webdyn
flexitron group

TITAN ROUTER

Application Note 71

Reading Modbus devices using the advanced Modbus-Expert feature

www.webdyn.com

1. Scenario Details

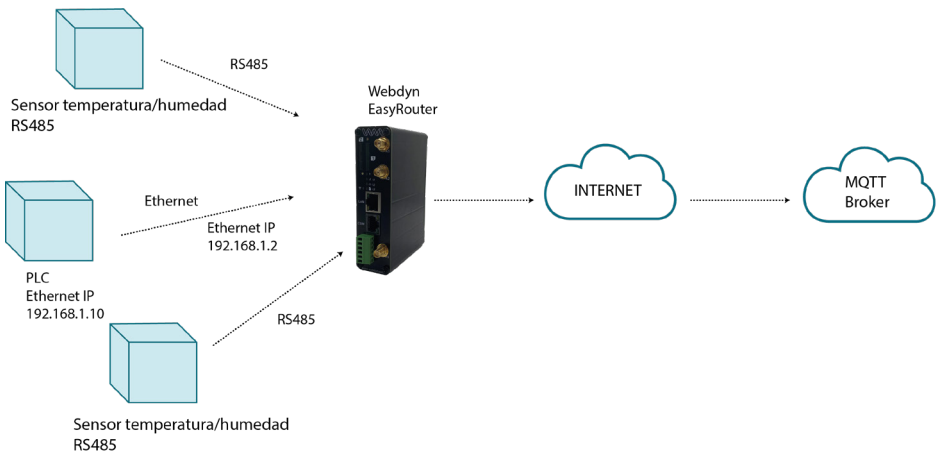
TITAN routers offer all the typical functionalities of 4G/3G/2G routers, as well as a series of added features that make them one of the most advanced routers on the market. One of these additional features is their ability to read Modbus protocol devices. Using the Modbus-Expert function of the Titan router, complex templates can be defined for reading devices, sending telemetry and facilitating interaction between them.

2. Description of the Scenario in the Example

- A Webdyn-Easy-Router device (hereinafter referred to as the Titan router) is connected to the Internet via a SIM card.

- The Titan router will have 3 devices connected: two Modbus RTU temperature and humidity sensors (connected via RS485) and a Modbus ETH PLC (connected via Ethernet cable).

- The Titan router must continuously read two temperature and humidity sensors. When it detects a temperature above 30 °C, it must write a "1" to a specific register in the PLC. When the temperature drops below 29 °C, you must write a "0". In addition, the Titan router must send the average temperatures and humidities from both sensors every 15 minutes to a web platform via the MQTT protocol.



3. Description of the Modbus registers of the devices

The memory map would look like this. For the two temperature/humidity sensors, which are the same, the memory map is as follows:

Register	Date	Command	Register type	Comments
30000	Temperature	3	Int16	Example. A value of 325 indicates a temperature of 32.5 °C
30001	Humidity	3	Int16	Example. A value of 607 indicates a humidity of 60.7%

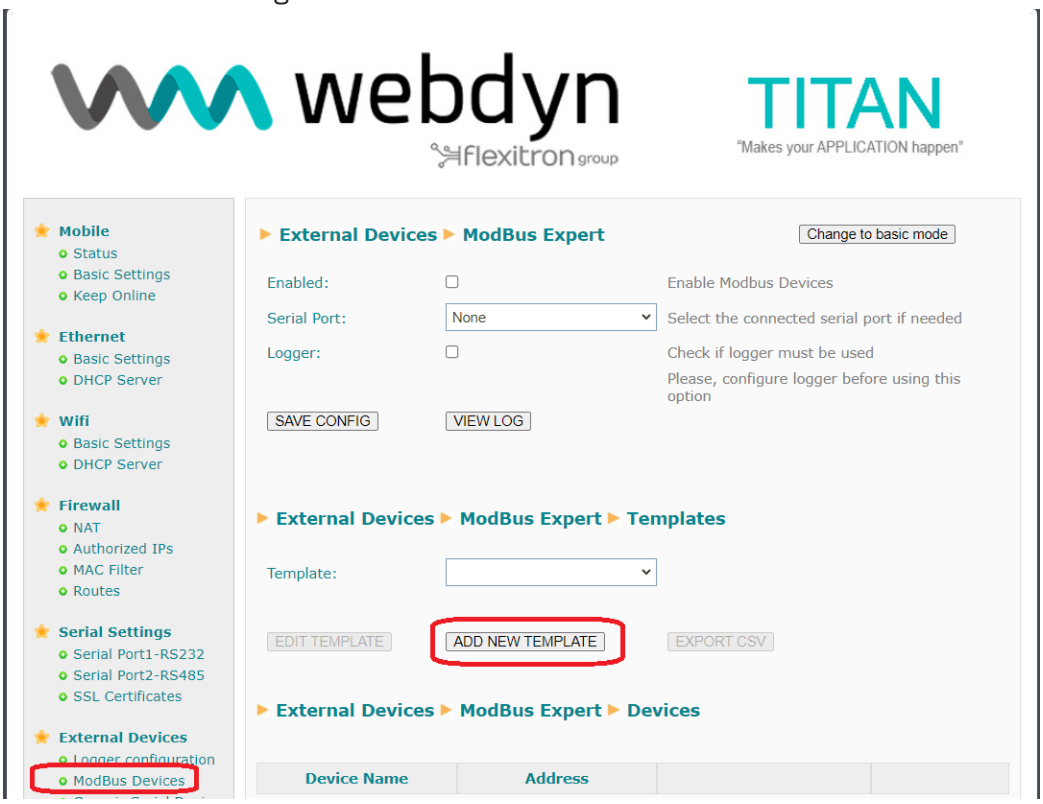
The memory map of the PLC where “1” must be written in the case of exceeding 30 °C or “0” if it falls below 29 °C:

Register	Date	Command	Register type	Comments
10000	Fan 1	1	Bit	Register to be written according to the temperature read by sensor 1
10001	Fan 2	1	Bit	Register to be written according to the temperature read by sensor 2
42122	Supply voltage	3	Float	PLC supply voltage

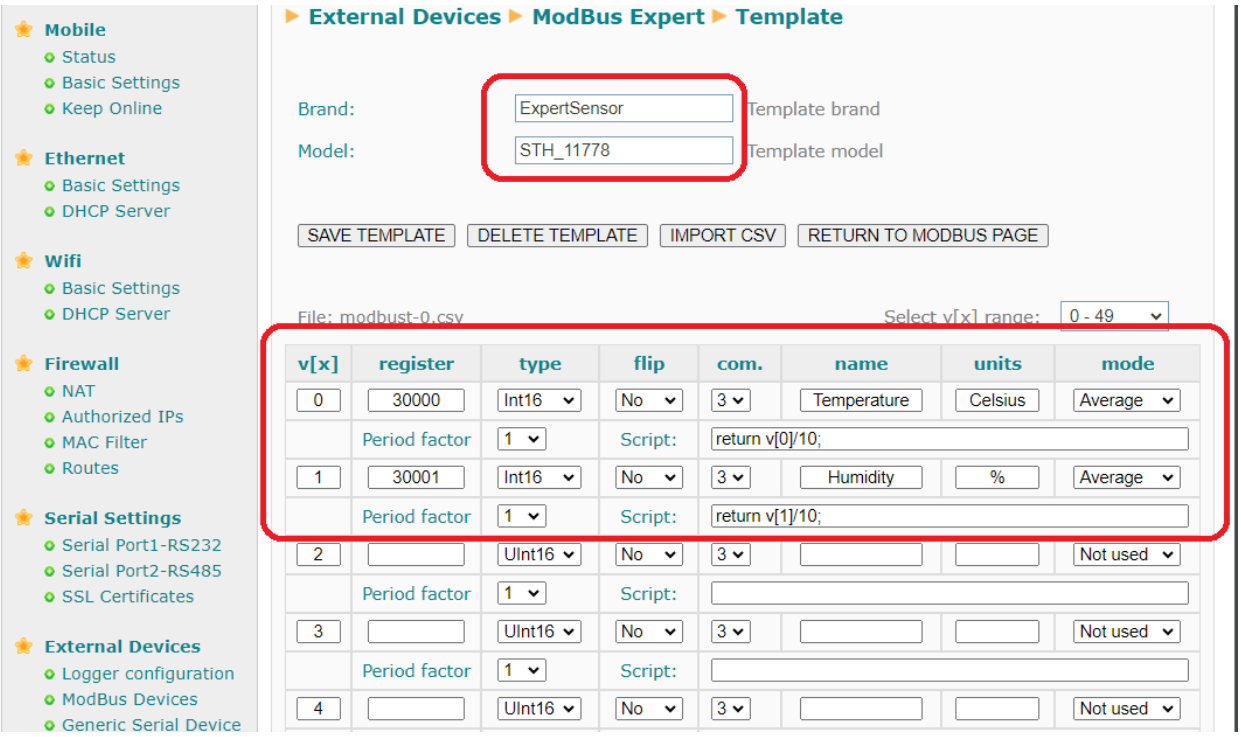
4. Creating device templates

As we are going to install 2 temperature/humidity sensors, and in anticipation of including more of the same devices in the future, it is a good idea to create a Modbus device template. That way, every time we want to introduce an additional temperature/humidity device into the scenario, the creation will be much faster.

To create a template, access the menu “Other -> Modbus Devices” and click on the “ADD NEW TEMPLATE” button, as shown in the following screenshot:



Once in that section, the make and model of the device in question must be entered and the table of registers completed according to the specifications indicated in the table above.



Register “30000”: the address of the Modbus register where the temperature is located.

Type “Int16”: The data type provided by the sensor, in this case, is a 16-bit integer. The sensor will return the temperature in the range of -200 to 900 (-20.0 °C to +90.0 °C).

Flip “no”: No word or byte exchanges are necessary in this case.

Com “3”: The Modbus command to be used for reading is 0x03.

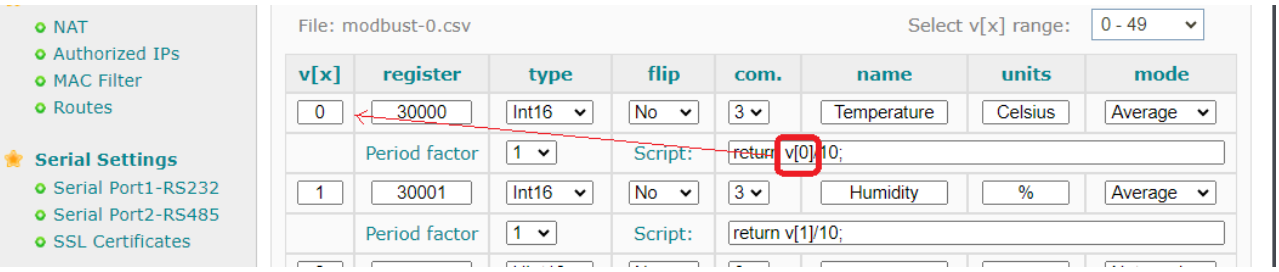
Name "temperature": Optional. The name of the register to be included in the JSON sent with the data to the MQTT platform.

Unit “celsius”: Optional. The name of the units to be included when sending the JSON with the data to the MQTT platform.

Script: Optional. Since the sensor returns data in integer format (no decimals), but you want to send the data to the platform in decimal format, each register must be divided by 10. Therefore, in the script, we indicate:

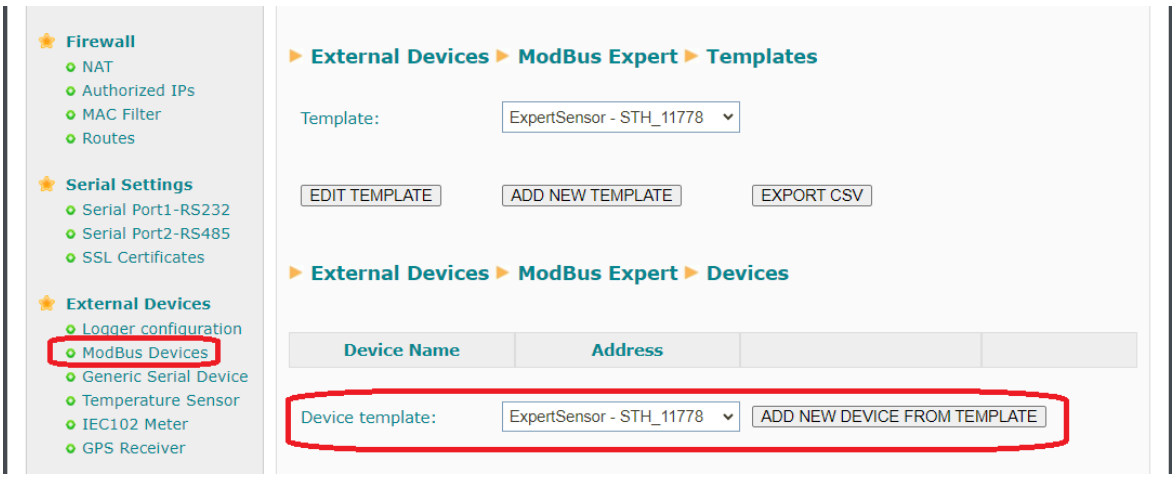
```
return v[0]/10;
```

where v[0] obviously indicates that it refers to the register v[0].

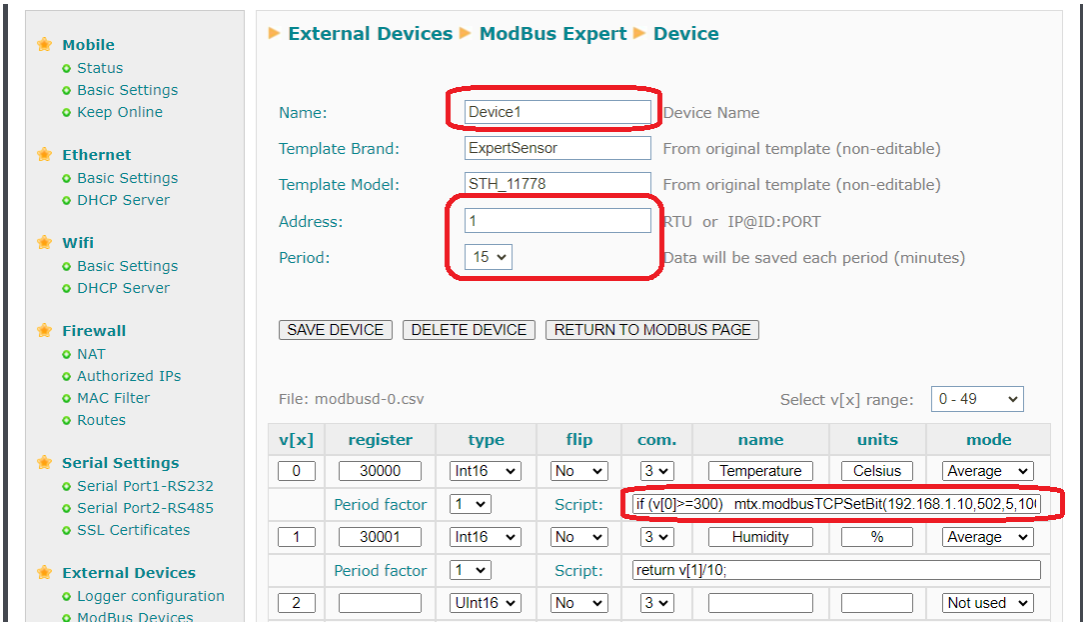


5. Creating the Modbus devices

The three Modbus devices involved in the scenario must now be created: the two sensors and a PLC. Since there is only one PLC, the device will be created directly without the need to create a template for it. To start creating the sensors, in the main Modbus Expert screen, select the template to be used and click on the ADD NEW DEVICE FROM TEMPLATE button.



Once on the new screen, the rest of the data must be filled in:

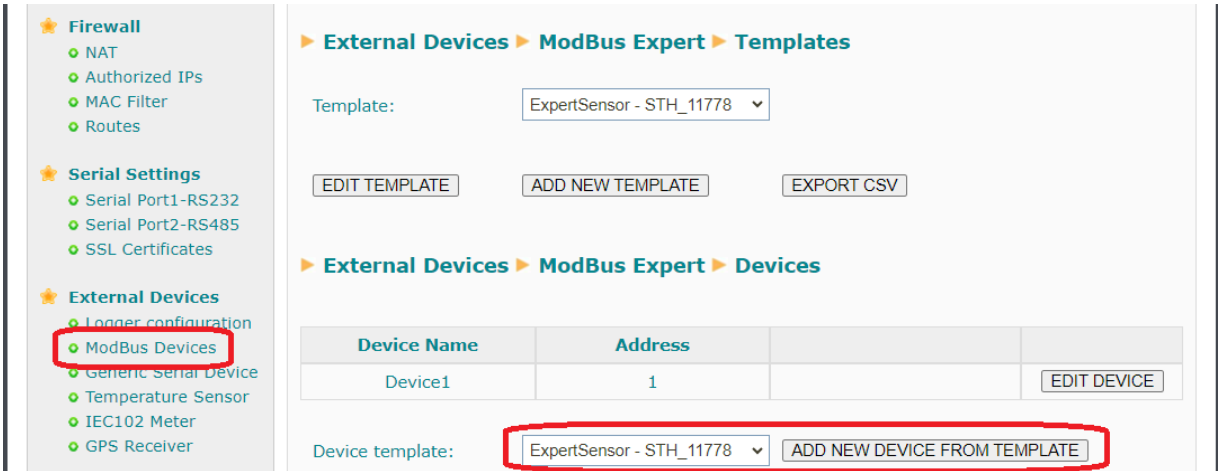


Enter the device name (e.g. Device1), the Modbus RTU address (1) and the base time interval at which you want to send the read registers to the platform (every 15 minutes).

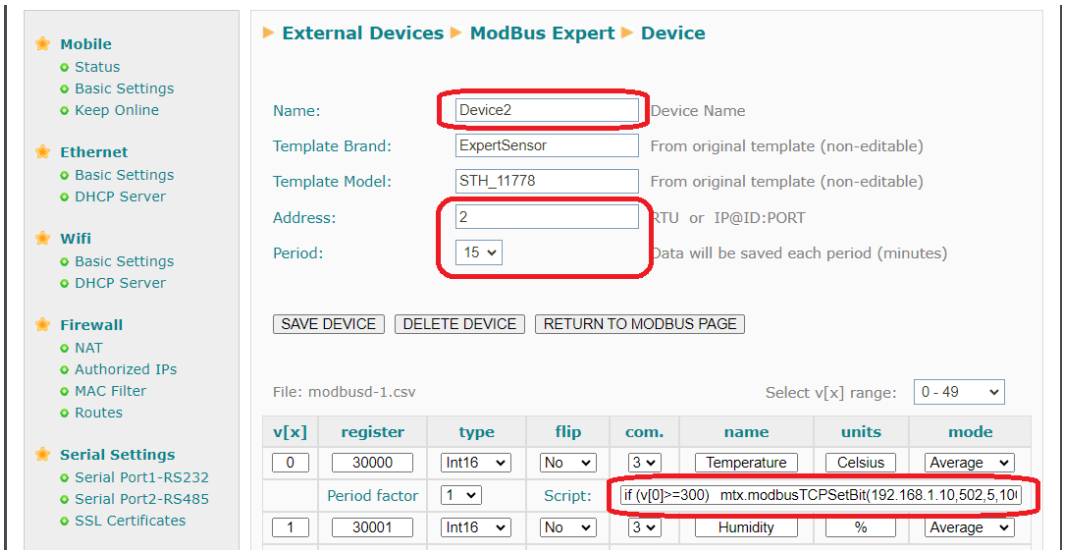
It is important to note that the v[0] register script has also been modified. In the script, it must be added that, in case the register read is >=300 (30 degrees), a 1 is written to PLC register 10000. If the register is <=290 (29 degrees), then a 0 must be written. Therefore, the script introduced is as follows:

```
if (v[0]>=300)
        mtx.modbusTCPSetBit(192.168.1.10,502,5,10000,1);
else if (v[0]<=290)
        mtx.modbusTCPSetBit(192.168.1.10,502,5,10000,0);
return v[0]/10;
```

For the second temperature/humidity sensor, exactly the same should be done.



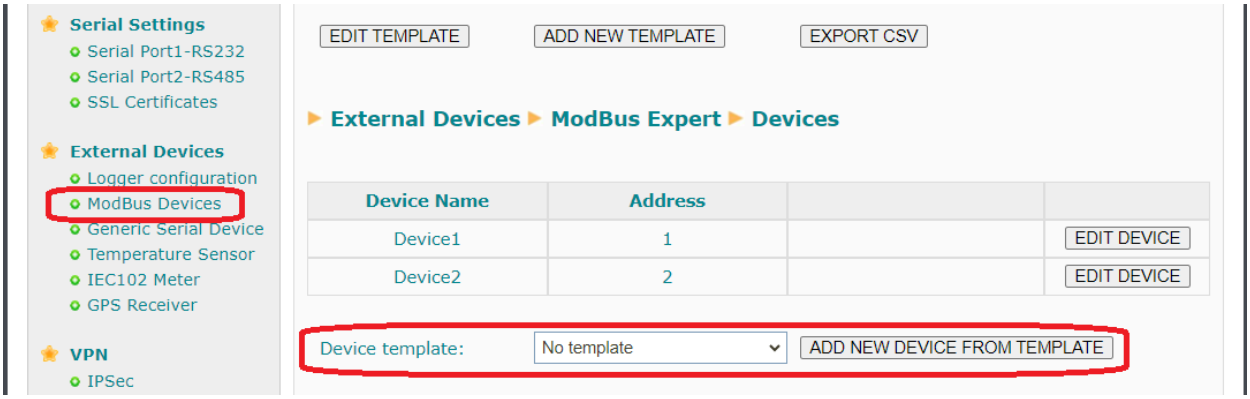
Although the script is slightly different in this case.



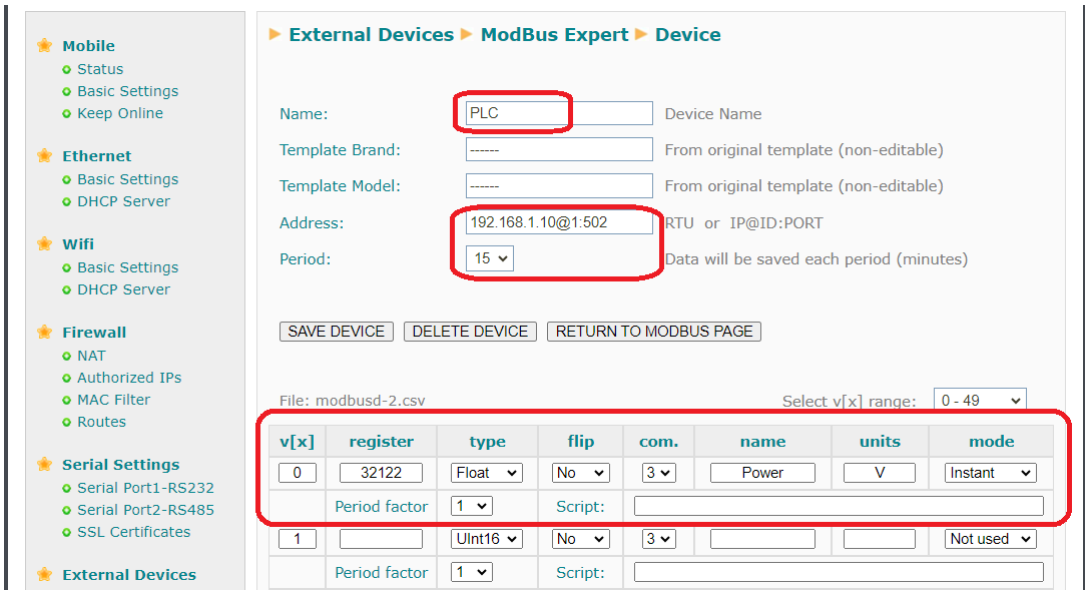
For this second sensor, the device must write to register 10001 instead of register 10000, i.e. this is the difference from sensor 1.

```
if (v[0]>=300)
        mtx.modbusTCPSetBit(192.168.1.10,502,5,10001,1);
else if (v[0]<=290)
        mtx.modbusTCPSetBit(192.168.1.10,502,5,10001,0);
return v[0]/10;
```

For the PLC, as there is only 1 unit, it can be created without using a template.

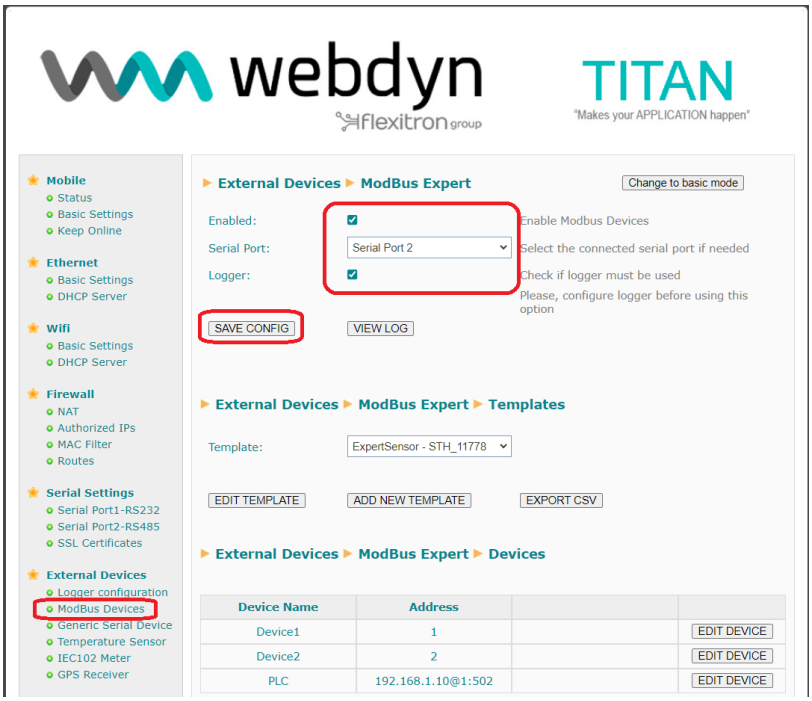


Indicate "PLC" as the name. As for the address, since this is a Modbus TCP device, indicate "192.168.1.10@1:502" (the PLC uses ID 1). The register submission period shall also be set at 15 minutes.



The PLC directly returns the value of its supply voltage in “float” format in register “32122”, so in this case we will not use a conversion script.

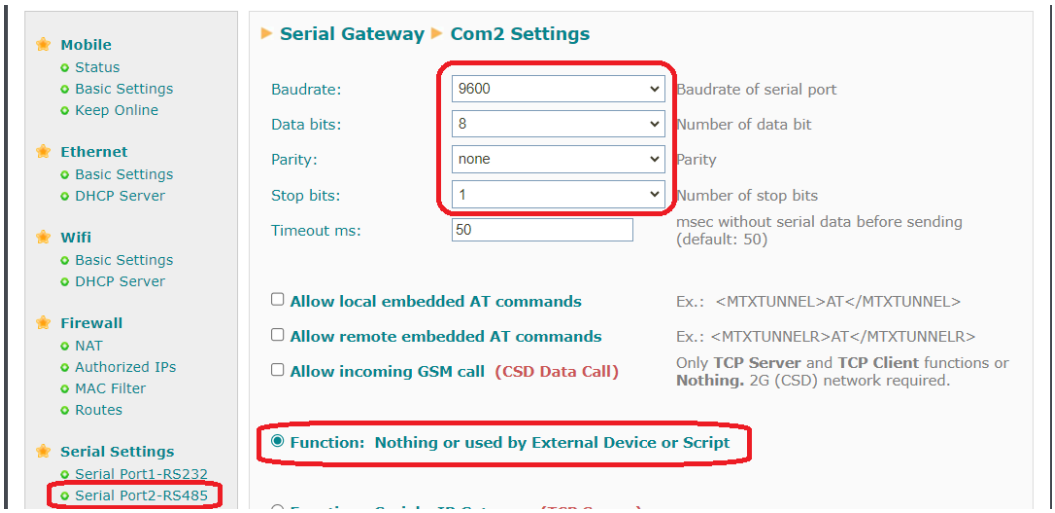
Returning to the main Modbus devices screen, the serial port to be used by the Modbus Expert service must be selected. In this case, the RS485 serial port (Serial Port 2) is being used and the Logger will also be activated, as it is intended to send the data to an MQTT platform.



6. Configuring the RS485 serial port

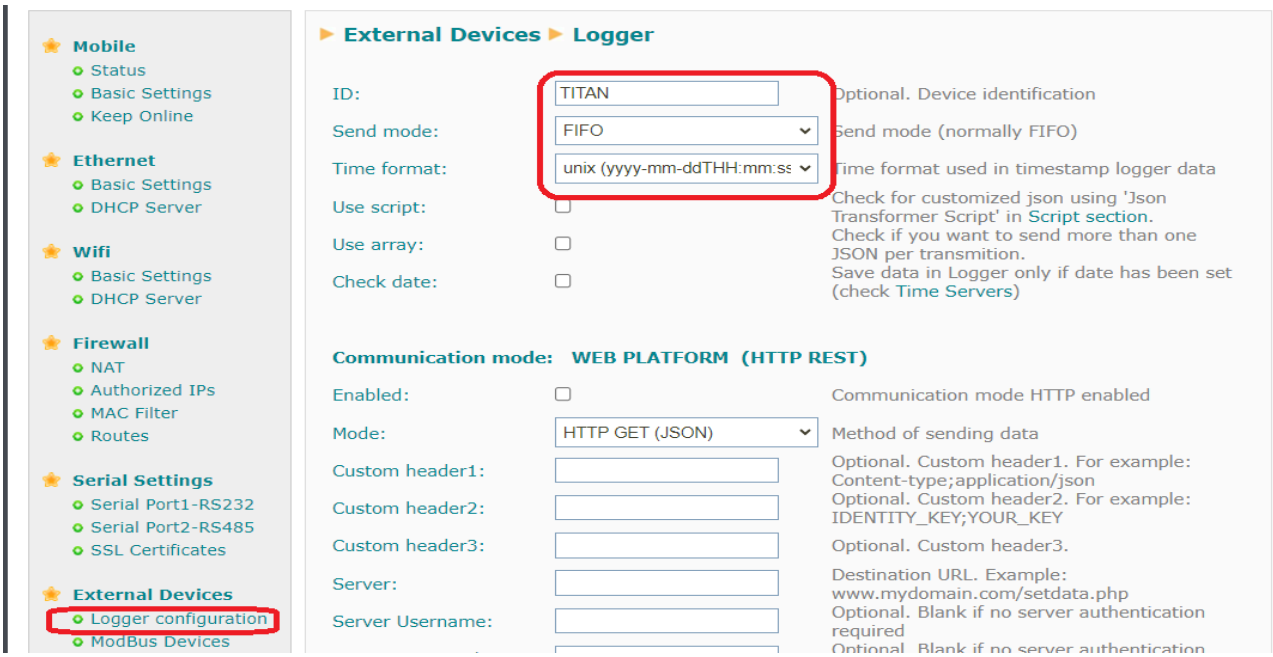
In the previous screen, Serial Port 2 (RS485 port) was selected as the serial port to be used with the Modbus Expert service. In this section, we will configure it. To do this, go to the menu “Serial Settings> Serial Port2-RS485”.

The serial port speed in this scenario will be 9600 baud, with 8 data bits, no parity (N), and 1 stop bit (1).

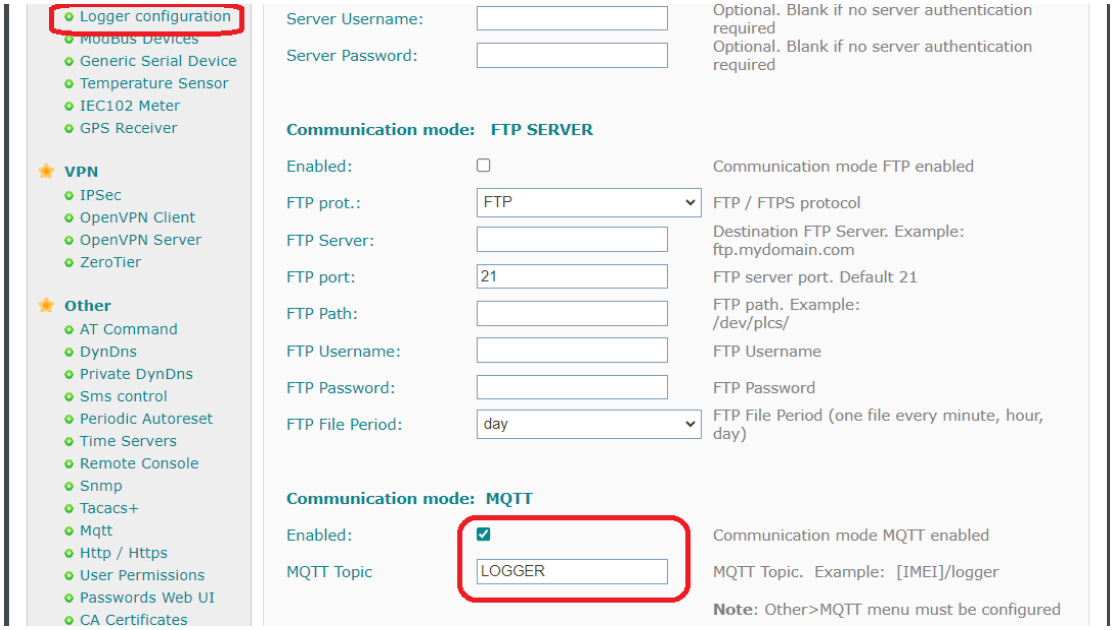


7. Configuring the datalogger

The Logger section must also be configured, as we want the router to read the Modbus registers autonomously and, every 15 minutes, these are sent to an MQTT platform. The Logger is configured in the menu 'External Devices -> Logger Configuration'.



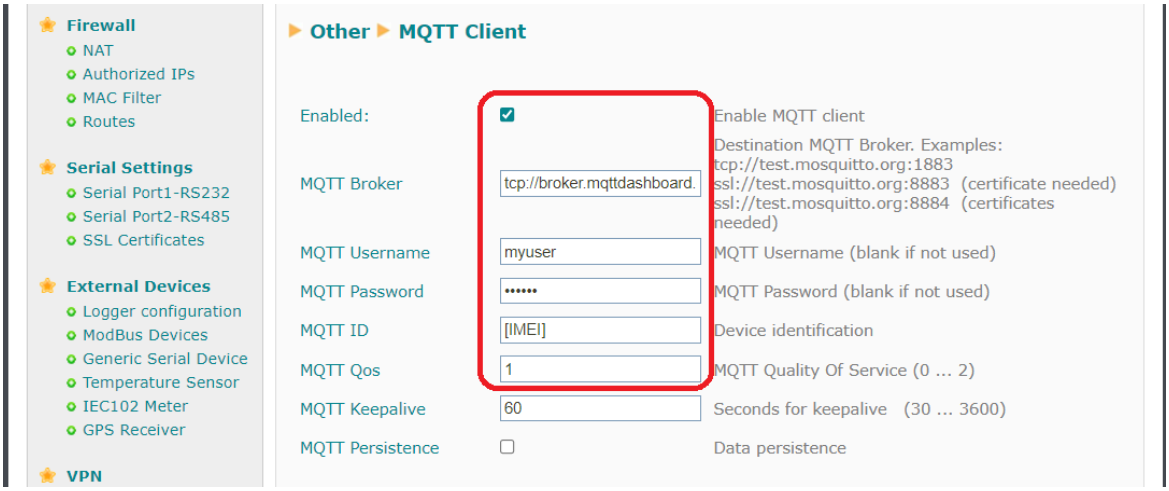
At the bottom of the same screen, the MQTT topic to which the data will be sent must be indicated. In this example, the topic with the text 'LOGGER' will be used.



Press the "SAVE CONFIG" button to save the Logger configuration.

8. MQTT Configuration

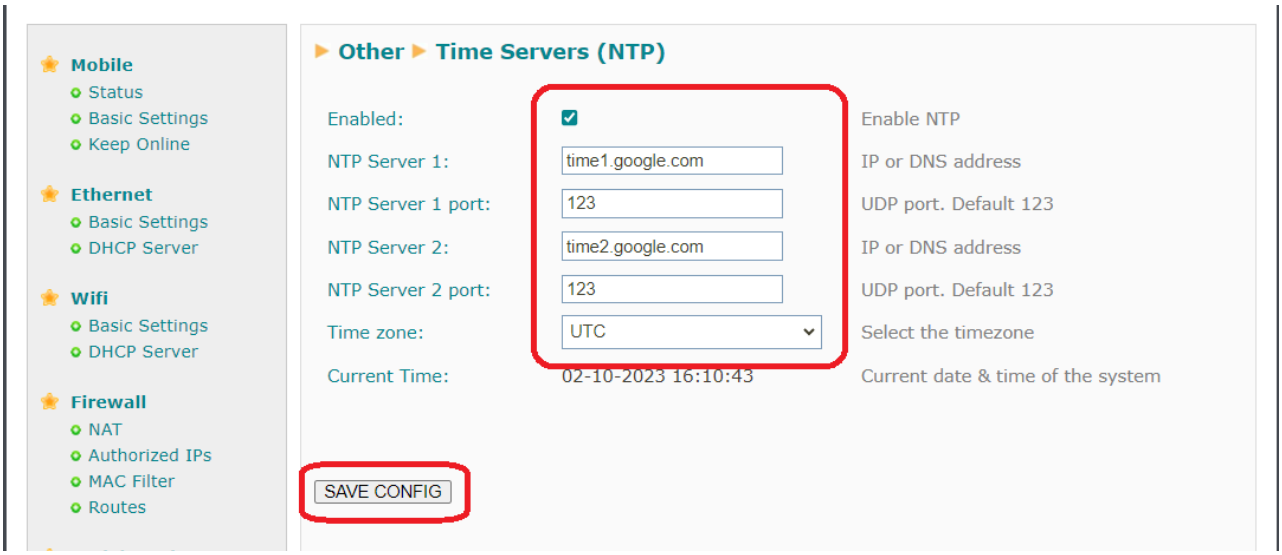
As the data will be sent to an MQTT broker, the connection to said broker needs to be configured beforehand. Configuration is done via the menu 'Other > MQTT'. In this case, the basic data is being configured.



At the end of the configuration, press the "SAVE CONFIG" button to save the MQTT configuration

9. Configuring the NTP time server

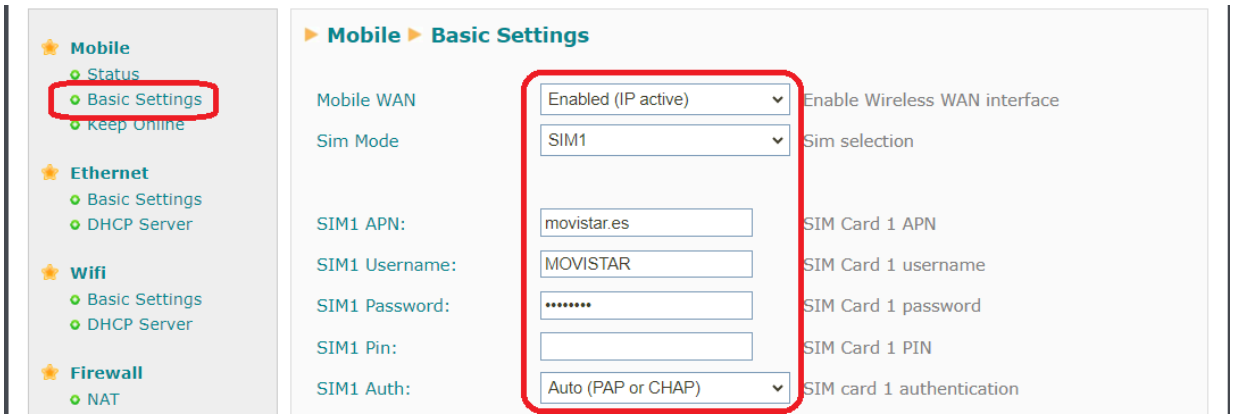
The Modbus registers read by the Titan router shall be stored in the internal datalogger and a timestamp with the router's own time will be added. It is therefore interesting to configure a time server so that the Titan router's time will always be correct and consequently the timestamp will also be correct. The time servers are configured from the configuration menu: "Other > Time servers"



When configuration is complete, press the "SAVE CONFIG" button to save the configuration.

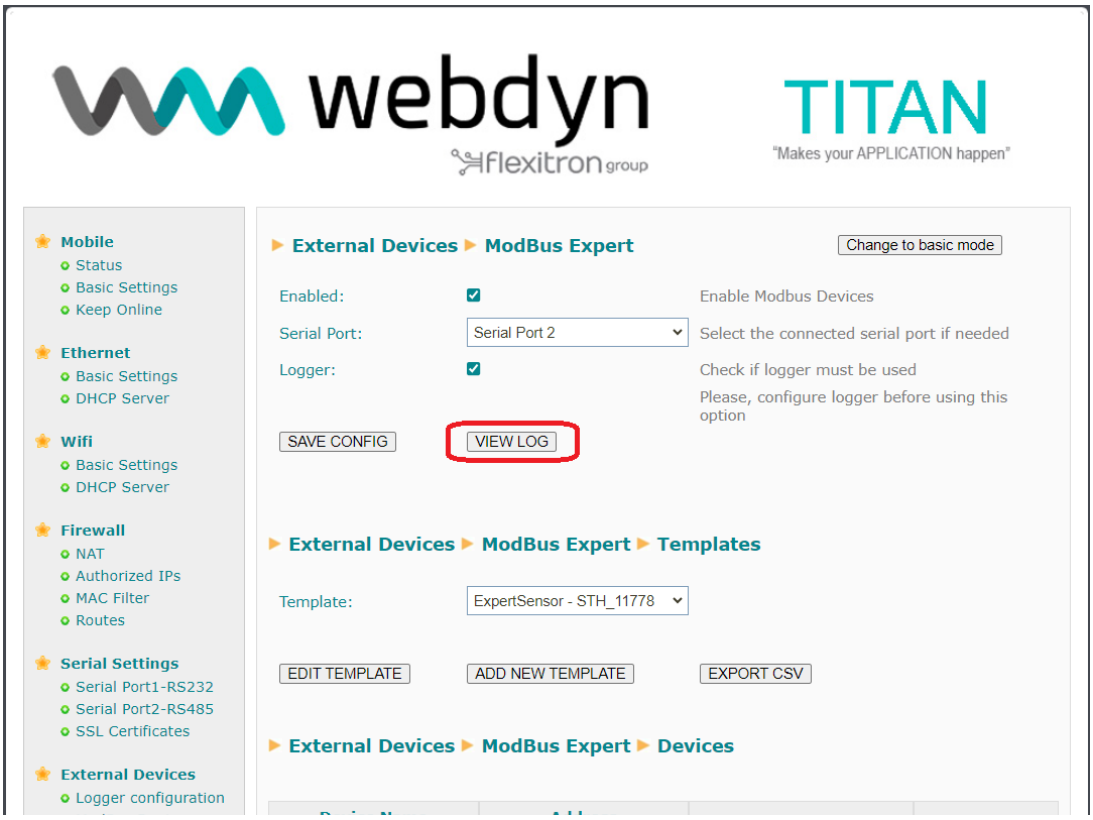
10. Configuring the Mobile Section

Finally, the “Mobile” section must be configured (in the menu “Mobile -> Basic Settings”), where the parameters of the SIM card(s) to be used by the Titan router will be specified. In this example, only SIM number 1 will be used. Once configuration of this section is complete, as usual, click on the “SAVE CONFIG” button.

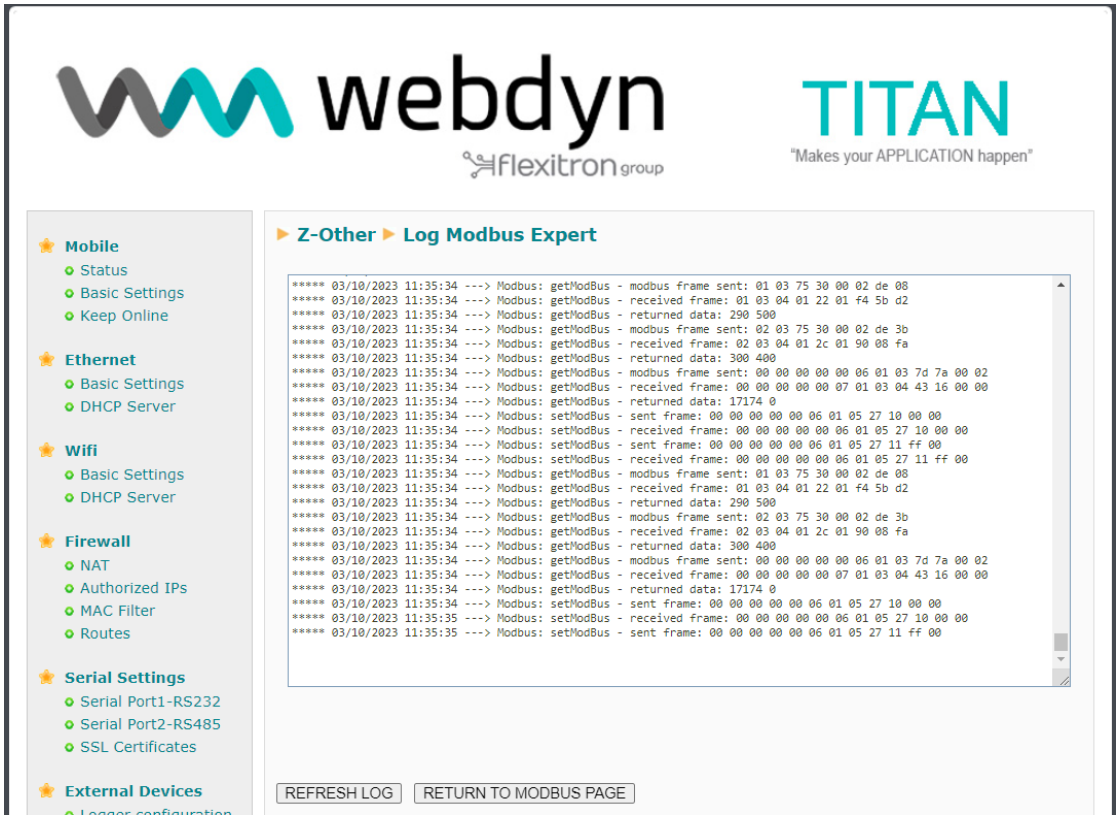


11. Testing the example

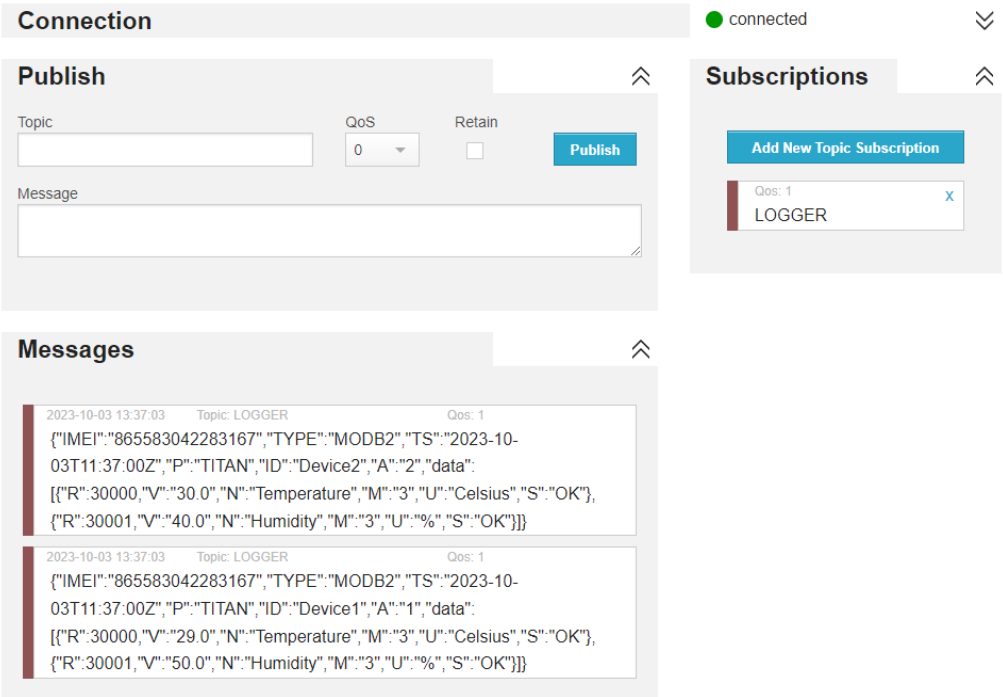
Finally, all that remains is to reboot the Titan router to get the new configuration up and running. The reboot can be carried out via the menu ‘Other -> Reboot’. After reboot, the Modbus Expert configuration page can be accessed in the menu ‘External Devices -> Modbus Devices’. The best way to verify that everything is working correctly is to click on the ‘VIEW LOG’ button.



In the log screen, the query/response Modbus frames can be seen.



Also, if everything is correct, the MQTT frames arriving at the Topic Logger can be verified, as shown in the following example:



The description of the JSON would be as follows:

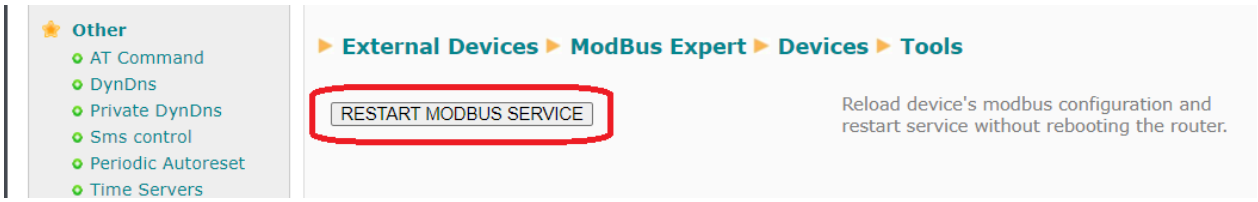
{“IMEI”:”865583042283167”,”TYPE”:”MODB2”,”TS”:”2023-10-03T11:46:00Z”,”P”:”TITAN”,”ID”:”Device2”,”A”:”2”,”data”:[{“R”:30000,”V”:”30.0”,”N”:”Temperature-”,”M”:”3”,”U”:”Celsius”,”S”:”OK”},{“R”:30001,”V”:”40.0”,”N”:”Humidity”,”M”:”3”,”U”:”%”,”S”:”OK”}]}

Where:

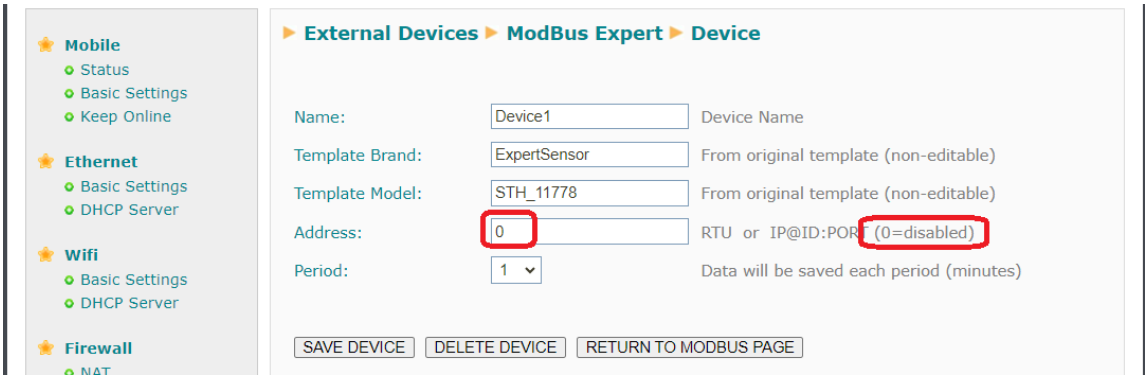
IMEI:	the router’s IMEI
TYPE:	type of frame
TS:	timestamp of the time the Modbus register was read.
P:	“ID” field of the “Logger” configuration section
ID:	“ID” field of the Modbus device section
A:	Modbus address of the device
R:	Register address
V:	Register value
N:	Register name
M:	Register mode (2: instantaneous value, 3: average value, 4: max value, 5: min value)
U:	Units
S:	Reading status (OK: correct reading, ERR: incorrect reading)

12. Other considerations

a) If the register configuration of any Modbus device needs to be changed or modified at any time, the Titan router does not need to be completely rebooted for it to take the new configuration. Simply press the “RESTART MODBUS SERVICE” button. This action will reload the configuration of all devices and restart the Modbus Expert service (if it was already started).



b) If you need to disable reading of any Modbus device (e.g. due to malfunction or to facilitate the correct operation of other devices in a more agile way), just enter a “0” in its “Address” change.



Any questions?

Please direct your enquiries to iotsupport@mtxm2m.com