TITAN

Application Note 4

Implementing a Data Logger for RS232/RS485 Devices

Implementing a Data Logger for Devices: 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.

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

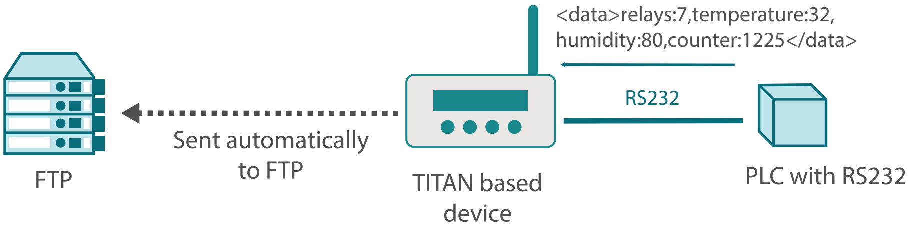
2. Description of the Example

In this example, a TITAN-based device will be configured to collect, store and send serial data from a PLC to a server via FTP. Each hour, the PLC sends an ASCII data frame via its RS232 serial port (the TITAN can also collect binary data) of the type:

```
<data>relays:7,temperature:32,humidity:80,counter:1225</data>
```

In addition to this, when an alert arises, it also sends a similar data frame through the serial port, indicating the alarm and the reason for it.

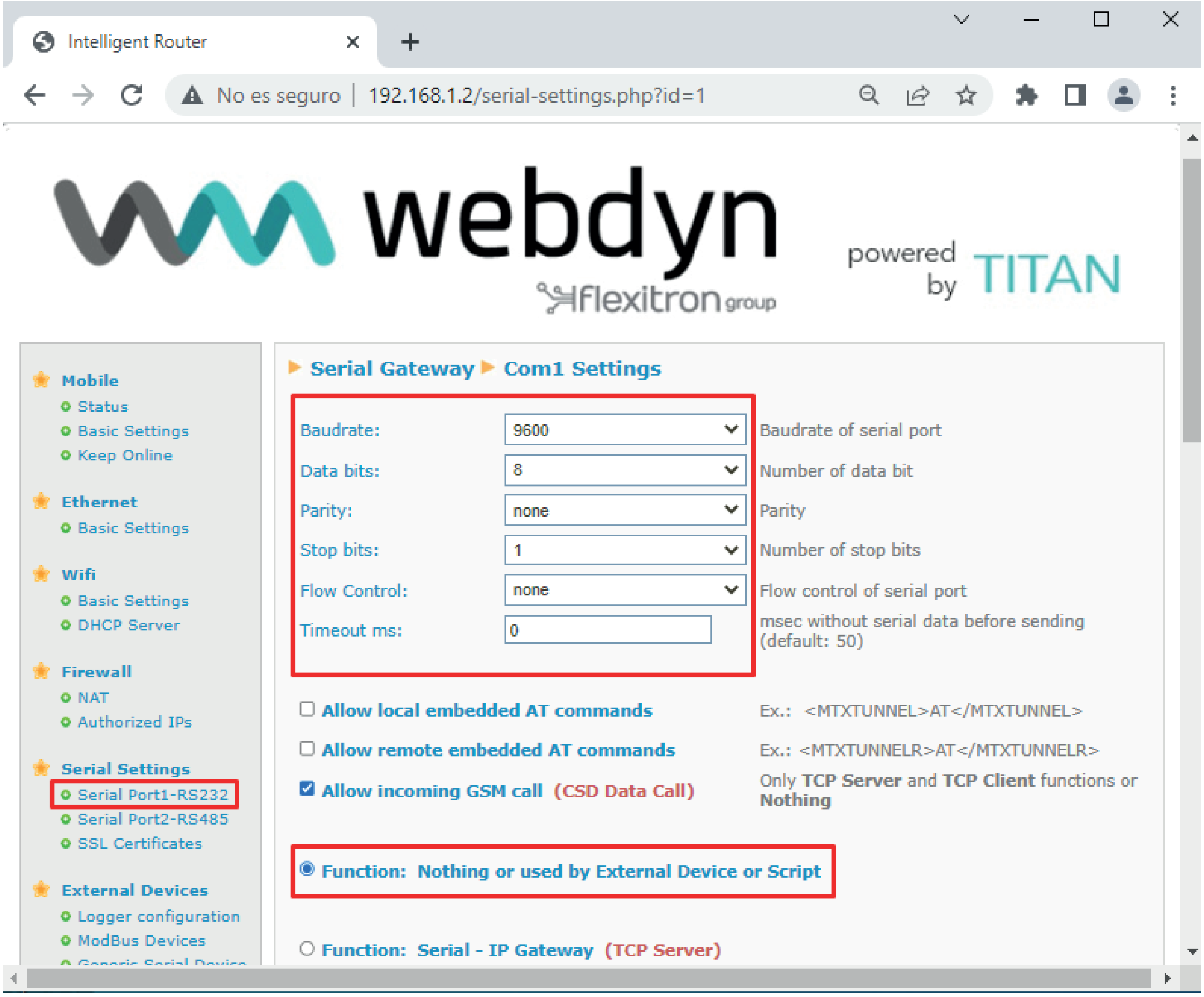
```
<alarm>alarm:1,subject:intrussion</data>
```



3. Configuring the Associated Serial Port

First we must decide which of the TITAN-based device's ports the device 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:



Serial Gateway ▸ Com1 Settings

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**

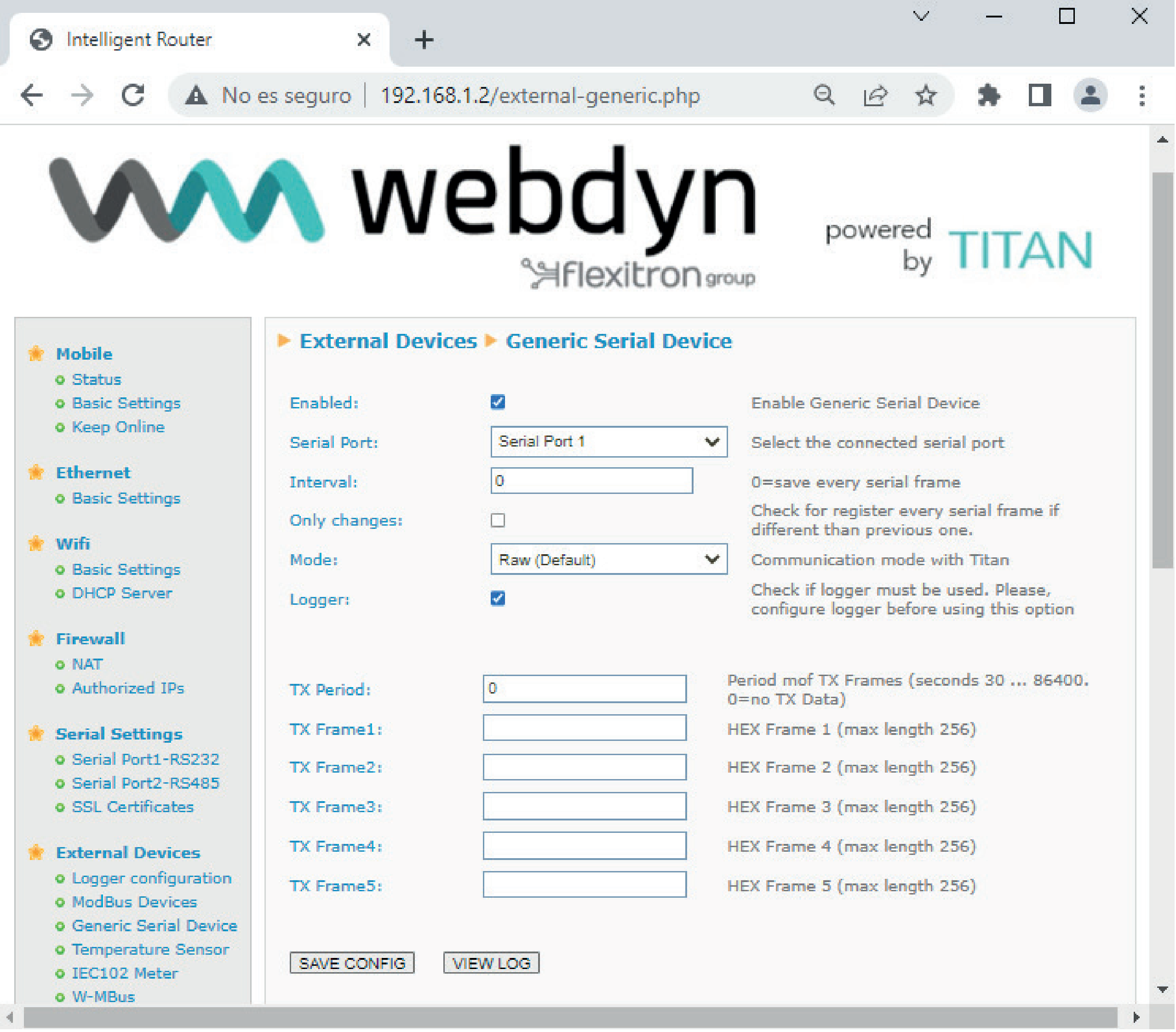
● Function: Nothing or used by External Device or Script

○ Function: Serial - IP Gateway (TCP Server)

4. Log Type Configuration

The next step is to configure the type of logging, i.e. we must indicate the serial port that will be used, the number of frames that we want to log (all, 1 out of 2, 1 out of 3, etc.) or if we only want to log the frames with changes, etc.

In our example we want to capture all serial frames coming from the PLC, which will send us a frame with data each hour, plus any alarms that occur. To do this, click on the link: “External Devices > Generic Serial Device” and configure it as shown below:



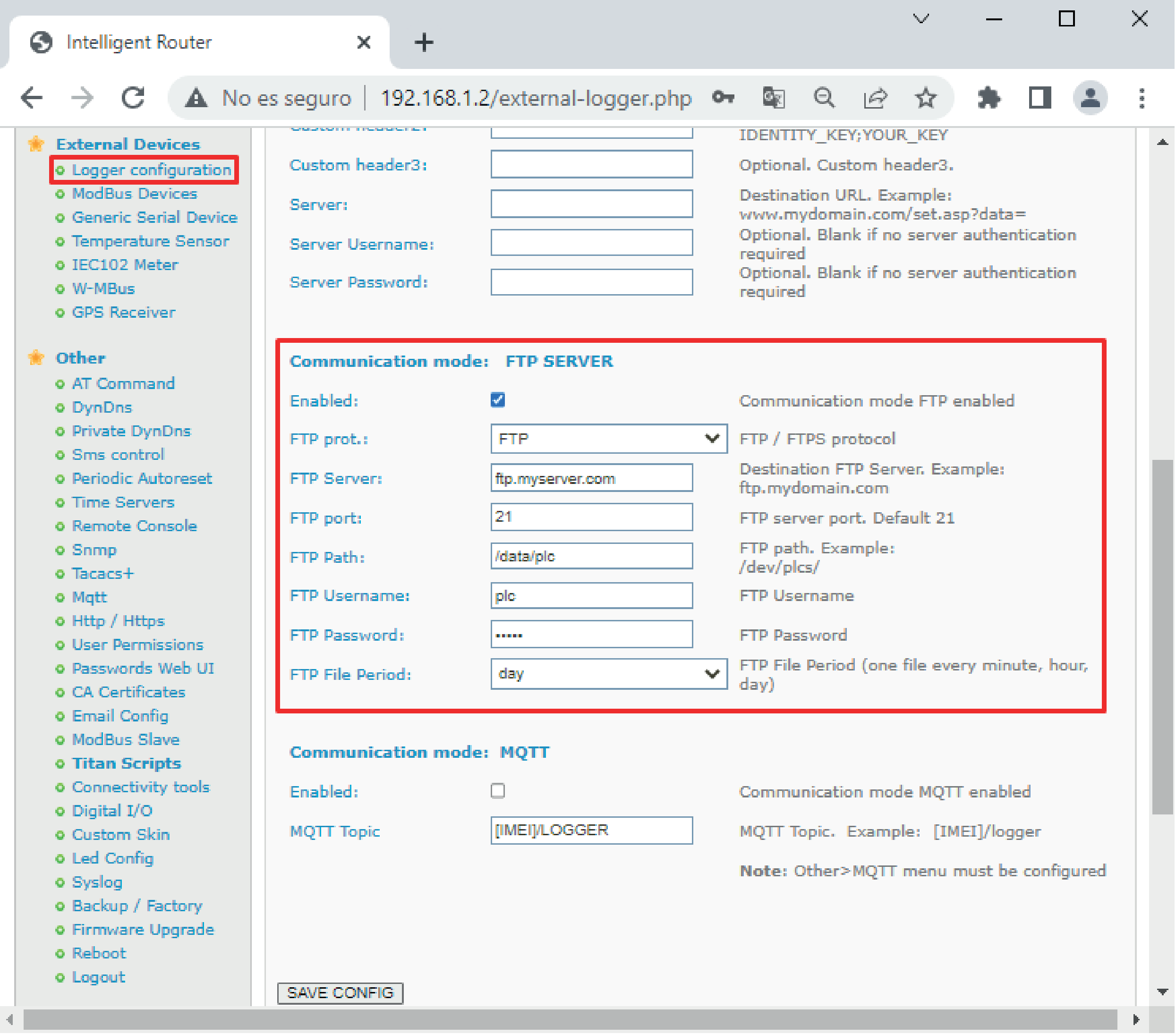
External Devices ▸ Generic Serial Device

Enabled:	☑	Enable Generic Serial Device
Serial Port:	Serial Port 1	Select the connected serial port
Interval:	0	0=save every serial frame
Only changes:	☐	Check for register every serial frame if different than previous one.
Mode:	Raw (Default)	Communication mode with Titan
Logger:	☑	Check if logger must be used. Please, configure logger before using this option
TX Period:	0	Period mof TX Frames (seconds 30 ... 86400. 0=no TX Data)
TX Frame1:		HEX Frame 1 (max length 256)
TX Frame2:		HEX Frame 2 (max length 256)
TX Frame3:		HEX Frame 3 (max length 256)
TX Frame4:		HEX Frame 4 (max length 256)
TX Frame5:		HEX Frame 5 (max length 256)

5. Logger Configuration (communication with FTP server)

The last step is to configure the Logger, i.e. storage and the sending method. In this case we will choose FTP, but it could be HTTP / HTTPS or MQTT / MQTTS

Here we are going to configure an FTP server with the IP (or DNS) ftp.myserver.com, we want it to deposit the data in the folder "/data/plc/". The Username will be "plc" and the Password "12345678". Lastly, we must specify how often we want to upload a file with collected data. We can set it to each minute, each hour or each day. We are going to select each hour. This would be the configuration, which can be accessed from the "External Devices > Logger Configuration" menu.



		IDENTITY_KEY;YOUR_KEY
Custom header3:		Optional. Custom header3.
Server:		Destination URL. Example: www.mydomain.com/set.asp?data=
Server Username:		Optional. Blank if no server authentication required
Server Password:		Optional. Blank if no server authentication required

Communication mode: FTP SERVER

Enabled:	☑	Communication mode FTP enabled
FTP prot.:	FTP	FTP / FTPS protocol
FTP Server:	ftp.myserver.com	Destination FTP Server. Example: ftp.mydomain.com
FTP port:	21	FTP server port. Default 21
FTP Path:	/data/plc	FTP path. Example: /dev/plcs/
FTP Username:	plc	FTP Username
FTP Password:	•••••	FTP Password
FTP File Period:	day	FTP File Period (one file every minute, hour, day)

Communication mode: MQTT

Enabled:	☐	Communication mode MQTT enabled
MQTT Topic	[IMEI]/LOGGER	MQTT Topic. Example: [IMEI]/logger
		Note: Other>MQTT menu must be configured

6. Other Considerations

- After configuring the TITAN-based device we will need to perform a reset to accept the new configuration.
- Files sent to the FTP server will have file names similar to:

  358709050113764-2015-09-20-13-44.txt

  Where:

358709050113764	IMEI of the TITAN-based Device.
2015-09-20-13-44	File creation date.

- The contents of the files will be JSON objects, with content similar to:

  {“TYPE”:”SERIAL”,”TS”:” 2021-11-14T06:55:50Z”,”IMEI”:”358709050113764”,”P”:
  “”,”DATA”:”3c646174613e72656c6179733a372c74656d7065726174757265
  3a33322c68756d69646974793a38302c636f756e7465723a313232353c2
  f646174613e0d0a”}

  The data are encoded in hexadecimal form as the TITAN-based device is also capable of logging binary characters (not ASCII). For example “3c64617461...” = “<data ...”