

TITAN

Application Note 55

Reading Modbus Devices and Sending the Data to an
MQTT Broker and Occasionally to a Modbus TCP-RTU
Remote Gateway

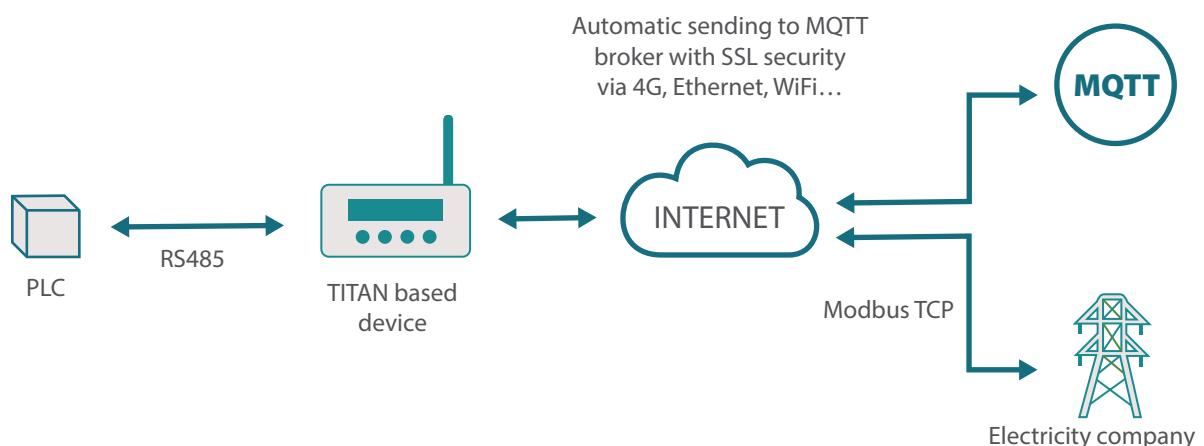
Reading Modbus devices and Sending the Data to an MQTT Broker and Occasionally to a Modbus TCP-RTU Remote Gateway

1. Scenario Details

TITAN-based devices have all the typical functionalities of 4G/3G/2G routers, as well as a series of added features that make them one of the most feature-packed routers on the market. One of the added features is the ability to autonomously interrogate Modbus RTU and TCP devices and subsequently to send the data to a WEB, FTP or MQTT server, together with the ability to establish an occasional Modbus TCP-RTU remote gateway. As always, this capability will be illustrated using a simple example.

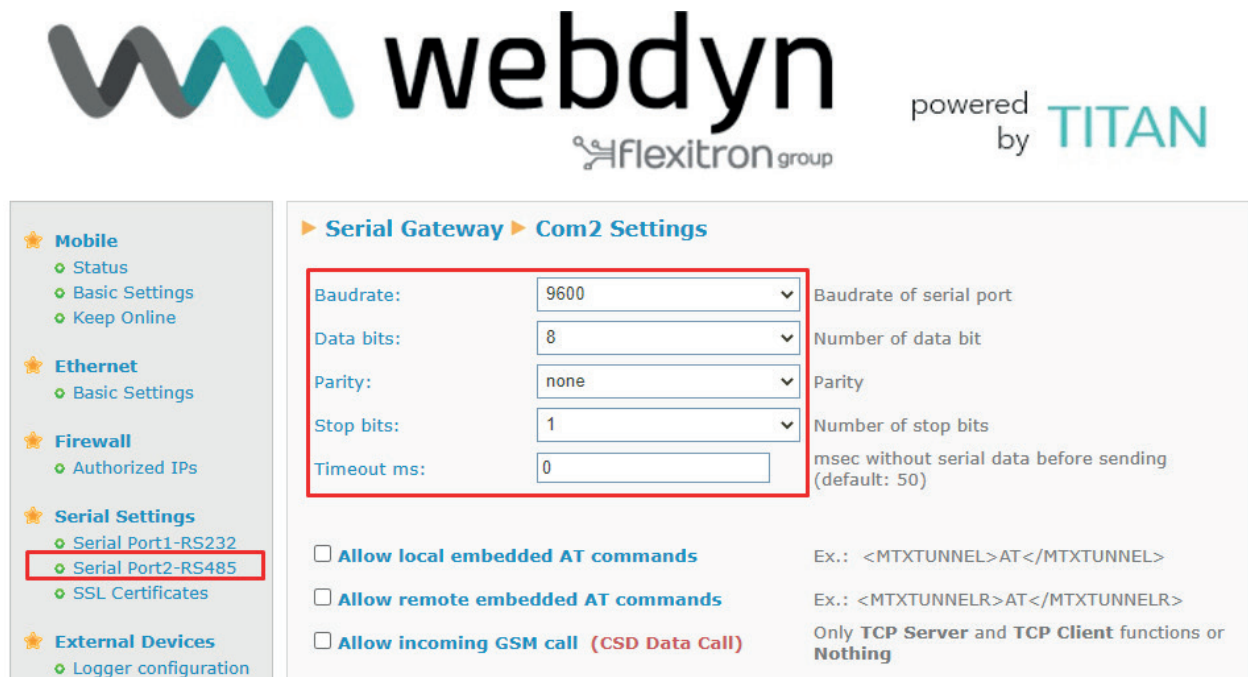
2. Description of the Example

- We have a PLC with an RS485 serial port configured to 9600.8;N.1.
- The PLC has Modbus communications in SLAVE mode and Modbus registers 10,11,12,13 and 14 need to be read from it.
- The device must be configured to read the PLC's Modbus registers every minute. After reading the PLC registers, the TITAN-based device must store the values in its internal memory and, if 4G coverage is available, send the data to an MQTT broker in a JSON object.
- The PLC allows its own firmware to be updated via Modbus TCP. For this reason, the TITAN-based device must be configured to establish a Modbus TCP - Modbus RTU gateway on port 502 via 4G.
- When a connection is made on TCP port 502 of the TITAN-based device, a Modbus TCP - Modbus RTU gateway will be established that will allow remote and direct access to the PLC to update the PLC firmware. When said gateway is established, the Titan device's autonomous communication to read the Modbus registers must be interrupted. When the connection through TCP port 502 is terminated (the PLC FW upgrade is complete), autonomous communication should be re-established.



3. Configuring the Serial Port of the TITAN-based Device to Which the Modbus Devices will be Connected

Let's imagine that the PLCs, which have an RS485 port, have the following serial port configuration: 9600,8,N,1. First we must configure the Serial Port2-RS485 section of the TITAN-based device. We will configure it as shown below:



The image shows the Webdyn web interface for configuring a TITAN-based device. The top header features the Webdyn logo (a stylized 'w' with a blue and green wave) and the text 'webdyn powered by TITAN', with 'flexitron group' below the logo. On the left is a sidebar menu with categories: Mobile (Status, Basic Settings, Keep Online), Ethernet (Basic Settings), Firewall (Authorized IPs), Serial Settings (Serial Port1-RS232, Serial Port2-RS485, SSL Certificates), and External Devices (Logger configuration). The 'Serial Port2-RS485' option is highlighted with a red box. The main content area is titled 'Serial Gateway > Com2 Settings'. It contains a table of serial port settings, all of which are enclosed in a red rectangular box: Baudrate (9600), Data bits (8), Parity (none), Stop bits (1), and Timeout ms (0). To the right of these settings are descriptive labels. Below the table are three checkboxes: 'Allow local embedded AT commands' (with example <MTXTUNNEL>AT</MTXTUNNEL>), 'Allow remote embedded AT commands' (with example <MTXTUNNELR>AT</MTXTUNNELR>), and 'Allow incoming GSM call (CSD Data Call)' (with note 'Only TCP Server and TCP Client functions or Nothing').

Serial Gateway > Com2 Settings

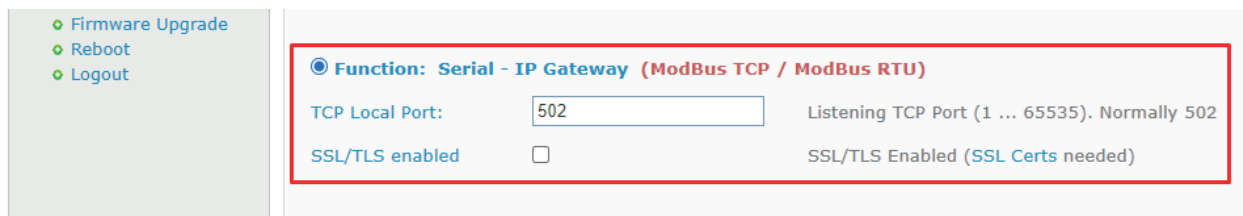
| | | |
|-------------|------|---|
| Baudrate: | 9600 | Baudrate of serial port |
| Data bits: | 8 | Number of data bit |
| Parity: | none | Parity |
| Stop bits: | 1 | Number of stop bits |
| Timeout ms: | 0 | msec without serial data before sending (default: 50) |

☐ **Allow local embedded AT commands** Ex.: <MTXTUNNEL>AT</MTXTUNNEL>

☐ **Allow remote embedded AT commands** Ex.: <MTXTUNNELR>AT</MTXTUNNELR>

☐ **Allow incoming GSM call (CSD Data Call)** Only TCP Server and TCP Client functions or Nothing

In the lower area of the same screen, we must also select the option “Function: Serial – IP Gateway (Modbus TCP / Modbus RTU)”.



The image shows the lower section of the Webdyn interface. The sidebar menu on the left has 'Firmware Upgrade', 'Reboot', and 'Logout' options. The main content area is titled 'Function: Serial - IP Gateway (ModBus TCP / ModBus RTU)'. This title and the settings below it are enclosed in a red rectangular box. The settings include 'TCP Local Port' set to 502 (with a note 'Listening TCP Port (1 ... 65535). Normally 502') and 'SSL/TLS enabled' which is unchecked (with a note 'SSL/TLS Enabled (SSL Certs needed)').


Function: Serial - IP Gateway (ModBus TCP / ModBus RTU)

TCP Local Port: 502 Listening TCP Port (1 ... 65535). Normally 502

SSL/TLS enabled ☐ SSL/TLS Enabled (SSL Certs needed)

4. Configuring the TITAN to Read Modbus Devices

To do this, click on the link: “External Devices > Modbus Devices” and configure it as shown below:

**webdyn**
flexitron group

powered by **TITAN**

- Mobile
 - Status
 - Basic Settings
 - Keep Online
- Ethernet
 - Basic Settings
- Firewall
 - Authorized IPs
- Serial Settings
 - Serial Port1-RS232
 - Serial Port2-RS485
 - SSL Certificates
- External Devices
 - Logger configuration
 - ModBus Devices**
 - Generic Serial Device
 - Temperature Sensor
 - IEC102 Meter
 - W-MBus
- Other
 - AT Command
 - DynDns
 - Private DynDns
 - Sms control
 - Periodic Autoreset
 - Time Servers
 - Remote Console

External Devices > ModBus RTU / TCP

Enabled: ☒

Enable Modbus Devices

Serial Port: Serial Port 2

Select the connected serial port if needed

Logger: ☒

Check if logger must be used
Please, configure logger before using this option

SAVE CONFIGVIEW LOG

| Dev. name / ID | Addr. | Command | Start @ | Num word/bit | Reg Type | Period | | |
|----------------|-------|---------|---------|--------------|----------|--------|-----|------|
| 1 | 1 | 0x03 | 10 | 5 | WORD | 1 | Del | Test |

Device name / ID:

Insert the device name or ID

Address:

Modbus RTU address or IP:port address

Command:

0x01

Modbus read command

Start:

Address of the first register

Number Words / Bits:

Words for command 0x03/0x04. Bits for 0x01/0x02

Reg Type:

WORD

Type of registers for command 0x03/0x04

Period:

1

Read period (minutes)

We want to read registers 10,11,12,13,14 from the PLC. Simply enter register 10 in the “Start” field and 5 in the “Number Words” field (as we want to read 5 records, from 10 to 14).

5. Configuring the Logger (communication with the MQTT server)

The next step is to configure the Logger. This is the system in the TITAN-based device that stores the data and sends it. In this example we are going to configure the device to send the data to an MQTT broker.

For the MQTT broker we will use the HiveMQ test broker and the client <http://www.hivemq.com/demos/websocket-client/>. The data will be sent in JSON format after each read (timestamp, device ID, etc.).

As seen in the following figure, we will use the “External Devices > Logger Configuration” menu to configure the section as shown below:

webdyn powered by **TITAN** flexitron group

External Devices > Logger

ID: Optional. Device identification

Send mode: Send mode (normally FIFO)

Time format: Time format used in timestamp logger data

Use script: ☐ Check for customized json using 'Json Transformer Script' in [Script section](#).

Use array: ☐ Check if you want to send more than one JSON per transmission.

Communication mode: MQTT

Enabled: ☒ Communication mode MQTT enabled

MQTT Topic: MQTT Topic. Example: [IMEI]/logger

Note: Other>MQTT menu must be configured

In this screen we have activated MQTT mode. All data collected by the TITAN-based device will be published in the [IMEI]/logger topic. Note that the device will replace the [IMEI] tag with your actual IMEI. This means that, if the device's IMEI is 357299070082380 (the IMEI can be found in the Mobile > Status menu), the topic used by the TITAN-based device to publish the collected Modbus data will be: 357299070082380/logger.

Conversely, if we activate logger mode, we must configure the MQTT client in the “Other -> Mqtt” section.

- Serial Port2-RS485
- SSL Certificates
- External Devices
 - Logger configuration
 - ModBus Devices
 - Generic Serial Device
 - Temperature Sensor
 - IEC102 Meter
 - W-MBus
- Other
 - AT Command
 - DynDns
 - Private DynDns
 - Sms control
 - Periodic Autoreset
 - Time Servers
 - Remote Console
 - Snmp
 - Tacacs+
 - Mqtt
 - Http / Https
 - User Permissions
 - Passwords Web UI
 - CA Certificates
 - Email Config
 - ModBus Slave
 - Titan Scripts
 - Connectivity tools
 - Digital I/O
 - Custom Skin
 - Led Config
 - Syslog
 - Backup / Factory
 - Firmware Upgrade
 - Reboot
 - Logout

Other ▶ MQTT Client

| | | |
|----------------------|--|--|
| Enabled: | <input checked="" type="checkbox"/> | Enable MQTT client |
| MQTT Broker | <input type="text" value="tcp://broker.mqttdashboard.com:1883"/> | Destination MQTT Broker. Examples: tcp://test.mosquitto.org:1883 ssl://test.mosquitto.org:8883 (certificate needed) ssl://test.mosquitto.org:8884 (certificates needed) |
| MQTT Username | <input type="text"/> | MQTT Username (blank if not used) |
| MQTT Password | <input type="text"/> | MQTT Password (blank if not used) |
| MQTT ID | <input type="text" value="[IMEI]"/> | Device identification |
| MQTT Qos | <input type="text" value="1"/> | MQTT Quality Of Service (0 ... 2) |
| MQTT Keepalive | <input type="text" value="60"/> | Seconds for keepalive (30 ... 3600) |
| MQTT Persistence | <input type="checkbox"/> | Data persistence |
| MQTT AT Topic | <input type="text" value="[IMEI]/AT"/> | This topic will be subscribed for receiving AT Commands (usefull for individual device) |
| MQTT AT Resp Topic | <input type="text" value="[IMEI]/ATR"/> | This topic will be used for publishing the AT Command Responses of AT Topic |
| MQTT AT Topic 2 | <input type="text"/> | This topic will be subscribed for receiving AT Commands (usefull for groups) |
| MQTT AT Resp Topic 2 | <input type="text"/> | This topic will be used for publishing the AT Command Responses of AT Topic 2 |
| MQTT AT Topic 3 | <input type="text"/> | This topic will be subscribed for receiving AT Commands (usefull for all devices) |
| MQTT AT Resp Topic 3 | <input type="text"/> | This topic will be used for publishing the AT Command Responses of AT Topic 3 |
| MQTT Script Topic 1 | <input type="text"/> | When data is received in this topic the 'Topic Script' will be executed. |
| MQTT Script Topic 2 | <input type="text"/> | When data is received in this topic the 'Topic Script' will be executed. |

Activate the Enabled field. In the “MQTT Broker” field we will enter: `tcp://broker.mqttdashboard.com:1883`, indicating the URL of the MQTT broker we want to connect to. Enter a unique identifier for our device in MQTT ID (a broker cannot have 2 devices with the same ID as this will cause continuous interruptions). If we want to be able to send AT commands to the TITAN-based device (e.g. to configure it or view the status from a mobile phone which is also connected to the MQTT broker), we must add the MQTT AT Topic and MQTT AT Response Topic fields.

All AT commands sent to the broker, to the MQTT AT Topic, will be executed by the TITAN-based device. Once executed, the result of the AT command will be published by the TITAN-based device in the MQTT AT Response Topic.

6. Other Considerations

- After configuring the TITAN-based device we will need to perform for the new configuration to take effect and start reading and sending.
- Each time the TITAN-based device sends a measurement to the MQTT broker, it does so using a JSON object of the following type:

PLC JSON Example

```
{"IMEI":"357299070082380","TYPE":"MODB","TS":"2022-07-15T10:14:02Z","ID":"1","A":"1","ST":"10","N":"5","V":[10,11,12,0,0],"P":"ID-1234"}
```

Where:

IMEI - indicates a unique identifier for the modem

TYPE: indicates the type of datum (MODB = Modbus reading)

TS: indicates the Timestamp (the time the measurement was read)

ID: name or identifier of the Modbus device

A: Modbus device's address

ST: indicates the address of the first Modbus register read

STX: an array that indicates the address of the first Modbus registers when reading groups of registers

N: indicates the number of words read

NX: an array that indicates the number of words read when reading groups of registers

PX: an array indicating the position of the initial register of each block within V.

V: An array containing the data read

P: indicate the ID field configured in the Logger

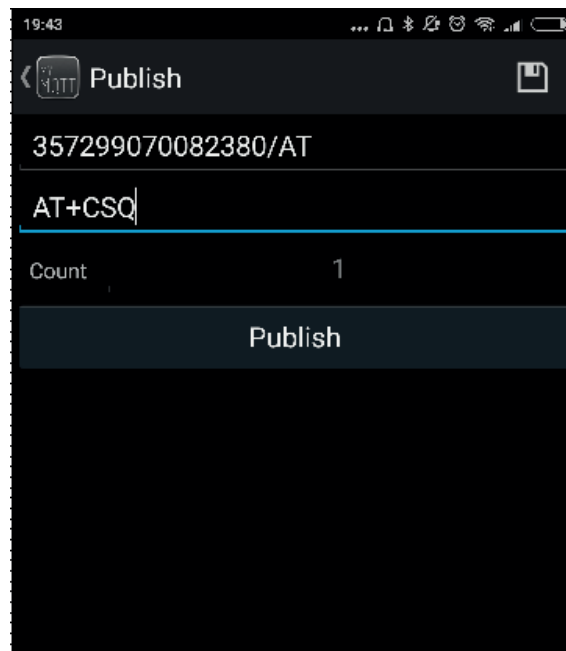
7. Remote Communication with a PLC via Modbus

In previous chapters we configured a Modbus TCP – Modbus RTU gateway on TCP port 502 within the TITAN-based device. Using this we can connect to the PLC both remotely and directly to update its firmware. By establishing a connection with the public IP of the TITAN based device (4G) via TCP port 502, autonomous reading of the Modbus registers will be suspended (to avoid collisions), and the Modbus TCP to Modbus RTU gateway will start operating, so allowing direct connection with the PLC remotely.

8. Communication with TITAN-based Devices Using AT Commands Sent by MQTT

As we mentioned above, we can send AT commands to the TITAN-based device via MQTT. We can use these to change the device's configuration, check conditions such as coverage, reset it, switch a relay, etc.

If we want to check the coverage, we can use the Android myMQTT program, sending the AT+CSQ command to the [IMEI]/AT topic, which in this example is 357299070082380/AT.



Clicking on the “Publish” button, the command will be received by the TITAN-based device and executed. The response will be published by the TITAN-based device in topic 357299070082380/ATR, as configured above.

