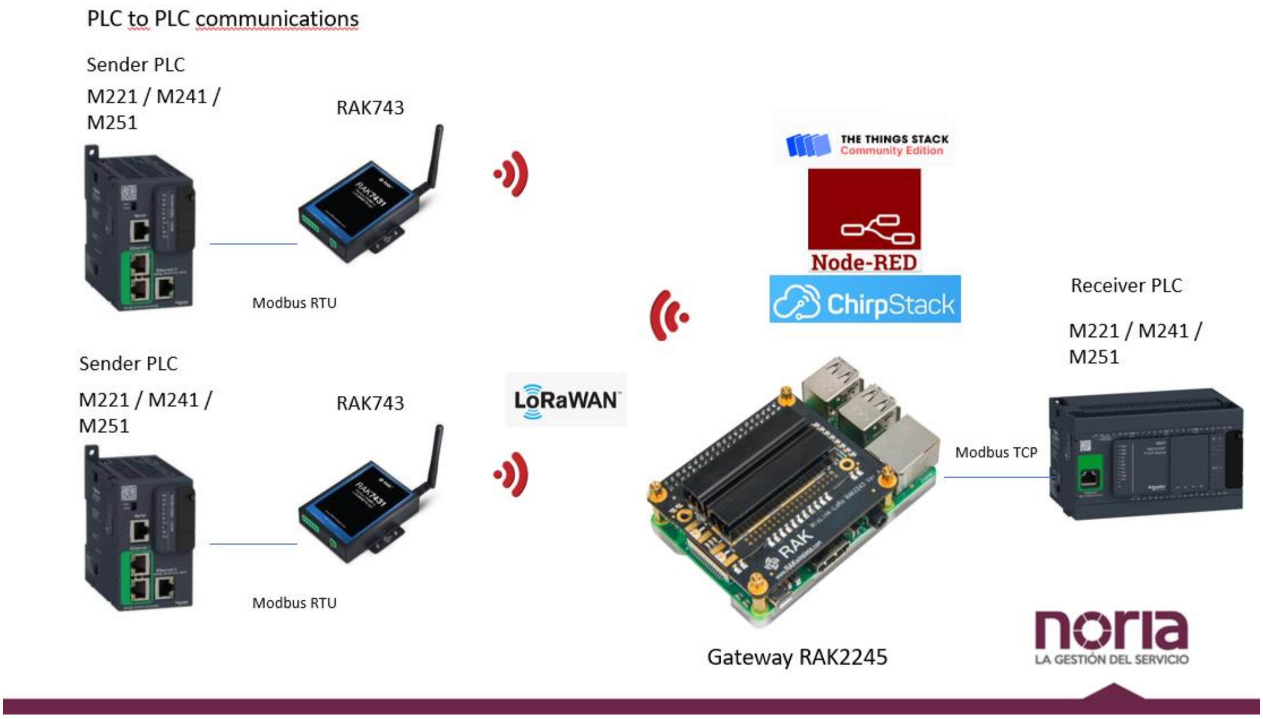
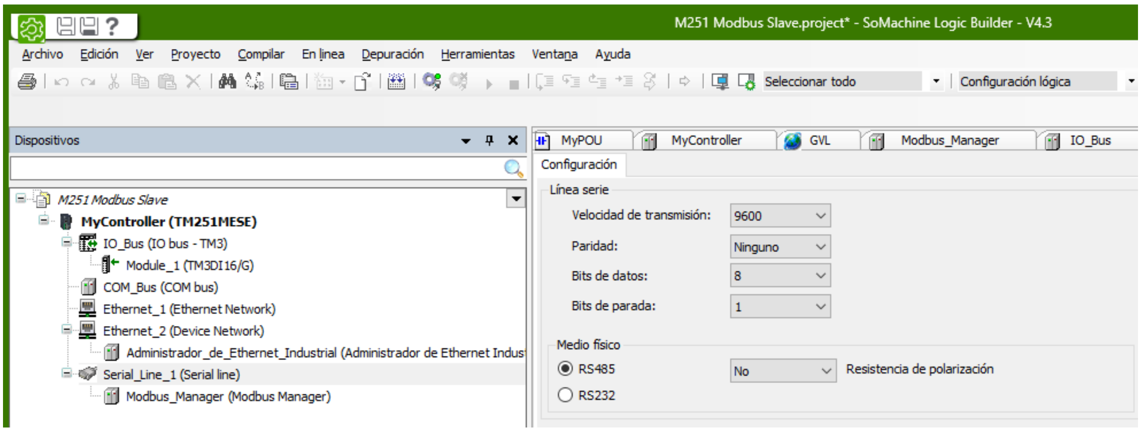
SCHNEIDER PLC to PLC communications with RAK7431 and LoRaWAN

System Arquitecture



Sender PLC configuration

We configure the sender PLC as a Modbus RTU slave device with address 1 and register %MW0 to hold the status of the digital inputs of sender PLC.


M251 Modbus Slave.project* - SoMachine Logic Builder - V4.3
Archivo Edición Ver Proyecto Compilar En línea Depuración Herramientas Ventana Ayuda
Seleccionar todo
Configuración lógica
Dispositivos
M251 Modbus Slave
MyController (TM251MESE)
IO_Bus (IO bus - TM3)
Module_1 (TM3DI16/G)
COM_Bus (COM bus)
Ethernet_1 (Ethernet Network)
Ethernet_2 (Device Network)
Administrador_de_Ethernet_Industrial (Administrador de Ethernet Indus
Serial_Line_1 (Serial line)
Modbus_Manager (Modbus Manager)
MyPOU
MyController
GVL
Modbus_Manager
IO_Bus
Configuración

Línea serie

Velocidad de transmisión: 9600

Paridad: Ninguno

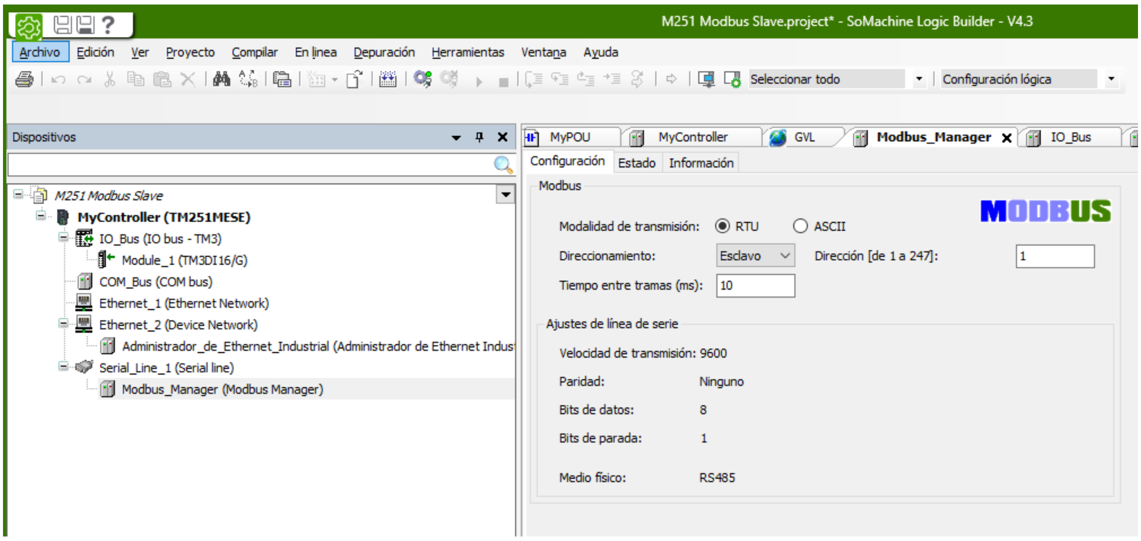
Bits de datos: 8

Bits de parada: 1

Medio físico

● RS485 No Resistencia de polarización

○ RS232


M251 Modbus Slave.project* - SoMachine Logic Builder - V4.3
Archivo Edición Ver Proyecto Compilar En línea Depuración Herramientas Ventana Ayuda
Seleccionar todo
Configuración lógica
Dispositivos
M251 Modbus Slave
MyController (TM251MESE)
IO_Bus (IO bus - TM3)
Module_1 (TM3DI16/G)
COM_Bus (COM bus)
Ethernet_1 (Ethernet Network)
Ethernet_2 (Device Network)
Administrador_de_Ethernet_Industrial (Administrador de Ethernet Indus
Serial_Line_1 (Serial line)
Modbus_Manager (Modbus Manager)
MyPOU
MyController
GVL
Modbus_Manager
IO_Bus
Configuración
Estado
Información
MODBUS

Modbus

Modalidad de transmisión: ● RTU ○ ASCII

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

Tiempo entre tramas (ms): 10

Ajustes de línea de serie

Velocidad de transmisión: 9600

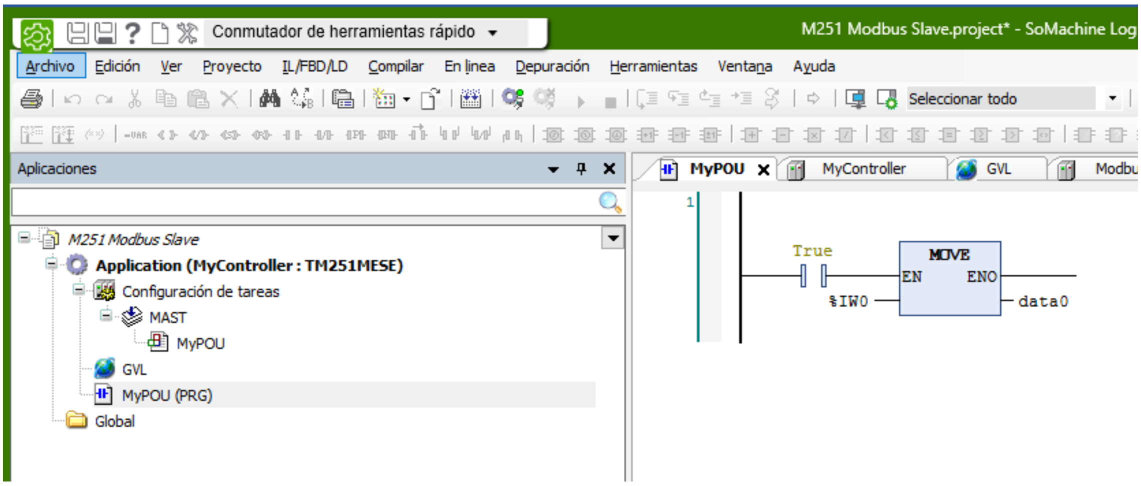
Paridad: Ninguno

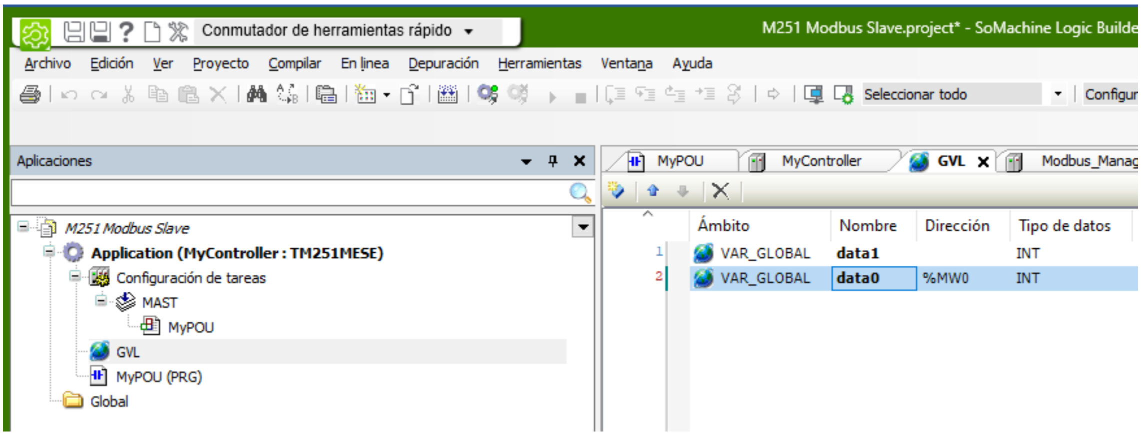
Bits de datos: 8

Bits de parada: 1

Medio físico: RS485

We update the digital input state on every scan


Conmutador de herramientas rápido
M251 Modbus Slave.project* - SoMachine Log
Archivo Edición Ver Proyecto IL/FBD/LD Compilar En línea Depuración Herramientas Ventana Ayuda
Seleccionar todo
Aplicaciones
M251 Modbus Slave
Application (MyController : TM251MESE)
Configuración de tareas
MAST
MyPOU
GVL
MyPOU (PRG)
Global
MyPOU
MyController
GVL
Modbu
True
MOVE
EN
ENO
%IW0
data0


M251 Modbus Slave.project* - SoMachine Logic Builde
Conmutador de herramientas rápido
Archivo Edición Ver Proyecto Compilar En línea Depuración Herramientas Ventana Ayuda
Seleccionar todo
Configur
Aplicaciones
MyPOU
MyController
GVL
Modbus_Manag

M251 Modbus Slave
- Application (MyController : TM251MESE)
  - Configuración de tareas
    - MAST
      - MyPOU
  - GVL
  - MyPOU (PRG)
- Global

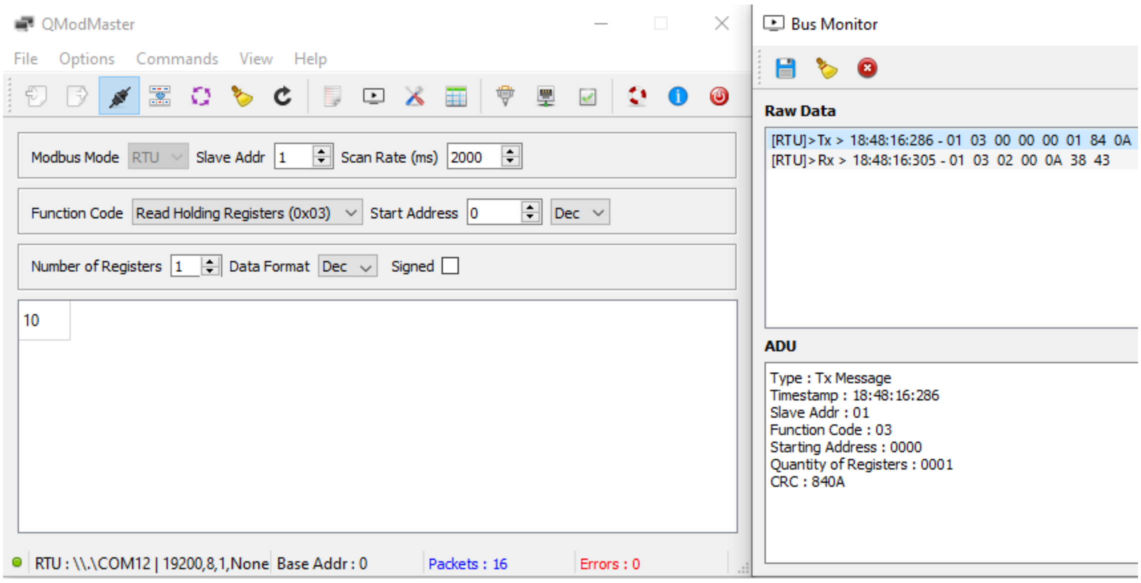
	Ámbito	Nombre	Dirección	Tipo de datos
1	VAR_GLOBAL	data1		INT
2	VAR_GLOBAL	data0	%MW0	INT

So we have such data available thru Modbus RTU

Modbus test

Let's see the Modbus response from PLC with the PC and qModMaster tool

Let's read from Modbus slave 1 (PLC) and address %MW0 (0) wich value is 10 for instance.



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

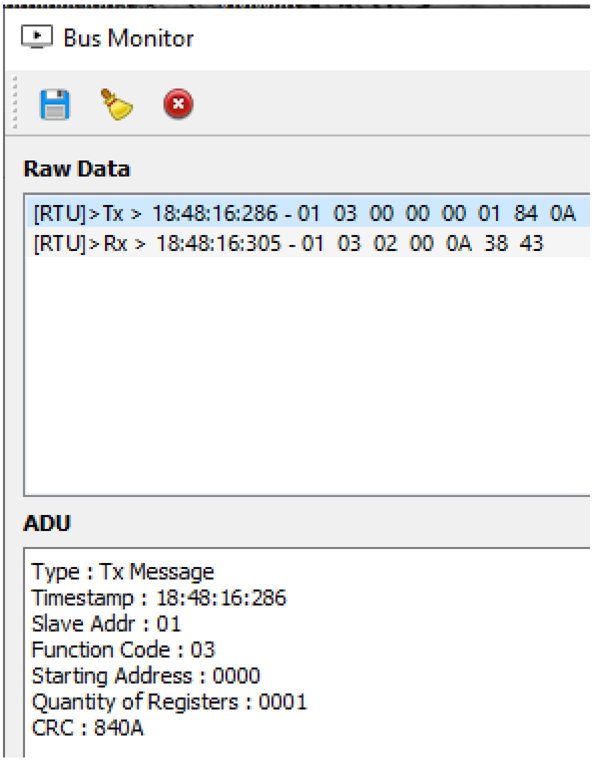
RTU : \\.\COM12 | 19200,8,1,None Base Addr : 0 Packets : 16 Errors : 0

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
```

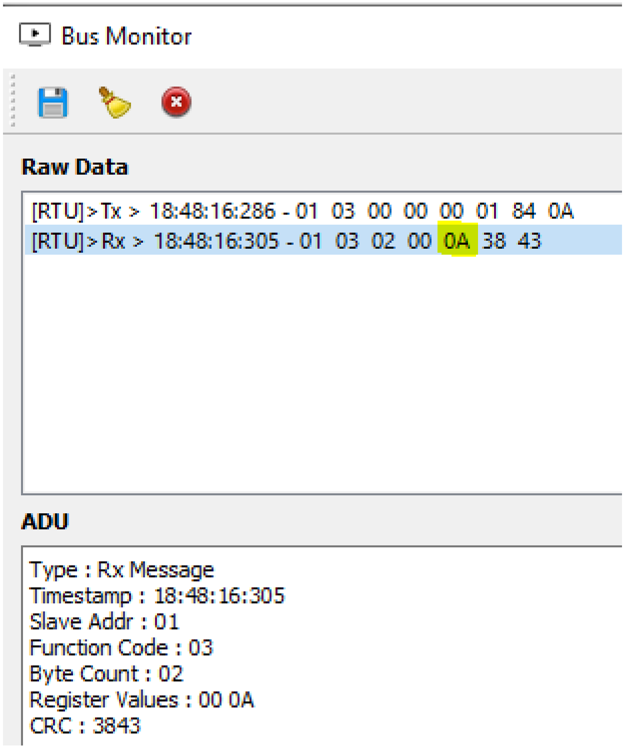


Raw Data

```
[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
```



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
```

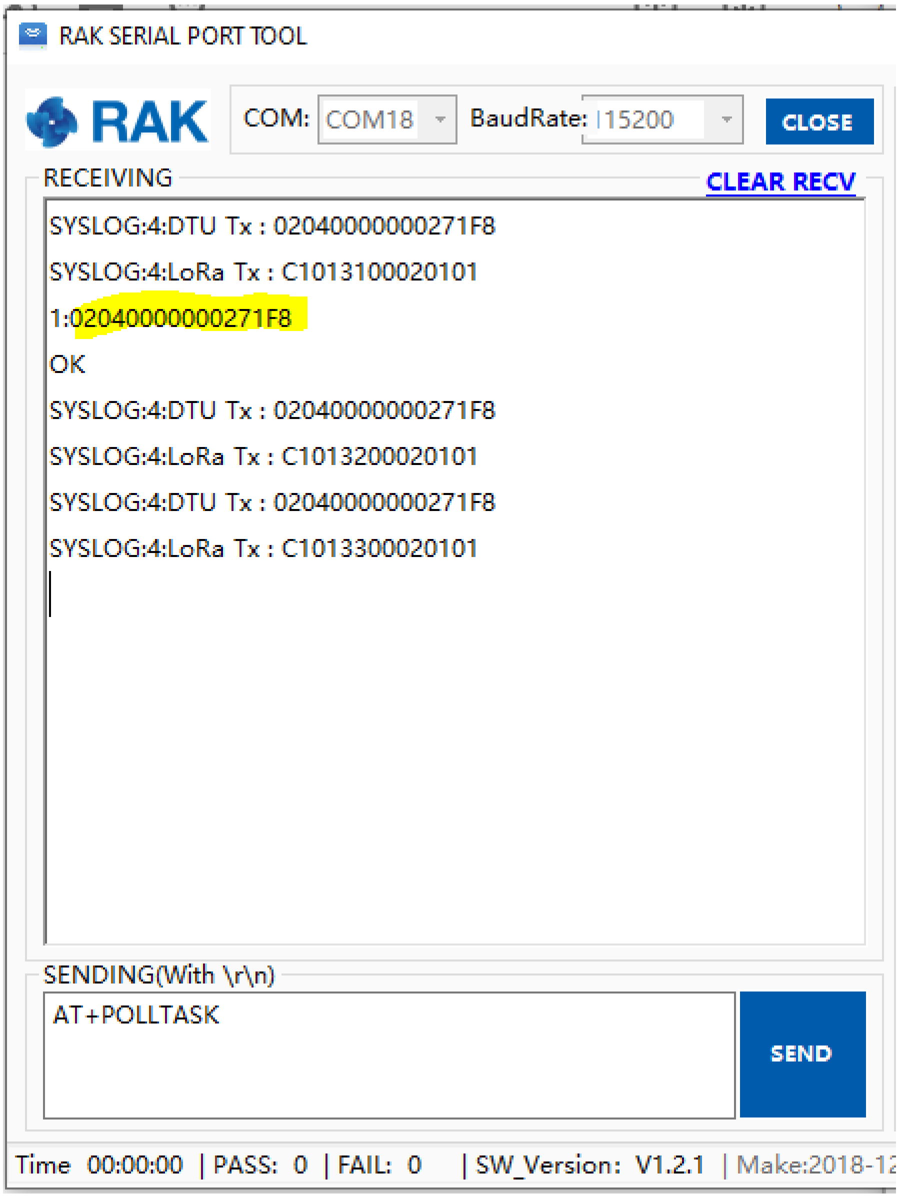
ADU

```
Type : Rx Message
Timestamp : 18:48:16:305
Slave Addr : 01
Function Code : 03
Byte Count : 02
Register Values : 00 0A
CRC : 3843
```

RAK7431 configuration

Let’s monitor the RAK7431 with RAK serial tool, wether there is a poll task scheduled

Yes, there is one



```
SYSLOG:4:DTU Tx : 02040000000271F8
SYSLOG:4:LoRa Tx : C1013100020101
1:02040000000271F8
OK
SYSLOG:4:DTU Tx : 02040000000271F8
SYSLOG:4:LoRa Tx : C1013200020101
SYSLOG:4:DTU Tx : 02040000000271F8
SYSLOG:4:LoRa Tx : C1013300020101
```

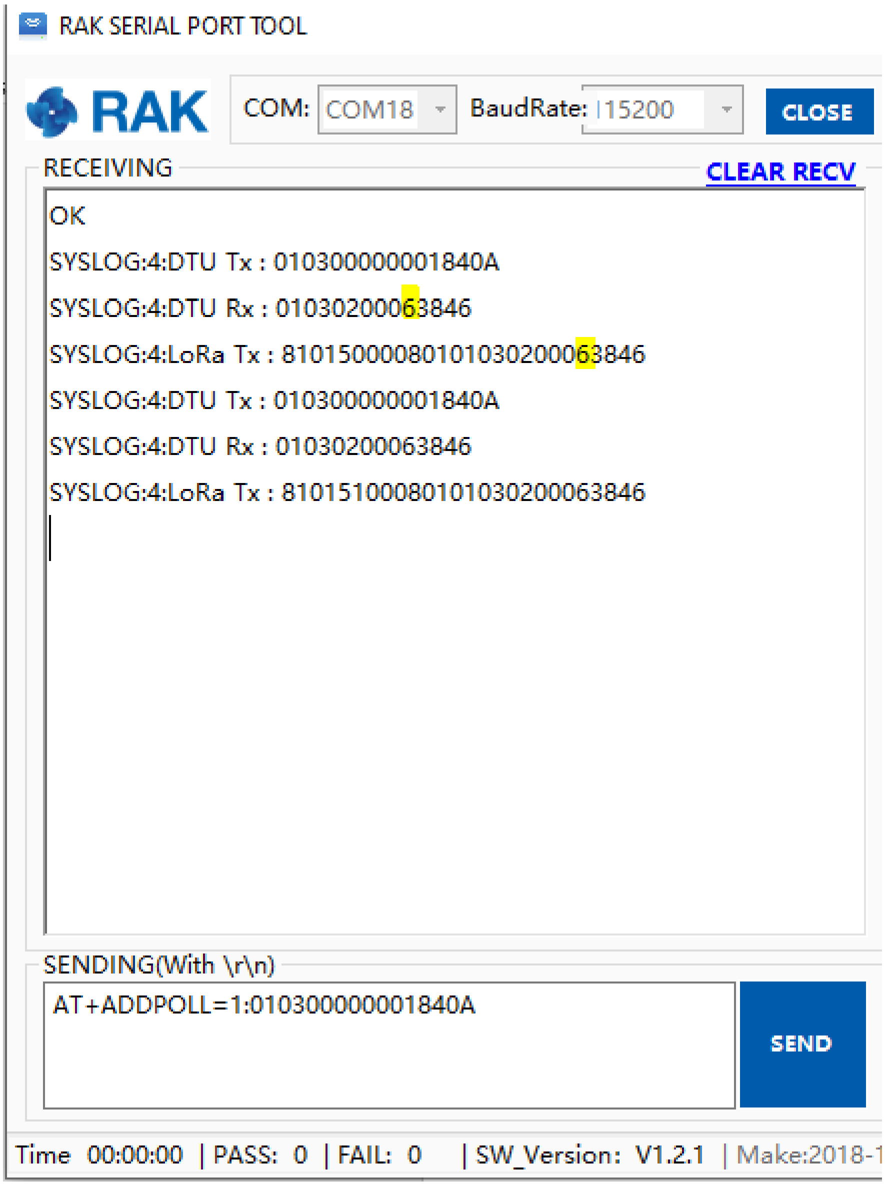
```
AT+POLLTASK
```

Time 00:00:00 | PASS: 0 | FAIL: 0 | SW_Version: V1.2.1 | Make:2018-12

We have to change this since we want to point to Modbus address 01 and message type 03 as we have seen on Modbus Test

Let's remove it
AT+RMPOLL=1

Now we add a new POLL task to measure Voltage
And this will be the right poll task according to the previous chapter
`AT+ADDPOLL=1:010300000001840A`



```
OK
SYSLOG:4:DTU Tx : 010300000001840A
SYSLOG:4:DTU Rx : 01030200063846
SYSLOG:4:LoRa Tx : 81015000080101030200063846
SYSLOG:4:DTU Tx : 010300000001840A
SYSLOG:4:DTU Rx : 01030200063846
SYSLOG:4:LoRa Tx : 81015100080101030200063846
```

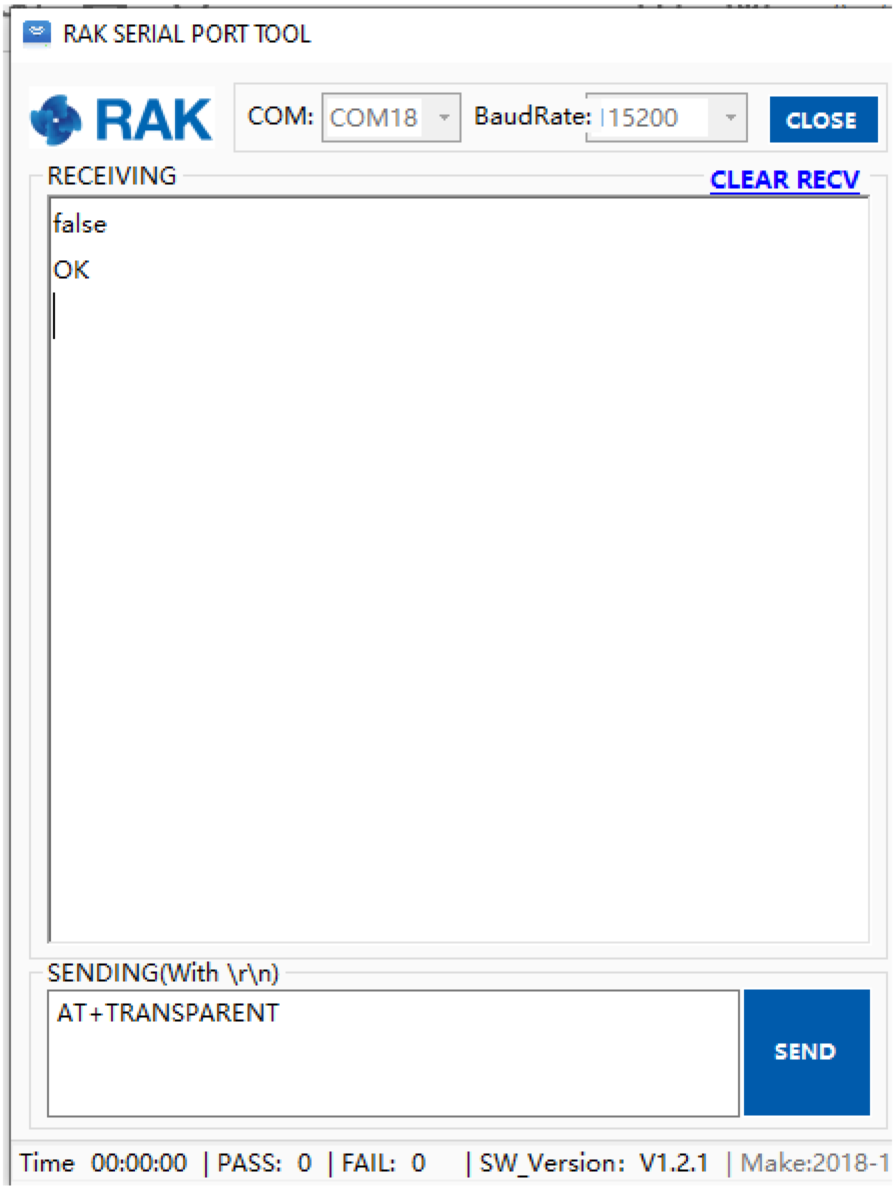
```
AT+ADDPOLL=1:010300000001840A
```

Time 00:00:00 | PASS: 0 | FAIL: 0 | SW_Version: V1.2.1 | Make:2018-1

And we receive the response from PLC register %MW0 that in this case is 6

Bit 1 and Bit 2 to ON state so this is 6

But we receive a long payload, Since we are in non transparent mode.



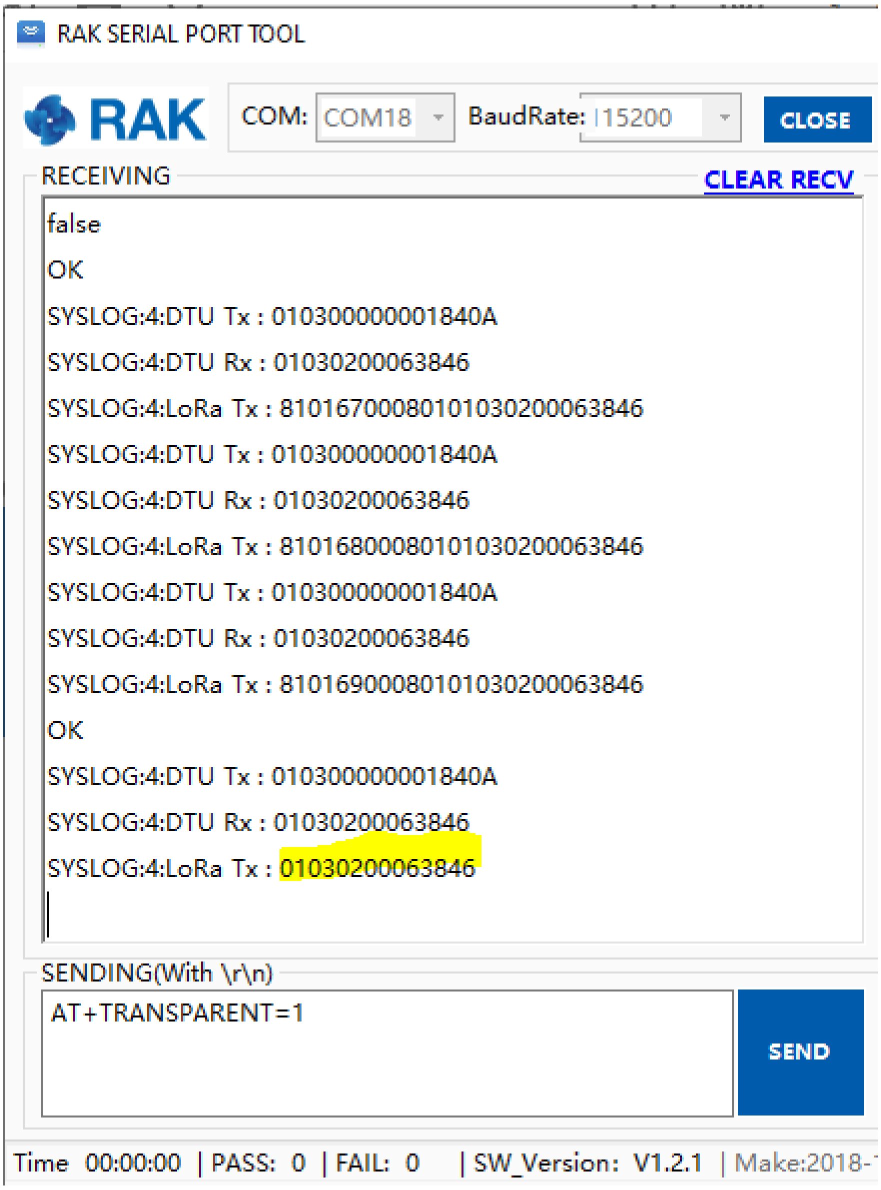
COM: COM18 BaudRate: 115200

```
false
OK
```

```
AT+TRANSPARENT
```

Time 00:00:00 | PASS: 0 | FAIL: 0 | SW_Version: V1.2.1 | Make:2018-1

Let’s change to transparent mode to reduce payload



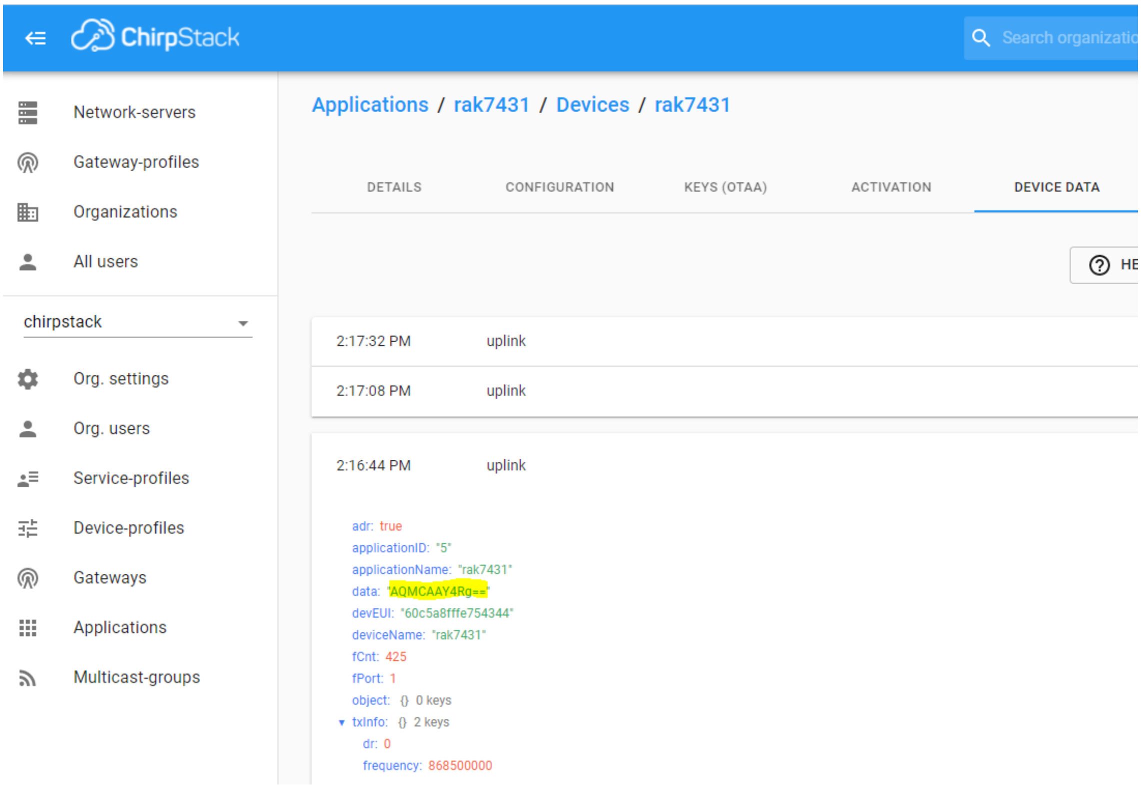
```
false
OK
SYSLOG:4:DTU Tx : 010300000001840A
SYSLOG:4:DTU Rx : 01030200063846
SYSLOG:4:LoRa Tx : 81016700080101030200063846
SYSLOG:4:DTU Tx : 010300000001840A
SYSLOG:4:DTU Rx : 01030200063846
SYSLOG:4:LoRa Tx : 81016800080101030200063846
SYSLOG:4:DTU Tx : 010300000001840A
SYSLOG:4:DTU Rx : 01030200063846
SYSLOG:4:LoRa Tx : 81016900080101030200063846
OK
SYSLOG:4:DTU Tx : 010300000001840A
SYSLOG:4:DTU Rx : 01030200063846
SYSLOG:4:LoRa Tx : 01030200063846
```

```
AT+TRANSPARENT=1
```

Time 00:00:00 | PASS: 0 | FAIL: 0 | SW_Version: V1.2.1 | Make:2018-

Chirpstack console

This is the view on Chirpstack console.



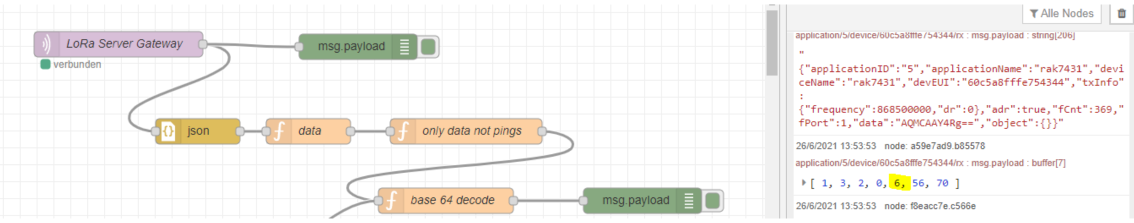
```
2:17:32 PM     uplink
2:17:08 PM     uplink
2:16:44 PM     uplink

adr: true
applicationID: "5"
applicationName: "rak7431"
data: "AQMCAAY4Rg=="
devEUI: "60c5a8fffe754344"
deviceName: "rak7431"
fCnt: 425
fPort: 1
object: {} 0 keys
▾ txInfo: {} 2 keys
   dr: 0
   frequency: 868500000
```

Now we see that we have the value of %MW0 on bytes 3 and 4

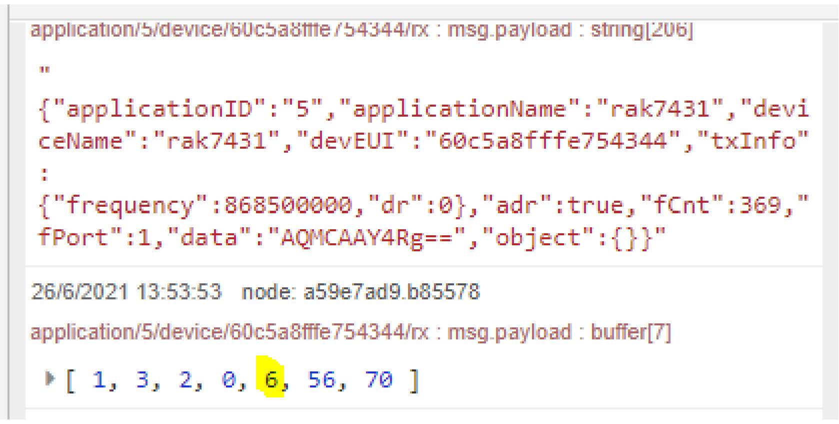
Node-RED

Node-RED is running on the same RAK 2245 gateway Raspberry Pi. Just install Node-RED after burning the Gateway operating system SD with the image from RAK Wireless web page.

Let’s decode this with Node-RED



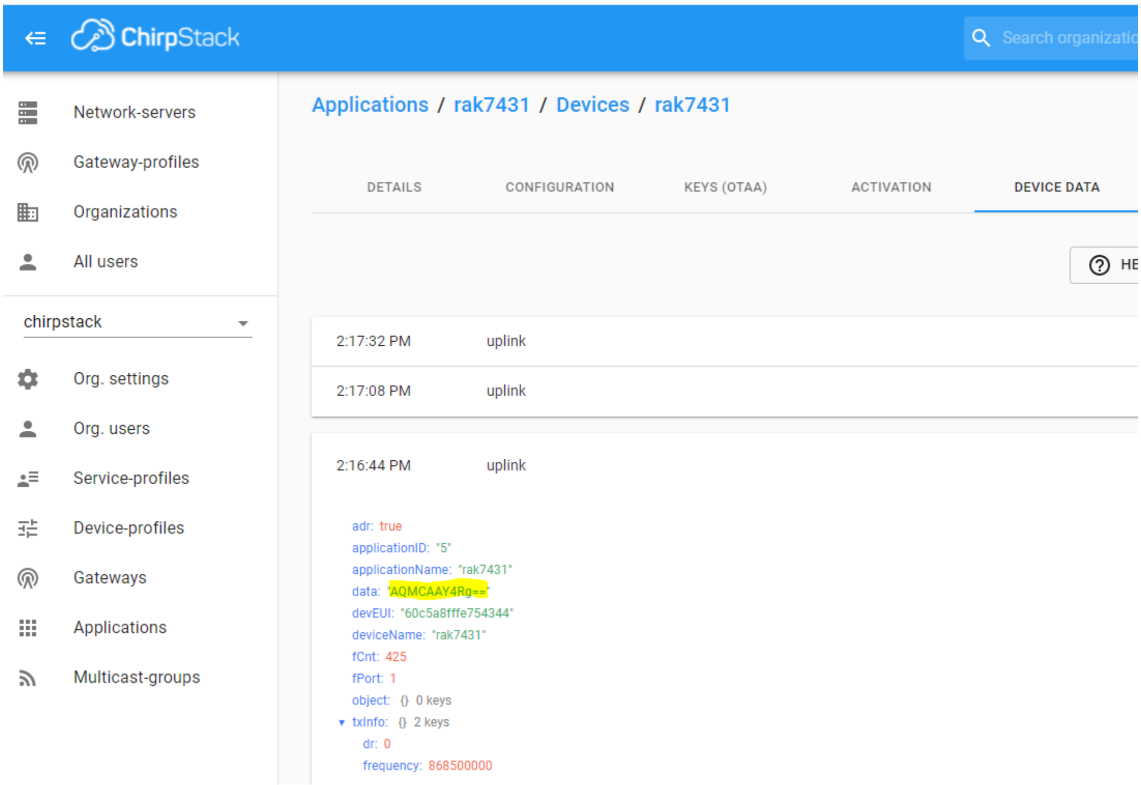
```
application/5/device/60c5a8fffe754344/rx : msg.payload : string[206]
"
{"applicationID":"5","applicationName":"rak7431","deviceName":"rak7431","devEUI":"60c5a8fffe754344","txInfo":{"frequency":868500000,"dr":0},"adr":true,"fCnt":369,"fPort":1,"data":"AQMCAAY4Rg==","object":{}}"
26/6/2021 13:53:53   node: a59e7ad9.b85578
application/5/device/60c5a8fffe754344/rx : msg.payload : buffer[7]
▸[ 1, 3, 2, 0, 6, 56, 70 ]
26/6/2021 13:53:53   node: f8eacc7e.c566e
```



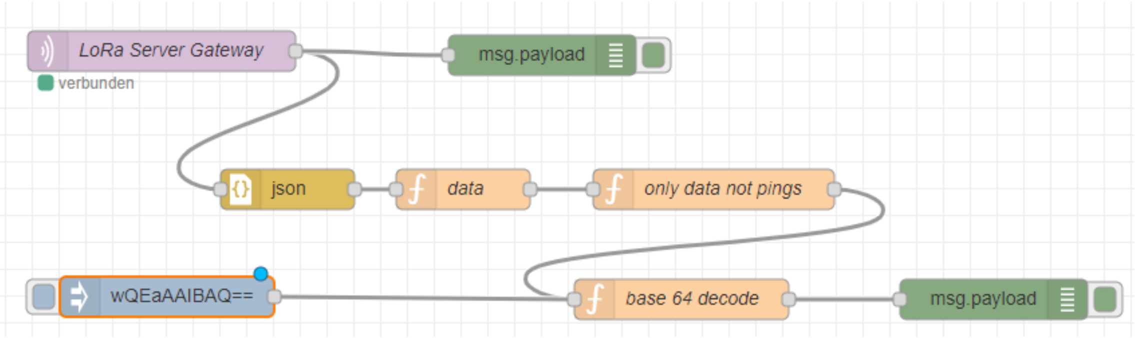
```
application/5/device/60c5a8fffe754344/rx : msg.payload : string[206]
"
{"applicationID":"5","applicationName":"rak7431","deviceName":"rak7431","devEUI":"60c5a8fffe754344","txInfo":{"frequency":868500000,"dr":0},"adr":true,"fCnt":369,"fPort":1,"data":"AQMCAAY4Rg==","object":{}}"
26/6/2021 13:53:53   node: a59e7ad9.b85578
application/5/device/60c5a8fffe754344/rx : msg.payload : buffer[7]
▸[ 1, 3, 2, 0, 6, 56, 70 ]
```

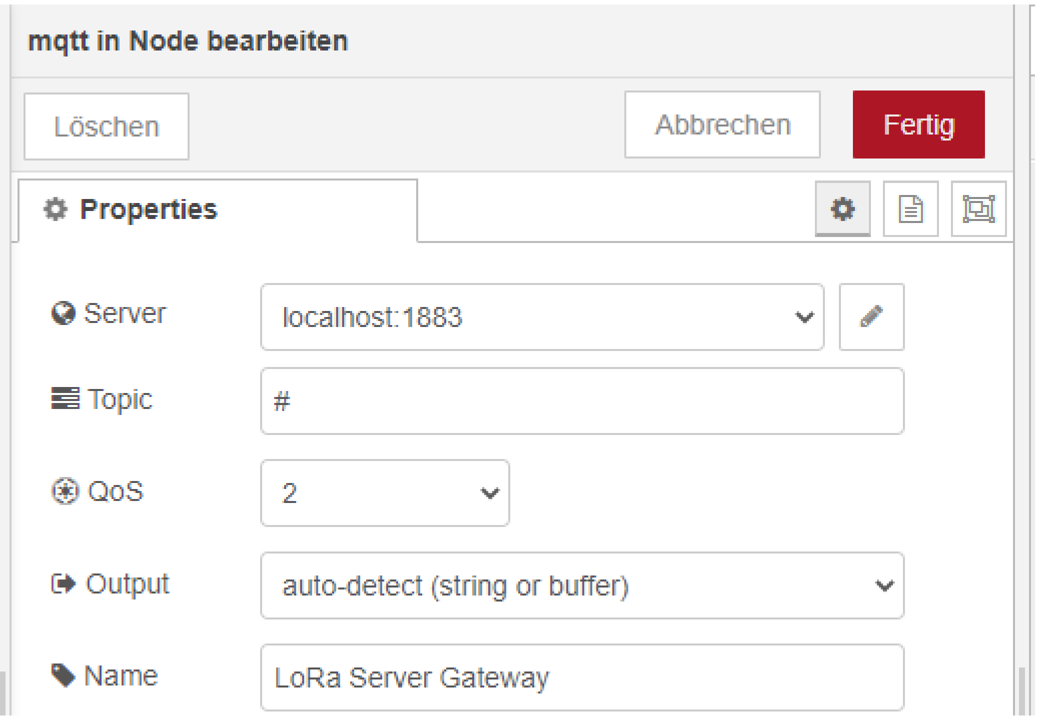
Whith this Flow

Take into account that payload is base 64 encoded



```
2:17:32 PM          uplink
2:17:08 PM          uplink
2:16:44 PM          uplink

   adr: true
   applicationID: "5"
   applicationName: "rak7431"
   data: "AQMCAAY4Rg=="
   devEUI: "60c5a8fffe754344"
   deviceName: "rak7431"
   fCnt: 425
   fPort: 1
   object: {} 0 keys
 ▾ txInfo: {} 2 keys
     dr: 0
     frequency: 868500000
```





mqtt in Node bearbeiten

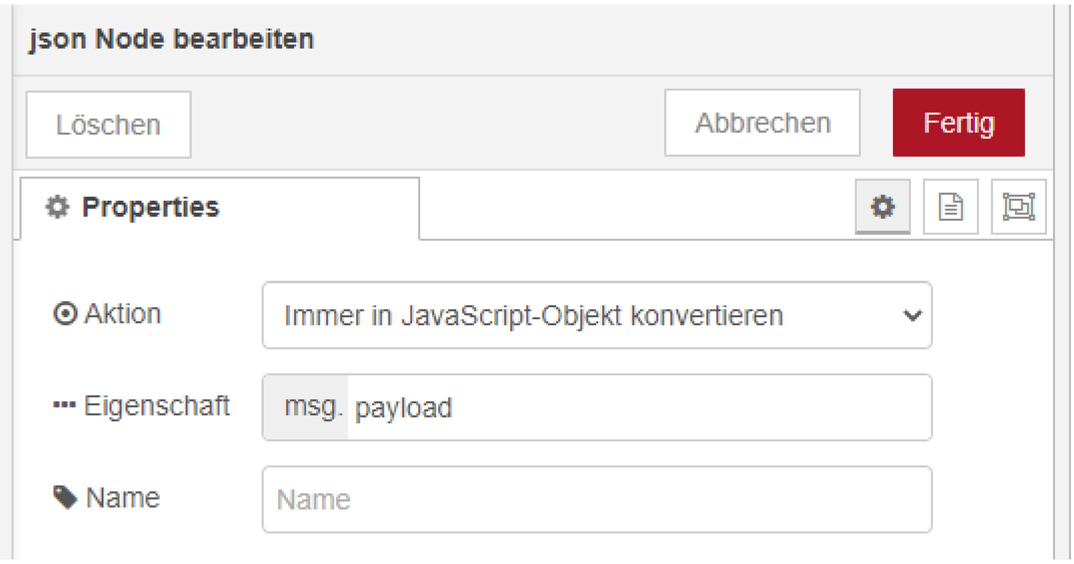
Server: localhost:1883

Topic: #

QoS: 2

Output: auto-detect (string or buffer)

Name: LoRa Server Gateway

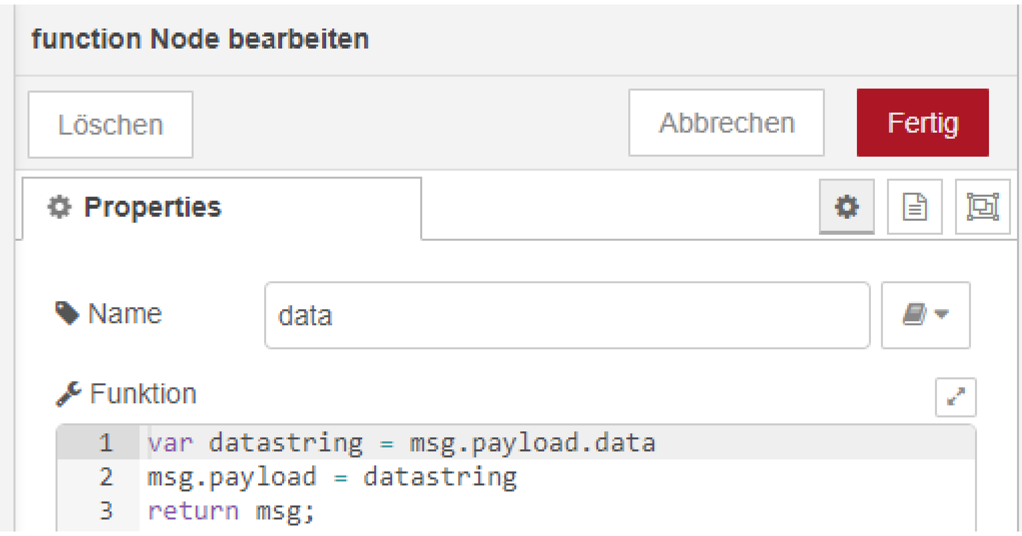


json Node bearbeiten

Aktion: Immer in JavaScript-Objekt konvertieren

Eigenschaft: msg. payload

Name: Name

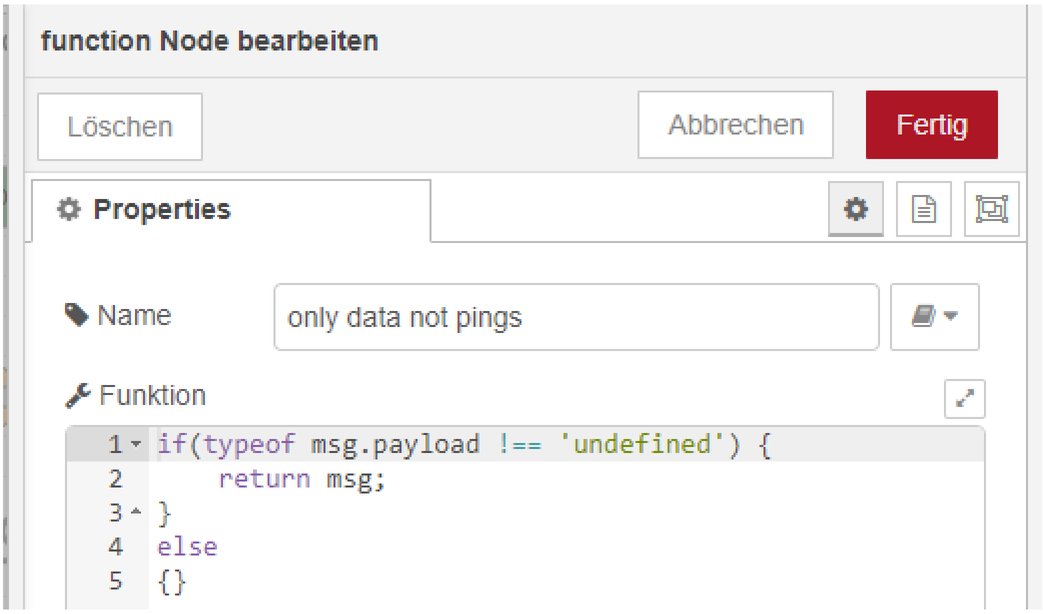


function Node bearbeiten

Name: data

Funktion

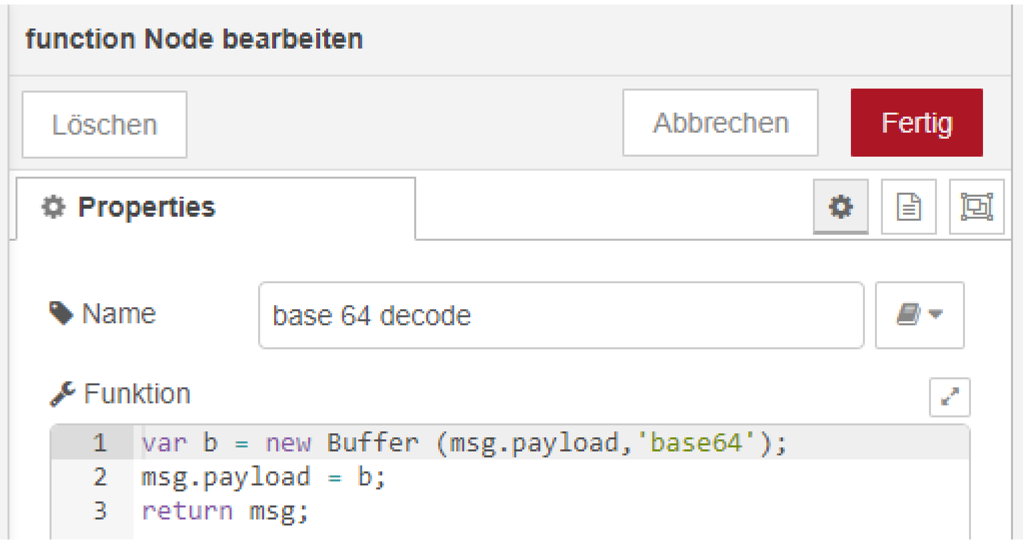
```
var datastring = msg.payload.data
msg.payload = datastring
return msg;
```



Name: only data not pings

Funktion

```
if(typeof msg.payload !== 'undefined') {
    return msg;
}
else
{}
```



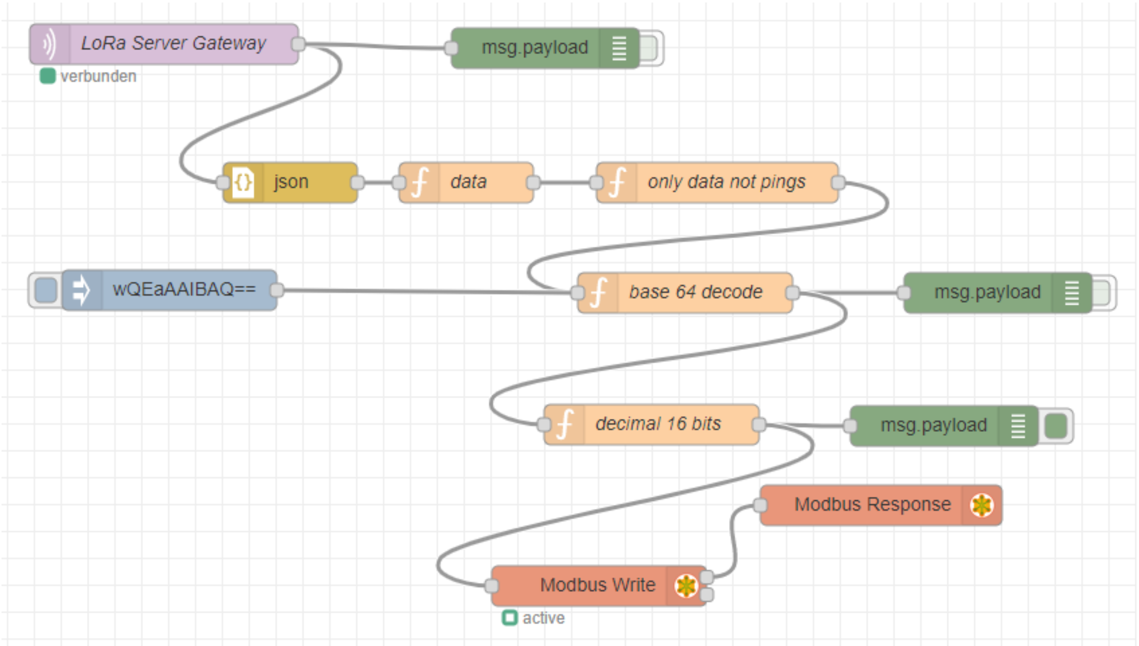
Name: base 64 decode

Funktion

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

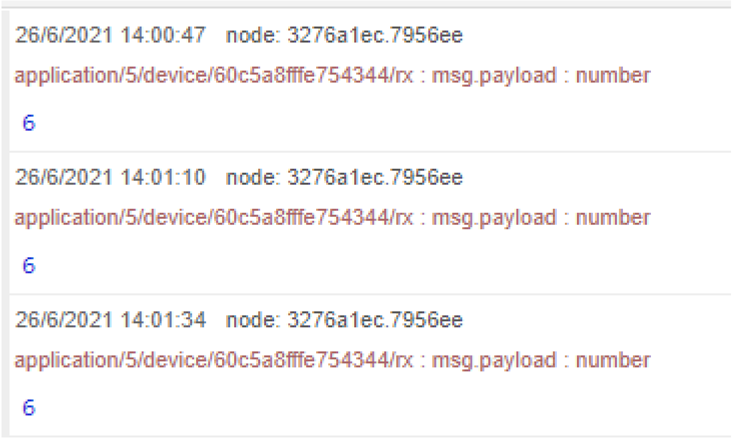
Now we decode such buffer

With bytes 3 and 4 to get a 16bit integer



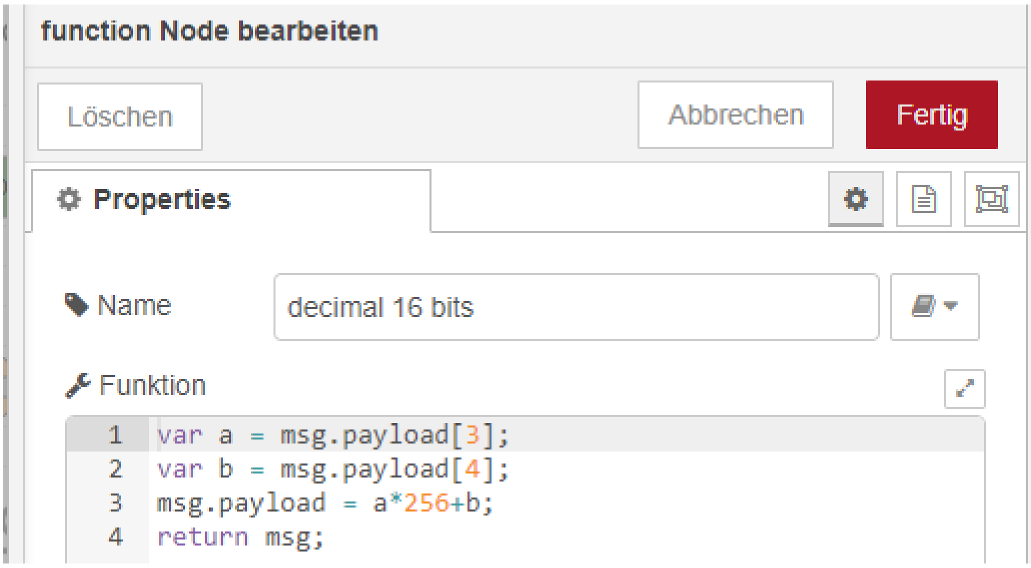
You can find the code here

https://github.com/xavierflorensa/Schneider-PLC-to-PLC-comms-with-RAK-wireless



```
26/6/2021 14:00:47   node: 3276a1ec.7956ee
application/5/device/60c5a8fffe754344/rx : msg.payload : number
 6
26/6/2021 14:01:10   node: 3276a1ec.7956ee
application/5/device/60c5a8fffe754344/rx : msg.payload : number
 6
26/6/2021 14:01:34   node: 3276a1ec.7956ee
application/5/device/60c5a8fffe754344/rx : msg.payload : number
 6
```

And inject to the receiver PLC thru Modbus TCP on PLC register %MW1

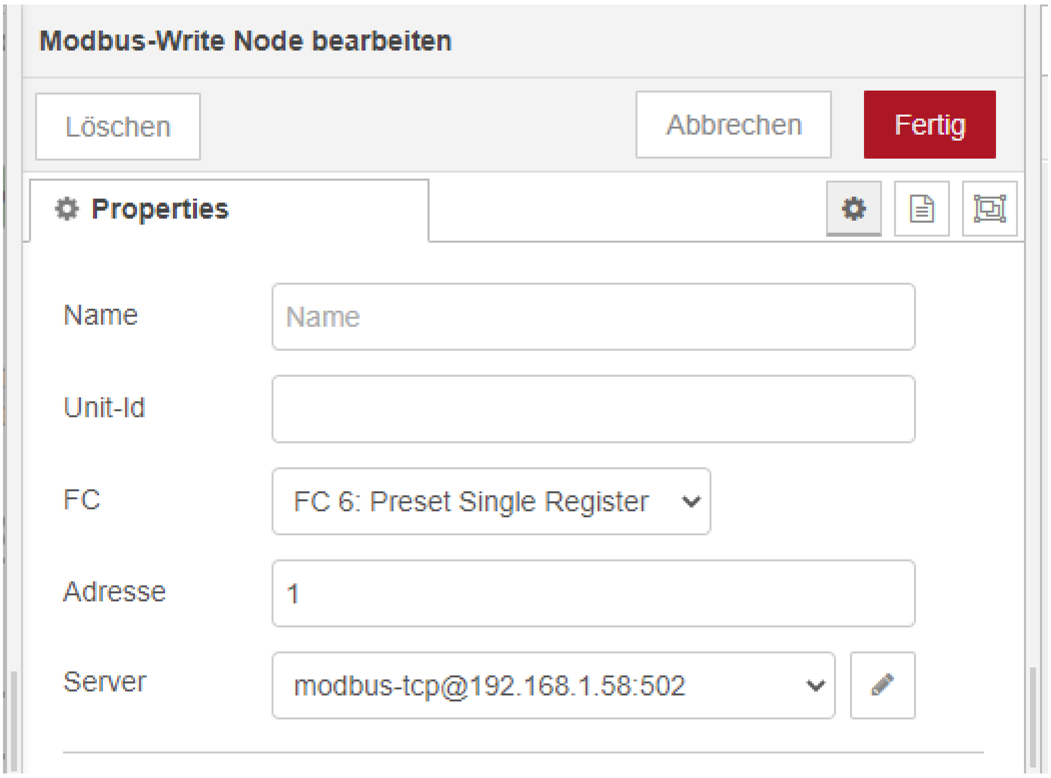


function Node bearbeiten

Name: decimal 16 bits

Funktion

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



Modbus-Write Node bearbeiten

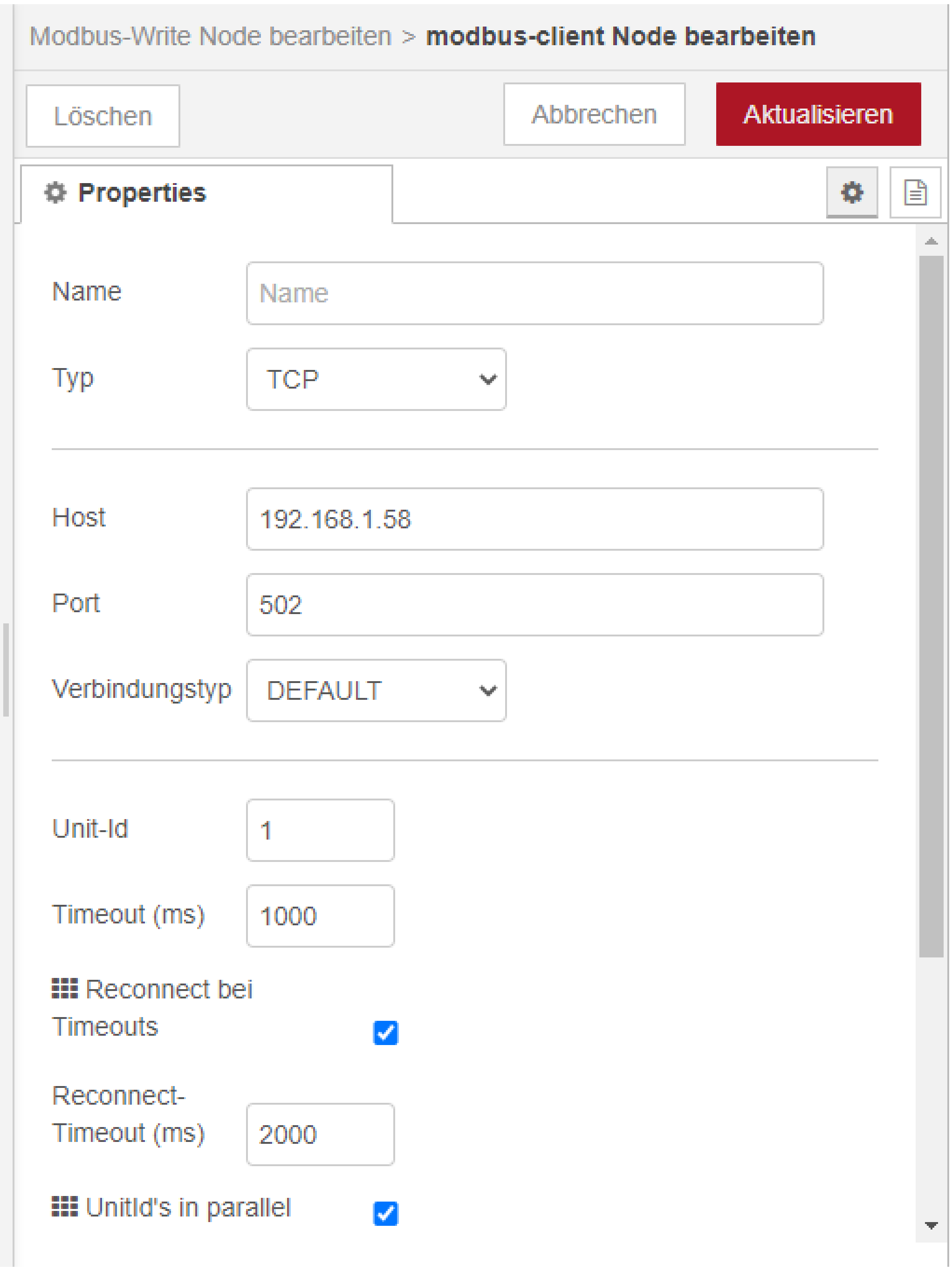
Name: Name

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**

Name: Name

Typ: TCP

Host: 192.168.1.58

Port: 502

Verbindungstyp: DEFAULT

Unit-Id: 1

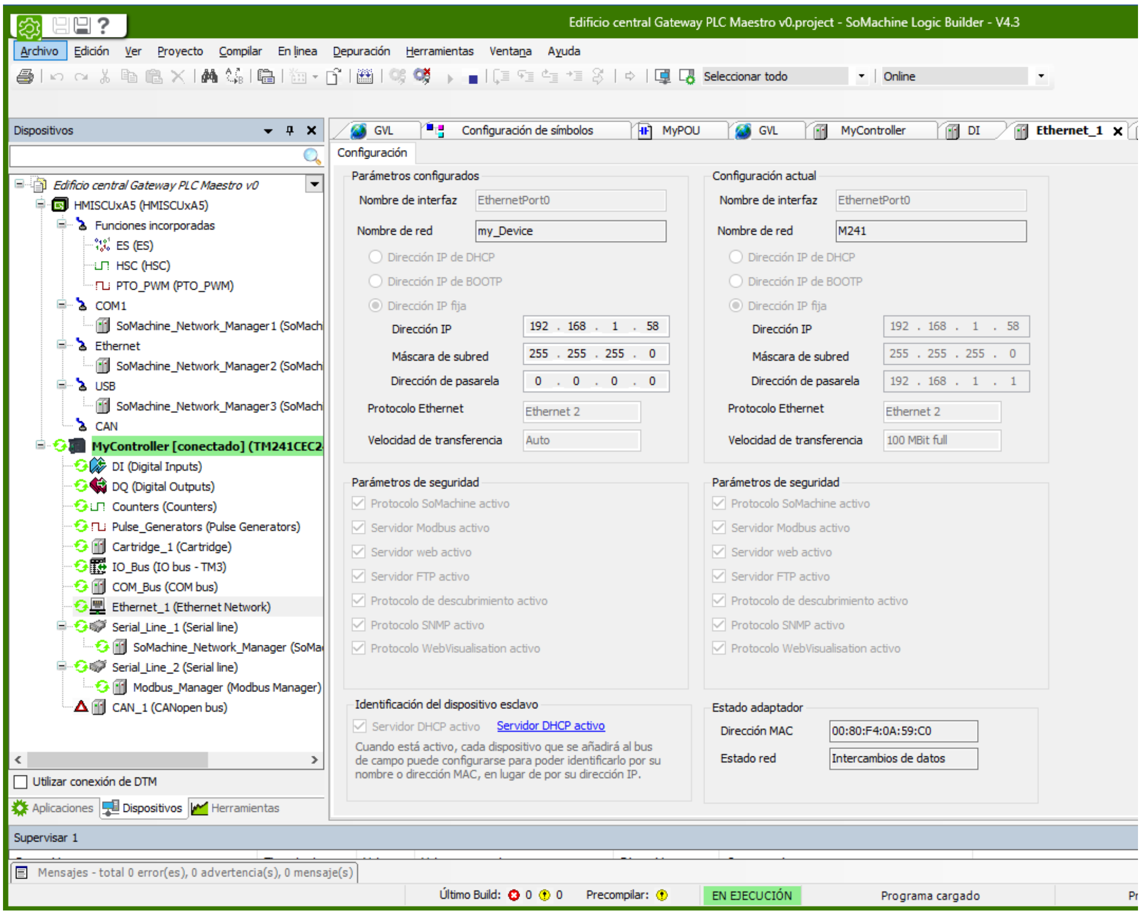
Timeout (ms): 1000

Reconnect bei Timeouts ☑

Reconnect-Timeout (ms): 2000

UnitId's in parallel ☑

Now let’s take a look at the receiver PLC, where you only have to set an IP address, allocate a global memory register for instance on %MW1 and create a program t osee the PLC digital outpus as an image of the sender PLC inputs.


Edificio central Gateway PLC Maestro v0.project - SoMachine Logic Builder - V4.3
Archivo
Edición
Ver
Proyecto
Compilar
En línea
Depuración
Herramientas
Ventana
Ayuda
Seleccionar todo
Online
Dispositivos
GVL
Configuración de símbolos
MyPOU
GVL
MyController
DI
Ethernet_1
Configuración
Edificio central Gateway PLC Maestro v0
HMISCUxA5 (HMISCUxA5)
Funciones incorporadas
ES (ES)
HSC (HSC)
PTO_PWM (PTO_PWM)
COM1
SoMachine_Network_Manager1 (SoMach
Ethernet
SoMachine_Network_Manager2 (SoMach
USB
SoMachine_Network_Manager3 (SoMach
CAN
MyController [conectado] (TM241CEC2
DI (Digital Inputs)
DQ (Digital Outputs)
Counters (Counters)
Pulse_Generators (Pulse Generators)
Cartridge_1 (Cartridge)
IO_Bus (IO bus - TM3)
COM_Bus (COM bus)
Ethernet_1 (Ethernet Network)
Serial_Line_1 (Serial line)
SoMachine_Network_Manager (SoMa
Serial_Line_2 (Serial line)
Modbus_Manager (Modbus Manager)
CAN_1 (CANopen bus)
□ Utilizar conexión de DTM
Aplicaciones
Dispositivos
Herramientas
Supervisar 1

Parámetros configurados

Nombre de interfaz: EthernetPort0

Nombre de red: my_Device

○ Dirección IP de DHCP

○ Dirección IP de BOOTP

● Dirección IP fija

Dirección IP: 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

Parámetros de seguridad

☑ Protocolo SoMachine activo
☑ Servidor Modbus activo
☑ Servidor web activo
☑ Servidor FTP activo
☑ Protocolo de descubrimiento activo
☑ Protocolo SNMP activo
☑ Protocolo WebVisualisation activo

Identificación del dispositivo esclavo

☑ Servidor DHCP activo Servidor DHCP activo

Cuando está activo, cada dispositivo que se añadirá al bus de campo puede configurarse para poder identificarlo por su nombre o dirección MAC, en lugar de por su dirección IP.

Configuración actual

Nombre de interfaz: EthernetPort0

Nombre de red: M241

○ Dirección IP de DHCP

○ Dirección IP de BOOTP

● Dirección IP fija

Dirección IP: 192 . 168 . 1 . 58

Máscara de subred: 255 . 255 . 255 . 0

Dirección de pasarela: 192 . 168 . 1 . 1

Protocolo Ethernet: Ethernet 2

Velocidad de transferencia: 100 MBit full

Parámetros de seguridad

☑ Protocolo SoMachine activo
☑ Servidor Modbus activo
☑ Servidor web activo
☑ Servidor FTP activo
☑ Protocolo de descubrimiento activo
☑ Protocolo SNMP activo
☑ Protocolo WebVisualisation activo

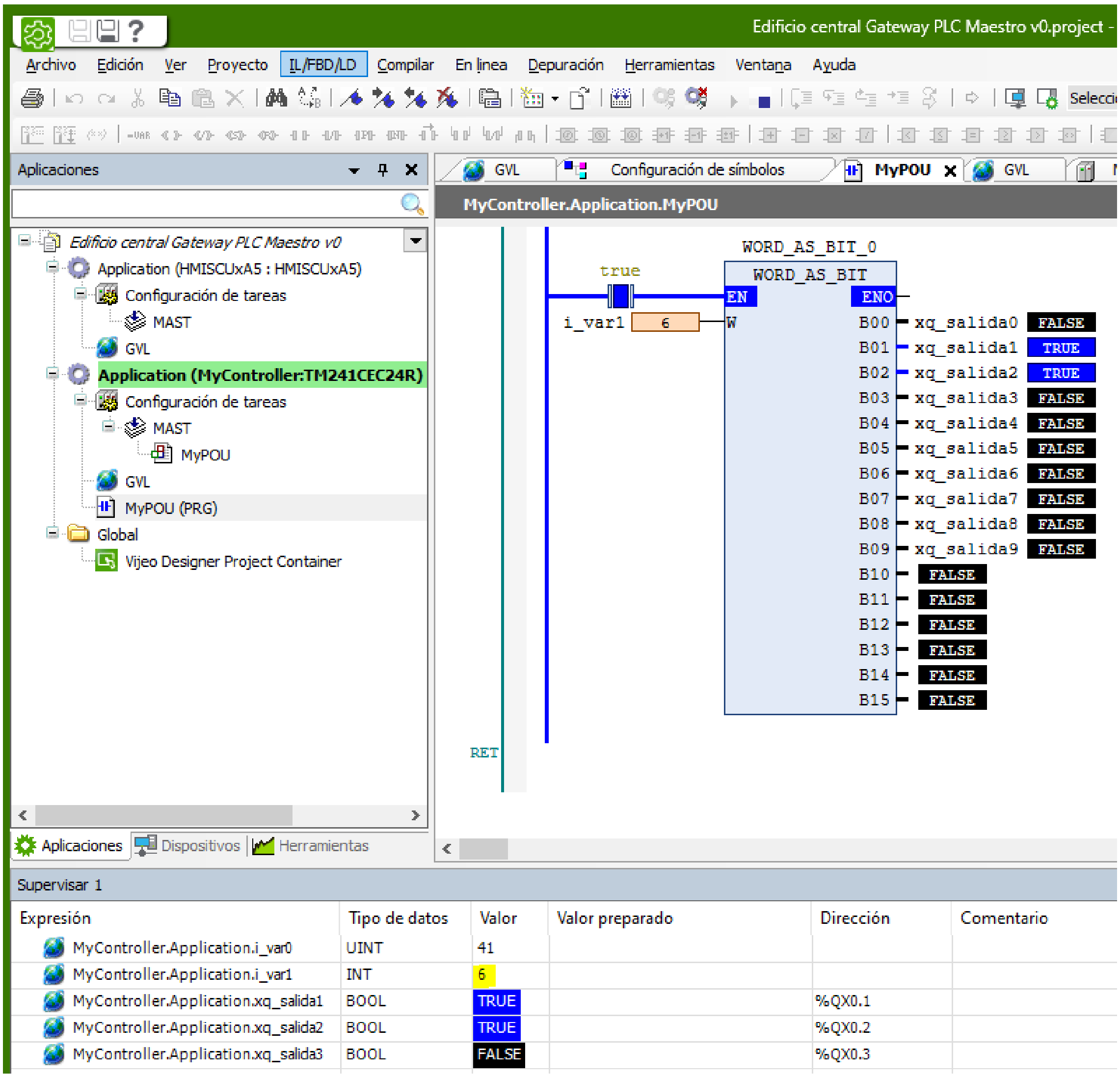
Estado adaptador

Dirección MAC: 00:80:F4:0A:59:C0

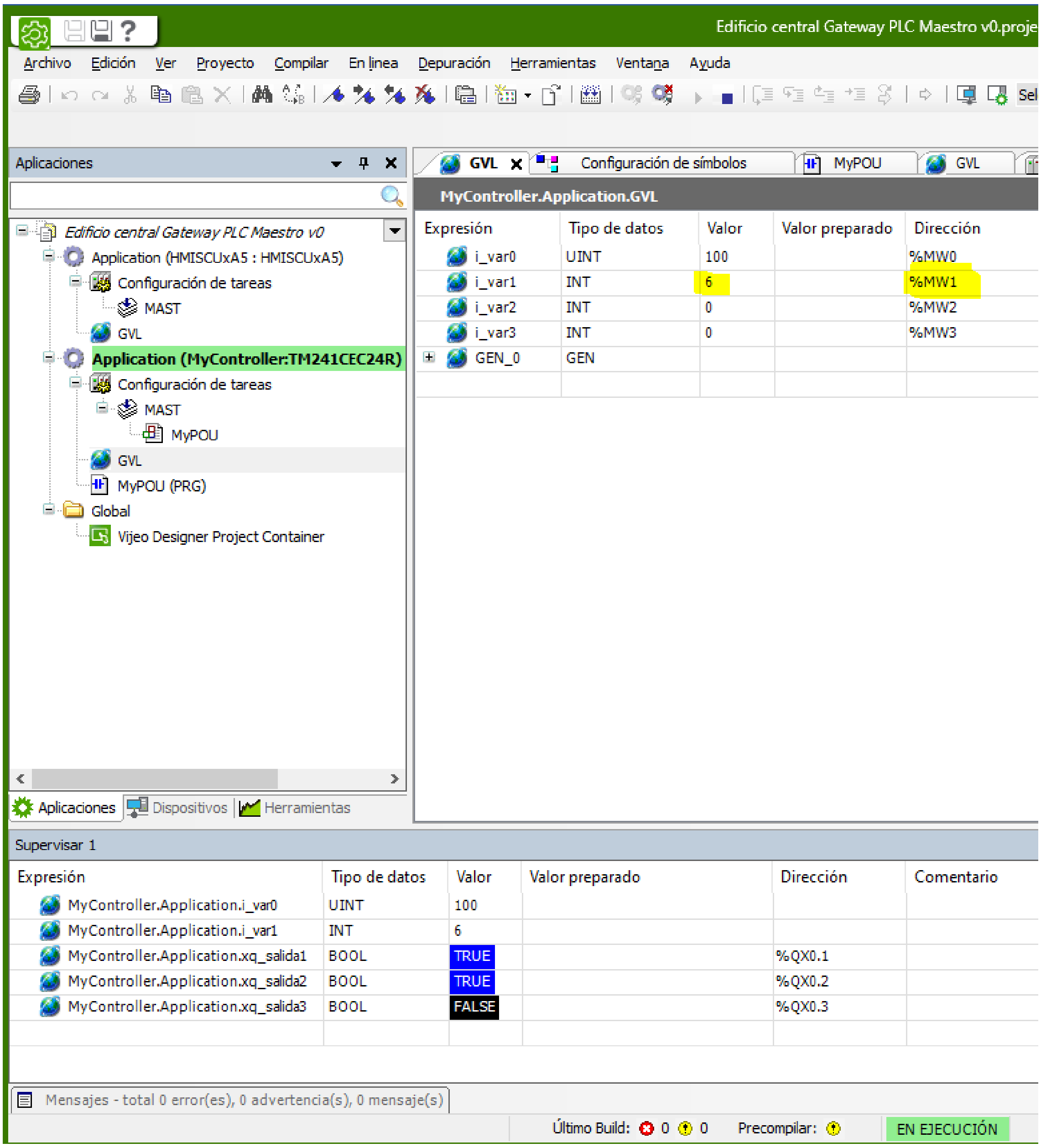
Estado red: Intercambios de datos

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

Último Build: 0 0 Precompilar: EN EJECUCIÓN Programa cargado



Expresión	Tipo de datos	Valor	Valor preparado	Dirección	Comentario
MyController.Application.i_var0	UINT	41			
MyController.Application.i_var1	INT	6			
MyController.Application.xq_salida1	BOOL	TRUE		%QX0.1	
MyController.Application.xq_salida2	BOOL	TRUE		%QX0.2	
MyController.Application.xq_salida3	BOOL	FALSE		%QX0.3	


Edificio central Gateway PLC Maestro v0.proje
Archivo Edición Ver Proyecto Compilar En línea Depuración Herramientas Ventana Ayuda
Aplicaciones
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
Aplicaciones Dispositivos Herramientas
GVL Configuración de símbolos MyPOU GVL
Último Build: 0 0 Precompilar:
EN EJECUCIÓN

MyController.Application.GVL

Expresión	Tipo de datos	Valor	Valor preparado	Dirección
i_var0	UINT	100		%MW0
i_var1	INT	6		%MW1
i_var2	INT	0		%MW2
i_var3	INT	0		%MW3
GEN_0	GEN			

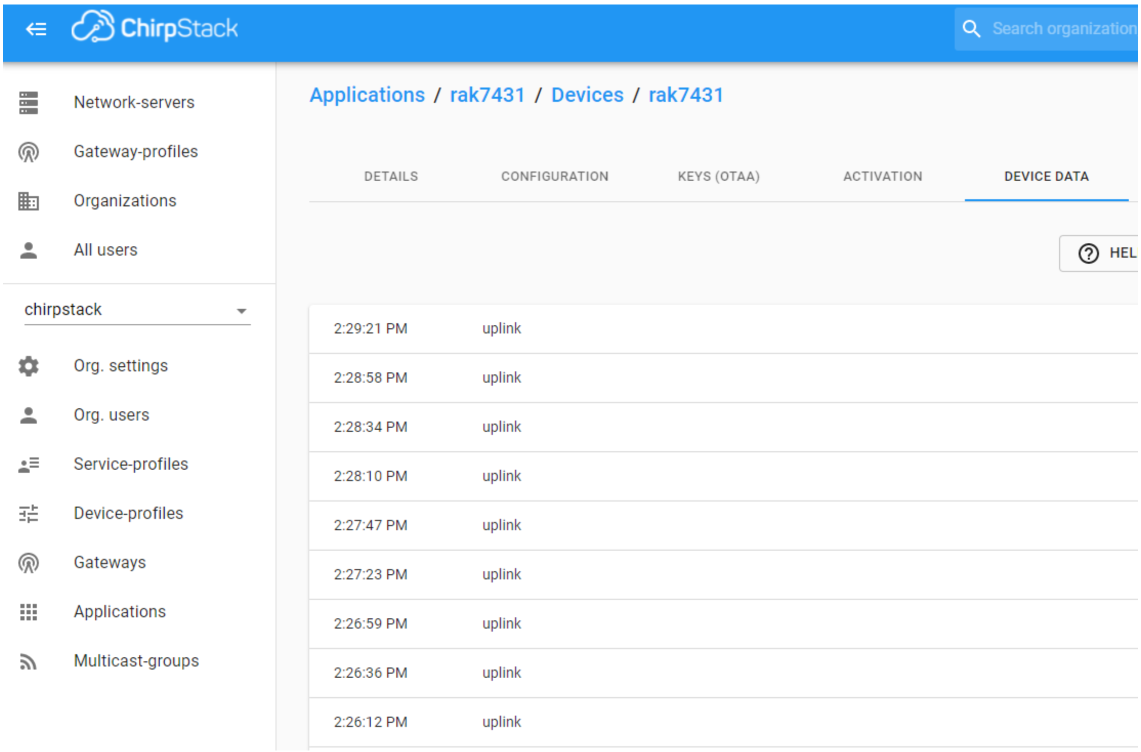
Supervisar 1

Expresión	Tipo de datos	Valor	Valor preparado	Dirección	Comentario
MyController.Application.i_var0	UINT	100			
MyController.Application.i_var1	INT	6			
MyController.Application.xq_salida1	BOOL	TRUE		%QX0.1	
MyController.Application.xq_salida2	BOOL	TRUE		%QX0.2	
MyController.Application.xq_salida3	BOOL	FALSE		%QX0.3	

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

So it Works!

Now let’s decrease the sending period since we are transmitting each 24 seconds



```
2:29:21 PM    uplink
2:28:58 PM    uplink
2:28:34 PM    uplink
2:28:10 PM    uplink
2:27:47 PM    uplink
2:27:23 PM    uplink
2:26:59 PM    uplink
2:26:36 PM    uplink
2:26:12 PM    uplink
```