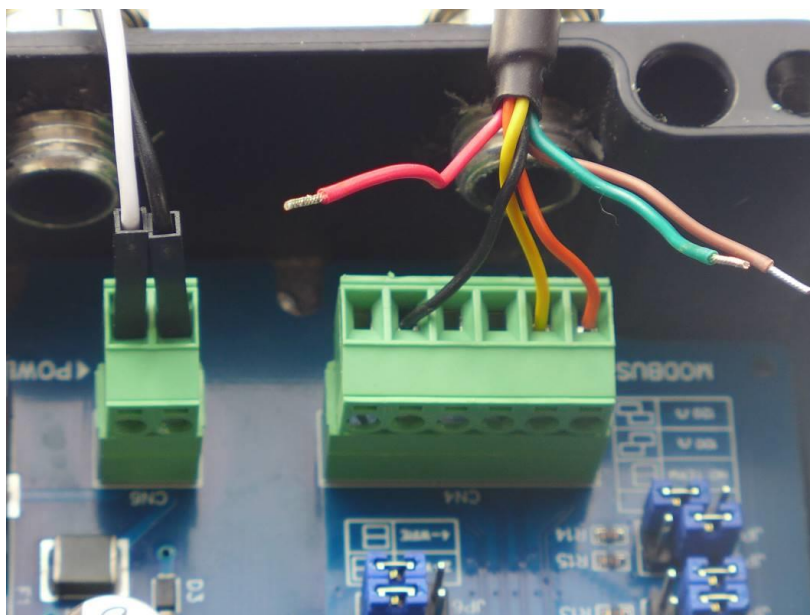
- Download the driver vs Operating System (if needed): <https://ftdichip.com/drivers/vcp-drivers>
- Then on the MODBUS ADAPTER side, we connect the USB-RS485 FTDI as per pictures 2 and 3. That is, only wires black, yellow and orange (thus leaving the red, green and brown unused).
- We set the jumpers of the MODBUS ADAPTER as per picture 4.
- We set the UDONG switch to the ON position which activates the UDONG green led and two green LEDs on the MODBUS ADAPTER (see picture 9 and 10).
- The hardware setup is complete we can now communicate.



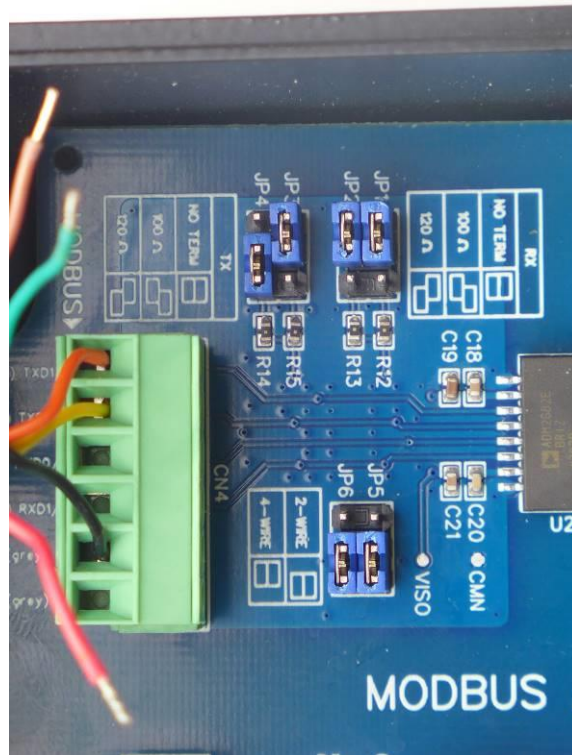
Picture 1: Hardware setup



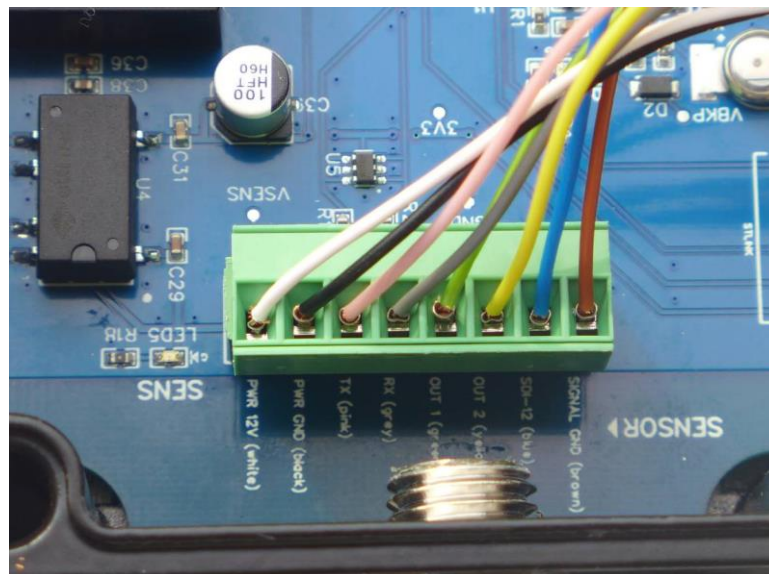
Picture 2: Modbus adapter connectors with MODBUS and POWER connectors on the left and SENSOR connector on the right



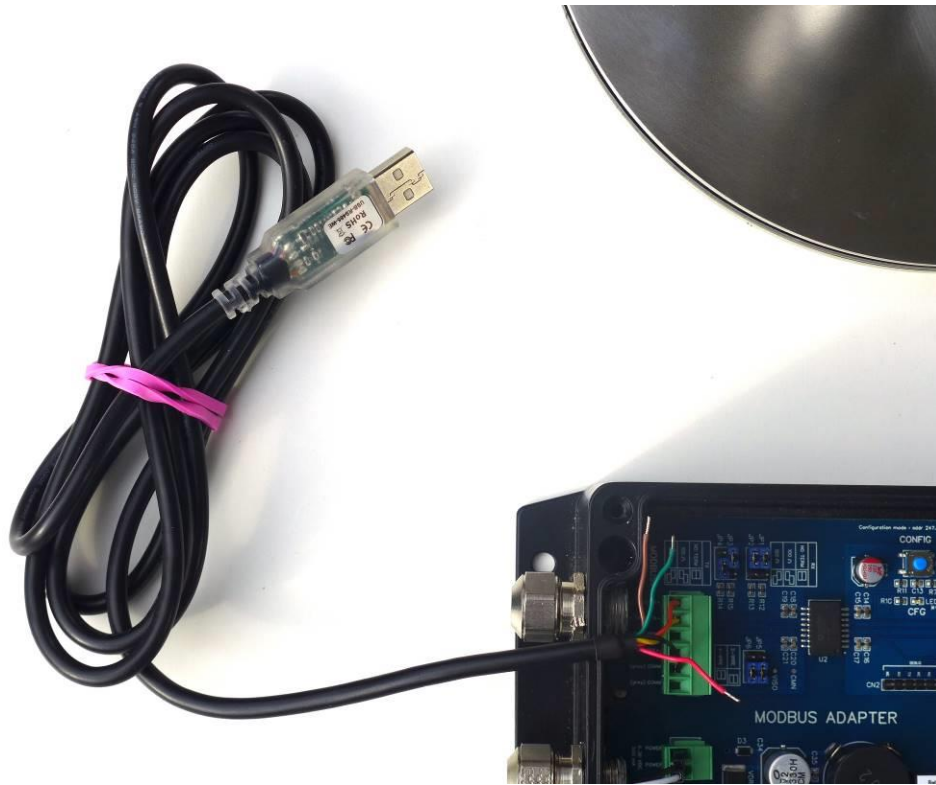
Picture 3: POWER connector (on the left), MODBUS connector (on the right)



Picture 4: Jumpers



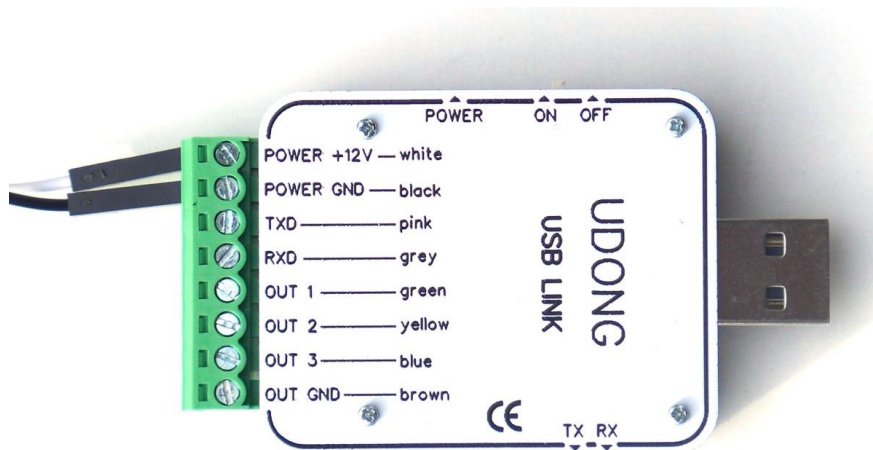
Picture 5: SENSOR connector



Picture 6: RS485/422 adapter connected to the MODBUS connector of the Modbus adapter



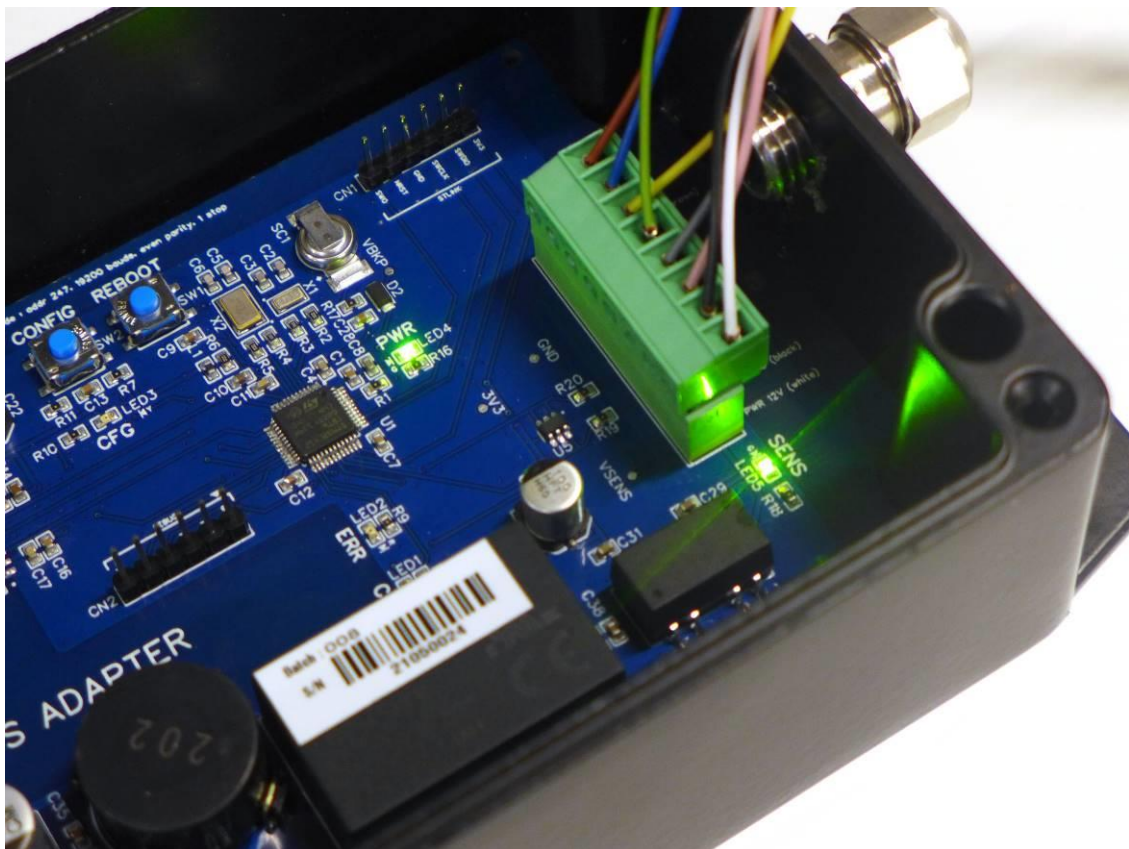
Picture 7: RS485/422 adapter



Picture 8: UDONG accessory



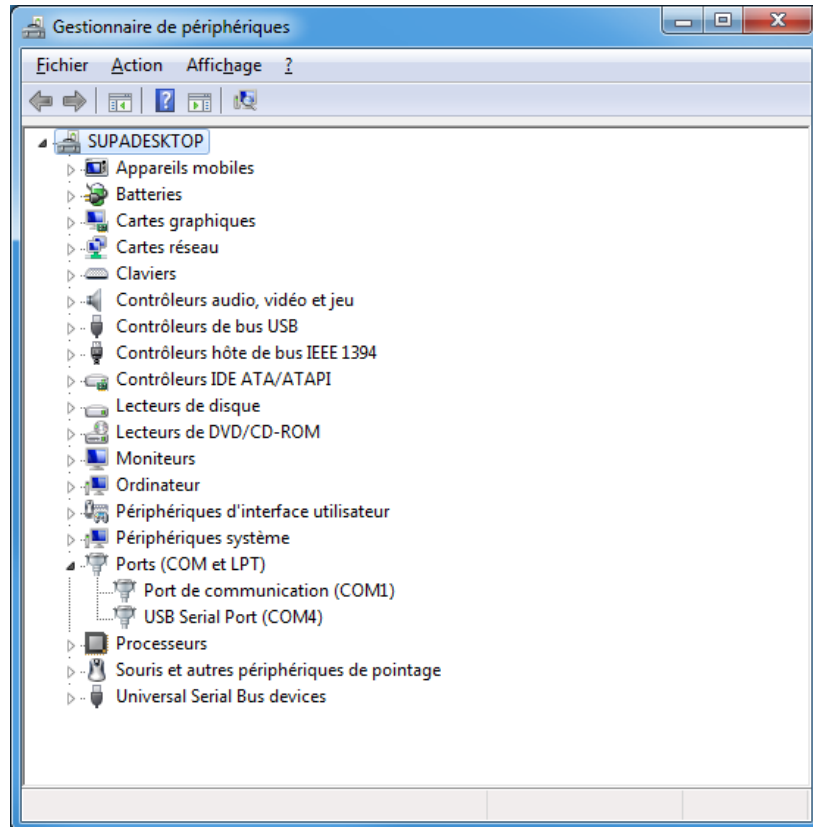
Picture 9: USB hub with UDONG accessory on the left



Picture 10: Modbus adapter LEDs

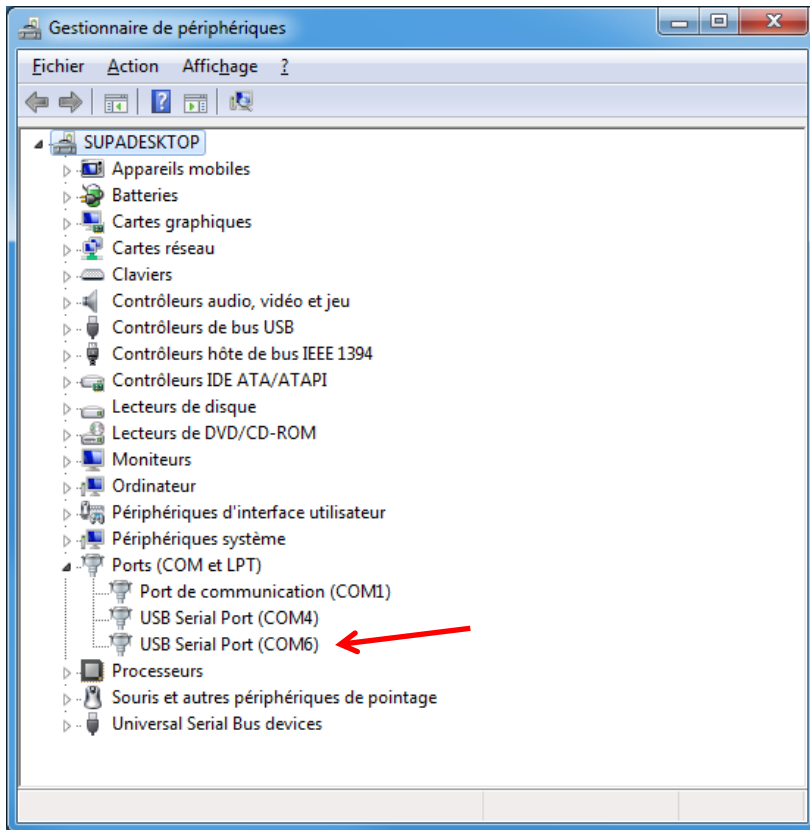
Communicating with the sensor

- Download a communication software, for example QModMaster. QModMaster is a free Qt-based implementation of a ModBus master application. A graphical user interface allows easy communication with ModBus RTU and TCP slaves. QModMaster also includes a bus monitor for examining all traffic on the bus. See <https://sourceforge.net/projects/qmodmaster/>
- Open the « Device Manager ».

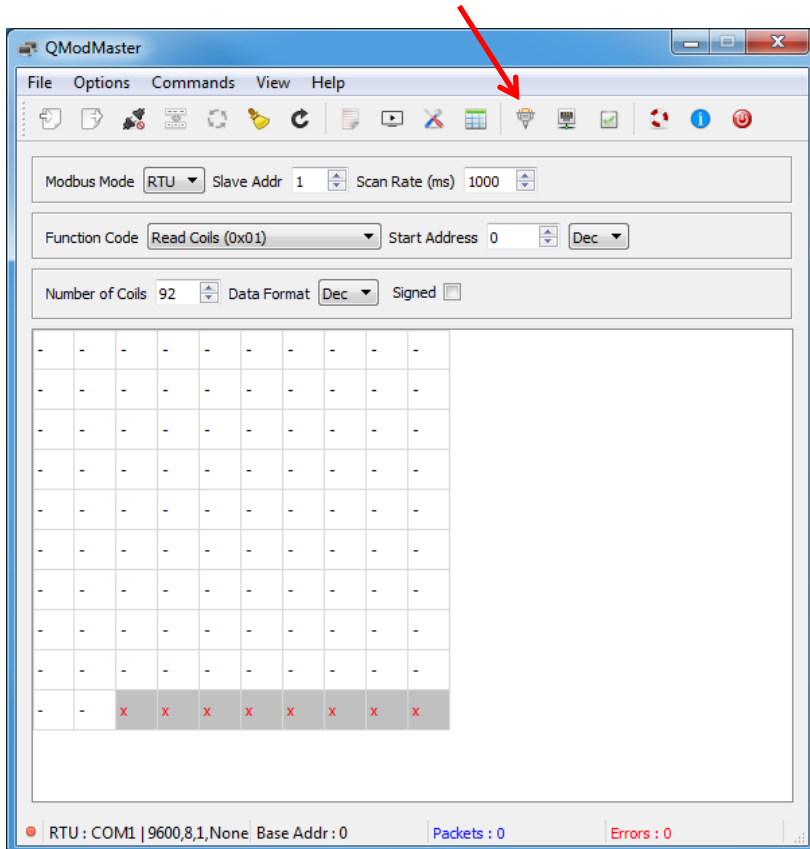


- Plug the RS485/422 adapter USB connector to the USB hub or PC and install the required drivers.

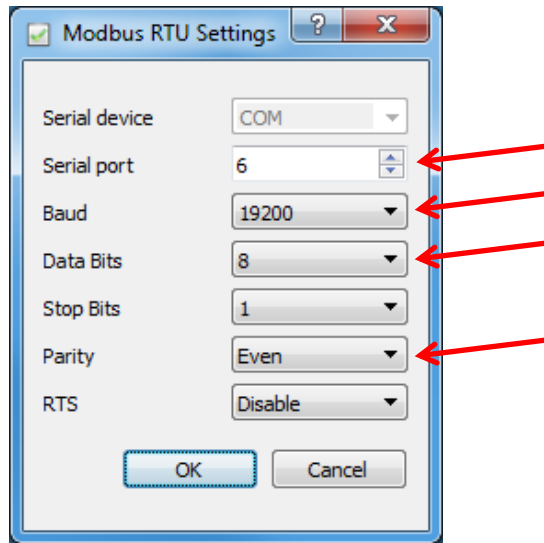
- When the drivers are successfully installed, a new serial port appears (here: COM6).



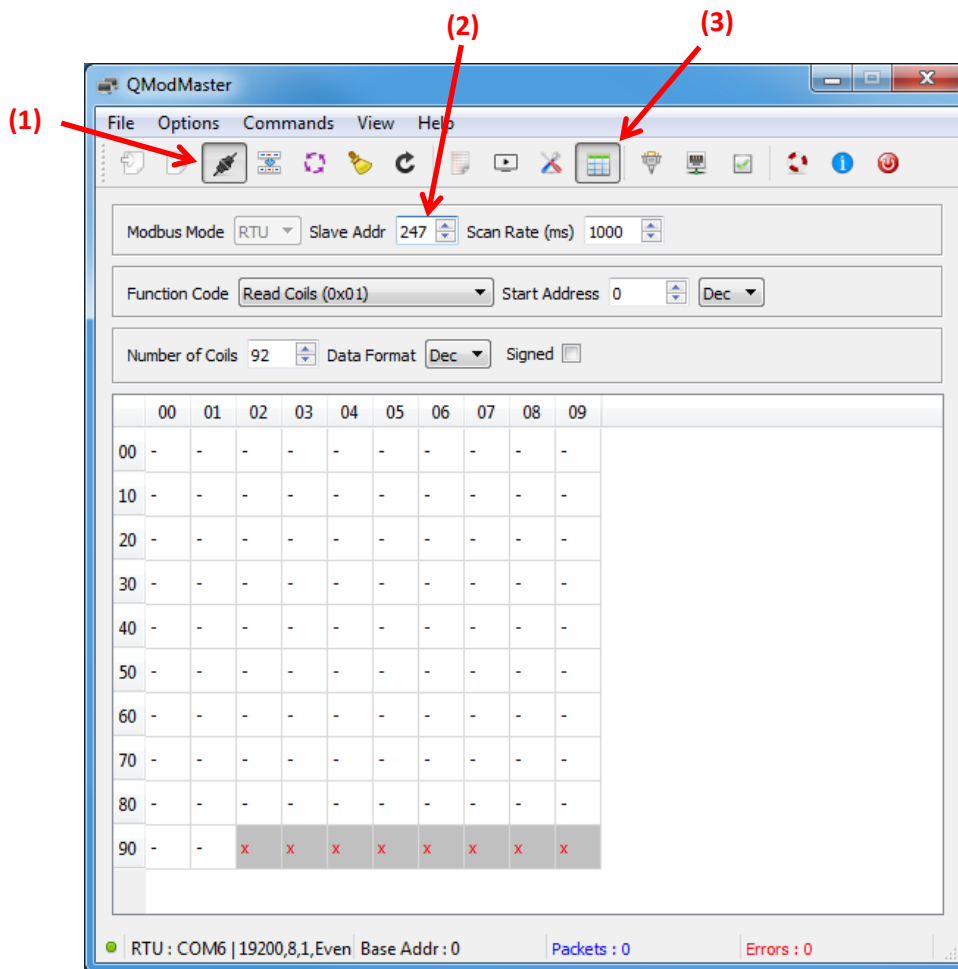
- Start QModMaster and open the Configuration Panel.



- Set the serial communication parameters. Use the relevant serial port number (here: 6).



- (1) Open the QModMaster serial port.
- (2) Set the Modbus address (e.g. 247).
- (3) Display grid header if needed.

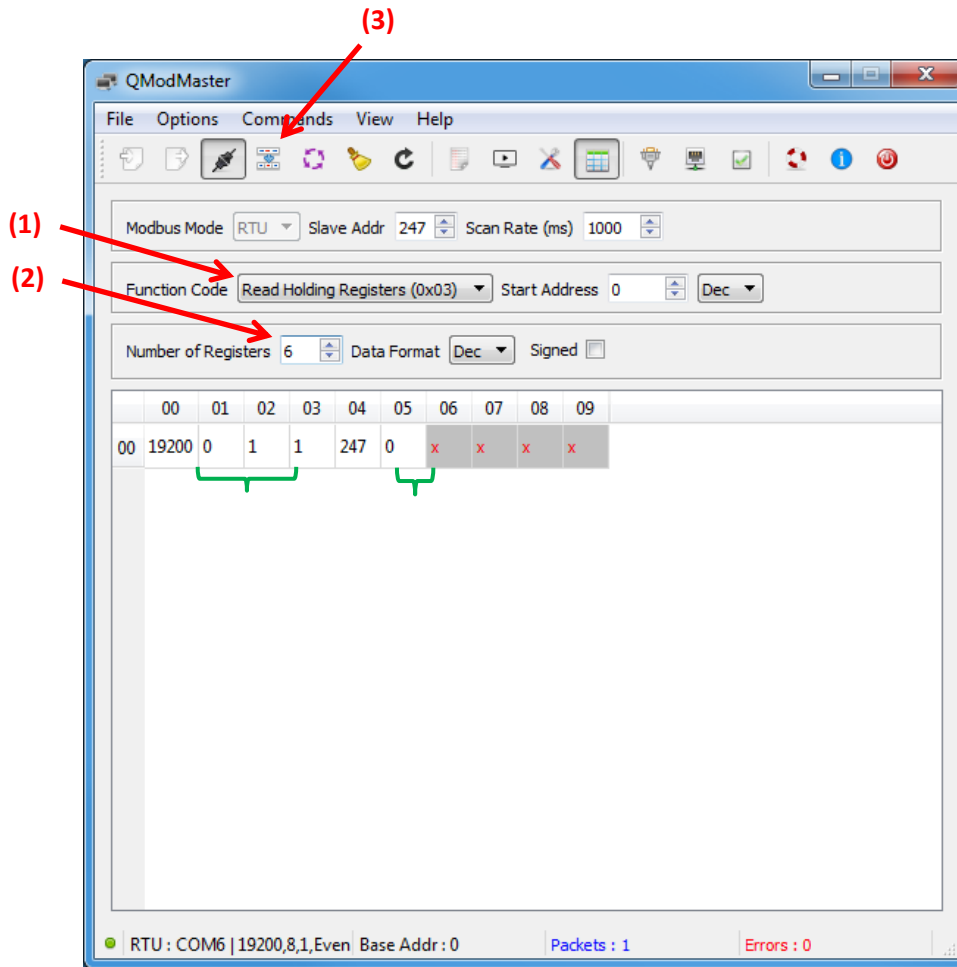


Tip : If you receive the « Connection failed » message below, your serial port is probably already opened by another application. Stop the other application and retry.

Connection failed
Could not connect to serial port.



- Try to read the configuration:
 - (1) Select the function code « Read Holding Registers ».
 - (2) Set the number of registers (e.g., 6 for the whole configuration).
 - (3) Click on « Read/Write » button.



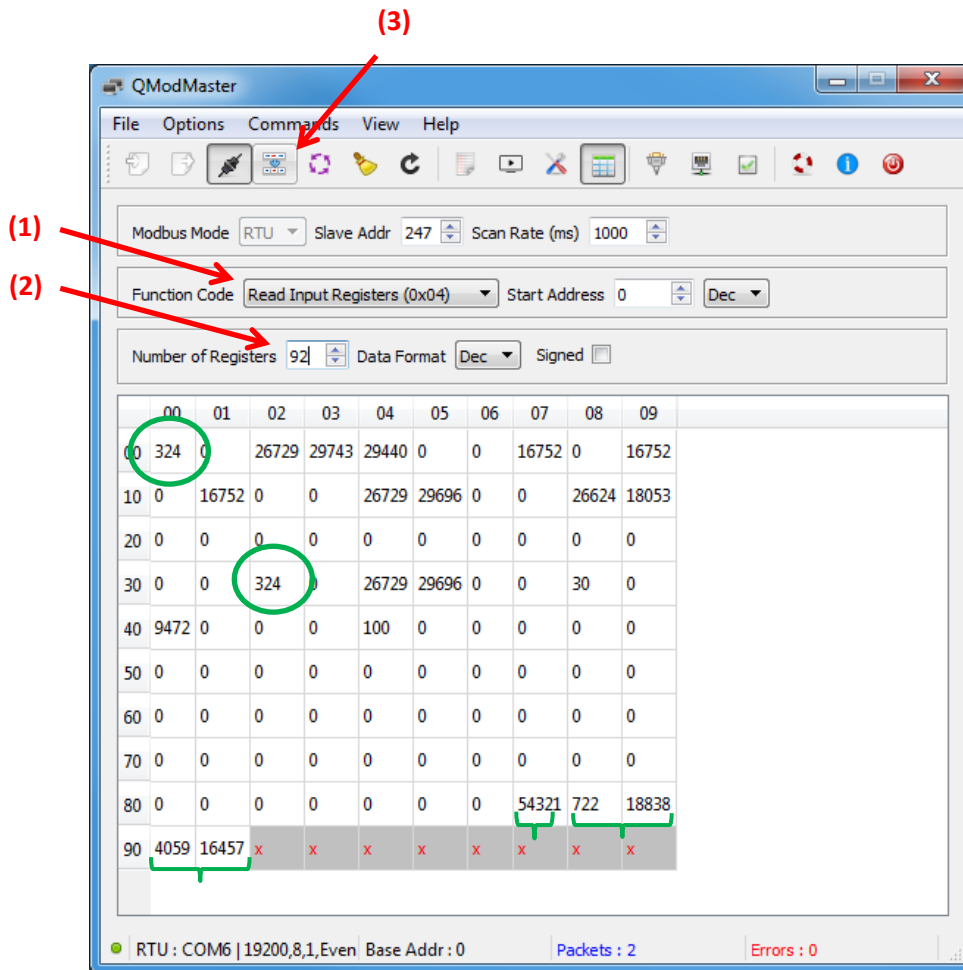
- You can see for example the « Baudrate » (Register 0) and « Address » (Register 4). To understand the registers definitions, values and formats please refer to the Modbus short notice.

Tip : If you receive « Read data failed » message below (Timeout or CRC), please check your hardware wiring (especially data wire swapping), jumper position and QModMaster serial configuration (especially baud rate and parity).

Read data failed.
Error : Timeout



- Try to read the measurements:
 - Select the function code « Read Input Registers »
 - Set the number of registers (e.g. 92 for the whole measurement) and click on « Read / Write » button



- You can see for example « Frame counters » (Registers 0 & 32) and « Test fixed values » (Register 87,88 & 90).
To understand the registers definitions, values and formats please refer to the Modbus short notice.

Tip: The interval between two measurements of the sensor depends on the sensor's "Averaging" setting: by default, the sensor sends one measurement every 600 seconds, i.e. a refresh of the displayed data every 10 minutes. To get more frequent refresh of the displayed data, you can change the "Averaging" parameters "Acquisition (A)", "Cycle (C)" and "Measure (M)" of the sensor. To do so, connect the sensor to a PC using the UDONG accessory and refer to the "Averaging panel" section of the User Guide. Example for quick test: set A=1s, C=2s and M=4s. In this case, do not forget to re-set the "Averaging" configuration of your sensor according to your needs after this test.

UPDATE THE MODBUS ADAPTER'S FIRMWARE

IAV Technologies constantly improves its products and provides upgrades of the ISAW firmwares for the sensors and accessories.

This chapter describes the procedure for quickly upgrading the Modbus adapter's firmware.

Hardware setup

You will need a USB-to-RS485 Serial Converter Cable, as well as the USB link accessory, which will be used as a 12V power supply. Use an FTDI TTL-232 cable if you want to know the Modbus adapter version before V1.19).

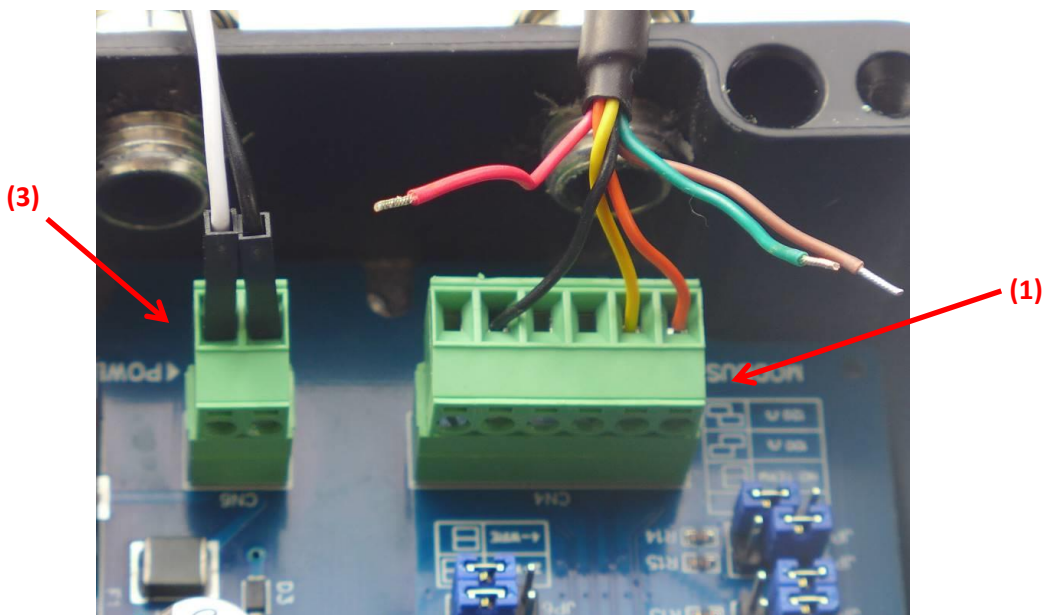


USB-to-RS485 Serial Converter Cable



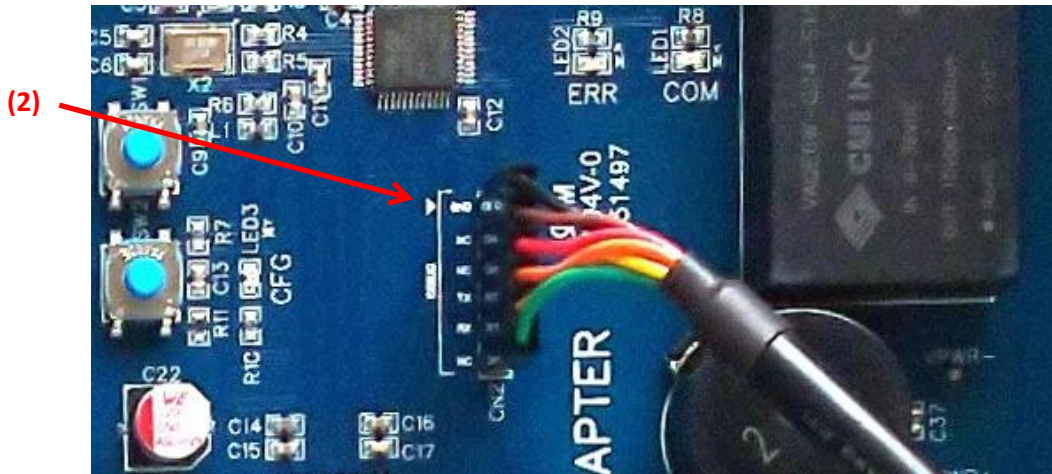
FTDI TTL-232 Cable (optional)

1. To communicate between the Modbus adapter and the PC, use any **RS485/422 adapter** (ex: FTDI USB-RS485-WE-1800-BT) connected to "MODBUS" connector of the MODBUS ADAPTER.



POWER connector (on the left), MODBUS connector (on the right)

- Optional: Connect the Modbus adapter to the PC using the FTDI 232 cable with the black wire (ground) facing the GND mark (t).**



- To power the Modbus adapter, use the white and black wires of the UDONG accessory connected to the "POWER" connector.**




Hardware setup with USB hub and Modbus adapter

Update procedure

Prerequisite: Install the ISAW-Toolbox software suite:

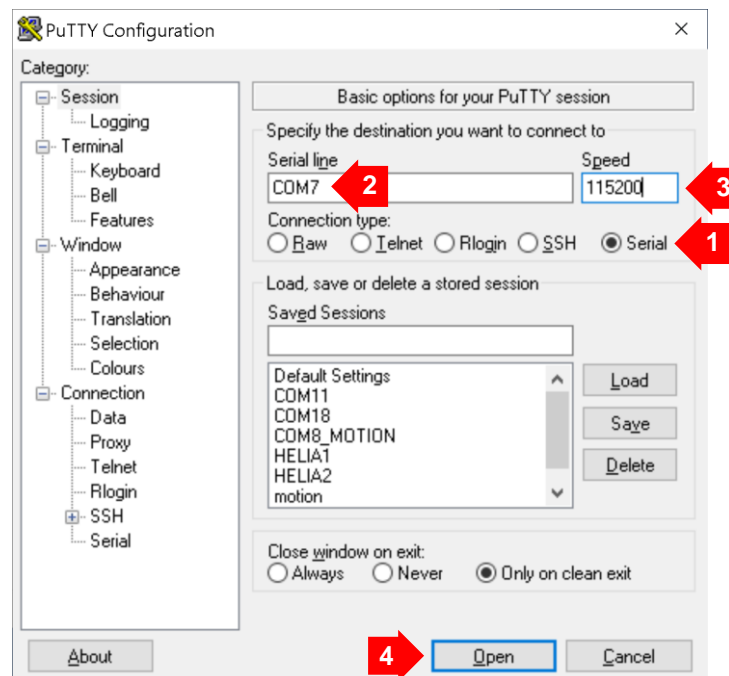
Download the ISAW-Toolbox software suite from the ISAW products website www.isaw-products.com and install the latest version of the toolbox.

Add an ISAW icon on your Desktop  to ensure direct access to the ISAW-Toolbox program.

1. Optional: Connect your terminal software

Open your favourite serial terminal (e.g., Putty¹, TeraTerm, HyperTerminal) on the serial port (e.g.: COM7) at 115200 bauds, 8 bits, 1 stop, no parity.

Tip: To identify the serial port the FTDI 232 cable is connected to, open the Control Panel >> Device Manager >> Ports interface. Unplug then plug the FTDI USB end and check the activated USB serial port.

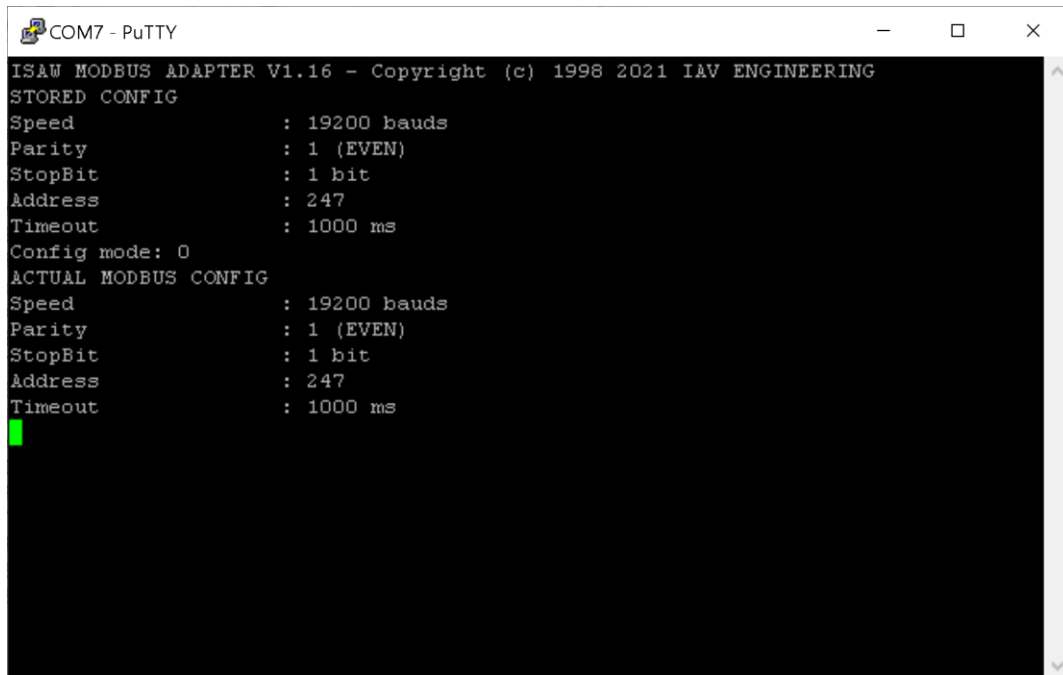


Press the REBOOT button on the Modbus adapter.



¹ Lightweight and non-intrusive "putty.exe" freeware available at <http://www.putty.org>.

The current configuration of the Modbus adapter is displayed, starting with the firmware's version:



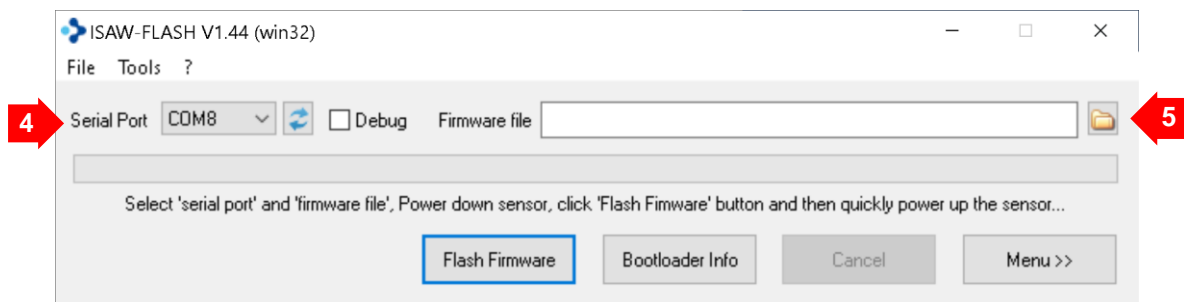
```

COM7 - PuTTY
ISAW MODBUS ADAPTER V1.16 - Copyright (c) 1998 2021 I&V ENGINEERING
STORED CONFIG
Speed      : 19200 bauds
Parity     : 1 (EVEN)
StopBit    : 1 bit
Address    : 247
Timeout    : 1000 ms
Config mode: 0
ACTUAL MODBUS CONFIG
Speed      : 19200 bauds
Parity     : 1 (EVEN)
StopBit    : 1 bit
Address    : 247
Timeout    : 1000 ms

```


2. Open the ISAW-Toolbox by double-clicking on the ISAW icon on your desktop.

3. Start the Flash utility by clicking on the corresponding item.

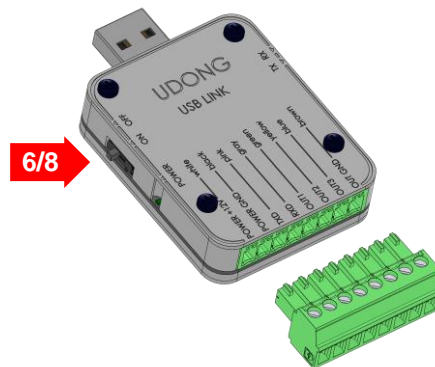


4. Select the FTDI 485 serial port in the Serial Port drop-down list box.

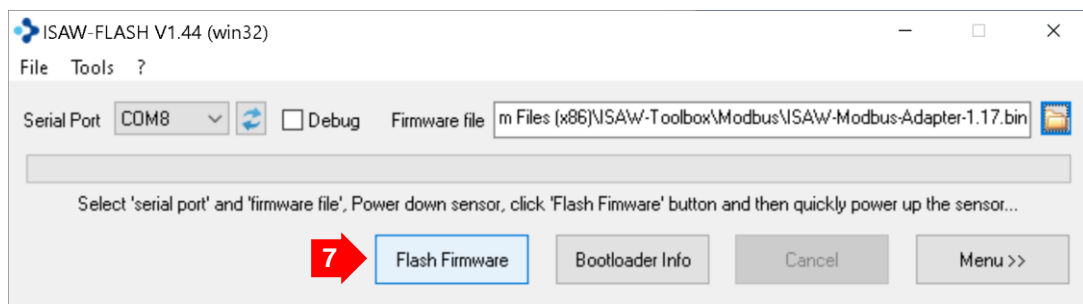
Tip: To identify the serial port the FTDI 485 cable is connected to, open the Control Panel >> Device Manager >> Ports interface. Unplug then plug the FTDI USB end and check the activated USB serial port.

5. **Select the firmware file:** Using the  button, select the last version of the Modbus firmware in the "ISAW-Toolbox > Modbus" directory: ISAW-Modbus-Adapter-x.xx.bin, where x.xx is the version number.

6. **Shut down the Modbus adapter power supply:** Set the USB dongle's power switch to OFF.



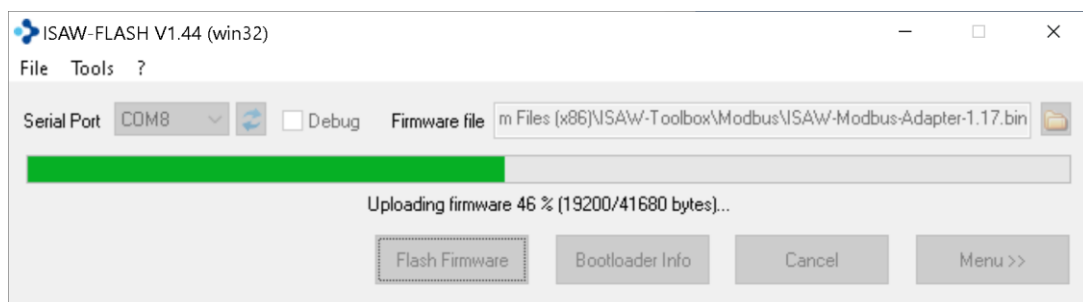
7. **Press the [Flash Firmware] button.**



At this stage, ISAW-Flash will automatically search for a powered device during ten seconds.

8. **Power-on the Modbus adapter** within these ten seconds: Switch the USB dongle's power back to "ON".
9. **Wait** during the firmware upload.

As soon as ISAW-Flash has found the powered Modbus adapter, the firmware upload starts automatically.



WARNING: Do not disconnect the power supply during the firmware upload.

When the firmware upload is successfully completed, ISAW-Flash will display a confirmation message.

10. **The Modbus adapter is now ready to use.**

Tip: Use your serial terminal to check if the firmware version is properly updated (see point 5.).
