

# ROUTER TITAN

## Nota de aplicación 69

---

Sending data to  
eSight platform

# Sending data to eSight platform

## 1. Detalles del escenario

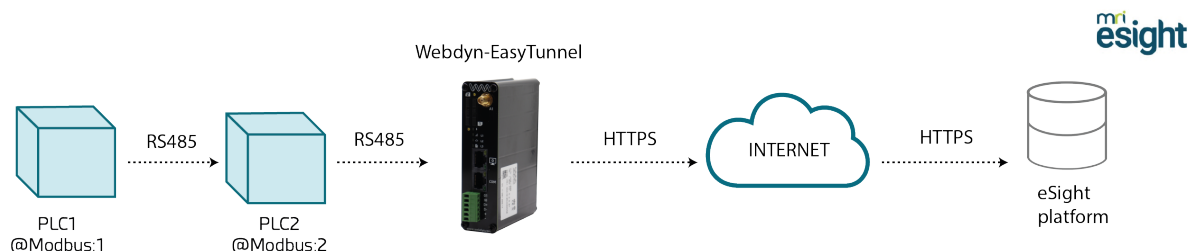
Los router Titan disponen de todas las funcionalidades típicas de un router 4G/3G/2G pero además cuentan con una serie de prestaciones adicionales que lo convierten en uno de los routers con más prestaciones del mercado.

Una de las prestaciones adicionales es la capacidad de datalogger donde el router Titan almacena en su memoria no volátil distintos tipos de registros en un formato JSON. Estos registros almacenados pueden provenir de lecturas MODBUS, o de capturas de datos SERIE por sus puertos RS232 / RS485, de posiciones GPS, etc. Estos registros de tipo JSON son almacenados por el router Titan en su memoria interna no volátil para ser posteriormente enviados a una plataforma remota mediante comunicaciones tipo HTTP, HTTPS, MQTT, MQTTS, FTP, FTPS.

Como se indica, el router Titan almacena en su memoria interna dichos registros JSON en un formato predeterminado y propietario. En ocasiones, esto puede ser un problema para la comunicación con aquellas plataformas que esperan recibir la información en un determinado formato (es decir, con un formato distinto de JSON al utilizado por defecto por el router Titan).

En esta nota de aplicación se desarrollará el ejemplo completo para el envío de datos a la conocida plataforma de MRI eSight (<https://www.mrisoftware.com/products/esight-energy-management-software>).

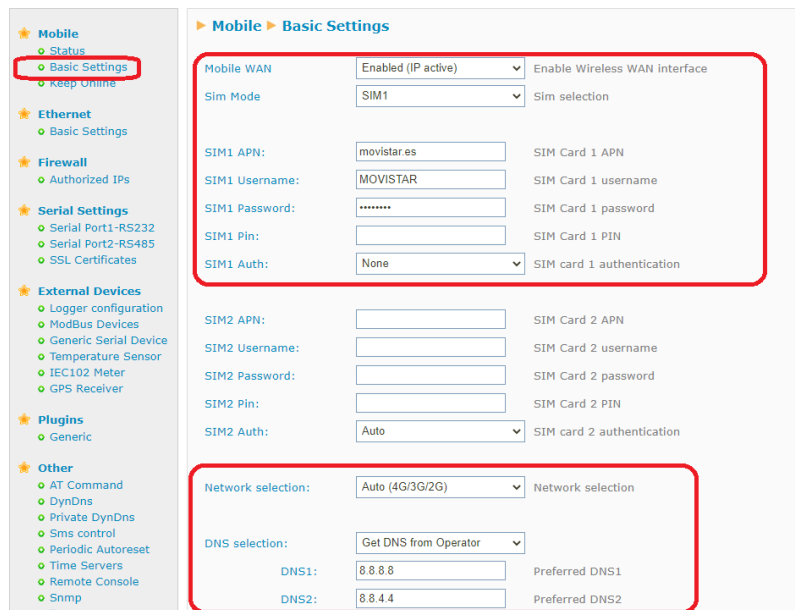
En este caso, vamos a suponer que se deben leer 2 registros de 2 PLCs con comunicaciones Modbus conectados a un Webdyn-EasyTunnel por su puerto RS485.



En concreto se pretende lo siguiente: cada minuto, el dispositivo Webdyn EasyTunnel debe leer los registros Modbus con dirección 40000 y 40001 del PLC1 y los registros 40000 y 40001 del PLC2. En ambos PLCs el registro 40000 corresponde con la temperatura medida y el registro 40001 con la humedad. Las lecturas realizadas deben almacenarse en la memoria interna no volátil del Webdyn EasyTunnel (en su datalogger) y éste debe enviar dichos datos leídos a la plataforma MRI eSight siempre que sea posible (haya cobertura, conectividad IP, ...) La comunicación con los PLCs es a través de un bus RS485 con una configuración 9600,8,N,1.

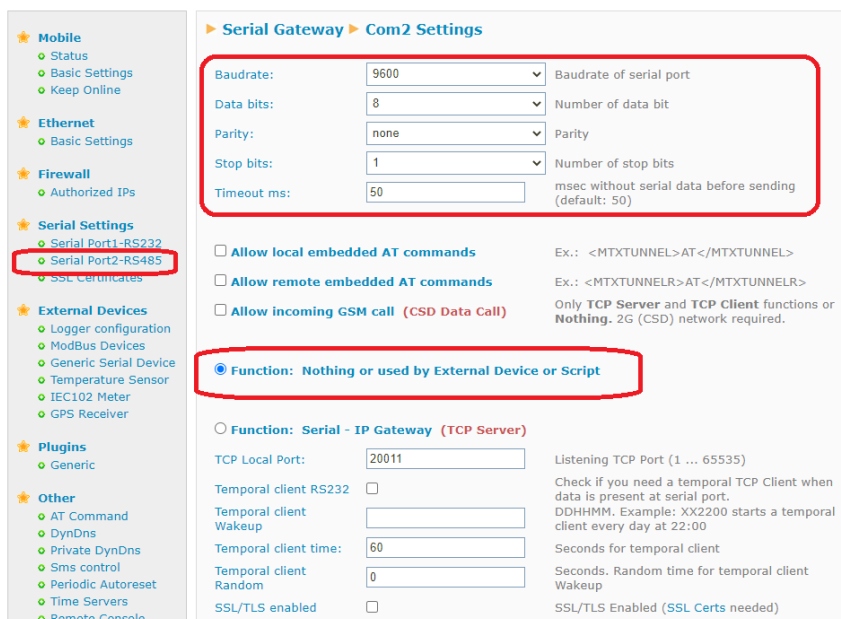
## 2. Configuración de la Mobile WAN

El Webdyn EasyTunnel debe comunicarse con la plataforma MRI eSight a través de comunicaciones 4G/3G/2G, por lo que la sección "Mobile>Basic Settings" debe configurarse apropiadamente en función de la tarjeta SIM utilizada



### 3. Configuración del puerto serie RS485

La comunicación con los dos PLCs se realizará a través de puerto serie RS485 a 9600,8,N,1, por lo que debe configurarse la sección “Serial Settings > Serial Port2-RS485” ajustando los parámetros como se indica a continuación.



### 4. Configuración del Logger

El siguiente paso es configurar el datalogger interno del Webdyn EasyTunnel. Para ello debe acudir al menú “External Devices > Logger configuration”. La configuración debería ser similar a la que se muestra en la siguiente figura:

- ★ **Mobile**
  - Status
  - Basic Settings
  - Keep Online
- ★ **Ethernet**
  - Basic Settings
  - DHCP Server
- ★ **Wifi**
  - Basic Settings
  - DHCP Server
- ★ **Firewall**
  - NAT
  - Authorized IPs
  - MAC Filter
  - Routes
- ★ **Serial Settings**
  - Serial Port1-RS232
  - Serial Port2-RS485
  - SSL Certificates
- ★ **External Devices**
  - **Logger configuration**
  - ModBus Devices
  - Generic Serial Device
  - Temperature Sensor

### ▶ External Devices ▶ Logger

ID:  Optional. Device identification

Send mode:  Send mode (normally FIFO)

Time format:  Time format used in timestamp logger data

Use script:

Use array:

Check date:  Check for customized json using 'Json Transformer Script' in **Script** section. Check if you want to send more than one JSON per transmission. Save data in Logger only if date has been set (check Time Servers)

**Communication mode: WEB PLATFORM (HTTP REST)**

Enabled:  Communication mode HTTP enabled

Mode:  Method of sending data

Custom header1:

Custom header2:

Custom header3:

Server:  Destination URL. Example: www.mydomain.com/setdata.php

Server Username:  Optional. Blank if no server authentication required

Server Password:  Optional. Blank if no server authentication required

Cabe destacar los siguientes parámetros:

- “ID” proporciona al registrador una identificación única. Sirve como la primera parte de un identificador único (código de importación eSight)
- “Send mode” seleccionar FIFO.
- “Time format” es necesario seleccionar formato “std”.
- “Check date” activamos la casilla.
- “Mode” seleccionamos HTTPS POST (JSON).
- “Server” aquí introduciremos la URL del servidor, ingrese su webhook eSight en el cuadro.

## 5. Configuración de la sección Modbus

En esta sección de configuración “External Devices > Modbus Devices” configuraremos las lecturas Modbus que deben realizarse sobre los 2 PLCs

- ★ **Mobile**
  - Status
  - Basic Settings
  - Keep Online
- ★ **Ethernet**
  - Basic Settings
  - DHCP Server
- ★ **Wifi**
  - Basic Settings
  - DHCP Server
- ★ **Firewall**
  - NAT
  - Authorized IPs
  - MAC Filter
  - Routes
- ★ **Serial Settings**
  - Serial Port1-RS232
  - Serial Port2-RS485
  - SSL Certificates
- ★ **External Devices**
  - Logger configuration
  - **ModBus Devices**
  - Generic Serial Device
  - Temperature Sensor
  - IEC102 Meter
  - EV Charge Point
- ★ **VPN**
  - IPsec
  - OpenVPN Client
  - OpenVPN Server

### ▶ External Devices ▶ ModBus RTU / TCP

Enabled:  Enable Modbus Devices

Serial Port:  Select the connected serial port if needed

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

Dev. name	Addr.	Comm.	Start @	Num word/bit	Reg Type	Period			
PLC1	1	0x04	40000	2	WORD	1	Edit	Del	Test
PLC2	2	0x04	40000	2	WORD	1	Edit	Del	Test

Device name / ID:  Insert the device name or ID

Address:  Modbus RTU address or IP:port address

Command:  Modbus read command

Start:  Address of the first register

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

Reg Type:  Type of registers for command 0x03/0x04

Period:  Read period (minutes)

(Max 40 modbus devices)

Debe habilitarse el servicio Modbus activando la casilla “Enabled”. Debe seleccionar “Serial Port 2”, pues la lectura se realizará a través del puerto RS485. Y también debe activarse la casilla “Logger”, pues los registros Modbus leídos de los PLCs deben almacenarse en el datalogger interno del Webdyn EasyTunnel.

También deben crearse dos dispositivos que en este ejemplo hemos llamado PLC1 y PLC2, servirán como segunda parte de un identificador único (código de importación eSight). En “address” debe indicarse la dirección modbus de cada uno de los PLCs, por lo que en PLC1 pondremos la dirección 1 y en PLC2 la dirección 2. El comando modbus que utilizaremos para la lectura de los registros 40000 y 40001 será el 0x04, por lo que seleccionaremos 0x04 en el campo “Command”. El registro iniciar a leer en ambos PLCs es el registro 40000, pues indicaremos en el campo “Start” el valor 40000. Pretendemos leer 2 registros modbus de cada PLC (el 40000 y 40001) por lo que en el campo “Number Words” indicaremos 2. El tipo de registro “Reg Type” lo seleccionaremos como “WORD” y en “Period” seleccionaremos 1, pues queremos lecturas de los registros Modbus cada minuto.

## 6. Configuración en eSight

El formato de JSON estándar generado por el Webdyn EasyTunnel con los datos leídos tendrá un formato como el que sigue:

```
{“IMEI”:”869101054286683”,”TYPE”:”MODB”,”TS”:”2022-11-11T12:17:00Z”,”ID”:”PLC1”,”A”:”1”,”ST”:”40000”,”N”:”2”,”V”:[225,62],”P”:”ID000”},
```

```
{“IMEI”:”869101054286683”,”TYPE”:”MODB”,”TS”:”2022-11-11T12:17:01Z”,”ID”:”PLC2”,”A”:”2”,”ST”:”40000”,”N”:”2”,”V”:[225,62],”P”:”ID000”}
```

Será necesario desarrollar un pequeño script para adaptarlo a las necesidades de la plataforma eSight. A continuación, se muestra un código que puede servir como ejemplo:

```
let inJSON = JSON.parse(inData);

let meter_id = inJSON.ID;
let site_name = inJSON.P;
let timestamp = inJSON.TS;
let value = inJSON.V[0]; // May need to change/remove indexing

let r = new eS.MeterReading();
r.MeterIdentifier = site_name + “_” + meter_id;
r.Timestamp = timestamp;
r.Value = value;

outData.push(r)
// Un-comment above line when device is setup
```

¿Más dudas?

Escríbanos sus consultas a [soporte@matrix.es](mailto:soporte@matrix.es)