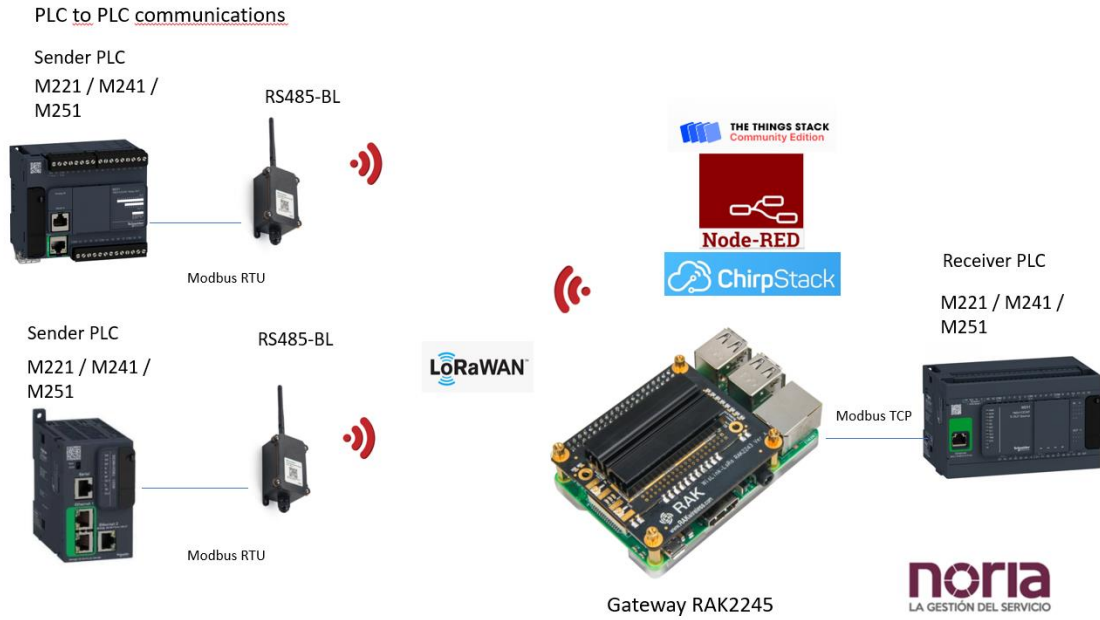


# LORAWAN PLC NETWORK

## PLC M221 as LoRaWAN Node



Let's apply a Modbus to LoRaWAN converter like RS485-BL or RS485-LN

Using a RJ45 patchcord with free wires for the serial Modbus RTU port.

Blue A+

White B-

Configuring PLC Modbus settings

- ✓ Mensajes
- MyController (TM221CE40T)
  - Entradas digitales
  - Salidas digitales
  - Entradas analógicas
  - ≡123 Contadores muy rápidos
  - Generadores de pulsos
  - Bus de E/S
- ETH1
  - Modbus TCP
  - Adaptador Ethernet/IP
- SL1 (línea serie)
  - Modbus



### Configuración de línea serie

#### Ajustes de protocolo

Protocolo Modbus

#### Configuración de línea serie

Velocidad de transmisión 19200

Paridad Ninguna

Bits de datos 8

Bits de parada 1

Medio físico

- RS-485  
 RS-232

Polarización No

Propiedades Configuración Programación

- ✓ Mensajes
- MyController (TM221CE40T)
  - Entradas digitales
  - Salidas digitales
  - Entradas analógicas
  - Contadores muy rápidos
  - Generadores de pulsos
  - Bus de E/S
  - ETH1
    - Modbus TCP
    - Adaptador Ethernet/IP
  - SL1 (línea serie)
    - Modbus

Modbus

Configuración del dispositivo

Dispositivo: Ninguno

Comando Init: [ ]


Ajustes de protocolo

Modo de transmisión:  RTU  ASCII

Direccionamiento:  Esclavo  Maestra Dirección [de 1 a 247]: 1

Timeout de respuesta (x 100 ms): 10

Tiempo entre tramas (ms): 10



## Reading %MW0 with M221

Let's have some value in %MW0

Configuración Programación Visualización Puesta en funcionamiento

Enviar Función de restauración Descargar datos ajenos al programa Copia de seguridad

IL > LD LD > IL - + T DEC 1 - Nuevo POU Comentario

Rung0

Time base of 1 s...

SB\_TB15 %S6

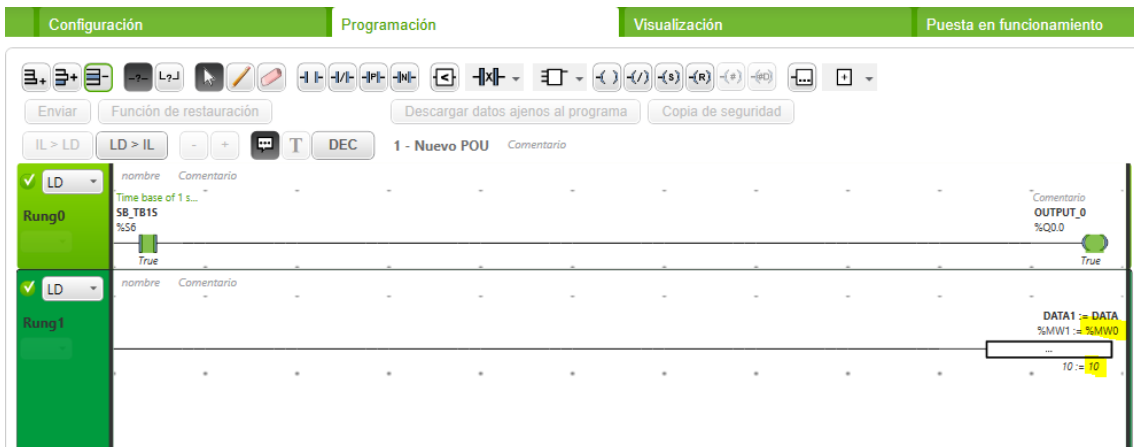
True

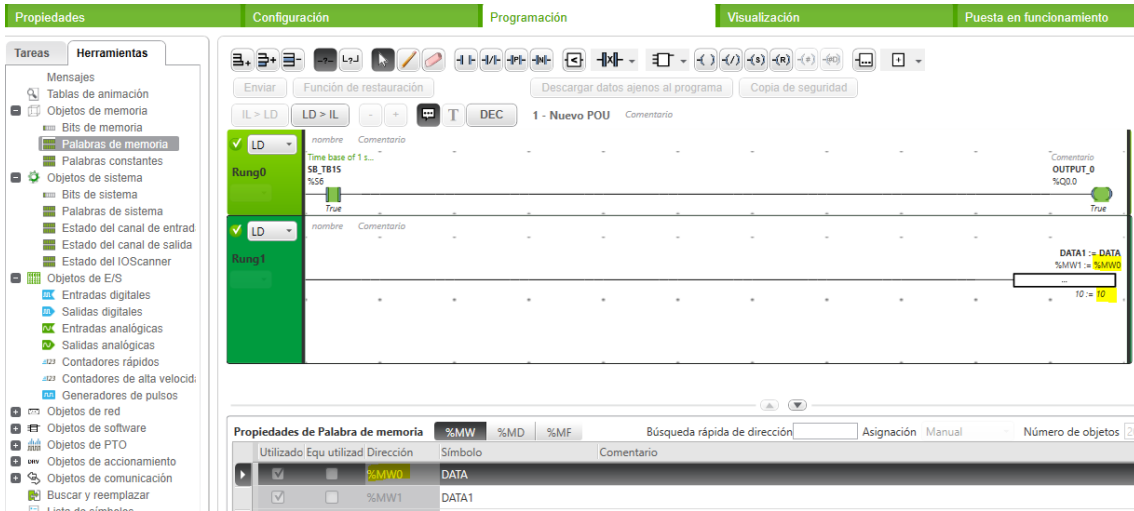
Comentario OUTPUT\_0 %Q0.0 True

Rung1

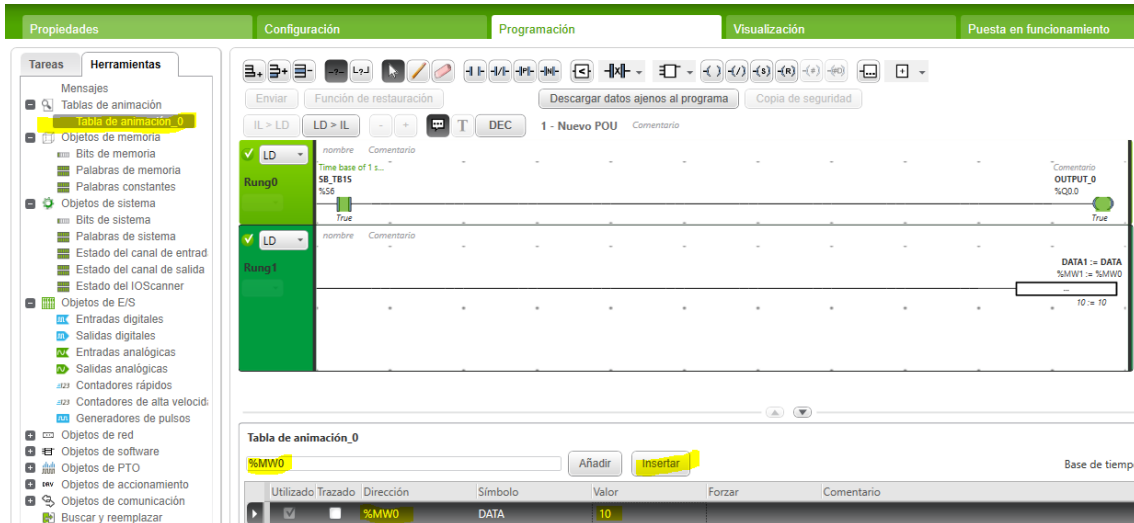
Comentario DATA1 := DATA %MW1 := %MW0

10 := 10



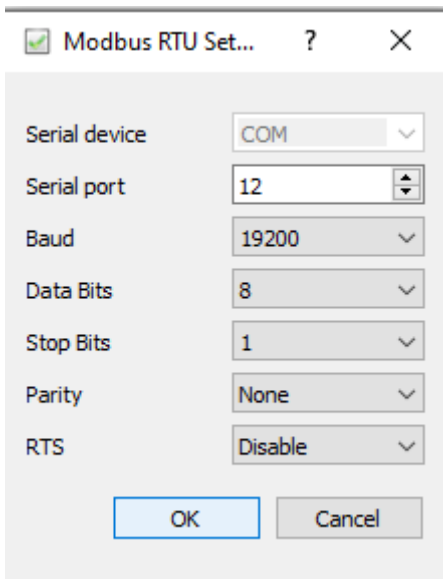


For example inserting the value of %MW0 with animation tables



Let's read with Qmod Master

With this settings



QModMaster

File Options Commands View Help

Modbus Mode: RTU Slave Addr: 1 Scan Rate (ms): 2000

Function Code: Read Holding Registers (0x03) Start Address: 0 Dec

Number of Registers: 1 Data Format: Dec Signed:

10

RTU: \\.\COM12 | 19200,8,1,None Base Addr: 0 Packets: 16 Errors: 0

**Bus Monitor**

Raw Data

```
[RTU]>Tx > 18:48:16:286 - 01 03 00 00 00 01 84 0A
[RTU]>Rx > 18:48:16:305 - 01 03 02 00 0A 38 43
```

**ADU**

```
Type : Tx Message
Timestamp : 18:48:16:286
Slave Addr : 01
Function Code : 03
Starting Address : 0000
Quantity of Registers : 0001
CRC : 840A
```

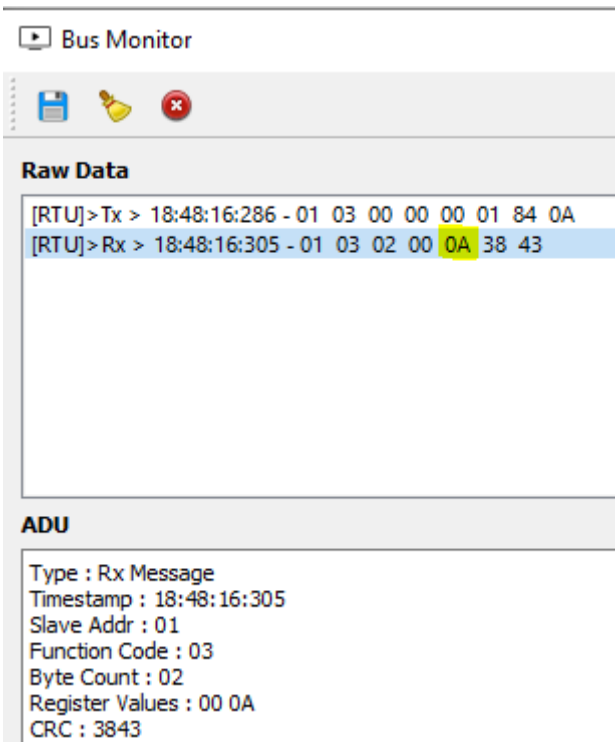
**Bus Monitor**

Raw Data

```
[RTU]>Tx > 18:48:16:286 - 01 03 00 00 00 01 84 0A
[RTU]>Rx > 18:48:16:305 - 01 03 02 00 0A 38 43
```

**ADU**

```
Type : Tx Message
Timestamp : 18:48:16:286
Slave Addr : 01
Function Code : 03
Starting Address : 0000
Quantity of Registers : 0001
CRC : 840A
```



So the right values for the Dragino RS485-BL will be:

AT+COMMAND1=01 03 00 00 00 01,1

AT+DATACUT1=7,1,4+5

## The things Stack

Let's setup the Dragino on TTS

DevEui= A8 40 41 95 C1 82 C9 43


AT+DEUI=?

AT+APPKEY=?

AT+APPEUI=?

## 1. Select the end device

Brand\*  Model\*  Hardware Ver.\*  Firmware Ver.\*  Profile (Region)\*



**RS485-BL**  
MAC V1.0.3, PHY V1.0.3 REV A, Over the air activation (OTAA), Class A  
LoRaWAN RS485/UART Converter -- Waterproof Battery Powered  
[Product website](#)

## 2. Enter registration data

Frequency plan ⓘ\*

AppEUI ⓘ\*

DevEUI ⓘ\*


AppKey ⓘ\*

Let's change the period of messages

```
AT+TDC=?  
600000  
OK
```

To 10 seconds

```
AT+TDC=10000  
OK
```



**rs485-bl**  
ID: rs485-bl

• Last seen 9 seconds ago ↑ 3 ↓ 1

Overview Live data Messaging Location Payload formatters Claiming General settings

| Time       | Type                        | Data preview   |
|------------|-----------------------------|--|
| ↑ 20:51:13 | Forward uplink data message | MAC payload: <input type="text" value="0D 48 01"/> FPort: 2 SNR: 8.75 RSSI: -29 Bandwidth: 125000  |
| ↑ 20:51:03 | Forward uplink data message | MAC payload: <input type="text" value="0D 4F 01"/> FPort: 2 SNR: 11.25 RSSI: -29 Bandwidth: 125000 |
| ↑ 20:50:53 | Forward uplink data message | MAC payload: <input type="text" value="0D 4A 01"/> FPort: 2 SNR: 11.75 RSSI: -30 Bandwidth: 125000 |
| ↑ 20:50:45 | Forward uplink data message | MAC payload: <input type="text" value="0D 54 01"/> FPort: 2 SNR: 8.25 RSSI: -30 Bandwidth: 125000  |

```
AT+BAUDR=?
```

```
9600
```

```
OK
```

```
AT+PARITY=?
```

```
0
```

```
OK
```

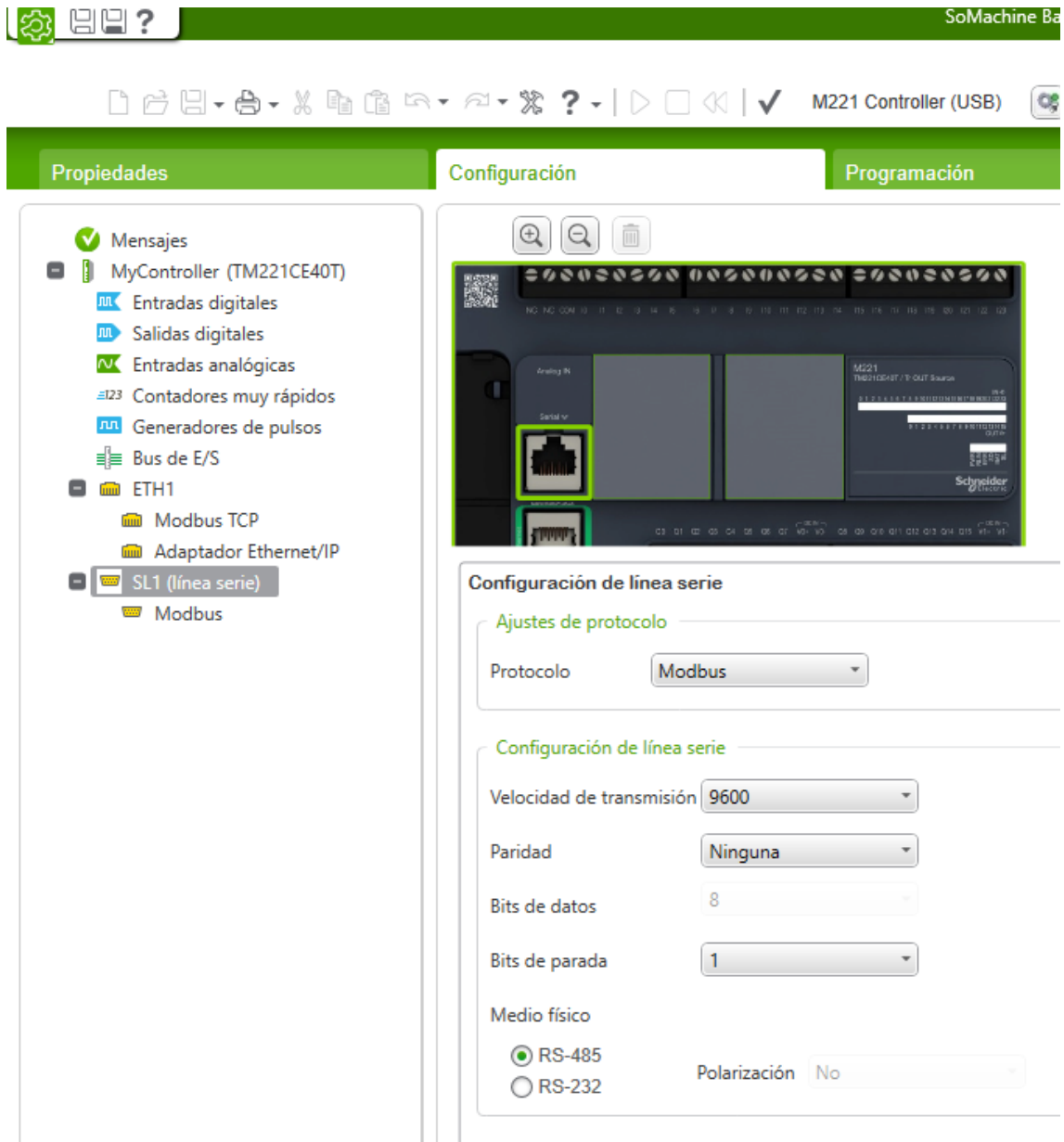
```
AT+STOPBIT=?
```

```
0
```

```
OK
```

Let's change the PLC serial speed to 9600 so it is the default value from Dragino





You can find the PLC code here

<https://github.com/xavierflorensa/Schneider-M221-as-LoRaWAN-node>

We test now with Dragino RS485 connected to the PLC

It Works

```

CMD1 = 01 03 00 00 00 01 84 0a
RETURN1 = 01 03 02 00 0a 38 43
Payload = 01 00 0a

```

Let's change the value to 6

THE THINGS NETWORK THE THINGS STACK Community Edition Overview Applications Gateways Organizations

rs485-bl-2 Applications > rs485-bl-2 > End devices > rak485-bl > Live data

Overview End devices Live data Payload formatters Integrations Collaborators API keys General settings

**rak485-bl**  
ID: rak485-bl  
Last seen 14 seconds ago ↑ 94 ↓ 8

Overview Live data Messaging Location Payload f

| Time       | Type                        | Data preview                         |
|------------|-----------------------------|--------------------------------------|
| ↑ 17:01:55 | Forward uplink data message | MAC payload: 00 40 01 00 06 FPort: 2 |
| ↑ 17:01:43 | Forward uplink data message | MAC payload: 00 49 01 00 06 FPort: 2 |
| ↑ 17:01:03 | Forward uplink data message | MAC payload: 00 46 01 00 06 FPort: 2 |
| ↑ 16:59:53 | Forward uplink data message | MAC payload: 00 49 01 00 06 FPort: 2 |
| ↑ 16:59:43 | Forward uplink data message | MAC payload: 00 49 01 00 06 FPort: 2 |
| ↑ 16:59:33 | Forward uplink data message | MAC payload: 00 49 01 00 06 FPort: 2 |

Let's change payload decoder

Applications > rs485-bl-2 > End devices > rak485-bl > Payload formatters

**rak485-bl**  
ID: rak485-bl  
Last seen 4 seconds ago ↑ 58 ↓ 4

Overview Live data Messaging Location Payload formatters

Uplink Downlink

### Setup

Formatter type\*  
Javascript

Formatter parameter\*

```

1 function Decoder(bytes, port) {
2   // Decode an uplink message from a buffer
3   // (array) of bytes to an object of fields.
4   var decoded = {};
5
6   if (port === 2) decoded.plc_mw0 = bytes[3]*256+bytes[4];
7
8
9   return decoded;
10 }

```

```
function Decoder(bytes, port) {
  // Decode an uplink message from a buffer
  // (array) of bytes to an object of fields.
  var decoded = {};

  if (port === 2) decoded.plc_mw0 = bytes[3]*256+bytes[4];

  return decoded;
}
```

The screenshot displays the 'Live data' view for the device 'rak485-bl'. The table below shows the following data:

| Time       | Type                        | Data preview                                    |
|------------|-----------------------------|---|
| ↑ 17:03:53 | Forward uplink data message | Payload: { plc_mw0: 6 } 00 46 01 00 06 FPort: 2 |
| ↑ 17:03:43 | Forward uplink data message | Payload: { plc_mw0: 6 } 00 46 01 00 06 FPort: 2 |
| ↑ 17:03:33 | Forward uplink data message | Payload: { plc_mw0: 6 } 00 46 01 00 06 FPort: 2 |
| ↑ 17:03:23 | Forward uplink data message | Payload: { plc_mw0: 6 } 00 46 01 00 06 FPort: 2 |

Now you can get these data thru mqtt by Edge computing on the receiver PLC and inject per Modbus to it.

Now let's inject the sender PLC input values on the receiver PLC and Chirpstack network server

Receiver PLC is connected to a Linux machine with Gateway hardware

CHIRP STACK network server 192.168.1.105

Node-RED 192.168.1.105

Fixed Gateway IP

Let's enter on the Chirpstack server:

<http://192.168.1.105:8080/>

user: admin

password: admin

Let's create a new application

The screenshot shows the 'Create Application' form in the ChirpStack web interface. The form is titled 'Applications / Create' and contains the following fields:

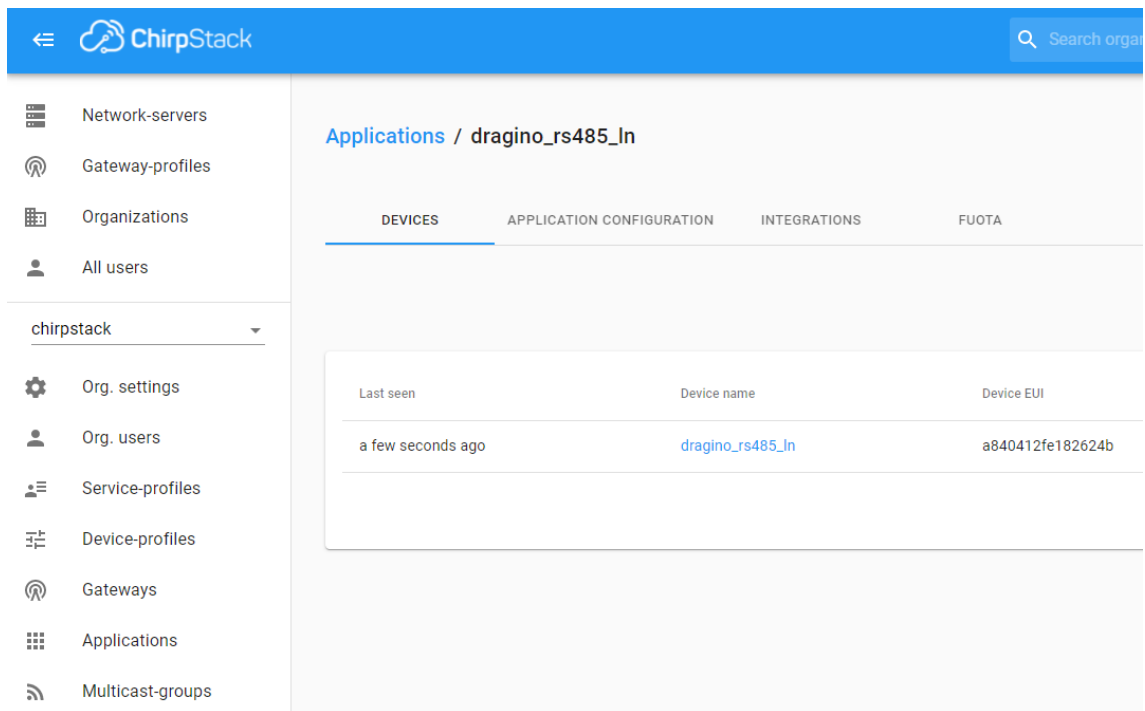
- Application name \***: Dragino-rs485-bl-2
- Application description \***: Dragino-rs485-bl-2
- Service-profile \***: service-profile

Below the service-profile field, there is a dropdown menu with 'service-profile' selected and 'None' as an option. A note below the dropdown states: 'By defining a payload codec, ChirpStack Application Server can encode and decode the binary device payload. Codec settings on the device-profile have priority over the application codec settings.'

The screenshot shows the 'Applications' list in the ChirpStack web interface. The table has the following columns: ID, Name, Service-profile, and Description. The 'Name' column contains the application names, and the 'Service-profile' column contains the service-profile names. The 'Description' column contains the application descriptions.

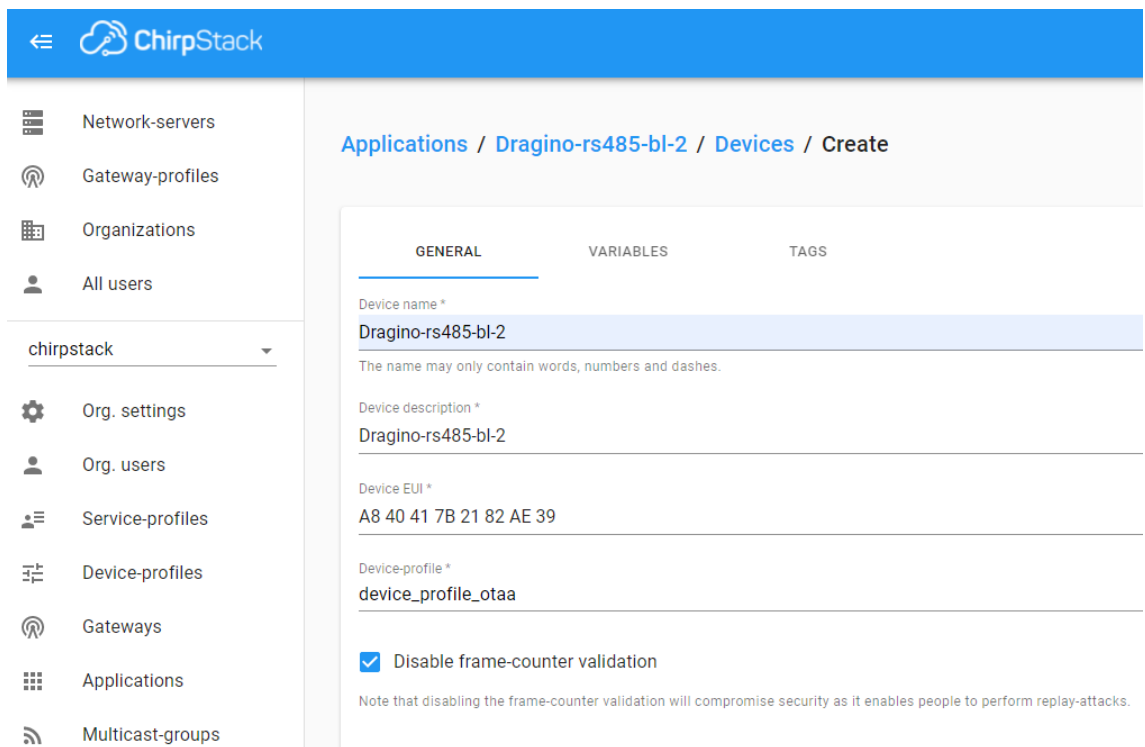
| ID | Name               | Service-profile | Description        |
|----|--------------------|-----------------|--------------------|
| 1  | app                | service-profile | app                |
| 40 | Dragino-rs485-bl-2 | service-profile | Dragino-rs485-bl-2 |
| 3  | dragino_rs485_in   | service-profile | dragino_rs485_in   |
| 2  | prova_jordi_tomas  | service-profile | Prova Jordi Tomas  |
| 39 | rak7431-rs485      | service-profile | rak7431-rs485      |

Let's add a new device.



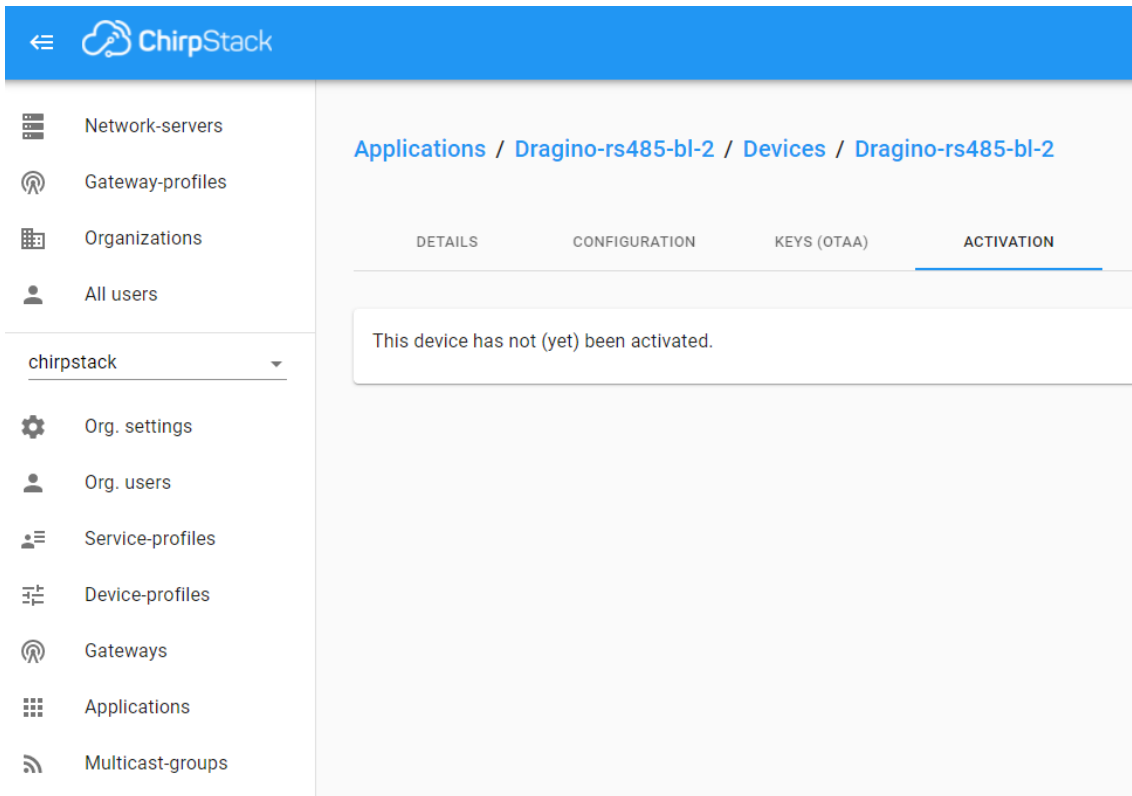
The screenshot shows the ChirpStack web interface. The top navigation bar is blue with the ChirpStack logo and a search box. The left sidebar contains a menu with items: Network-servers, Gateway-profiles, Organizations, All users, and a dropdown for 'chirpstack' which includes Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area is titled 'Applications / dragino\_rs485\_In' and has four tabs: DEVICES, APPLICATION CONFIGURATION, INTEGRATIONS, and FUOTA. The 'DEVICES' tab is active, showing a table with columns 'Last seen', 'Device name', and 'Device EUI'. A single device is listed with 'Last seen' as 'a few seconds ago', 'Device name' as 'dragino\_rs485\_In', and 'Device EUI' as 'a840412fe182624b'.

We use same device as before with same DEVEUI than in The Things stack



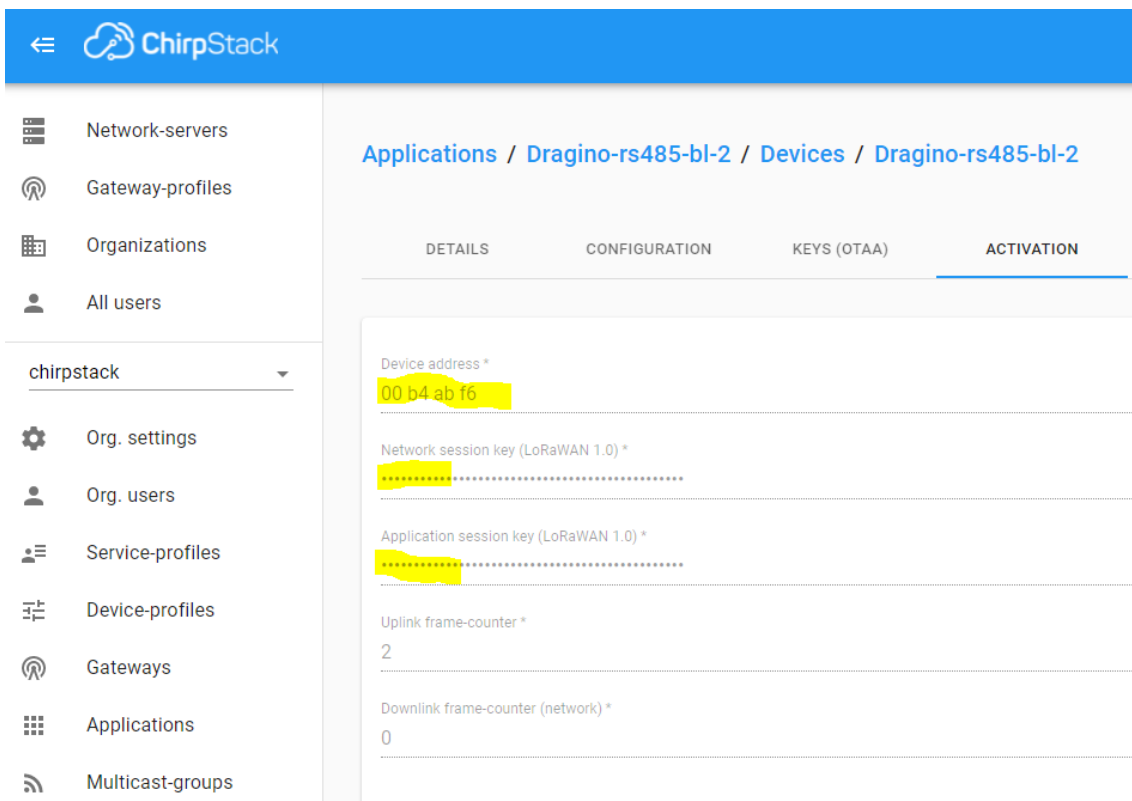
The screenshot shows the ChirpStack web interface for creating a new device. The top navigation bar is blue with the ChirpStack logo and a search box. The left sidebar contains a menu with items: Network-servers, Gateway-profiles, Organizations, All users, and a dropdown for 'chirpstack' which includes Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area is titled 'Applications / Dragino-rs485-bl-2 / Devices / Create' and has three tabs: GENERAL, VARIABLES, and TAGS. The 'GENERAL' tab is active, showing a form with the following fields: 'Device name \*' with value 'Dragino-rs485-bl-2', 'Device description \*' with value 'Dragino-rs485-bl-2', 'Device EUI \*' with value 'A8 40 41 7B 21 82 AE 39', and 'Device-profile \*' with value 'device\_profile\_otaa'. There is a checked checkbox for 'Disable frame-counter validation' and a note below it: 'Note that disabling the frame-counter validation will compromise security as it enables people to perform replay-attacks.'

And we have to enter the same AppKey than in The things stack



The screenshot shows the ChirpStack web interface. The left sidebar contains a navigation menu with items: Network-servers, Gateway-profiles, Organizations, All users, chirpstack (selected), Org. settings, Org. users, Service-profiles, Device-profiles, Gateways, Applications, and Multicast-groups. The main content area displays the breadcrumb path: Applications / Dragino-rs485-bl-2 / Devices / Dragino-rs485-bl-2. Below this is a tabbed interface with four tabs: DETAILS, CONFIGURATION, KEYS (OTAA), and ACTIVATION (which is selected). A message box in the center of the page reads: "This device has not (yet) been activated."

Then we see that the device has been activated, with a different Device address than in the Things Stack



The screenshot shows the ChirpStack web interface with the 'Activation' tab selected. The breadcrumb path is Applications / Dragino-rs485-bl-2 / Devices / Dragino-rs485-bl-2. The configuration details are as follows:

| Field                                   | Value       |
|---|-------------|
| Device address *                        | 00 b4 ab f6 |
| Network session key (LoRaWAN 1.0) *     | [Redacted]  |
| Application session key (LoRaWAN 1.0) * | [Redacted]  |
| Uplink frame-counter *                  | 2           |
| Downlink frame-counter (network) *      | 0           |

And starts transmitting

The screenshot shows the ChirpStack web interface. The left sidebar contains navigation options: Network-servers, Gateway-profiles, Organizations, All users, chirpstack (selected), Org. settings, Org. users, Service-profiles, Device-profiles, and Gateways. The main content area displays the breadcrumb path: Applications / Dragino-rs485-bl-2 / Devices / Dragino-rs485-bl-2. Below this, there are tabs for DETAILS, CONFIGURATION, KEYS (OTAA), ACTIVATION, and DEVICE DATA (selected). The DEVICE DATA tab shows a table with four rows of uplink data:

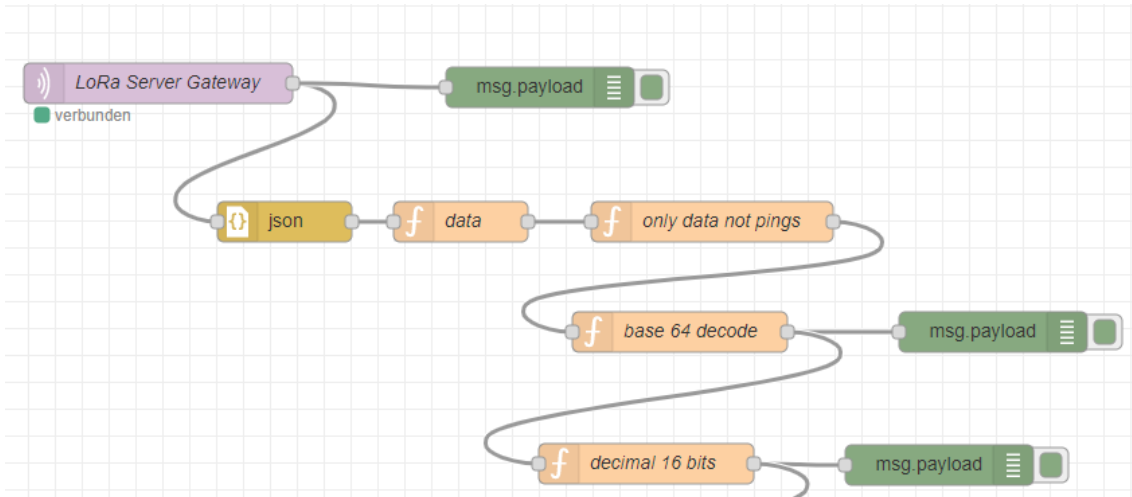
| Time       | Type   |
|------------|--------|
| 5:49:24 PM | uplink |
| 5:49:14 PM | uplink |
| 5:49:04 PM | uplink |
| 5:48:54 PM | uplink |

And the data is base 64 encoded

The screenshot shows the ChirpStack web interface with the same navigation sidebar. The breadcrumb path is Applications / Dragino-rs485-bl-2 / Devices / Dragino-rs485-bl-2. The tabs are DETAILS, CONFIGURATION, KEYS (OTAA), and ACTIVATION. The main content area displays a detailed view of an uplink event at 5:50:24 PM. The data is as follows:

```
adr: true
applicationID: "40"
applicationName: "Dragino-rs485-bl-2"
data: "DTDBAAA="
devEUI: "a840417b2182ae39"
deviceName: "Dragino-rs485-bl-2"
fCnt: 16
fPort: 2
txInfo: {} 2 keys
  dr: 0
  frequency: 868300000
```

Let's use Node-RED on same machine



```

application/40/device/a840417b2182ae39/rx: msg.payload : string[212]
{"applicationID":"40","applicationName":"Dragino-rs485-b1-2","deviceName":"Dragino-rs485-b1-2","devEUI":"a840417b2182ae39","txInfo":{"frequency":868300000,"dr":0},"adr":true,"fCnt":50,"fPort":2,"data":"DTkBAAA="}

28/6/2021 17:56:04 node: 7ca59ec2.d53c
application/40/device/a840417b2182ae39/rx: msg.payload : buffer[5]
▶ [ 13, 57, 1, 0, 0 ]

28/6/2021 17:56:04 node: 838f75f0.107028
application/40/device/a840417b2182ae39/rx: msg.payload : number
0

```

So we receive the right payload

### mqtt in Node bearbeiten

Löschen
Abbrechen
Fertig

⚙️ Properties
⚙️ 📄 🖨️

🌐 Server  ✎

📄 Topic

🔊 QoS  ▼

➡️ Output  ▼

🏷️ Name



mqtt in Node bearbeiten > **mqtt-broker Node bearbeiten**

Löschen Abbrechen Aktualisieren

**Properties** ⚙️ 📄

🔑 Name

**Verbindung** Sicherheit Nachrichten

🌐 Server  Port

Sichere Verbindung (SSL/TLS) aktivieren

🔑 Client-ID

🕒 Keepalive-Zeit (en)   Bereinigte Sitzung verwenden

Traditionelle MQTT 3.1-Unterstützung verwenden

**json Node bearbeiten**

Löschen Abbrechen Fertig

**Properties** ⚙️ 📄 🖨️

🕒 Aktion  ▼

⋮ Eigenschaft

**function Node bearbeiten**

Löschen Abbrechen Fertig

**Properties** ⚙️ 📄 🖨️

🔑 Name  📄 ▼

🔧 Funktion 🔗

```
1 var datastring = msg.payload.data
2 msg.payload = datastring
3 return msg;
```

function Node bearbeiten

Löschen Abbrechen Fertig

⚙ Properties

🔑 Name only data not pings

🔧 Funktion

```
1 if(typeof msg.payload !== 'undefined') {
2   return msg;
3 }
4 else
5 {}
```

function Node bearbeiten

Löschen Abbrechen Fertig

⚙ Properties

🔑 Name base 64 decode

🔧 Funktion

```
1 var b = new Buffer (msg.payload,'base64');
2 msg.payload = b;
3 return msg;
```

function Node bearbeiten

Löschen Abbrechen Fertig

⚙ Properties

🔑 Name decimal 16 bits

🔧 Funktion

```
1 var a = msg.payload[3];
2 var b = msg.payload[4];
3 msg.payload = a*256+b;
4 return msg;
```

## Modbus-Write Node bearbeiten

Löschen

Abbrechen

Fertig

### ⚙ Properties



Name

Name

Unit-Id

FC

FC 6: Preset Single Register ▾

Adresse

1

Server

modbus-tcp@192.168.1.58:502 ▾



Modbus-Write Node bearbeiten > **modbus-client Node bearbeiten**

Löschen Abbrechen **Aktualisieren**

**Properties**

Name

Typ

---

Host

Port

Verbindungstyp

---

Unit-Id

Timeout (ms)

Reconnect bei Timeouts

Reconnect-Timeout (ms)

Now let's activate input 1 and 2 on the sender PLC

Which is 6 coded in binary

```

28/6/2021 17:58:44 node: e31463f074198
gateway8276ff61b6de1eeventip: msg.payload: buffer[100]
> [ 18, 18, 64, 246, 171, 188, 0, 128, 66, 0 - ]
28/6/2021 17:58:44 node: e31463f074198
application040device04041762182ae39ix: msg.payload: string[12]
{"applicationID":"48","applicationName":"Dragino-rs485-b1-2","deviceName":"Dragino-rs485-b1-2","devEUI":"a84041762182ae39","txInfo":{"frequency":868500000,"dr":8,"adr":true,"fcnt":66,"fport":2,"data":"DTXBAAV-1"}}
28/6/2021 17:58:44 node: 7ca59ec2d53c
application040device04041762182ae39ix: msg.payload: buffer[5]
> [ 13, 57, 1, 0, 6 ]
28/6/2021 17:58:44 node: 8387590107925
application040device04041762182ae39ix: msg.payload: number
4

```

```

application/40/device/a840417b2182ae39/rx : msg.payload : string[212]

{"applicationID":"40","applicationName":"Dragino-rs485-b1-2","deviceName":"Dragino-rs485-b1-2","devEUI":"a840417b2182ae39","txInfo":{"frequency":868100000,"dr":0},"adr":true,"fCnt":68,"fPort":2,"data":"DTkBAAY="}

28/6/2021 17:59:04 node: 7ca59ec2.d53c
application/40/device/a840417b2182ae39/rx : msg.payload : buffer[5]
▶ [ 13, 57, 1, 0, 6 ]

28/6/2021 17:59:04 node: 838f75f0.107028
application/40/device/a840417b2182ae39/rx : msg.payload : number
6

```

And on the receiver PLC:

We have the number 6 on the register %MW1 and outputs 1 and 2 on ON state

The screenshot shows the SoMachine Logic Builder interface. The left pane displays a project tree for 'Edificio central Gateway PLC Maestro v0'. The main area shows the configuration for 'Ethernet\_1' with the following parameters:

- Nombre de interfaz: EthernetPort0
- Nombre de red: my\_Device
- Dirección IP fija: 192 . 168 . 1 . 58
- Máscara de subred: 255 . 255 . 255 . 0
- Dirección de pasarela: 0 . 0 . 0 . 0
- Protocolo Ethernet: Ethernet 2
- Velocidad de transferencia: Auto

The 'Supervisar 1' table at the bottom shows the following data:

| Expresión                           | Tipo de datos | Valor | Valor preparado | Dirección | Comentario |
|-------------------------------------|---------------|-------|-----------------|-----------|------------|
| MyController.Application.i_var0     | UINT          | 5     |                 |           |            |
| MyController.Application.i_var1     | INT           | 6     |                 |           |            |
| MyController.Application.xq_salida1 | BOOL          | TRUE  |                 | %QX0.1    |            |
| MyController.Application.xq_salida2 | BOOL          | TRUE  |                 | %QX0.2    |            |

Edificio central Gateway PLC Maestro v0.project\* - SoMachine Logic Builder - V4.3

Archivo Edición Ver Proyecto IL/FBD/LD Compilar En línea Depuración Herramientas Ventana Ayuda

Aplicaciones MyController Application.MyPOU

Edificio central Gateway PLC Maestro v0

- Application (HMISCUxA5 : HMISCUxA5)
  - Configuración de tareas
  - MAST
  - GVL
  - Application (MyController:TM241CEC24R)
    - Configuración de tareas
    - MAST
    - MyPOU
    - GVL
    - MyPOU (PRG)
    - Global
    - Vijeo Designer Project Container

2

3

ADD

EN ENO

50

i\_var0 2

i\_sinusoidal -48

WORD\_AS\_BIT\_0

EN ENO

W

i\_var1 6

B00 xq\_salida0 FALSE

B01 xq\_salida1 TRUE

B02 xq\_salida2 TRUE

B03 xq\_salida3 FALSE

B04 xq\_salida4 FALSE

B05 xq\_salida5 FALSE

B06 xq\_salida6 FALSE

B07 xq\_salida7 FALSE

B08 xq\_salida8 FALSE

B09 xq\_salida9 FALSE

B10 FALSE

B11 FALSE

B12 FALSE

B13 FALSE

Supervisor 1

| Expresión                           | Tipo de datos | Valor | Valor preparado | Dirección | Comentario |
|-------------------------------------|---------------|-------|-----------------|-----------|------------|
| MyController.Application.i_var0     | UINT          | 2     |                 |           |            |
| MyController.Application.i_var1     | INT           | 6     |                 |           |            |
| MyController.Application.xq_salida1 | BOOL          | TRUE  |                 | %QX0.1    |            |
| MyController.Application.xq_salida2 | BOOL          | TRUE  |                 | %QX0.2    |            |

Mensajes - total 0 error(es), 3 advertencia(s), 11 mensaje(s)

Último Build: 0 3 Precompilar: EN EJECUCIÓN Programa cargado

And that's all

You can find the Node-RED code here

<https://github.com/xavierflorencia/Schneider-M221-as-LoRaWAN-node>

