

TITAN

Application Note 48

Serial Frame Datalogger for Proprietary Protocols -
RS232 / RS485

Serial Frame Datalogger for Proprietary Protocols - RS232 / RS485

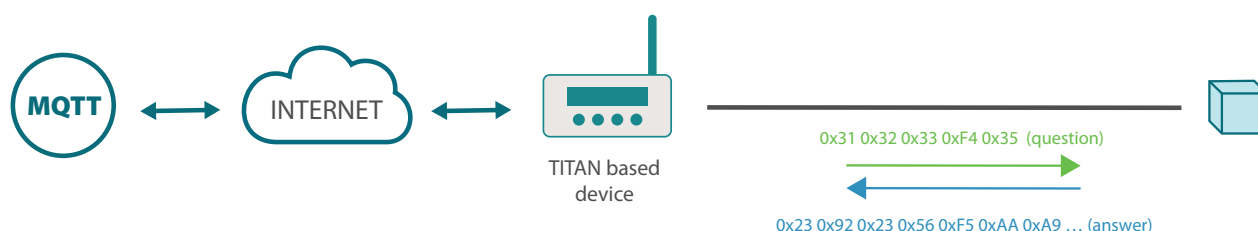
1. Scenario Details

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 store data from a serial device connected to an RS485 or RS232 port by interrogating them with custom serial frames, which can be useful for proprietary protocols.

As always, this capability will be illustrated using a simple example.

2. Description of the Example Scenario

- We have a device (PLC) with an RS232 serial port (115200,8,N,1). We must read a series of internal registers from that device every 10 minutes and send them to an MQTT platform.
- In order to be able to read the registers of the device, we must use a proprietary protocol, i.e. we must send certain frames of bytes (proprietary protocol) through the device's serial port so that it sends back the values of the internal registers we need.
- The TITAN-based device must therefore periodically send preconfigured frames of bytes to the serial device through its RS232 serial port, collect the responses and, adding the time required to collect the data, send said data to an MQTT broker encapsulated in a JSON object.
- To read the registers, two serial frames will be sent via RS232 through the serial port of the TITAN-based device every 10 minutes: 313233F435 and 41A12D42421F4343.



3. Configuring the Associated Serial Port

First we must decide which of the TITAN-based device's serial ports the serial device being interrogated will be plugged into.

We will assume that the PLC is to be connected to the COM1 (RS232) port of the TITAN-based device. The PLC's serial port is set to 115200,8,n,1, we must therefore click in the "Serial Settings > Serial Port1-232" menu and configure the screen as follows:

Intelligent Router x +

No es seguro | 192.168.1.2/serial-settings.php?id=1

webdyn powered by **TITAN**
flexitron group

Serial Gateway > Com1 Settings

Serial Settings

- Serial Port1-RS232
- Serial Port2-RS485
- SSL Certificates

Mobile

- Status
- Basic Settings
- Keep Online

Ethernet

- Basic Settings

Wifi

- Basic Settings
- DHCP Server

Firewall

- NAT
- Authorized IPs

External Devices

- Logger configuration
- ModBus Devices
- Generic Serial Device
- Temperature Sensor
- IEC102 Meter
- W-MBus
- GPS Receiver

Baudrate: 9600 Baudrate of serial port

Data bits: 8 Number of data bit

Parity: none Parity

Stop bits: 1 Number of stop bits

Flow Control: none Flow control of serial port

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**

☒ **Function: Nothing or used by External Device or Script**

☐ **Function: Serial - IP Gateway (TCP Server)**

TCP Local Port: 20010 Listening TCP Port (1 ... 65535)

☐ **Temporal client RS232** Check if you need a temporal TCP Client when data is present at serial port.

Temporal client Wakeup DDHHMM. Example: XX2200 starts a temporal client every day at 22:00

4. Configuring the Serial Datalogger

The next step is to configure the serial datalogger, i.e. we must configure the TITAN-based device to select the serial port that will be used with the datalogger, as well as the frames (in hex format) that the TITAN-based device must send via its serial port to interrogate the PLC, and also the send interval.

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

The screenshot shows a web browser window with the URL `192.168.1.2/external-generic.php`. The page header features the 'webdyn' logo, 'powered by TITAN', and the 'Flexitron group' logo. A left sidebar contains a navigation menu with categories: Mobile, Ethernet, Wifi, Firewall, Serial Settings, External Devices, and Other. The 'Generic Serial Device' option under 'External Devices' is highlighted with a red box. The main content area is titled 'External Devices > Generic Serial Device' and contains two red-bordered boxes highlighting the configuration fields. The first box includes 'Enabled' (checked), 'Serial Port' (set to 'Serial Port 1'), 'Interval' (set to '0'), 'Only changes' (unchecked), 'Mode' (set to 'Raw (Default)'), and 'Logger' (checked). The second box includes 'TX Period' (set to '600'), 'TX Frame1' (set to '313233F535'), and 'TX Frame2' (set to '41A12D42421F4343'). To the right of these fields are explanatory text blocks: 'Enable Generic Serial Device', 'Select the connected serial port', '0=save every serial frame', 'Check for register every serial frame if different than previous one.', 'Communication mode with Titan', 'Check if logger must be used. Please, configure logger before using this option', 'Period mof TX Frames (seconds 30 ... 86400. 0=no TX Data)', and 'HEX Frame 1 (max length 256)' through 'HEX Frame 5 (max length 256)'. At the bottom of the configuration area are 'SAVE CONFIG' and 'VIEW LOG' buttons.

Namely:

- Check the box to enable the serial datalogger (Enabled box)
- Choose the Port 1 as the serial port.
- Choose an Interval of “0”, as all response frames must be collected
- Do not select the “Only Changes” frame, as it would not make sense in the current application
- Mode: Raw

- Check the box to enable the serial frames received by the internal datalogger to be stored (Logger box)
- TX Period: indicate the interval at which frames are sent. 600 seconds implies that the serial frames are sent every 10 minutes.
- TX FrameX: indicate the serial frames to be sent each interval (in this example 600 seconds). The serial frames must be entered in hex format.
- Click on “SAVE CONFIG” at the end of the configuration process.

5. Logger Configuration (communication with the MQTT server)

The next step is to configure the Logger. This is the data storage and transmission system used by the TITAN-based device itself. In this example it will be configured to send the serial data to an MQTT broker.

As can be seen in the following figure, we will go to the “External Devices > Logger Configuration” menu and configure the section as follows:

The screenshot displays the Webdyn configuration interface. On the left sidebar, the 'External Devices' menu is expanded, and 'Logger configuration' is highlighted with a red box. The main configuration area shows two communication modes: 'FTP SERVER' and 'MQTT'. The 'MQTT' mode is selected and highlighted with a red box. In the 'MQTT' section, 'Enabled' is checked, and 'MQTT Topic' is set to '/LOGGER'. The 'FTP SERVER' section is also visible but not selected. At the bottom, there is a 'SAVE CONFIG' button.

By doing this we are activating the mode for sending data via MQTT to the “/LOGGER” topic in the MQTT broker.

Next, the Other > MQTT section must be configured to activate the MQTT client correctly, configuring parameters such as the MQTT broker to be used and the MQTT ID, etc.

The screenshot shows a web browser window with the address bar displaying "192.168.1.2/other-mqtt.php". The page header features the "webdyn" logo, "flexitron group" branding, and "powered by TITAN". A left sidebar contains a navigation menu with categories: Mobile (Status, Basic Settings, Keep Online), Ethernet (Basic Settings), Wifi (Basic Settings, DHCP Server), Firewall (NAT, 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, GPS Receiver), and Other. The main content area is titled "Other > MQTT Broker" and "Other > MQTT Client". The MQTT Broker section includes fields for Enabled (checkbox), TCP Port (1883), Anonymous User (checkbox), User, and Password. The MQTT Client section, highlighted with a red box, includes fields for Enabled (checked checkbox), MQTT Broker (tcp://broker.mqttdashboard.), MQTT Username, MQTT Password, MQTT ID ([IMEI]), MQTT Qos (1), and MQTT Keepalive (60). Each field has a descriptive label to its right.

Section	Parameter	Value	Description
MQTT Broker	Enabled	<input type="checkbox"/>	Enable MQTT broker service
	TCP Port	1883	Listening port (for example 1883)
	Anonymous User	<input type="checkbox"/>	Allow anonymous user (no user / password is needed)
	User		Username (used if anonymous is not selected)
	Password		Password (used if anonymous is not selected)
MQTT Client	Enabled	<input checked="" type="checkbox"/>	Enable MQTT client
	MQTT Broker	tcp://broker.mqttdashboard.	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		MQTT Username (blank if not used)
	MQTT Password		MQTT Password (blank if not used)
	MQTT ID	[IMEI]	Device identification
	MQTT Qos	1	MQTT Quality Of Service (0 ... 2)
	MQTT Keepalive	60	Seconds for keepalive (30 ... 3600)

6. Other Considerations

- After configuring the TITAN-based device we will need to perform a reset to accept the new configuration.
- The JSON objects sent to the MQTT broker, containing the serial reads of the responses received from the PLC, will have content similar to:

```
{ "TYPE": "SERIAL", "TS": "11/04/2021 17:19:44", "IMEI": "358709050113764", "P": "", "DATA":  
"3c646174613e72656c6179733a372c74656d70657261747572653a33322c68756  
d69646974793a38302c636f756e7465723a313232353c2f646174613e0d0a" }
```

where the data in hex format are stored in the DATA field in hex format.