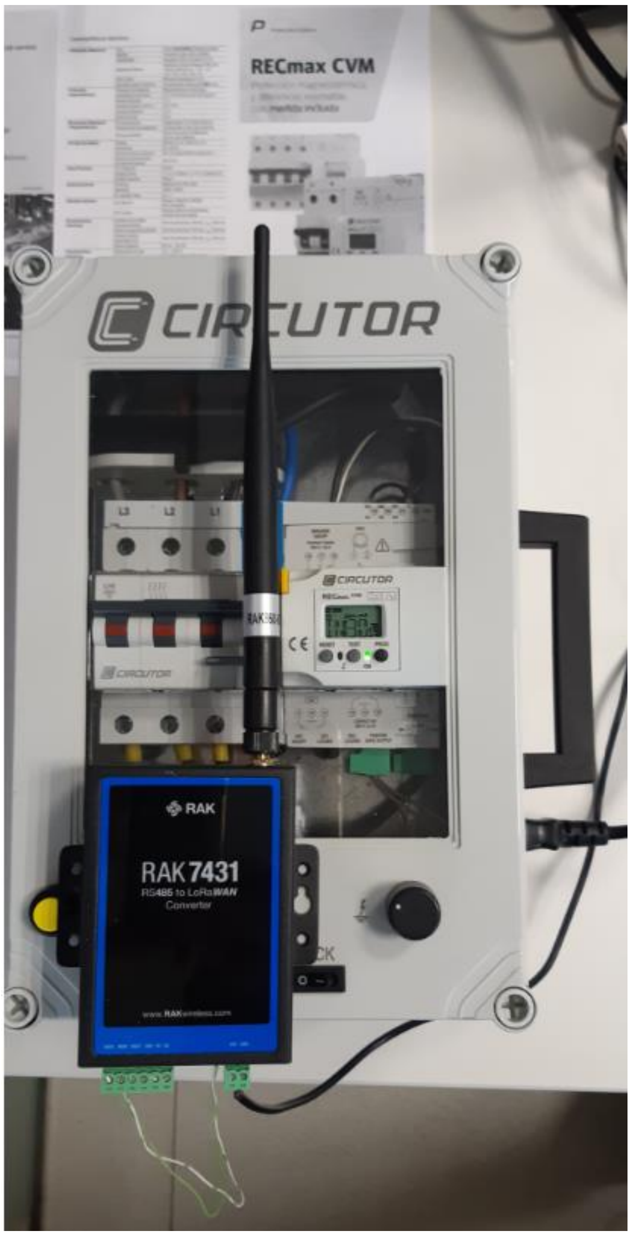
Earth leakage circuit breaker with automatic reclosing and measurement system RECmax-CVM and LoRaWAN RAK 7431 to LoRaWAN converter



We start reading some energy values

Voltage

Table 22: Modbus memory map (Table 1)

Parameter	Symbol	Instantaneous	Maximum	Minimum	Units
Phase voltage L1	V1	00-01	106-107	164-165	V x 10
Current L1	A1	02-03	108-109	166-167	mA
Active power L1	kW 1	04-05	10A-10B	168-169	W

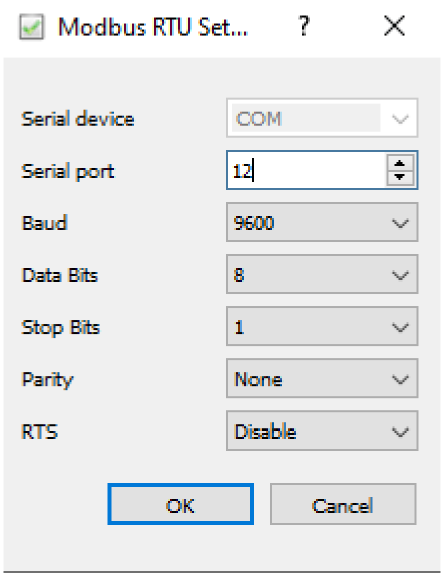
These are the default comm's parametres

Table 43:Modbus memory map: Communications

Communications			
Configuration variable	**Address**	**Valid data margin**	**Default value**
Protocol	2742	**0 :** Modbus	0
Peripheral number	2743	**0 - 255**	1
Transmission speed	2744	**0:** 9600 **- 1:**19200	0
Parity	2745	**0:** No parity **1:** Odd parity **2:** Even parity	0
Data bits	2746	**0 :** 8-bit **1:** 7 bits	0
Stop bits	2747	**0 :** 1 stop bit **1:** 2 stop bits	0

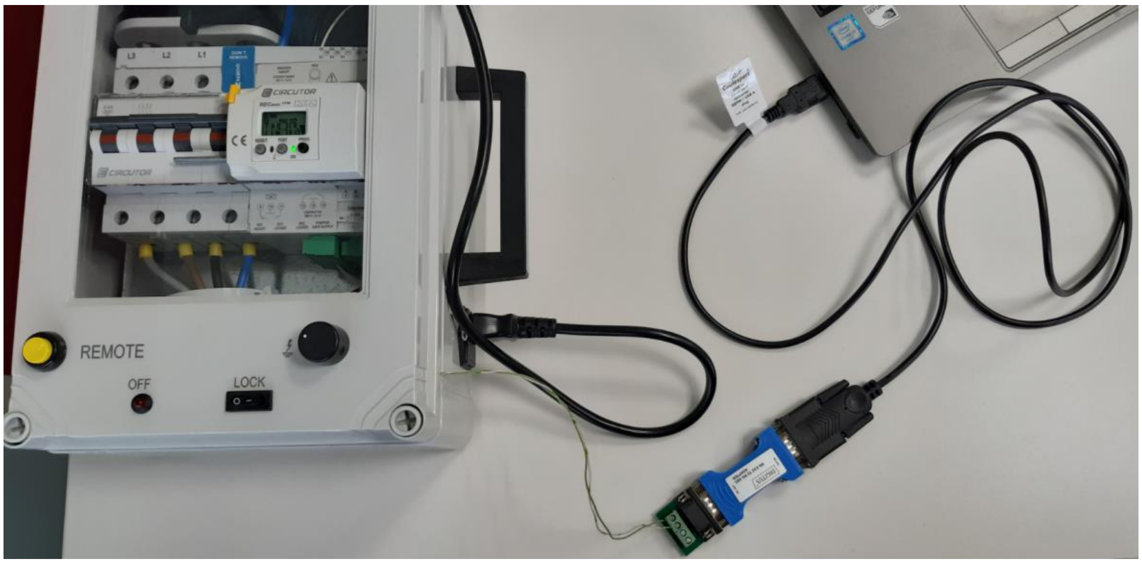
So let's start with software qModMaster

Use this configuration

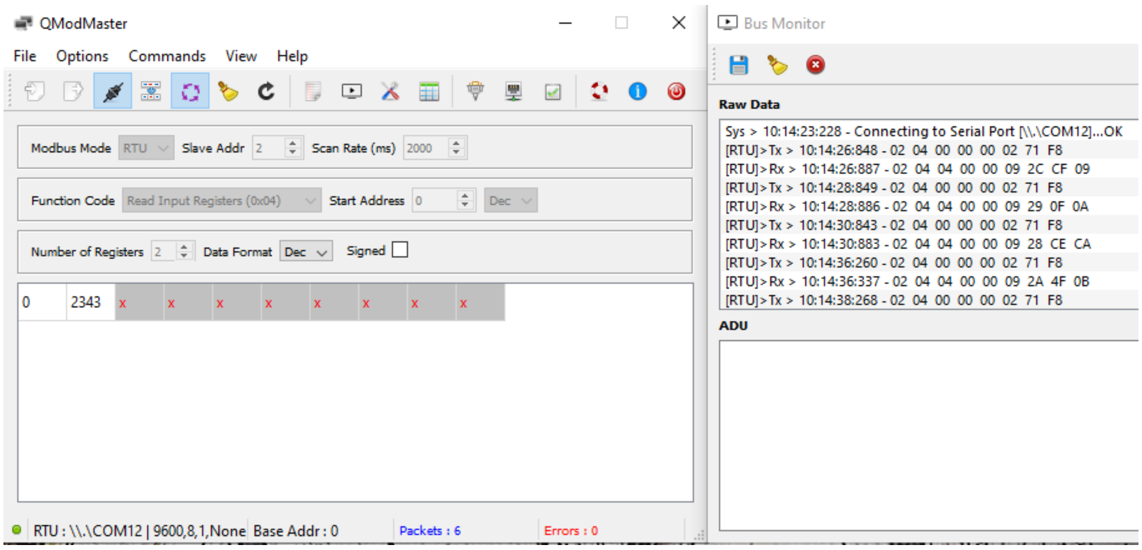

Modbus RTU Set...

Serial device	COM
Serial port	12
Baud	9600
Data Bits	8
Stop Bits	1
Parity	None
RTS	Disable

Let’s read the voltage Phase 1



Peripheral slave was 2 on the Device after looking at the front LCD


QModMaster

Modbus Mode: RTU Slave Addr: 2 Scan Rate (ms): 2000

Function Code: Read Input Registers (0x04) Start Address: 0 Dec

Number of Registers: 2 Data Format: Dec Signed: ☐

0	2343	x	x	x	x	x	x	x	x

RTU : \\.\COM12 | 9600,8,1,None Base Addr : 0 Packets : 6 Errors : 0

Bus Monitor

Raw Data

```
Sys > 10:14:23:228 - Connecting to Serial Port [\\.\COM12]...OK
[RTU]>Tx > 10:14:26:848 - 02  04  00  00  00  02  71  F8
[RTU]>Rx > 10:14:26:887 - 02  04  04  00  00  09  2C  CF  09
[RTU]>Tx > 10:14:28:849 - 02  04  00  00  00  02  71  F8
[RTU]>Rx > 10:14:28:886 - 02  04  04  00  00  09  29  0F  0A
[RTU]>Tx > 10:14:30:843 - 02  04  00  00  00  02  71  F8
[RTU]>Rx > 10:14:30:883 - 02  04  04  00  00  09  28  CE  CA
[RTU]>Tx > 10:14:36:260 - 02  04  00  00  00  02  71  F8
[RTU]>Rx > 10:14:36:337 - 02  04  04  00  00  09  2A  4F  0B
[RTU]>Tx > 10:14:38:268 - 02  04  00  00  00  02  71  F8
```

ADU

So the voltatge is 234,3 Volts (We have to divide the data by 10)

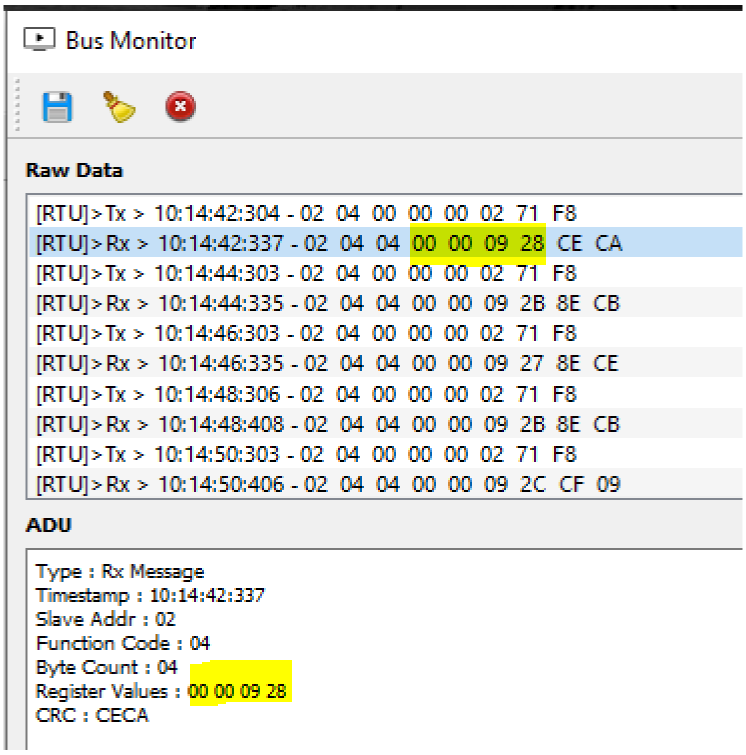
So the Modbus command is

02 04 00 00 00 02 71 F8

And this will be the right poll task

```
AT+ADDPOLL=1:020400000002 71F8
```

And the answer is


Bus Monitor

Raw Data

```
[RTU]>Tx > 10:14:42:304 - 02  04  00  00  00  02  71  F8
[RTU]>Rx > 10:14:42:337 - 02  04  04  00  00  09  28  CE  CA
[RTU]>Tx > 10:14:44:303 - 02  04  00  00  00  02  71  F8
[RTU]>Rx > 10:14:44:335 - 02  04  04  00  00  09  2B  8E  CB
[RTU]>Tx > 10:14:46:303 - 02  04  00  00  00  02  71  F8
[RTU]>Rx > 10:14:46:335 - 02  04  04  00  00  09  27  8E  CE
[RTU]>Tx > 10:14:48:306 - 02  04  00  00  00  02  71  F8
[RTU]>Rx > 10:14:48:408 - 02  04  04  00  00  09  2B  8E  CB
[RTU]>Tx > 10:14:50:303 - 02  04  00  00  00  02  71  F8
[RTU]>Rx > 10:14:50:406 - 02  04  04  00  00  09  2C  CF  09
```

ADU

```
Type : Rx Message
Timestamp : 10:14:42:337
Slave Addr : 02
Function Code : 04
Byte Count : 04
Register Values : 00 00 09 28
CRC : CECA
```

02 04 04 00 00 09 28 CE CA

28Hex is 40 in decimal

Yes since 9*256+40=2344 Volts (The difference is due to precision on the readings)

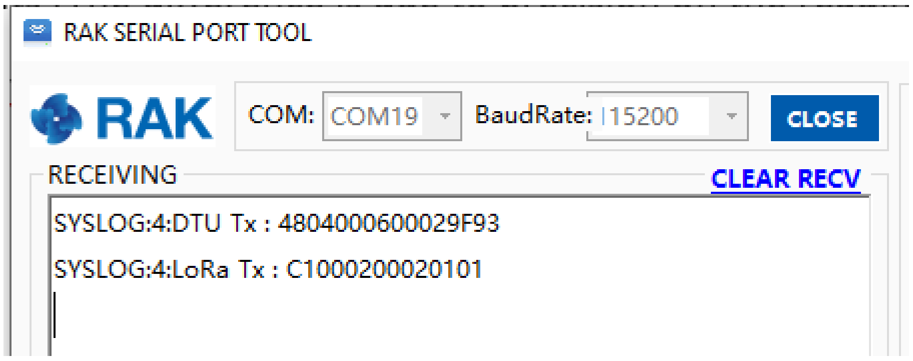
Now let's transmit these info thru LoRaWAN

Configuring RAK7431 to send the readings

Let's use the serial sofdtware tool and the USB connection

First we check for any configured POLL instruction

Since there is a Poll running



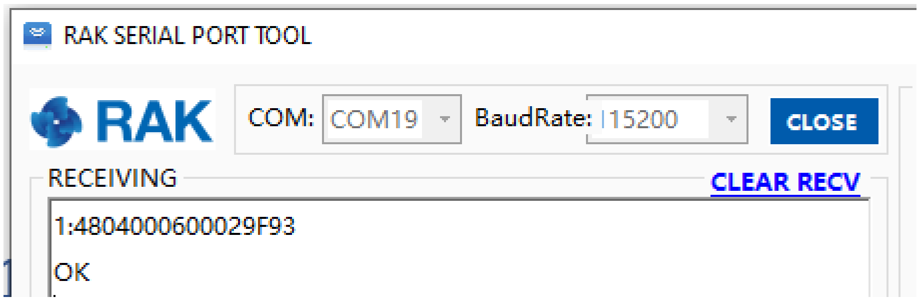
```
SYSLOG:4:DTU Tx : 4804000600029F93
SYSLOG:4:LoRa Tx : C1000200020101
```

According to the AT commands manual

https://docs.rakwireless.com/Product-Categories/WisNode/RAK7431/AT-Command-Manual/#data-interface-commands

13. **AT+POLLTASK**

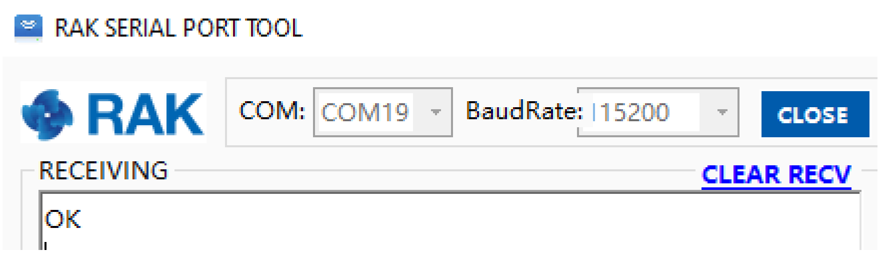
There was only one Poll task



```
1:4804000600029F93
OK
```

Let's remove it

AT+RMPOLL=1



```
OK
```

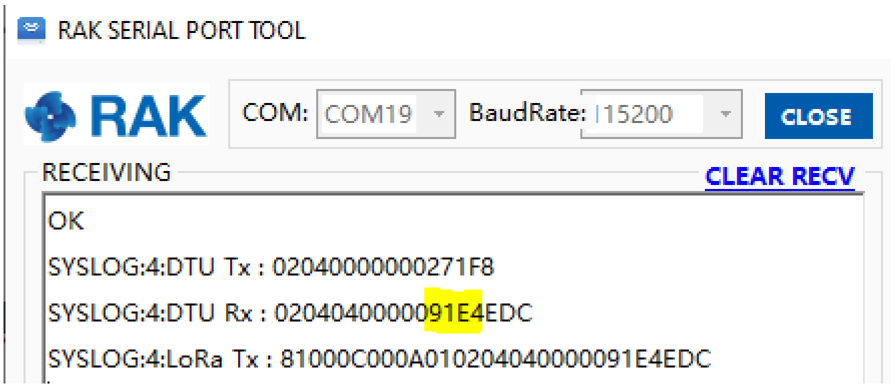
13. **AT+POLLTASK**

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=1:020400000000271F8
```

We are not in transparent mode



```
OK
SYSLOG:4:DTU Tx : 020400000000271F8
SYSLOG:4:DTU Rx : 0204040000091E4EDC
SYSLOG:4:LoRa Tx : 81000C000A0102040400000091E4EDC
```

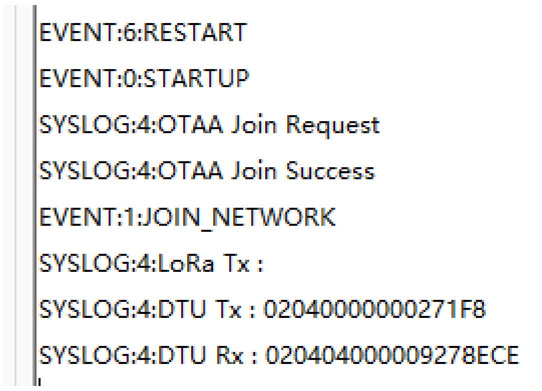
So we get the reading from Leakage Device, and this is 9 1E

1E Hex is 30

So (9*256+30)/10 = 233,4 Volts!

It is transmiting thru LoRa but there is no gateway available.

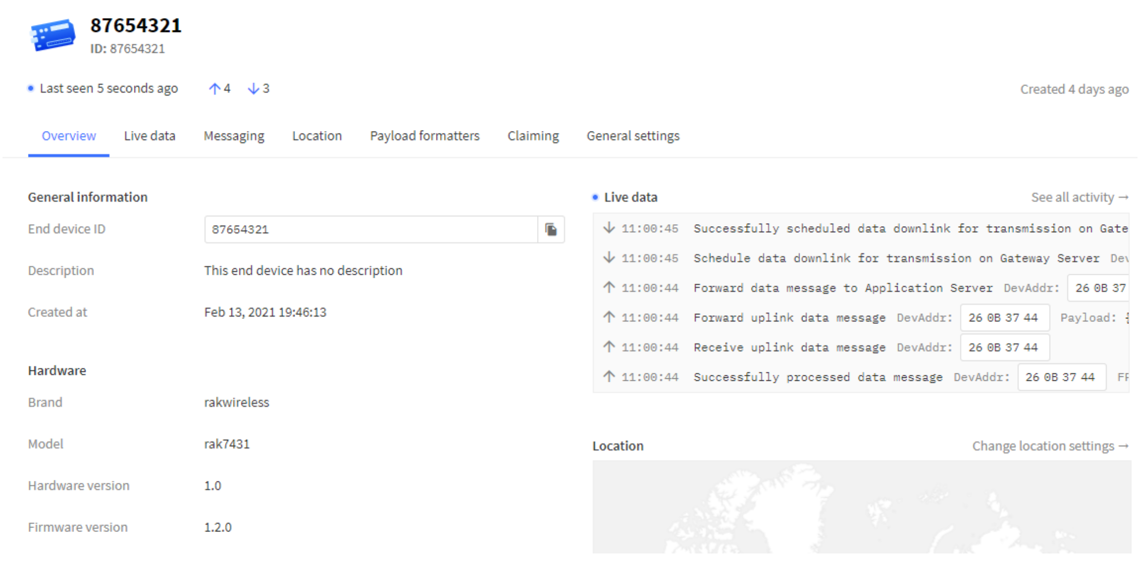
Let’s connect gateway



```
EVENT:6:RESTART
EVENT:0:STARTUP
SYSLOG:4:OTAA Join Request
SYSLOG:4:OTAA Join Success
EVENT:1:JOIN_NETWORK
SYSLOG:4:LoRa Tx :
SYSLOG:4:DTU Tx : 020400000000271F8
SYSLOG:4:DTU Rx : 0204040000009278ECE
```

Let's go to the ttn v3 console

https://eu1.cloud.thethings.network/



87654321
ID: 87654321

● Last seen 5 seconds ago ↑4 ↓3 Created 4 days ago

Overview | Live data | Messaging | Location | Payload formatters | Claiming | General settings

General information

End device ID	87654321
Description	This end device has no description
Created at	Feb 13, 2021 19:46:13

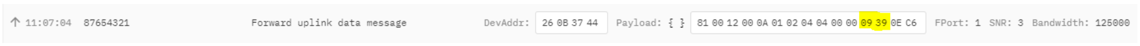
Hardware

Brand	rakwireless
Model	rak7431
Hardware version	1.0
Firmware version	1.2.0

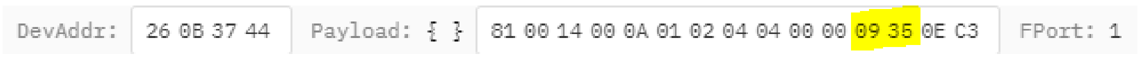
● **Live data** See all activity →

```
↓ 11:00:45  Successfully scheduled data downlink for transmission on Gate
↓ 11:00:45  Schedule data downlink for transmission on Gateway Server  Dev
↑ 11:00:44  Forward data message to Application Server  DevAddr:  26 0B 37
↑ 11:00:44  Forward uplink data message  DevAddr:  26 0B 37 44   Payload: {
↑ 11:00:44  Receive uplink data message  DevAddr:  26 0B 37 44
↑ 11:00:44  Successfully processed data message  DevAddr:  26 0B 37 44   FF
```

Location Change location settings →

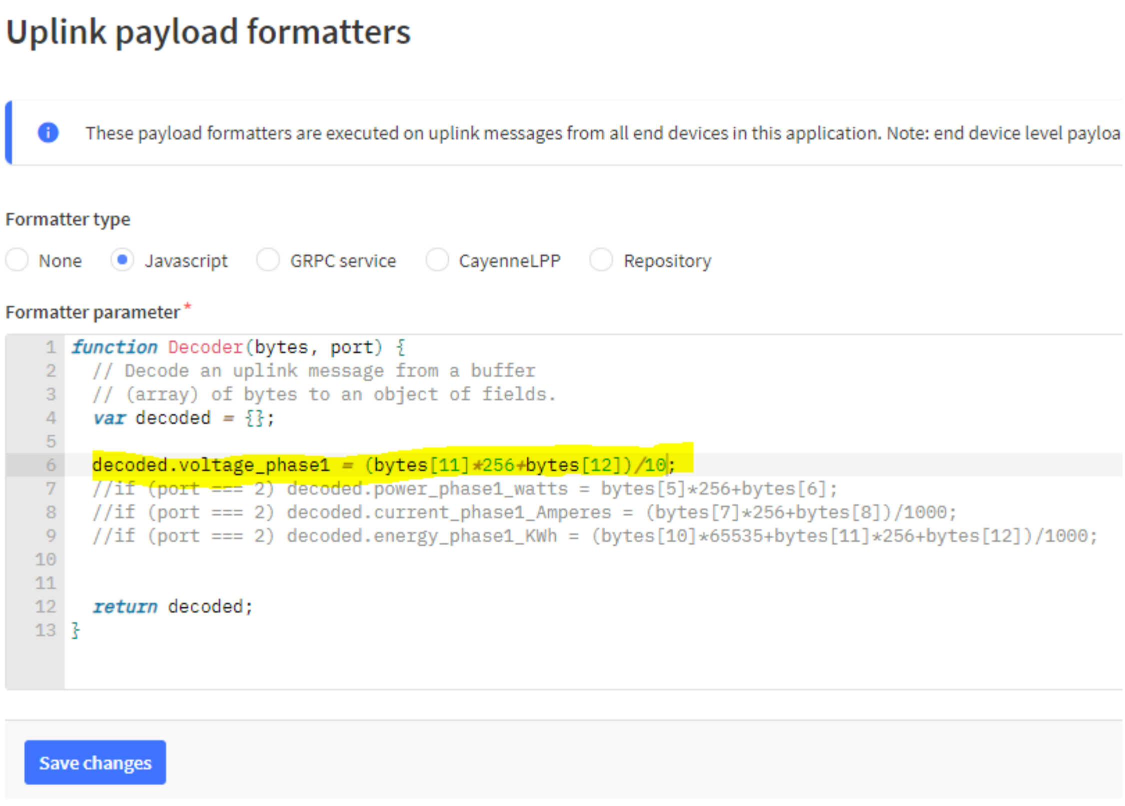


```
↑ 11:07:04  87654321   Forward uplink data message   DevAddr: 26 0B 37 44  Payload: { }  81 00 12 00 0A 01 02 04 04 00 00 09 39 0E C6  FPort: 1  SNR: 3  Bandwidth: 125000
```



```
DevAddr:  26 0B 37 44   Payload: { }  81 00 14 00 0A 01 02 04 04 00 00 09 35 0E C3   FPort: 1
```

Let's take a look at the payload decoder



Uplink payload formatters

ⓘ These payload formatters are executed on uplink messages from all end devices in this application. Note: end device level payloa

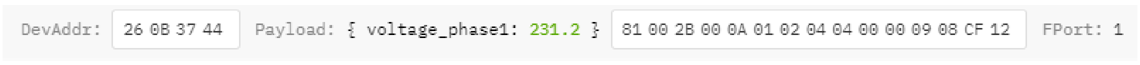
Formatter type

None [✓ Javascript] GRPC service CayenneLPP Repository

Formatter parameter *

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

  decoded.voltage_phase1 = (bytes[11]*256+bytes[12])/10;
  //if (port === 2) decoded.power_phase1_watts = bytes[5]*256+bytes[6];
  //if (port === 2) decoded.current_phase1_Amperes = (bytes[7]*256+bytes[8])/1000;
  //if (port === 2) decoded.energy_phase1_KWh = (bytes[10]*65535+bytes[11]*256+bytes[12])/1000;


  return decoded;
}
```

If we were on transparent mode, then we will have to choose another byte position on the payload decoder

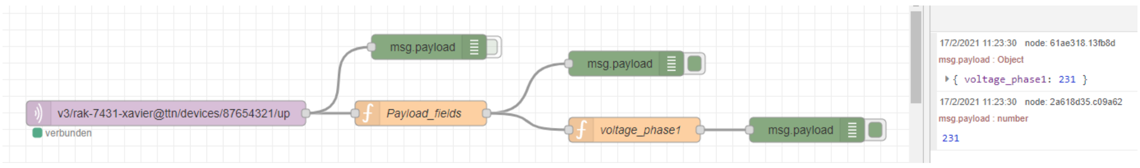


```
DevAddr: 26 0B 37 44   Payload: { voltage_phase1: 231.2 }  81 00 2B 00 0A 01 02 04 04 00 00 09 08 CF 12   FPort: 1
```

So we are sending the voltage thru LoRa

NODE-RED INTEGRATION

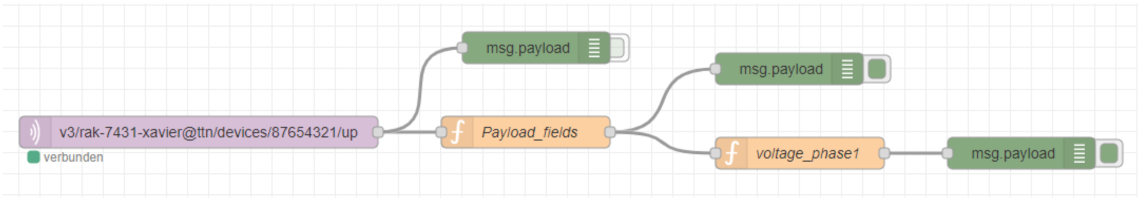
Now let's go to the Node-RED Integration

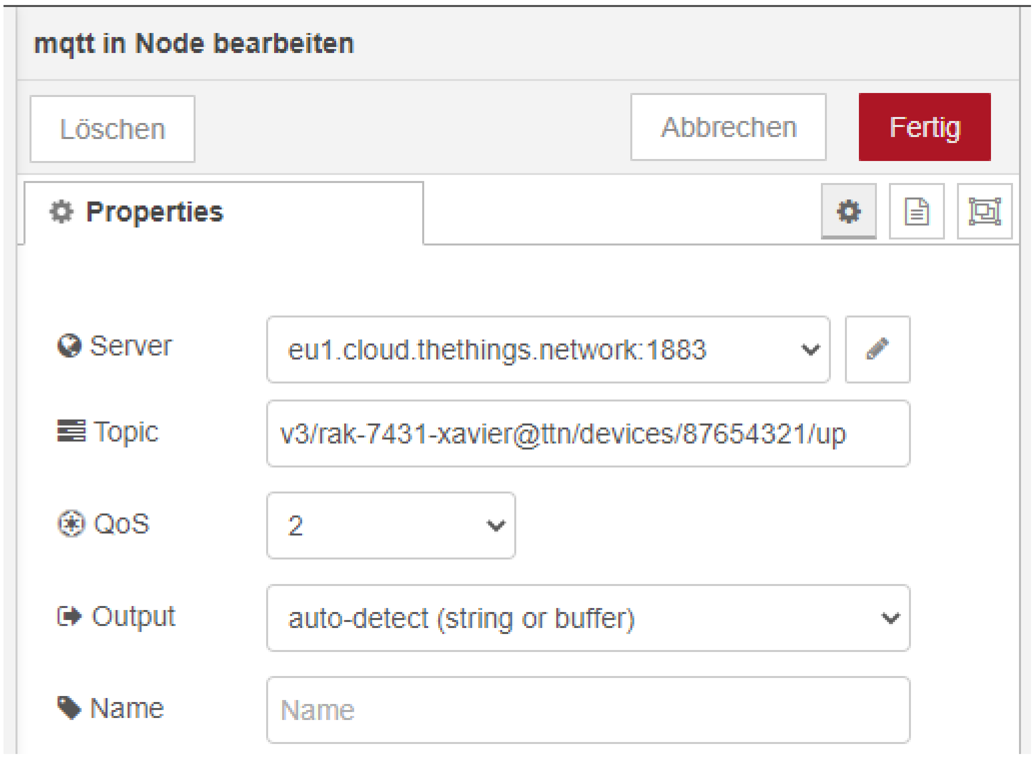


```
17/2/2021 11:23:30   node: 61ae318.13fb8d
msg.payload : Object
▸{ voltage_phase1: 231 }
17/2/2021 11:23:30   node: 2a618d35.c09a62
msg.payload : number
231
```

You can find the code here

https://github.com/xavierflorensa/RAK-7431-CIRCUTOR-RECmaxCVM




mqtt in Node bearbeiten

Löschen | Abbrechen | Fertig

Properties

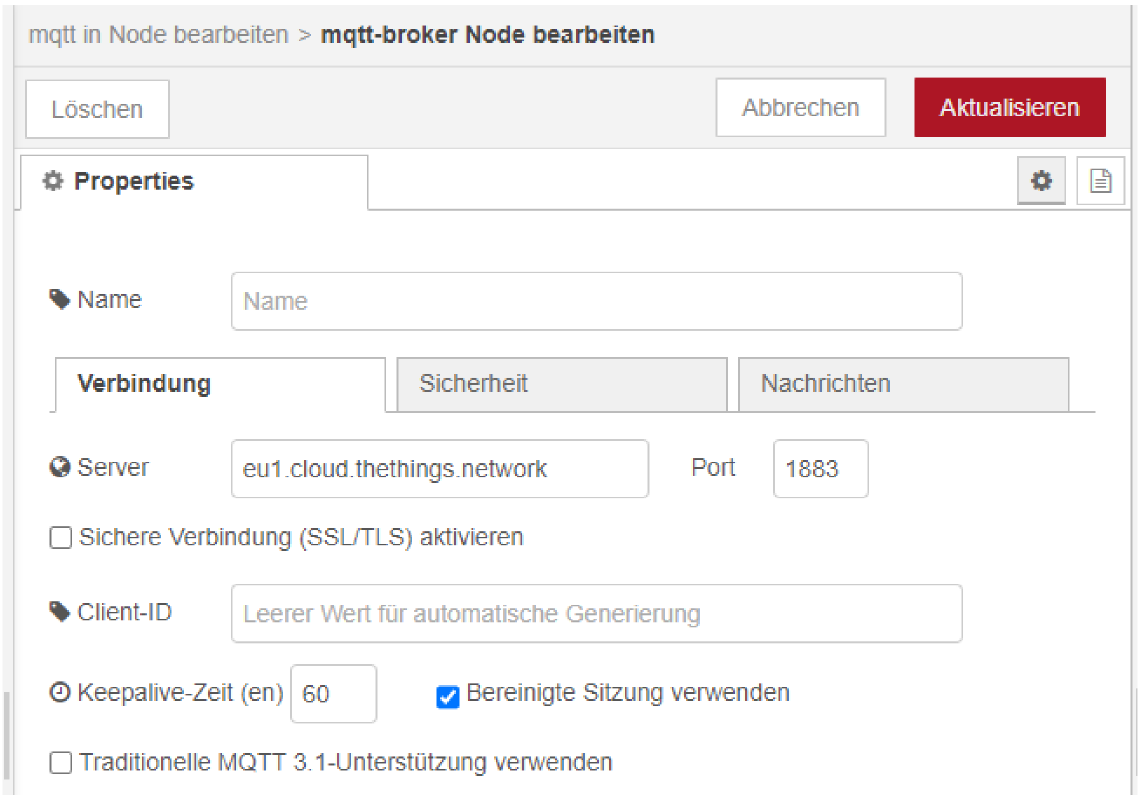
Server: eu1.cloud.thethings.network:1883

Topic: v3/rak-7431-xavier@ttn/devices/87654321/up

QoS: 2

Output: auto-detect (string or buffer)

Name: Name


mqtt in Node bearbeiten > mqtt-broker Node bearbeiten

Löschen | Abbrechen | Aktualisieren

Properties

Name: Name

Verbindung | Sicherheit | Nachrichten

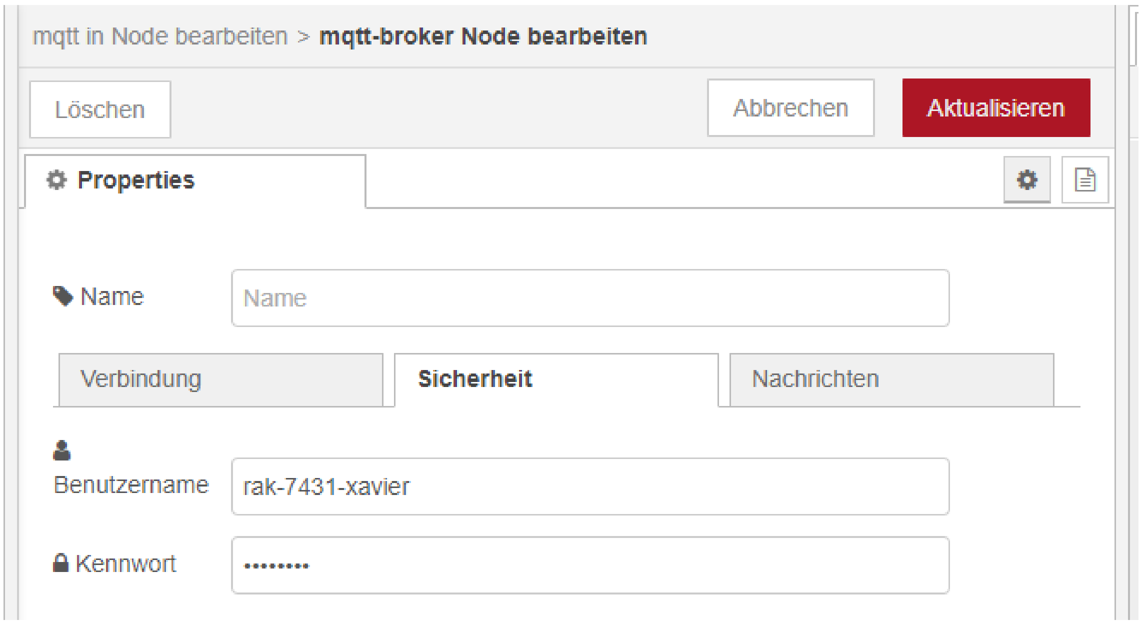
Server: eu1.cloud.thethings.network Port: 1883

☐ Sichere Verbindung (SSL/TLS) aktivieren

Client-ID: Leerer Wert für automatische Generierung

Keepalive-Zeit (en): 60 ☑ Bereinigte Sitzung verwenden

☐ Traditionelle MQTT 3.1-Unterstützung verwenden


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

Löschen | Abbrechen | Aktualisieren

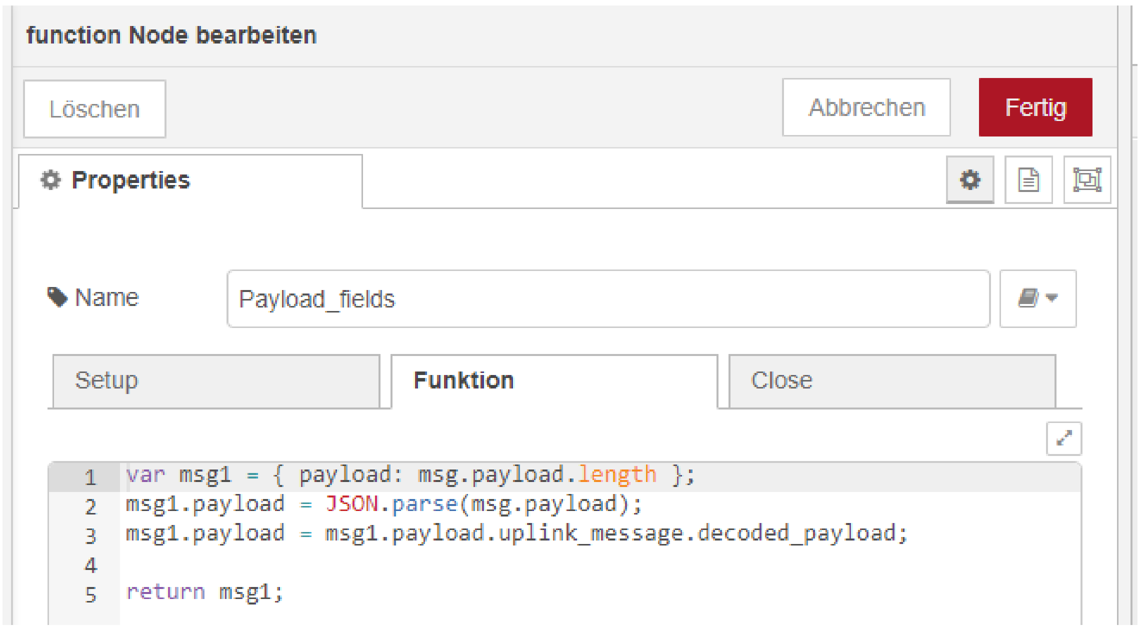
Properties

Name: Name

Verbindung | **Sicherheit** | Nachrichten

Benutzername: rak-7431-xavier

Kennwort: ••••••••


function Node bearbeiten

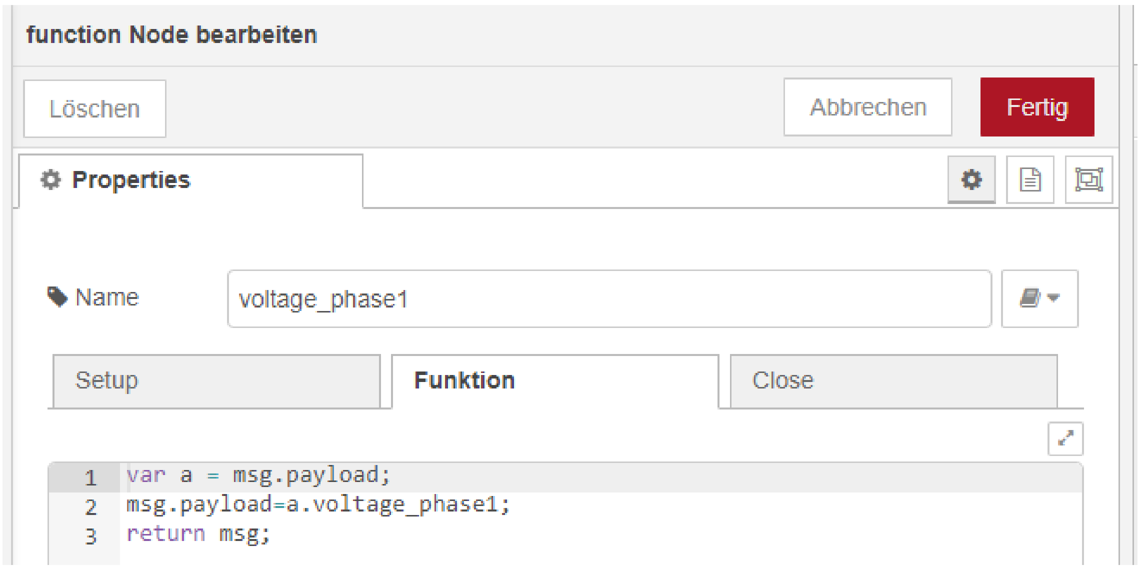
Löschen | Abbrechen | Fertig

Properties

Name: Payload_fields

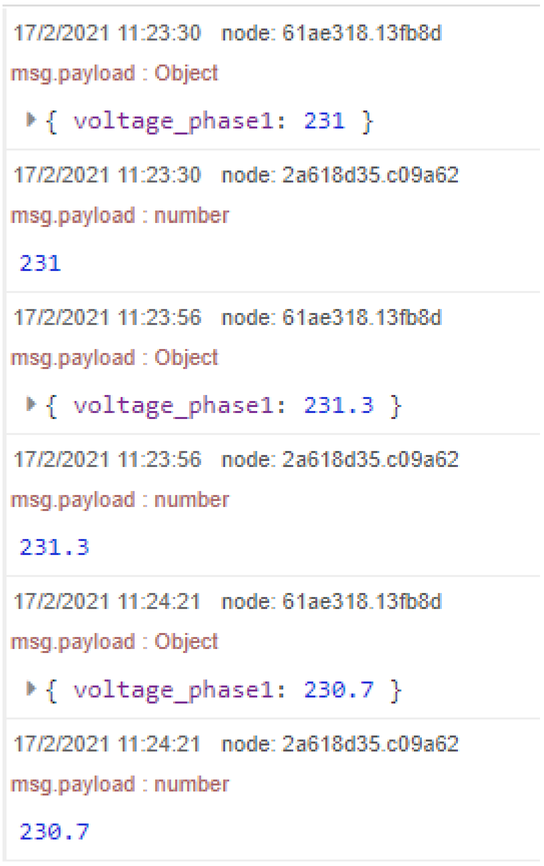
Setup | **Funktion** | Close

```
var msg1 = { payload: msg.payload.length };
msg1.payload = JSON.parse(msg.payload);
msg1.payload = msg1.payload.uplink_message.decoded_payload;

return msg1;
```



Name: voltage_phase1

```
var a = msg.payload;
msg.payload=a.voltage_phase1;
return msg;
```

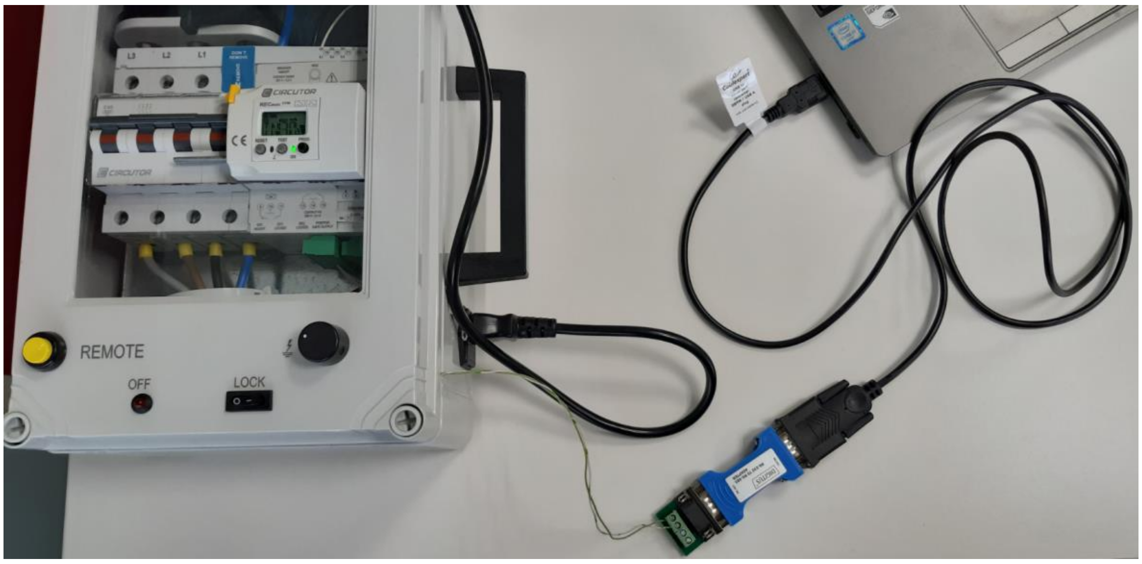


```
17/2/2021 11:23:30   node: 61ae318.13fb8d
msg.payload : Object
 ▸{ voltage_phase1: 231 }
17/2/2021 11:23:30   node: 2a618d35.c09a62
msg.payload : number
 231
17/2/2021 11:23:56   node: 61ae318.13fb8d
msg.payload : Object
 ▸{ voltage_phase1: 231.3 }
17/2/2021 11:23:56   node: 2a618d35.c09a62
msg.payload : number
 231.3
17/2/2021 11:24:21   node: 61ae318.13fb8d
msg.payload : Object
 ▸{ voltage_phase1: 230.7 }
17/2/2021 11:24:21   node: 2a618d35.c09a62
msg.payload : number
 230.7
```

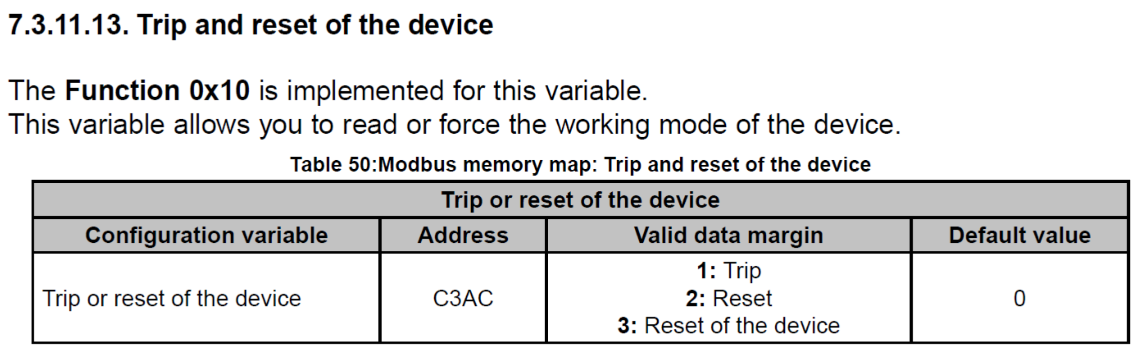
DOWNLINK TO REMOTELY OPEN CLOSE THE CIRCUIT BREAKER

Now let’s try the downlink to open/close then circuit breaker remotely thru LoRaWAN

First of all, we go back to the PC-circuit breaker connection



Opening circuit breaker



7.3.11.13. Trip and reset of the device

The **Function 0x10** is implemented for this variable.
This variable allows you to read or force the working mode of the device.

Table 50:Modbus memory map: Trip and reset of the device

Trip or reset of the device			
Configuration variable	**Address**	**Valid data margin**	**Default value**
Trip or reset of the device	C3AC	**1:** Trip **2:** Reset **3:** Reset of the device	0

First of all, it is worth to know that RECMAX CVM is designed to give preference to protecion first and on a second priority to Communications.

So if you want to get a reliable open/close control you have to first raise a flag before performing the open/close.

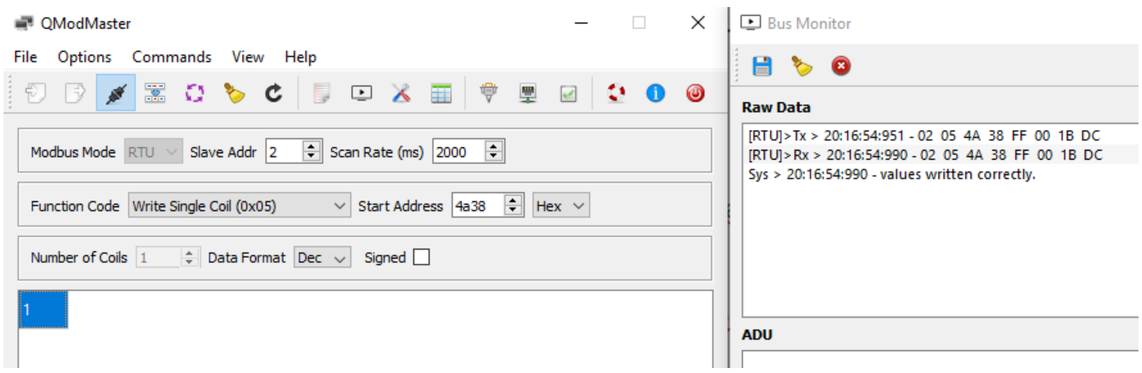
Then you can lower the flag after the open/close to give priority to protection.

Setting up the communication priority Flag

You have to use function 5 to write a 1 on address 0x4A38

If you want to reset the flag you have to repeat same operation: (use function 5 to write a 1 on address 0x4A38)

For instance:



Modbus Mode RTU Slave Addr 2 Scan Rate (ms) 2000

Function Code Write Single Coil (0x05) Start Address 4a38 Hex

Number of Coils 1 Data Format Dec Signed ☐

1

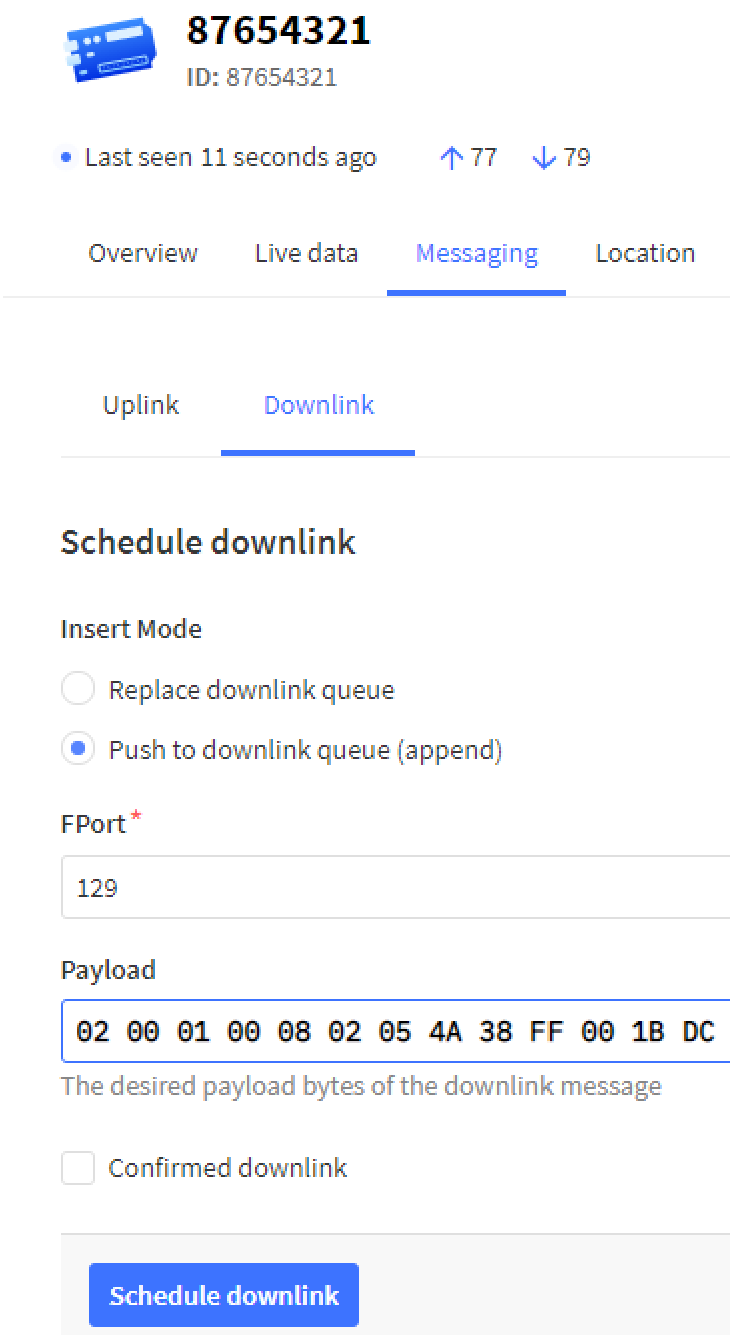
Raw Data

```
[RTU]>Tx > 20:16:54:951 - 02  05  4A  38  FF  00  1B  DC
[RTU]>Rx > 20:16:54:990 - 02  05  4A  38  FF  00  1B  DC
Sys > 20:16:54:990 - values written correctly.
```

ADU

If you want to do this with a downlink, just do it this way:

020001000802054A38FF001BDC



87654321

ID: 87654321

Last seen 11 seconds ago ↑ 77 ↓ 79

Schedule downlink

Insert Mode

- ○ Replace downlink queue
- ● Push to downlink queue (append)

FPort*

129

Payload

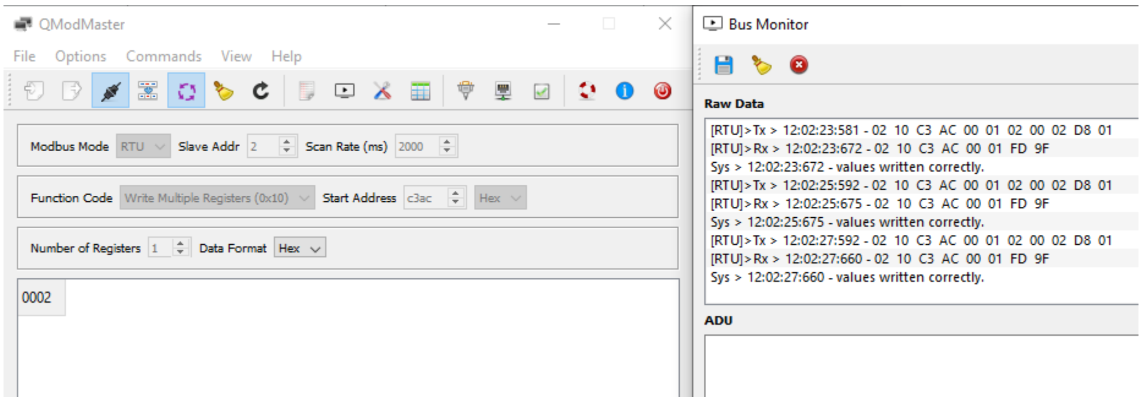
02 00 01 00 08 02 05 4A 38 FF 00 1B DC

The desired payload bytes of the downlink message

☐ Confirmed downlink

After powering REC MAX CVM off this flag is reset.

Closing circuit breaker



QModMaster

Modbus Mode: RTU, Slave Addr: 2, Scan Rate (ms): 2000

Function Code: Write Multiple Registers (0x10), Start Address: c3ac, Hex

Number of Registers: 1, Data Format: Hex

0002

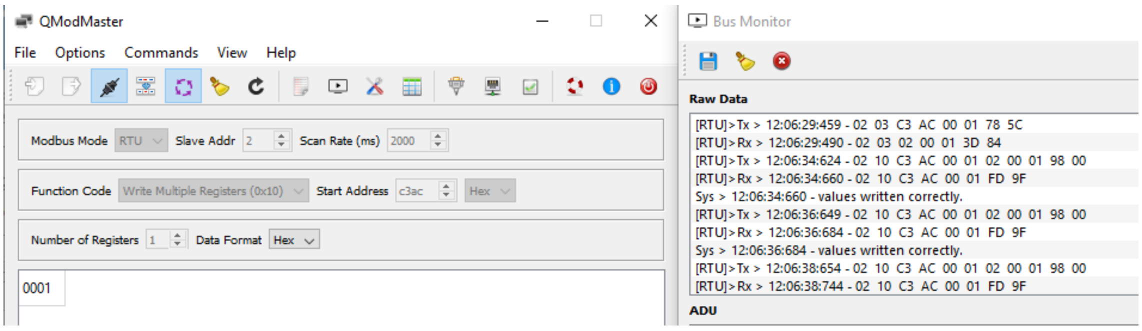
Bus Monitor

Raw Data

```
[RTU]>Tx > 12:02:23:581 - 02  10  C3  AC  00  01  02  00  02  D8  01
[RTU]>Rx > 12:02:23:672 - 02  10  C3  AC  00  01  FD  9F
Sys > 12:02:23:672 - values written correctly.
[RTU]>Tx > 12:02:25:592 - 02  10  C3  AC  00  01  02  00  02  D8  01
[RTU]>Rx > 12:02:25:675 - 02  10  C3  AC  00  01  FD  9F
Sys > 12:02:25:675 - values written correctly.
[RTU]>Tx > 12:02:27:592 - 02  10  C3  AC  00  01  02  00  02  D8  01
[RTU]>Rx > 12:02:27:660 - 02  10  C3  AC  00  01  FD  9F
Sys > 12:02:27:660 - values written correctly.
```

ADU

Opening circuit breaker



QModMaster

Modbus Mode: RTU, Slave Addr: 2, Scan Rate (ms): 2000

Function Code: Write Multiple Registers (0x10), Start Address: c3ac, Hex

Number of Registers: 1, Data Format: Hex

0001

Bus Monitor

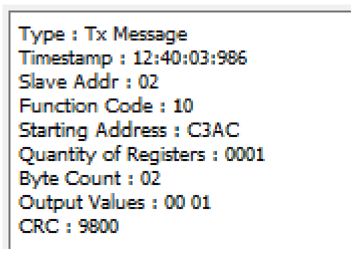
Raw Data

```
[RTU]>Tx > 12:06:29:459 - 02  03  C3  AC  00  01  78  5C
[RTU]>Rx > 12:06:29:490 - 02  03  02  00  01  3D  84
[RTU]>Tx > 12:06:34:624 - 02  10  C3  AC  00  01  02  00  01  98  00
[RTU]>Rx > 12:06:34:660 - 02  10  C3  AC  00  01  FD  9F
Sys > 12:06:34:660 - values written correctly.
[RTU]>Tx > 12:06:36:649 - 02  10  C3  AC  00  01  02  00  01  98  00
[RTU]>Rx > 12:06:36:684 - 02  10  C3  AC  00  01  FD  9F
Sys > 12:06:36:684 - values written correctly.
[RTU]>Tx > 12:06:38:654 - 02  10  C3  AC  00  01  02  00  01  98  00
[RTU]>Rx > 12:06:38:744 - 02  10  C3  AC  00  01  FD  9F
```

ADU

Let's try with a downlink

Opening circuit breaker



```
Type : Tx Message
Timestamp : 12:40:03:986
Slave Addr : 02
Function Code : 10
Starting Address : C3AC
Quantity of Registers : 0001
Byte Count : 02
Output Values : 00 01
CRC : 9800
```

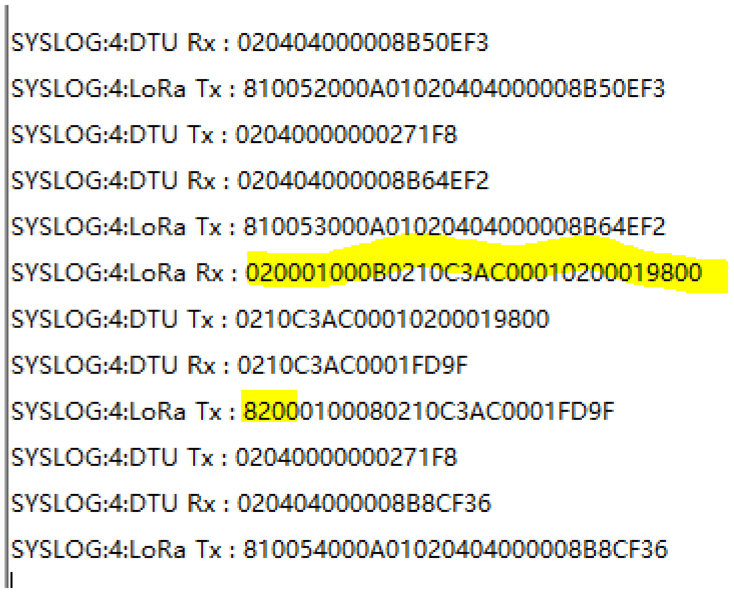
0210C3AC00010200019800

The right downlink would be

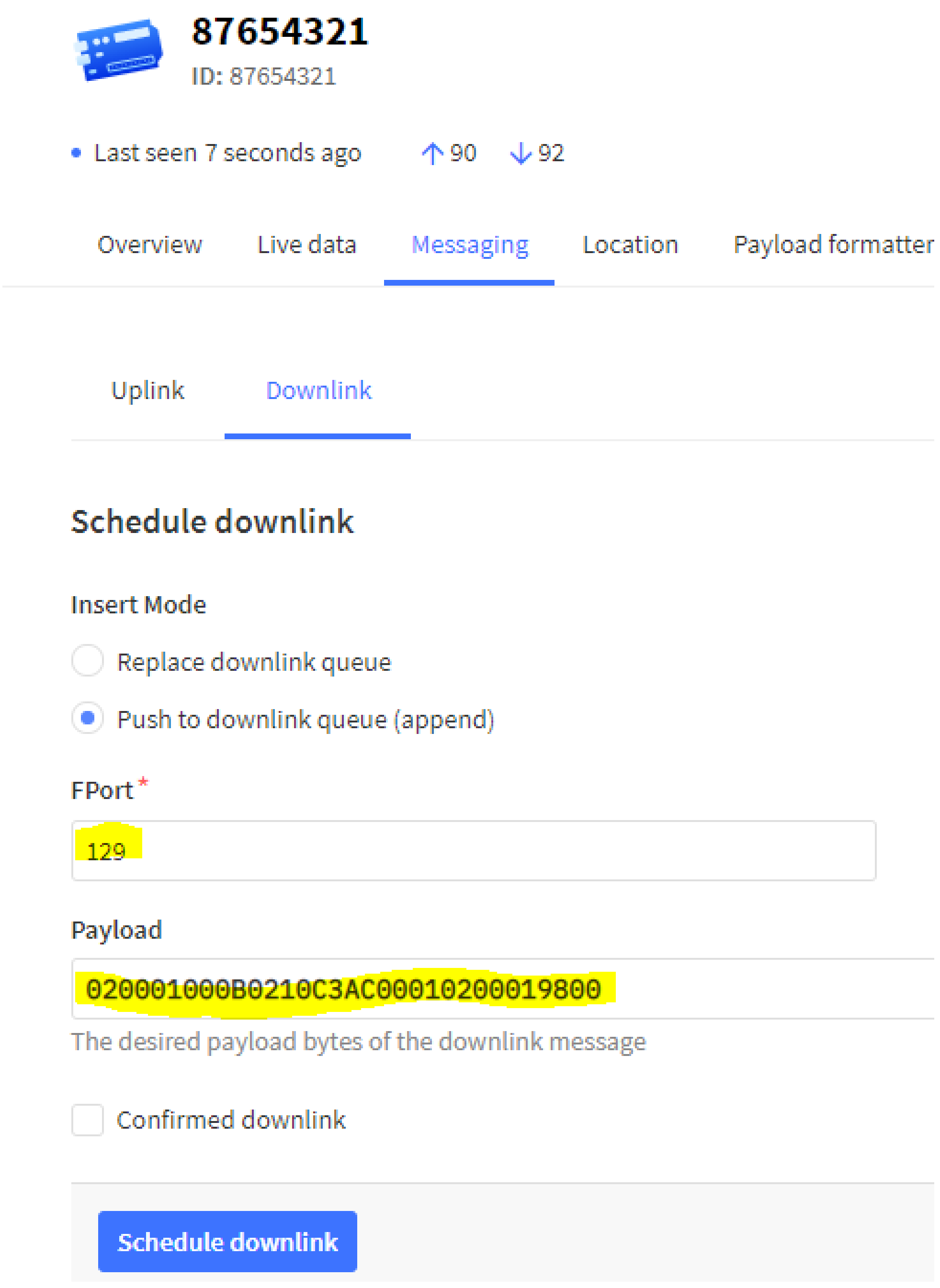
and the number of bytes should be in Hex (11 bytes of Modbus message in Hex is 000B)

020001000B0210C3AC00010200019800

Success



```
SYSLOG:4:DTU Rx : 020404000008B50EF3
SYSLOG:4:LoRa Tx : 810052000A01020404000008B50EF3
SYSLOG:4:DTU Tx : 02040000000271F8
SYSLOG:4:DTU Rx : 020404000008B64EF2
SYSLOG:4:LoRa Tx : 810053000A01020404000008B64EF2
SYSLOG:4:LoRa Rx : 020001000B0210C3AC00010200019800
SYSLOG:4:DTU Tx : 0210C3AC00010200019800
SYSLOG:4:DTU Rx : 0210C3AC0001FD9F
SYSLOG:4:LoRa Tx : 82000100080210C3AC0001FD9F
SYSLOG:4:DTU Tx : 02040000000271F8
SYSLOG:4:DTU Rx : 020404000008B8CF36
SYSLOG:4:LoRa Tx : 810054000A01020404000008B8CF36
```



87654321

ID: 87654321

• Last seen 7 seconds ago ↑ 90 ↓ 92

Schedule downlink

Insert Mode

○ Replace downlink queue

● Push to downlink queue (append)

FPort *

129

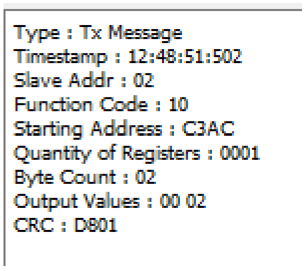
Payload

020001000B0210C3AC000102000198000

The desired payload bytes of the downlink message

☐ Confirmed downlink

Closing circuit breaker



```
Type : Tx Message
Timestamp : 12:48:51:502
Slave Addr : 02
Function Code : 10
Starting Address : C3AC
Quantity of Registers : 0001
Byte Count : 02
Output Values : 00 02
CRC : D801
```

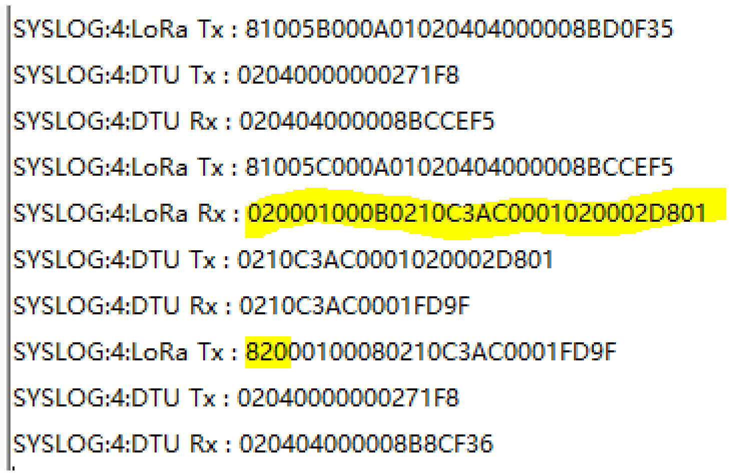
Modbus

0210C3AC0001020002D801

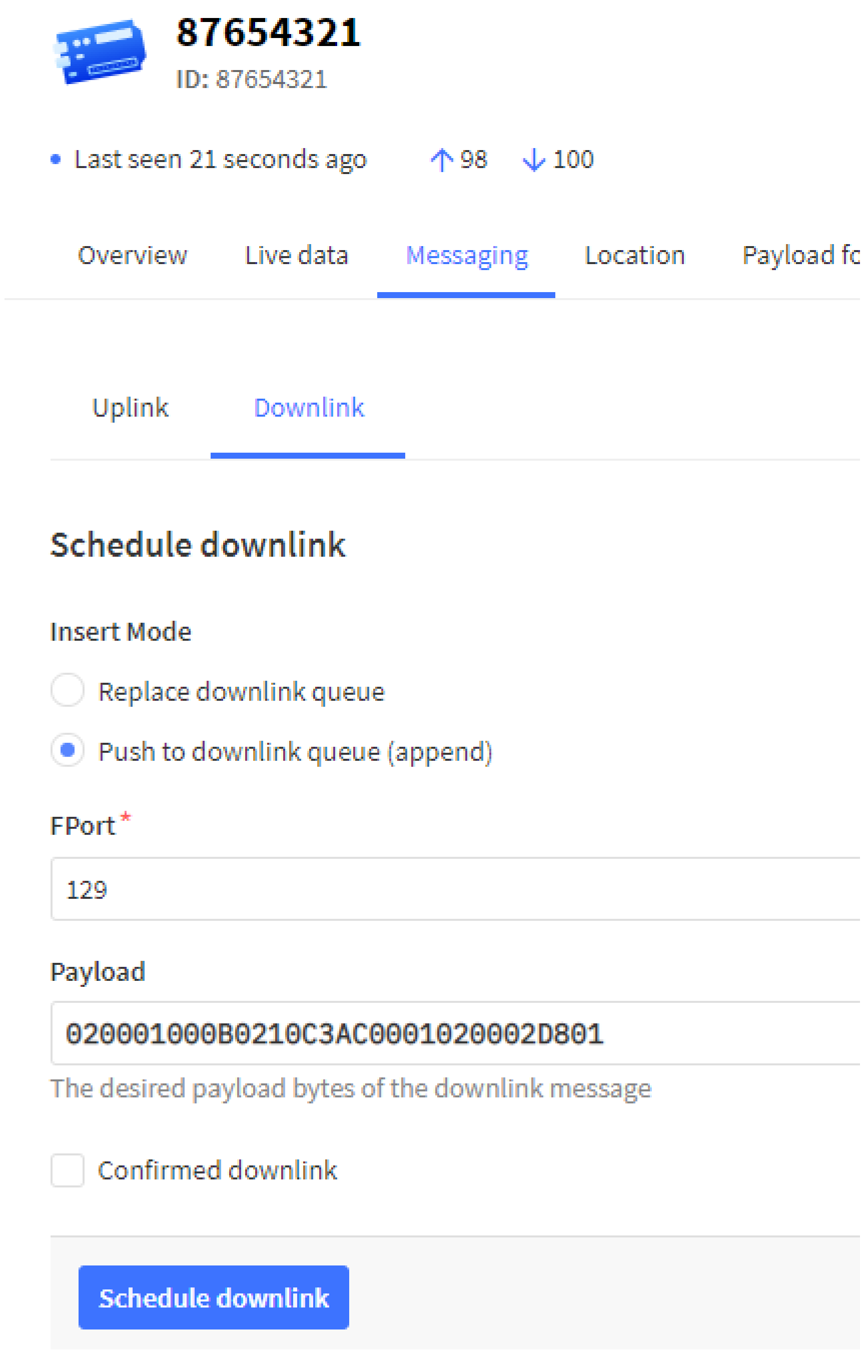
The right downlink would be

020001000B0210C3AC0001020002D801

Success



```
SYSLOG:4:LoRa Tx : 81005B000A01020404000008BD0F35
SYSLOG:4:DTU Tx : 0204000000271F8
SYSLOG:4:DTU Rx : 020404000008BCCEF5
SYSLOG:4:LoRa Tx : 81005C000A01020404000008BCCEF5
SYSLOG:4:LoRa Rx : 020001000B0210C3AC0001020002D801
SYSLOG:4:DTU Tx : 0210C3AC0001020002D801
SYSLOG:4:DTU Rx : 0210C3AC0001FD9F
SYSLOG:4:LoRa Tx : 8200010008 0210C3AC0001FD9F
SYSLOG:4:DTU Tx : 0204000000271F8
SYSLOG:4:DTU Rx : 020404000008B8CF36
```



87654321
ID: 87654321

• Last seen 21 seconds ago ↑ 98 ↓ 100

Overview | Live data | Messaging | Location | Payload fc

Uplink | Downlink

Schedule downlink

Insert Mode

( ) Replace downlink queue
(●) Push to downlink queue (append)

FPort*

129

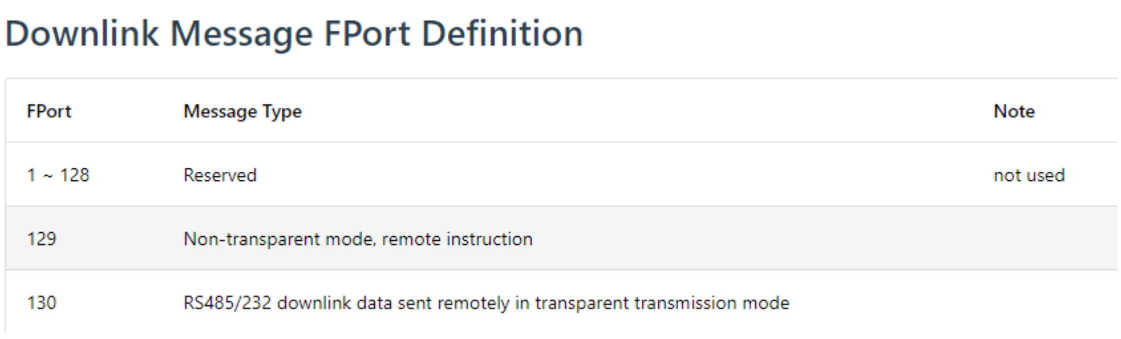
Payload

```
020001000B0210C3AC0001020002D801
```

The desired payload bytes of the downlink message

☐ Confirmed downlink

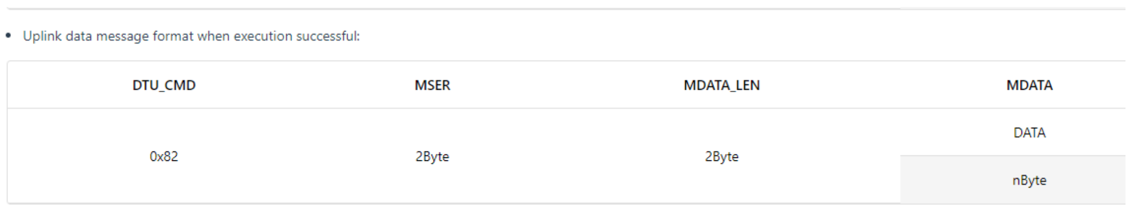
Schedule downlink



Downlink Message FPort Definition

FPort	Message Type	Note
1 ~ 128	Reserved	not used
129	Non-transparent mode, remote instruction	
130	RS485/232 downlink data sent remotely in transparent transmission mode	

We should receive a reponse message like



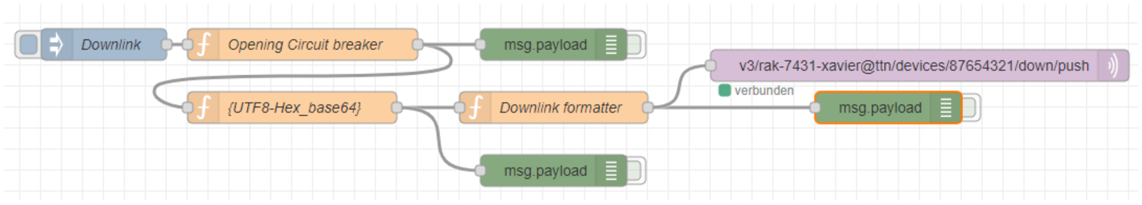
- Uplink data message format when execution successful:

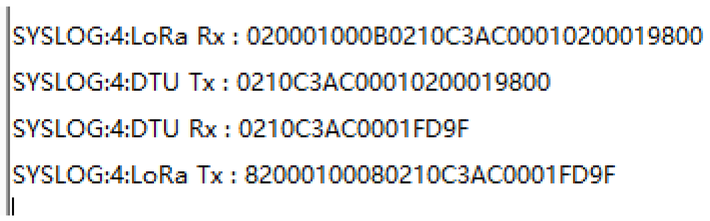
DTU_CMD	MSER	MDATA_LEN	MDATA
			DATA
0x82	2Byte	2Byte	nByte

Let’s try with Node-RED

Downlink with Node-RED

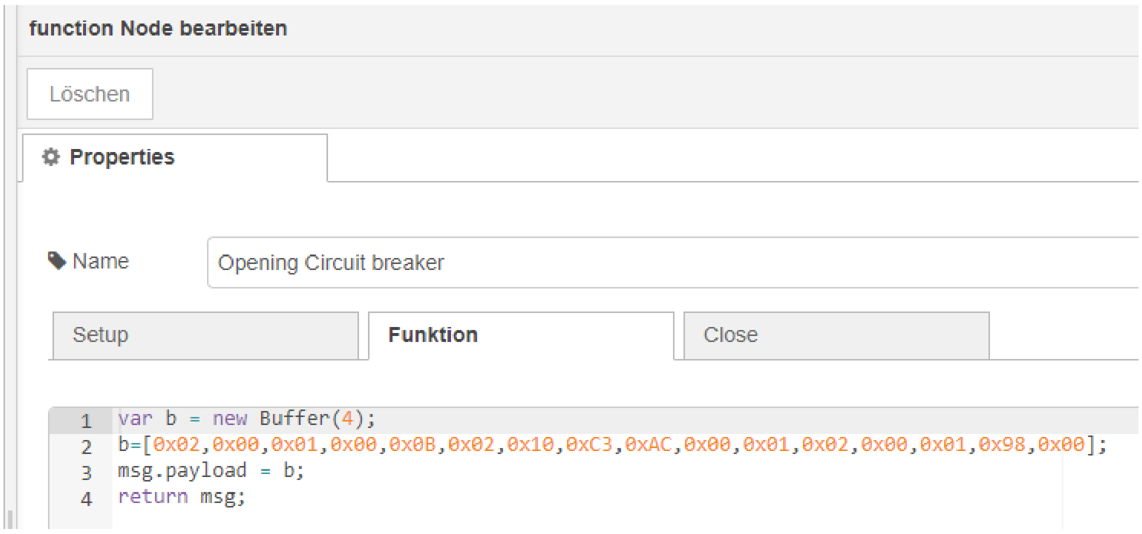
Opening circuit breaker with Node-RED





```
SYSLOG:4:LoRa Rx : 020001000B0210C3AC00010200019800
SYSLOG:4:DTU Tx : 0210C3AC00010200019800
SYSLOG:4:DTU Rx : 0210C3AC0001FD9F
SYSLOG:4:LoRa Tx : 82000100080210C3AC0001FD9F
```

Success


function Node bearbeiten

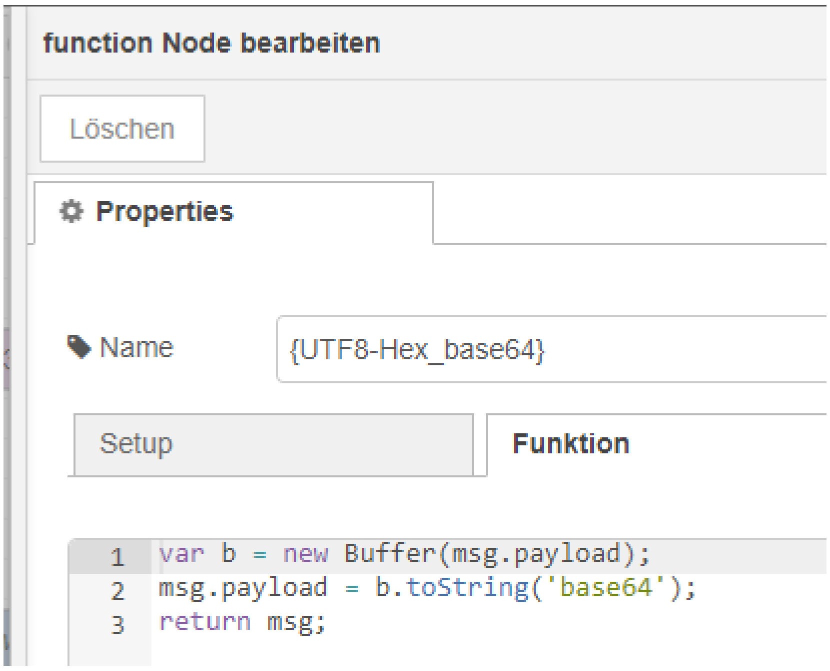
Löschen

Properties

Name: Opening Circuit breaker

Setup | Funktion | Close

```
var b = new Buffer(4);
b=[0x02,0x00,0x01,0x00,0x0B,0x02,0x10,0xC3,0xAC,0x00,0x01,0x02,0x00,0x01,0x98,0x00];
msg.payload = b;
return msg;
```


function Node bearbeiten

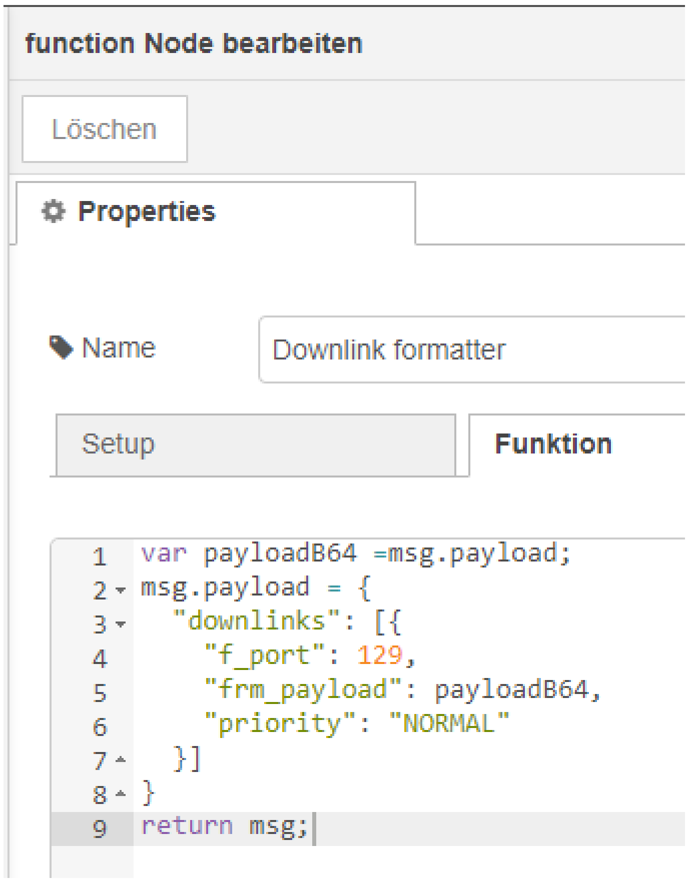
Löschen

Properties

Name: {UTF8-Hex_base64}

Setup | Funktion

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


function Node bearbeiten

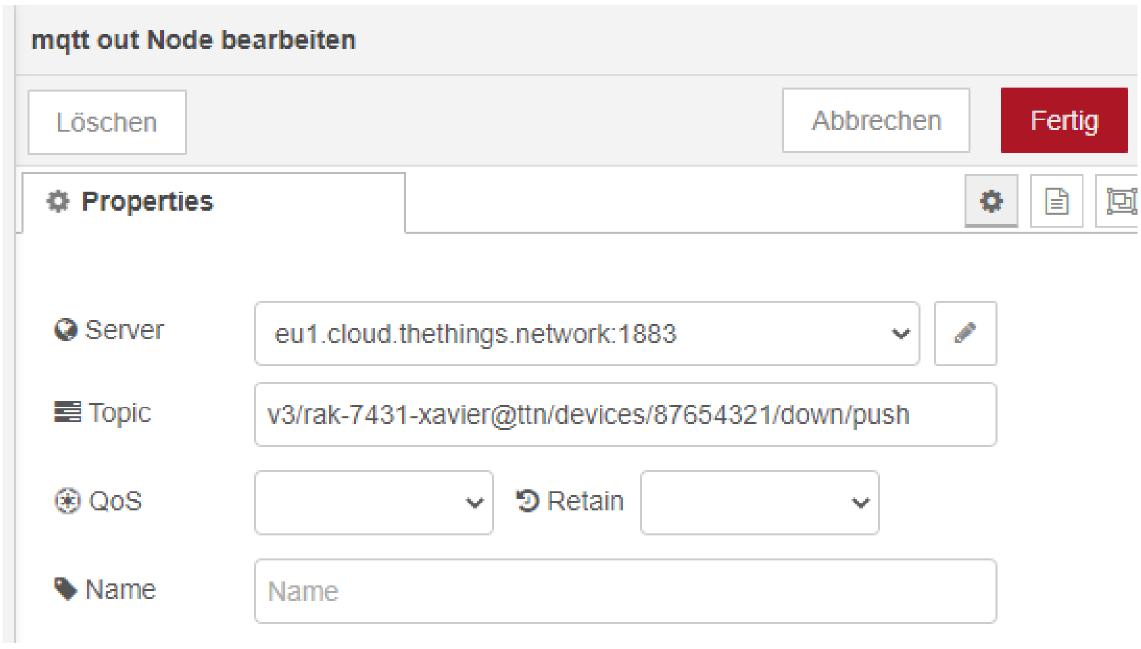
Löschen

Properties

Name: Downlink formatter

Setup | Funktion

```
var payloadB64 =msg.payload;
msg.payload = {
  "downlinks": [{
    "f_port": 129,
    "frm_payload": payloadB64,
    "priority": "NORMAL"
  }]
}
return msg;
```


mqtt out Node bearbeiten

Löschen | Abbrechen | Fertig

Properties

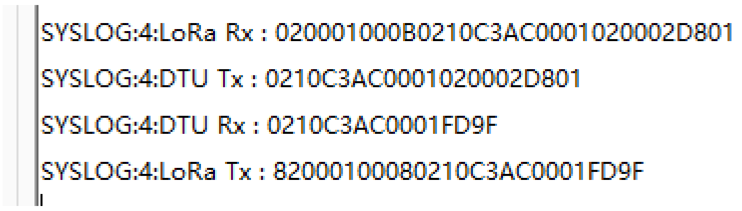
Server: eu1.cloud.thethings.network:1883

Topic: v3/rak-7431-xavier@ttn/devices/87654321/down/push

QoS: | Retain:

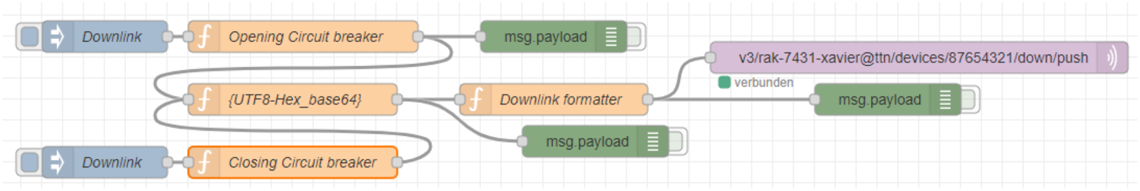
Name: Name

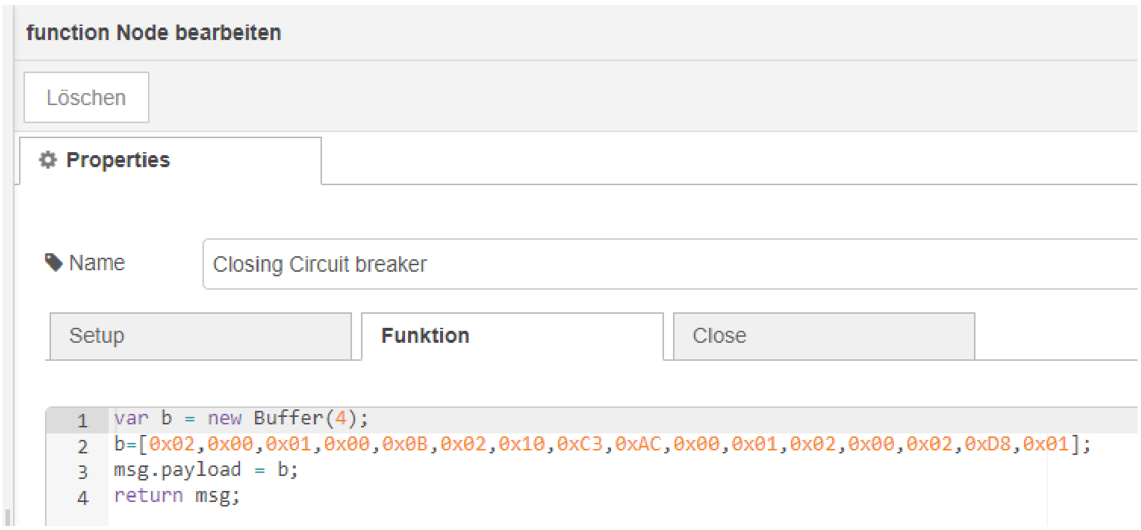
Downlink to close Circuit breaker with Node-RED



```
SYSLOG:4:LoRa Rx : 020001000B0210C3AC0001020002D801
SYSLOG:4:DTU Tx : 0210C3AC0001020002D801
SYSLOG:4:DTU Rx : 0210C3AC0001FD9F
SYSLOG:4:LoRa Tx : 82000100080210C3AC0001FD9F
```

Success





function Node bearbeiten

Löschen

Properties

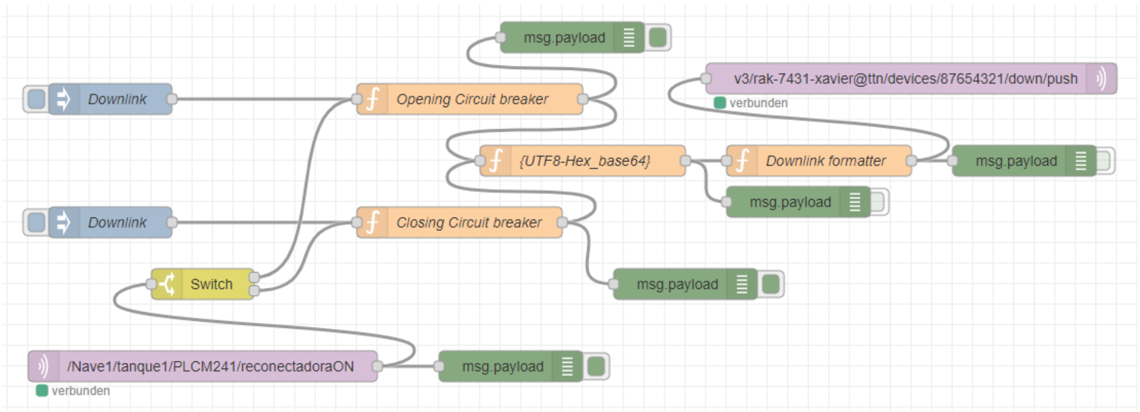
Name: Closing Circuit breaker

Setup | Funktion | Close

```
var b = new Buffer(4);
b=[0x02,0x00,0x01,0x00,0x0B,0x02,0x10,0xC3,0xAC,0x00,0x01,0x02,0x00,0x02,0xD8,0x01];
msg.payload = b;
return msg;
```

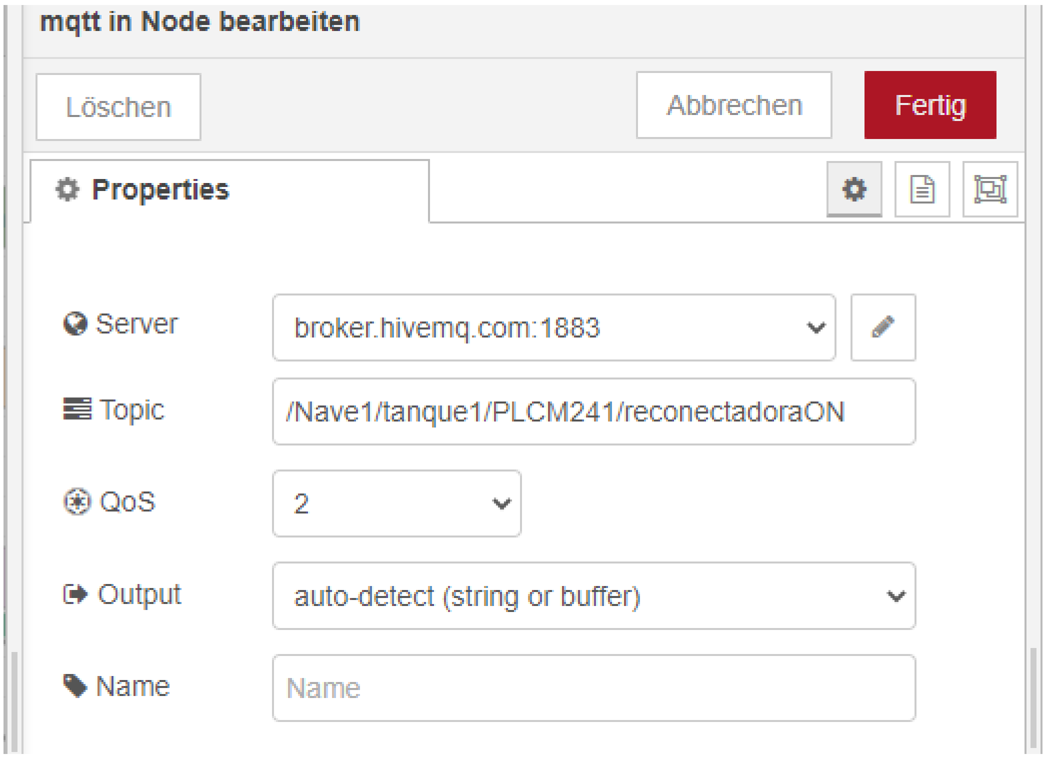
Now Let’s use the Mobile phone to open and close te circuit

Opening and closing with MQTT Thru IoT OnOff

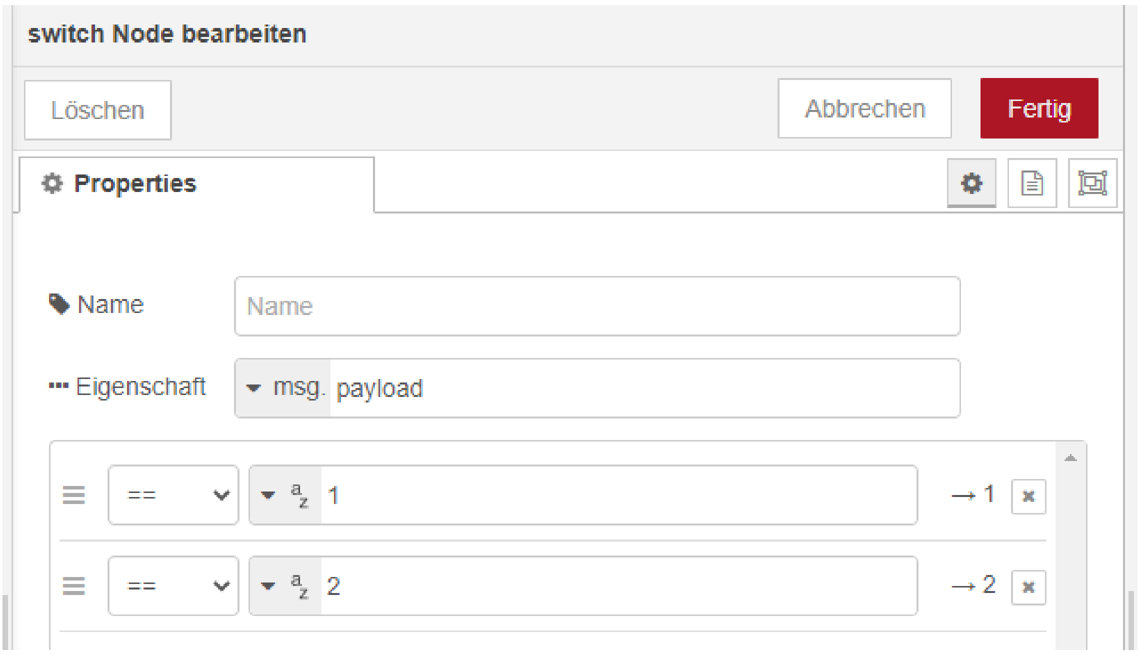


You can find the code here:

https://github.com/xavierflorensa/RAK-7431-CIRCUTOR-RECmaxCVM/blob/main/node-red%20flow%20rak%207431%20RECMaxCVM%20ttn%20IoT%20ON%20OFF.txt

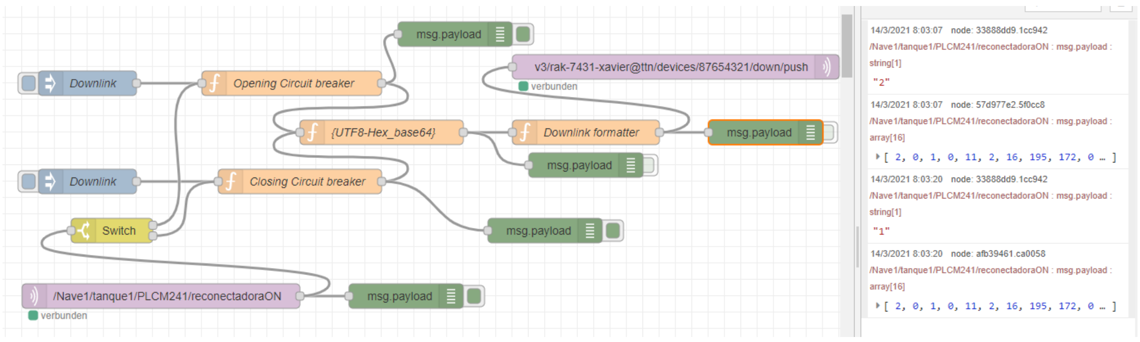

mqtt in Node bearbeiten

Server: broker.hivemq.com:1883
Topic: /Nave1/tanque1/PLCM241/reconectadoraON
QoS: 2
Output: auto-detect (string or buffer)
Name: Name

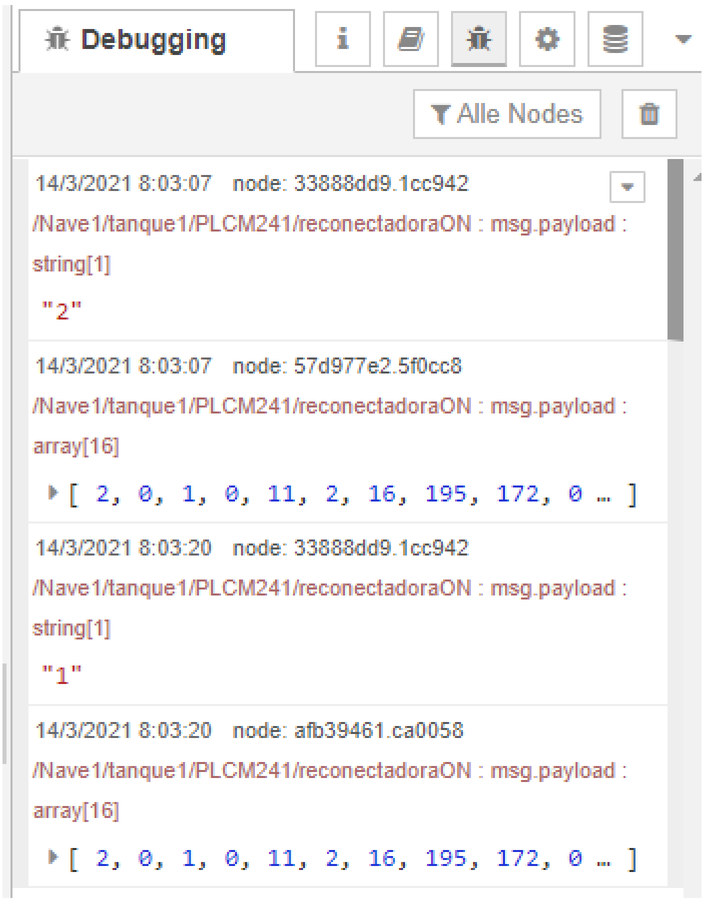

switch Node bearbeiten

Name: Name
Eigenschaft: msg. payload

==	1	→ 1
==	2	→ 2

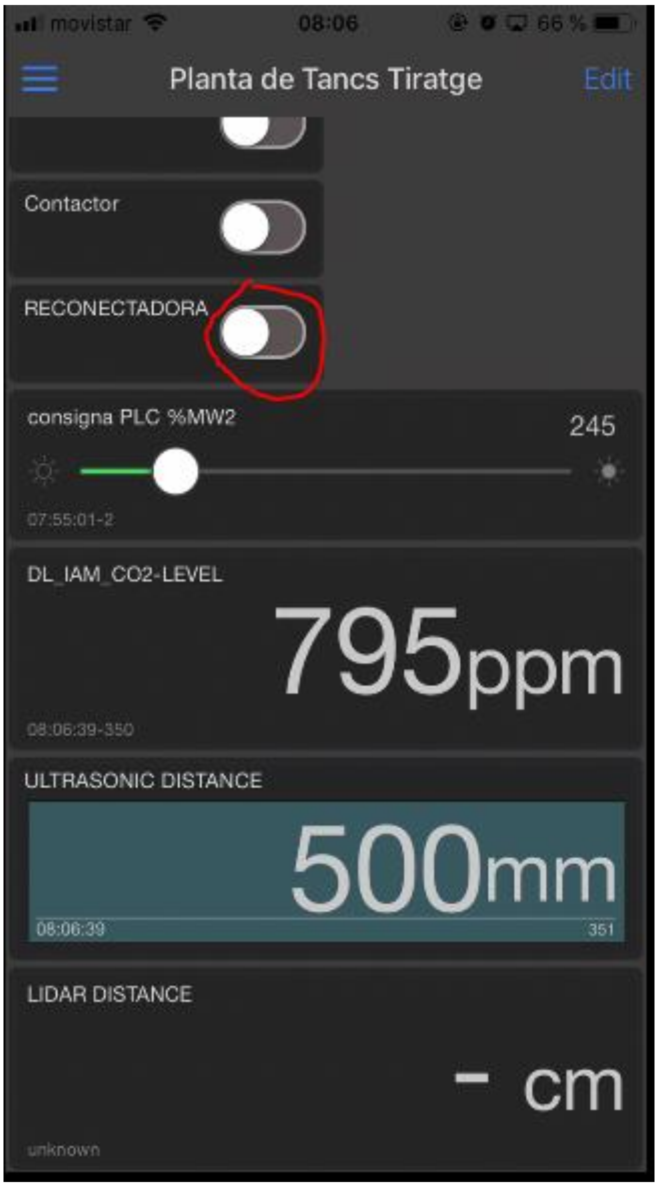


```
14/3/2021 8:03:07   node: 33888dd9.1cc942
/Nave1/tanque1/PLCM241/reconectadoraON : msg.payload :
string[1]
 "2"
14/3/2021 8:03:07   node: 57d977e2.5f0cc8
/Nave1/tanque1/PLCM241/reconectadoraON : msg.payload :
array[16]
 ▸[ 2, 0, 1, 0, 11, 2, 16, 195, 172, 0 … ]
14/3/2021 8:03:20   node: 33888dd9.1cc942
/Nave1/tanque1/PLCM241/reconectadoraON : msg.payload :
string[1]
 "1"
14/3/2021 8:03:20   node: afb39461.ca0058
/Nave1/tanque1/PLCM241/reconectadoraON : msg.payload :
array[16]
 ▸[ 2, 0, 1, 0, 11, 2, 16, 195, 172, 0 … ]
```

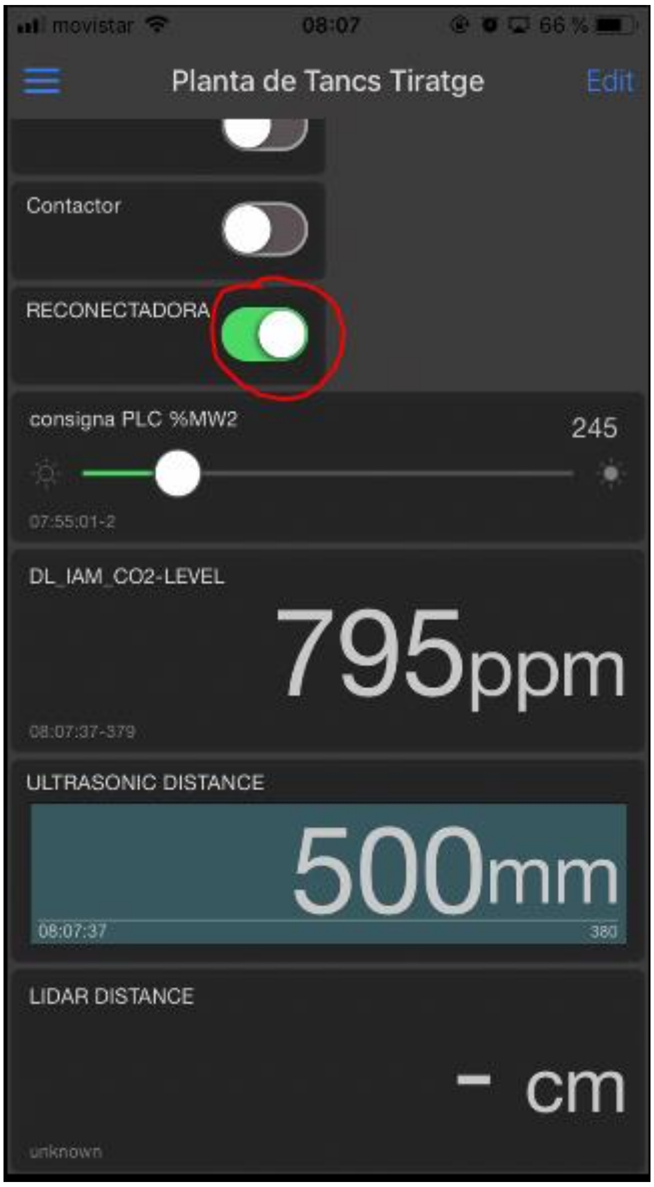


```
14/3/2021 8:03:07   node: 33888dd9.1cc942
/Nave1/tanque1/PLCM241/reconectadoraON : msg.payload :
string[1]
 "2"

14/3/2021 8:03:07   node: 57d977e2.5f0cc8
/Nave1/tanque1/PLCM241/reconectadoraON : msg.payload :
array[16]
 ▸[ 2, 0, 1, 0, 11, 2, 16, 195, 172, 0 … ]

14/3/2021 8:03:20   node: 33888dd9.1cc942
/Nave1/tanque1/PLCM241/reconectadoraON : msg.payload :
string[1]
 "1"

14/3/2021 8:03:20   node: afb39461.ca0058
/Nave1/tanque1/PLCM241/reconectadoraON : msg.payload :
array[16]
 ▸[ 2, 0, 1, 0, 11, 2, 16, 195, 172, 0 … ]
```

To open circuit breaker



To close circuit breaker



Contactor

RECONECTADORA

consigna PLC %MW2 245

07:55:01-2

DL_IAM_CO2-LEVEL

795ppm

08:07:37-379

ULTRASONIC DISTANCE

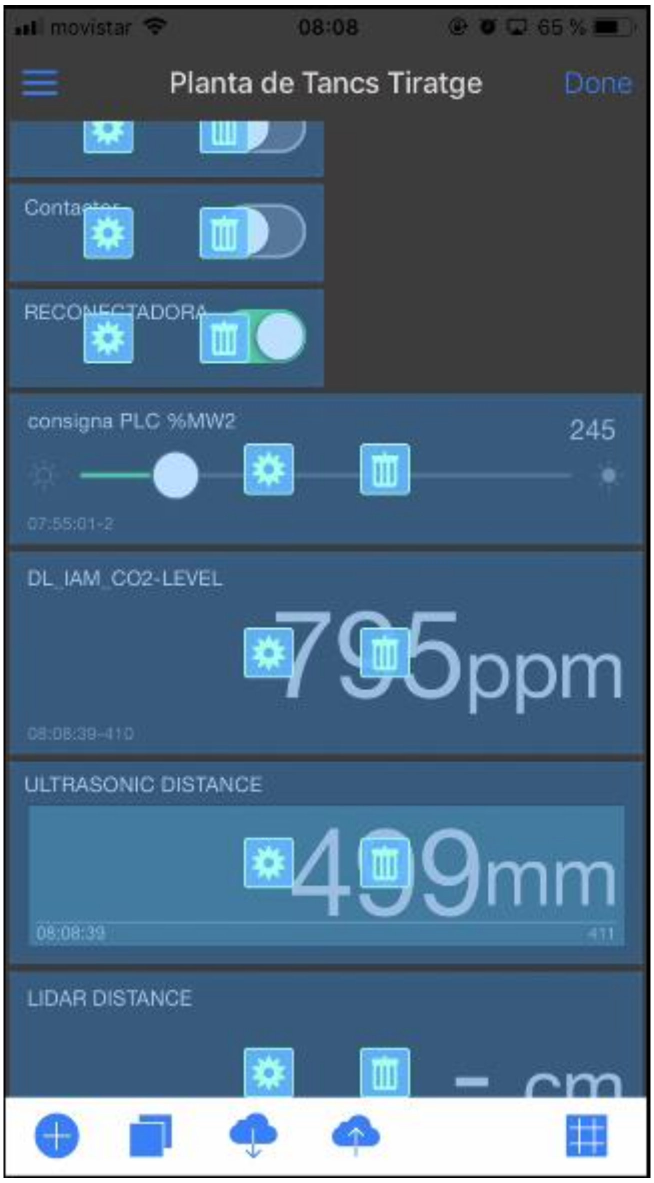
500mm

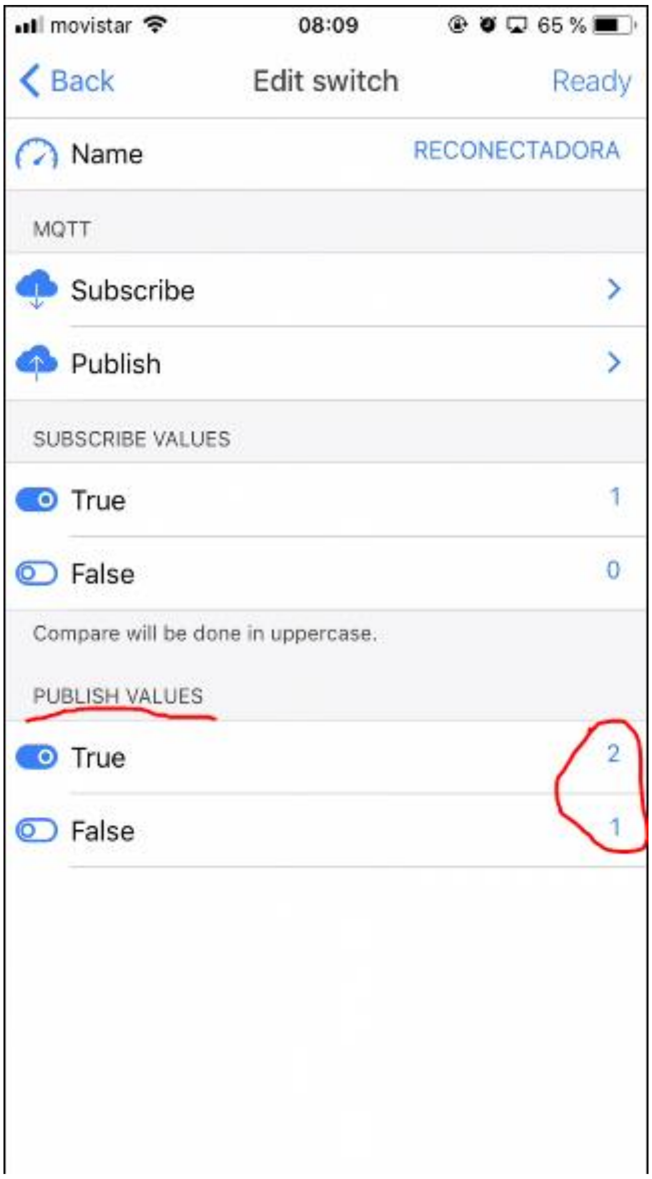
08:07:37 380

LIDAR DISTANCE

- cm

unknown


movistar
08:08
65 %
Planta de Tancs Tiratge
Done
Contactor
RECONECTADORA
consigna PLC %MW2
245
07:55:01-2
DL_IAM_CO2-LEVEL
795ppm
08:08:39-410
ULTRASONIC DISTANCE
499mm
08:08:39
411
LIDAR DISTANCE
– cm



movistar 08:09 65 %

Back Edit switch Ready

Name RECONECTADORA

MQTT

Subscribe

Publish

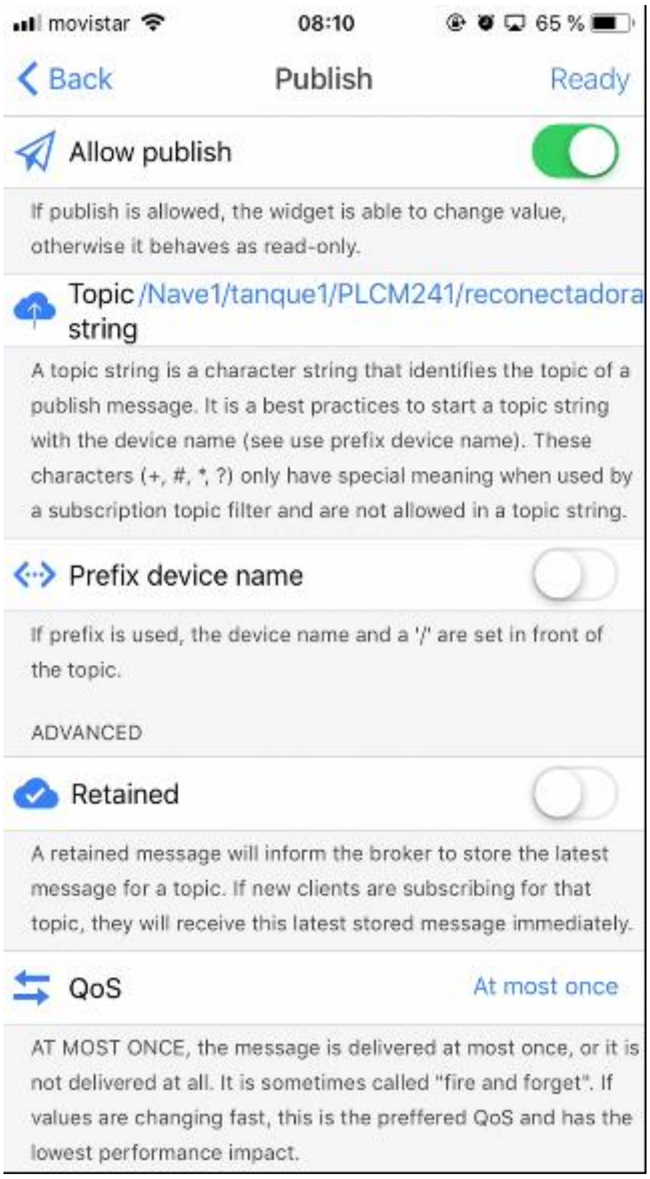
SUBSCRIBE VALUES

True	1
False	0

Compare will be done in uppercase.

PUBLISH VALUES

True	2
False	1



Allow publish

If publish is allowed, the widget is able to change value, otherwise it behaves as read-only.

Topic string /Nave1/tanque1/PLCM241/reconectadora

A topic string is a character string that identifies the topic of a publish message. It is a best practices to start a topic string with the device name (see use prefix device name). These characters (+, #, *, ?) only have special meaning when used by a subscription topic filter and are not allowed in a topic string.

Prefix device name

If prefix is used, the device name and a '/' are set in front of the topic.

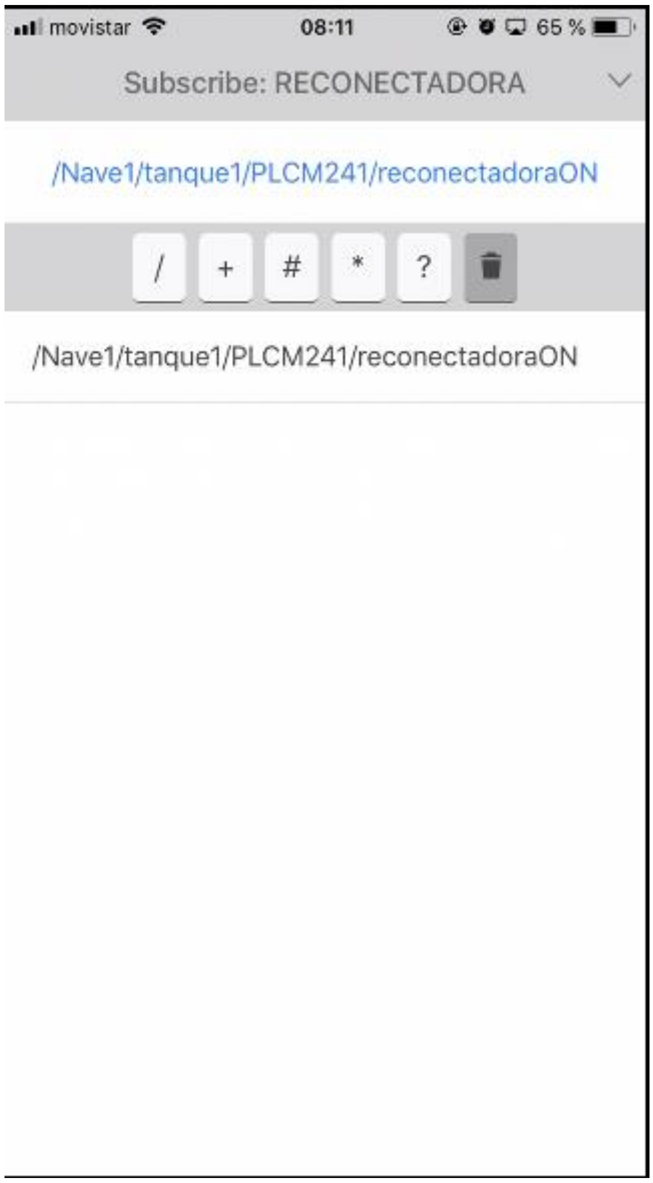
ADVANCED

Retained

A retained message will inform the broker to store the latest message for a topic. If new clients are subscribing for that topic, they will receive this latest stored message immediately.

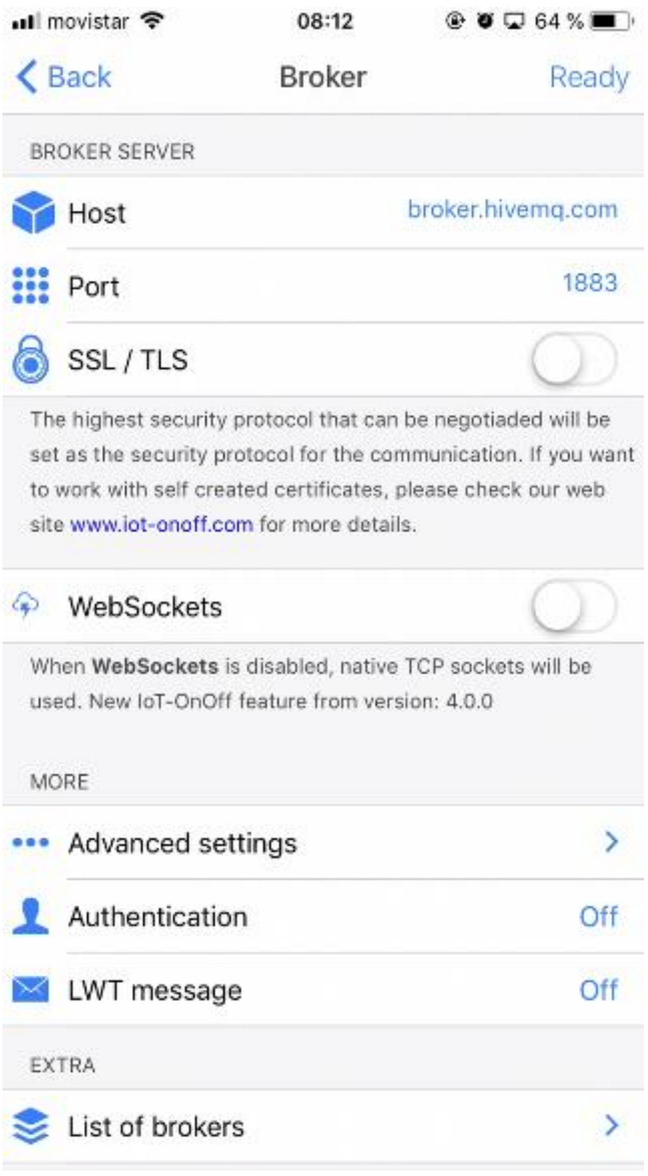
QoS At most once

AT MOST ONCE, the message is delivered at most once, or it is not delivered at all. It is sometimes called "fire and forget". If values are changing fast, this is the preffered QoS and has the lowest performance impact.



/Nave1/tanque1/PLCM241/reconectadoraON

/Nave1/tanque1/PLCM241/reconectadoraON


movistar 08:12 64 %

Back **Broker** Ready

BROKER SERVER

Host broker.hivemq.com

Port 1883

SSL / TLS

The highest security protocol that can be negotiaded will be set as the security protocol for the communication. If you want to work with self created certificates, please check our web site www.iot-onoff.com for more details.

WebSockets

When **WebSockets** is disabled, native TCP sockets will be used. New IoT-OnOff feature from version: 4.0.0

MORE

Advanced settings

Authentication Off

LWT message Off

EXTRA

List of brokers

As you can see on this video

[REC MAX CVM RAK 7431 to trip circuit breaker with TTN]