

TITAN

Nota de aplicación 32

Lectura de dispositivos Modbus con envío de lecturas
a un bróker MQTTS (con seguridad SSL/TLS)

Lectura de dispositivos Modbus con envío de lecturas a un bróker MQTTS (con seguridad SSL/TLS)

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 WEB, FTP o MQTT.

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

2. Descripción del ejemplo

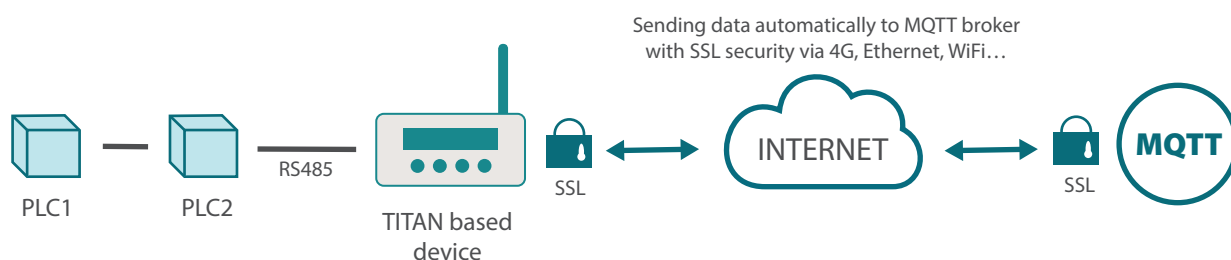
En este ejemplo se va a configurar un equipo basado en TITAN para recoger, almacenar y enviar por MQTT los registros Modbus de dos PLCs. Dichas lecturas se realizarán cada 10 minutos.

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 displays the webdyn configuration interface, powered by TITAN. 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'. A red box highlights the serial port configuration fields: Baudrate (9600), Data bits (8), Parity (none), Stop bits (1), and Timeout ms (0). Below these fields 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

Hacemos click en el link: “External Devices > Modbus Devices” y configuramos la pantalla como se muestra a continuación:

**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
 - Snmp
 - Tacacs+

External Devices > ModBus RTU / TCP

Enabled: ☒

Serial Port: Serial Port 2

Logger: ☒

Enable Modbus Devices

Select the connected serial port if needed

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	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:

Address:

Command: 0x01

Start:

Number Words / Bits:

Reg Type: WORD

Period: 1

Insert the device name or ID

Modbus RTU address or IP:port address

Modbus read command

Address of the first register

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

Type of registers for command 0x03/0x04

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 MQTT)

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 para enviar los datos a un bróker MQTT.

Como bróker MQTT utilizaremos el bróker de test Mosquitto ubicado en test.mosquitto.org. Ahí se enviarán los datos en formato JSON de cada lectura realizada (timestamp, ID equipo, ...). Para hacerlo totalmente seguro, utilizaremos comunicaciones SSL/TLS, con necesidad de utilizar certificado de Servidor y certificado de Cliente.

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

Firmware Upgrade
Reboot
Logout

External Devices > Logger

ID: ID-1234 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 transimition.

Communication mode: MQTT

Enabled: ☒ Communication mode MQTT enabled

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

Note: Other>MQTT menu must be configured

En esta pantalla hemos activado el modo de envío MQTT. Todos los datos recogidos por el equipo basado en TITAN serán publicados en el topic [IMEI]/logger. Nótese que el equipo substituirá el tag [IMEI] por su IMEI real. Es decir, si el IMEI del equipo es 357299070082380 (el IMEI puede verse en el menú Mobile > Status) el topic usado por el equipo basado en TITAN para publicar los datos Modbus recogidos será: 357299070082380/logger.

Por otro lado, si activamos el modo del logger, deberemos configurar el cliente MQTT en la sección "Other > Mqtt".

Other > MQTT Client

Enabled: ☒ Enable MQTT client

MQTT Broker: 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: Device identification

MQTT Qos: MQTT Quality Of Service (0 ... 2)

MQTT Keepalive: Seconds for keepalive (30 ... 3600)

MQTT Persistence: ☐ Data persistence

MQTT AT Topic: This topic will be subscribed for receiving AT Commands (usefull for individual device)

MQTT AT Resp Topic: This topic will be used for publishing the AT Command Responses of AT Topic

MQTT AT Topic 2: This topic will be subscribed for receiving AT Commands (usefull for groups)

MQTT AT Resp Topic 2: This topic will be used for publishing the AT Command Responses of AT Topic 2

MQTT AT Topic 3: This topic will be subscribed for receiving AT Commands (usefull for all devices)

MQTT AT Resp Topic 3: This topic will be used for publishing the AT Command Responses of AT Topic 3

MQTT Script Topic 1: When data is received in this topic the 'Topic Script' will be executed.

MQTT Script Topic 2: When data is received in this topic the 'Topic Script' will be executed.

Activaremos el campo "Enabled". En el campo "MQTT Broker" pondremos: `ssl://test.mosquitto.org:8884`, indicando la URL del bróker MQTTS al que queremos conectarnos. En MQTT ID pondremos un identificador único para nuestro dispositivo (si conectamos 2 equipos con el mismo ID, se irán desconectando continuamente, ya que un bróker no puede tener 2 dispositivos con el mismo ID. Si queremos poder enviar comandos AT al equipo basado en TITAN (por ejemplo, para configurarlo o ver el estado desde un teléfono móvil que también esté conectado al bróker MQTT) complementaremos los campos MQTT AT Topic y MQTT AT Response Topic.

Todos los comandos AT enviados al bróker al topic MQTT AT Topic serán ejecutados por el equipo basado en TITAN. Una vez ejecutado, el resultado del comando AT será publicado por el equipo basado en TITAN en el topic MQTT AT Response Topic.

Seguidamente configuraremos la sección de certificados.

► Other ► MQTT Client ► Certificates for MQTTS

If you have a specific CA Server Certificate, please, configure the section: [Other > CA Certificates](#)

Mandatory only if client authentication is required by MQTTS broker. PEM format.

Client certificate: file 'mqttts_client.crt'	Seleccionar archivo	Ninguno archivo selec.	Upload	not uploaded
Client KEY: file 'mqttts_client.key'	Seleccionar archivo	Ninguno archivo selec.	Upload	not uploaded

DELETE CERTIFICATES

Para ello es útil seguir las instrucciones indicadas en: <http://test.mosquitto.org/>

Lo primero que haremos es bajarnos el certificado CA de mosquitto.org.crt (PEM format) que puede encontrarse en la página anterior. Ese certificado tendremos que incluirlo en la sección “Other > CA certificates”, pues el equipo basado en TITAN lo utilizará para autenticar al bróker MQTTS en el momento de la conexión.



webdyn

flexitron group

powered by TITAN

- ★ Mobile
 - ◆ Status
 - ◆ Basic Settings
 - ◆ Keep Online
- ★ Ethernet
 - ◆ Basic Settings
- ★ Firewall
 - ◆ Authorized IPs
- ★ Serial Settings

► Other ► CA-Root Certificates

Custom CA-Root Certificates (PEM format)

User CA-Root-1	Seleccionar archivo	mosquitto.org.crt	Upload	uploaded
Certificate fingerprint (SHA1): 06:90:7E:A4:EC:A0:F7:70:28:12:DC:51:1F:ED:62:A7:99:AC:71:9A				
User CA-Root-2	Seleccionar archivo	Ninguno archivo selec.	Upload	not uploaded

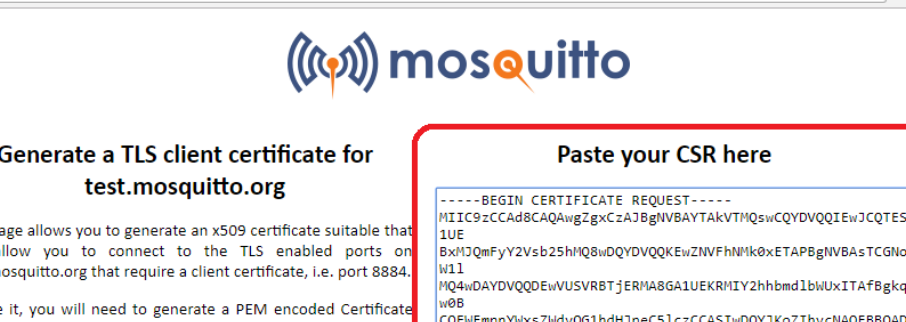
Otra opción, en caso de no querer subir el certificado, sería seleccionar la casilla “Allow all certificates”, pero no es una opción recomendable por insegura. Sólo válida para pruebas.

Lo segundo es ejecutar las sentencias que se encuentran en este enlace: <http://test.mosquitto.org/ssl/> para lo cual deberemos tener instalado openssl en nuestro PC. Es decir, ejecutaremos:

```
openssl genrsa -out client.key
```

```
openssl req -out client.csr -key client.key -new
```

Tras lo cual tenemos los ficheros client.key y client.csr.



The screenshot shows a web browser window with the address bar displaying `test.mosquitto.org/ssl/`. The page features the Mosquitto logo and a heading "Generate a TLS client certificate for test.mosquitto.org". Below this, there is a paragraph explaining the purpose of the page and a "Generate a CSR using the openssl utility" section. A red box highlights the "Paste your CSR here" section, which contains a text area with a sample CSR and a "Submit" button. At the bottom, there is a "Donate" button and a row of payment logos (MasterCard, Visa, Bitcoin, etc.).

Intelligent Router x Generate client certificate: x

← → ↺ No es seguro | test.mosquitto.org/ssl/ ☆ 🛡️ ⚙️ 📄

mosquitto

Generate a TLS client certificate for test.mosquitto.org

This page allows you to generate an x509 certificate suitable that will allow you to connect to the TLS enabled ports on test.mosquitto.org that require a client certificate, i.e. port 8884.

To use it, you will need to generate a PEM encoded Certificate Signing Request (CSR) and paste it into the form. After you submit the form, the certificate will be generated for you to download.

Generate a CSR using the openssl utility

Generate a private key:

```
openssl genrsa -out client.key
```

Generate the CSR:

```
openssl req -out client.csr -key client.key -new
```

When you are generating the CSR, please do not use the default values. At a minimum, the CSR must include the Country, Organisation and Common Name fields.

You should paste the contents of client.csr into the form.

Paste your CSR here

```
-----BEGIN CERTIFICATE REQUEST-----
MIIC9zCCA8CAQAwZgxhCzA3BgNVBAYTAKVTMswCQYDVQEEwJQCTESMBAGA
1UE
BxMjQmFyY2Vsb25hMQ8wDQYDVQQKEWZNVFhMk0xETAPBgNVBAsTCGNoYW5nZ
W11
MQ4wDAYDVQQDEwVUSVRBTjERMA8GA1UEKRMiY2hhbmdlbWUxITAFBgkqhkiG
9w0B
CQEWEmpnYixsZWdvQ61hdHJpcC51czCCASIQDYJOKZ1hvcNAQEBBQADggEPA
DCC
AQoCggEBAMsdvQAIwFHYBKkun30btzL7X44I9eHwOfjIPT7+RJIeqKuoM7s1
A+c
ygP3UNMvWfGxJKz4nMw0Bpp/bN1bY3dCBnLS9n39Z68vm1dvKvDdEhhkZqX1R
tAu
Jkic1ZFSrcPedi8D12Q8CuXQMUnFYhKGcMdb6VMcq77tX3nfOD1wygDDoEsqj
u9i
-----
```

Submit

[Donate](#)

MasterCard Visa Bitcoin

► **Other** ► **MQTT Client** ► **Certificates for MQTTS**

If you have a specific CA Server Certificate, please, configure the section: **Other > CA Certificates**

Mandatory only if client authentication is required by MQTTS broker. PEM format.

Client certificate: file 'mqttps_client.crt'	Seleccionar archivo Ninguno archivo selec.	Upload	uploaded
Client KEY: file 'mqttps_client.key'	Seleccionar archivo Ninguno archivo selec.	Upload	uploaded

contact@webdyn.com | webdyn.com
V1.0 subject to changes | Webdyn © by Flexitron Group

Ejemplo JSON PLC2

```
{“IMEI”:”357299070082380”,”TYPE”:”MODB”,”TS”:”2022-07-15T10:14:02Z”,”ID”:”2”,”A”:”2”,”ST”:”10”,”N”:”5”,”V”:”[10,11,12,0,0]”,”P”:”ID-1234”}
```

Ejemplo JSON PLC1

```
{“IMEI”:”357299070082380”,”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-1234”}
```

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 primeros 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

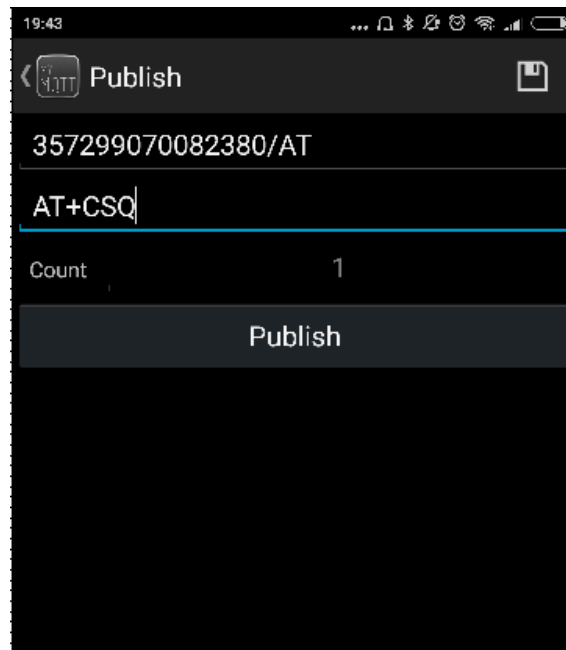
Nótese que hay una diferencia significativa entre los datos enviados por PLC1 y PLC2. PLC1 dispone de grupos, por tanto, el campo ST y N se sustituyen en el JSON por STX y NX, arrays donde están almacenados los registros iniciales y número de registros de cada bloque. También aparece el registro PX que 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).

7. Comunicación con el equipo basado en TITAN mediante comandos AT enviados por MQTT

Como se comentó anteriormente, es posible enviar al equipo basado en TITAN comandos AT vía MQTT. Con ello podemos cambiar una configuración del equipo, comprobar estados como la cobertura,

resetearlo, conmutar un relé, etc ...

Por ejemplo, si queremos consultar la cobertura, podemos usar el programa myMQTT de Android, enviando el comando AT+CSQ al topic [IMEI]/AT, que en el caso de este ejemplo es 357299070082380/AT.



Una vez pulsado el botón “Publish”, el comando será recibido por el equipo basado en TITAN y ejecutado. La respuesta será publicada por el equipo basado en TITAN en el topic 357299070082380/ATR, tal y como configuramos anteriormente.

