

TITAN

Nota de aplicación 27

Lectura de dispositivos Modbus con envío de lecturas
a la nube por HTTP

Lectura de dispositivos Modbus con envío a la nube por HTTP

1. Detalles del escenario

Los equipos basados en TITAN disponen de todas las funcionalidades típicas de un router 4G/3G/2G pero además cuentan con una serie de prestaciones adicionales que lo convierten en uno de los equipos con más prestaciones del mercado.

Una de las prestaciones adicionales es la capacidad de interrogar de forma autónoma a dispositivos Modbus RTU o TCP, con el posterior envío de los datos a un servidor HTTP, FTP o MQTT.

Como siempre, se ilustrará esta capacidad con un ejemplo sencillo.

2. Descripción del ejemplo

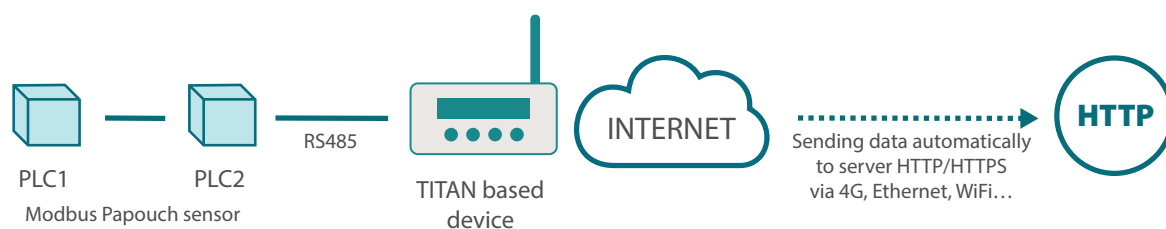
En este ejemplo se va a configurar un equipo basado en TITAN para recoger, almacenar y enviar por HTTP los registros Modbus de 2 PLCs. Dichas lecturas se realizarán cada 1 minuto.

Del PLC1 deben leerse los registros Modbus:

1;10;11;12;55;56;69;70;72;73;74;75;76;77;78;79;80;100;101;102;103;104;105;106;107;108;109;120;121;122;123;124;130;131;132;133;152;153;154;160;161;162;163;164;165;166;170

Del PLC2 deben leerse los registros:

10;11;12;13;14



Es decir, del PLC1 se necesitan obtener un mapa de registros variado, no siempre registros consecutivos. El PLC2 es más sencillo y únicamente necesitamos 5 registros consecutivos

Los PLCs son dispositivos RS485, por lo que se utilizará Modbus RTU, pero el escenario podría ser perfectamente válido para equipos Modbus TCP (con Ethernet) o una mezcla de ambos (Modbus TCP y Modbus RTU).

3. Configuración del Puerto Serie del equipo basado en TITAN donde se conectarán los dispositivos Modbus

Imaginemos que los PLCs, que disponen de puerto RS485, tienen una configuración de puerto serie: 9600,8,N,1. La primera tarea a realizar es configurar el apartado Serial Port2-RS485 del equipo basado en TITAN. Lo configuraremos como se muestra en la siguiente figura:

The screenshot shows the webdyn configuration interface. The left sidebar contains a menu with categories: Mobile, Ethernet, Firewall, Serial Settings, and External Devices. Under Serial Settings, 'Serial Port2-RS485' is highlighted with a red box. The main content area is titled 'Serial Gateway > Com2 Settings'. It contains a table of serial port settings, which is also highlighted with a red box. The settings are: Baudrate (9600), Data bits (8), Parity (none), Stop bits (1), and Timeout ms (0). Below this table are three checkboxes: 'Allow local embedded AT commands', 'Allow remote embedded AT commands', and 'Allow incoming GSM call (CSD Data Call)'. At the bottom, a red box highlights the 'Function' dropdown, which is set to 'Nothing or used by External Device or Script'.

webdyn powered by **TITAN**
flexitron group

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**

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

4. Configuración del Titan para la lectura de dispositivos Modbus

We click on the link: “External Devices > Modbus Devices” and configure the screen as shown below:

webdyn powered by **TITAN**
flexitron group

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 CONFIG VIEW LOG

Dev. name / ID	Addr.	Command	Start @	Num word/bit	Reg Type	Period		
1	1	0x03	1;10;55;69;72;100;120;130;152;160;170	1;3;2;2;9;19;5;4;3;7;1	WORD	1	Del	
2	2	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)

Es decir, del PLC 2 queremos leer los registros 10,11,12,13,14. Pues simplemente es necesario indicar en el campo “Start” el registro 10 y en el “Number Words” 5 (pues queremos leer 5 registros, del 10 al 14).

El PLC1 es más complejo, ya que disponemos de un mapa de registro no consecutivo. Por ello, los diferentes bloques de registros a leer los pondremos separados por “;” (punto y coma). Es decir, si queremos leer los registros:

1;10;11;12;55;56;69;70;72;73;74;75;76;77;78;79;80;100;101;102;103;104;105;106;107;108;109;120;121;122;123;124;130;131;132;133;152;153;154;160;161;162;163;164;165;166;170

Tendremos que poner en el campo Start (el registro inicial de cada bloque)

1;10;55;69;72;100;120;130;152;160;170

Y en el campo “Number Words” (el nº de registros a leer de cada bloque)

1;3;2;2;9;10;5;4;3;7;1

5. Configuración del logger (comunicación con servidor HTTP)

El siguiente paso es configurar el Logger, esto es, el sistema de almacenamiento y envío de datos realizado por el propio equipo basado en TITAN. En este ejemplo vamos a configurar el equipo basado en TITAN para enviar los datos a un servidor HTTP (podría utilizarse también HTTPS o FTP o MQTT / MQTTS).

Como servidor HTTP utilizaremos la URL <http://www.mydomain.com/set.asp?data=> donde recibiremos en la variable “data” los datos en formato JSON de cada lectura realizada (timestamp, ID equipo, ...)

Tal y como se ve en la siguiente figura, accederemos a través del menú “External Devices > Logger Configuration” y configuraremos dicha sección como se muestra a continuación:

webdyn powered by **TITAN**
flexitron group

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

External Devices > Logger

ID: ID-869101054287764 Optional. Device identification

Send mode: LIFO Send mode (normally FIFO)

Time format: unix (yyyy-mm-ddTHH:mm:ss) 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: WEB PLATFORM (HTTP REST)

Enabled: ☒ Communication mode HTTP enabled

Mode: HTTP GET (JSON) Method of sending data

Custom parameters: Optional. Ex: &a=1&b=2 only for "HTTP GET/PUT (PARAMETERS)" modes

Custom header1: Optional. Custom header1. For example: Content-type:application/json

Custom header2: Optional. Custom header2. For example: IDENTITY_KEY;YOUR_KEY

Custom header3: Optional. Custom header3.

Server: www.mydomain.com/set.as 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

6. Consideraciones varias

- Tras la configuración del equipo basado en TITAN es necesario realizar un reset para coger la nueva configuración y empiecen las lecturas y envíos.
- Cada vez que el equipo basado en TITAN envía una medida a un servidor HTTP lo hace mediante un objeto JSON del siguiente tipo:

Ejemplo JSON PLC2

```
{“IMEI”:” 869101054287764”,“TYPE”:”MODB”,“TS”:”2022-07-15T10:14:02Z”,“ID”:”2”,“A”:”2”,“ST”:”10”,“N”:”5”,“V”:”[10,11,12,0,0]”,“P”:”ID- 869101054287764”}
```

Ejemplo JSON PLC1

```
{“IMEI”:” 869101054287764”,“TYPE”:”MODB”,“TS”:”2022-07-15T10:09:01Z”,“ID”:”1”,“A”:”1”,“STX”:”[1,10,55,69,72,100,120,130,152,160,170]”,“NX”:”[1,3,2,2,9,10,5,4,3,7,1]”, “PX”:”[0,1,4,6,8,17,27,32,36,39,46]”, “V”:”[1,10,11,12,55,56,69,70,72,73,74,75,76,77,78,79,80,100,101,102,103,104,105,106,107,108,109,120,121,122,123,124,130,131,132,133,152,153,154,160,161,162,163,164,165,166,170]”,“P”:”ID- 869101054287764”}
```

Donde:

IMEI: indica un identificador único para el módem

TYPE: indica el tipo de dato (MODB = lectura modbus)

TS: indica el Timestamp (es decir, la hora de lectura de la medida)

ID: nombre o identificador del dispositivo modbus

A: address del dispositivo modbus

ST: indica la dirección del primero registro modbus leído

STX: array que indica la dirección de los 1os registros modbus en caso de leer grupos de registros

N: indica el número de words leídos

NX: array que indica el número de words leídos en caso de leer grupos de registros

PX: array que indica la posición del registro inicial de cada bloque dentro de V.

V: Array con los datos leídos

P: indicaría el campo ID configurado en el Logger

Nótese que hay una diferencia significativa entre los datos enviados por PLC1 y PLC2. PLC1 dispone de grupos, por tanto, el campo ST y N se sustituyen en el JSON por STX y NX, arrays donde están almacenados los registros iniciales y número de registros de cada bloque. También aparece el registro PX que indica la posición inicial del grupo dentro del array V (realmente PX no es necesario, puesto que se puede calcular, pero se incluye para facilitar la operación de decodificación en el servidor).