

# WoMaster

WR322-EC and LR140  
Quick start & OTA integration

Brian



# RS232 and RS485 connection on LR140



RS485A

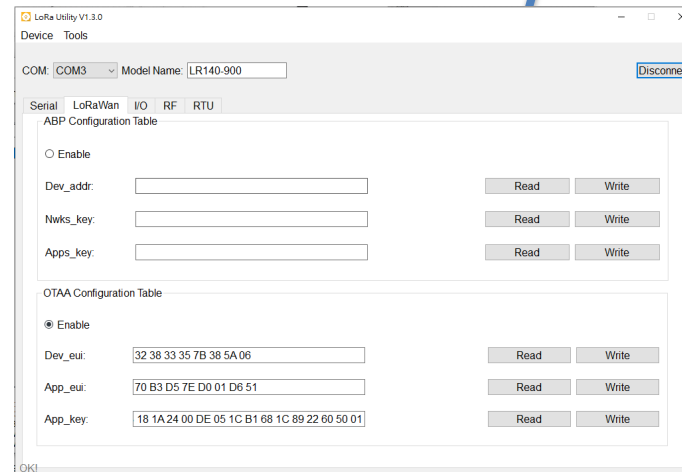
RS485B

USB-To-RS232

ES104



LoRa Windows Utility



# RTU settings on LR140

LoRa Utility V1.3.0  
Device Tools

COM: COM3 Model Name: LR140-900 [Disconnect]

Serial LoRaWAN I/O RF RTU

**RTU Parameters**

RTU Name : ES104  
Slave ID: 1  
Address(PLC): 501  
Length: 5 [Add]

Timestamp: 2000-01-03 08:04:43 週一 [Check] [Set]

Countdown Time: 1 min [Check] [Set]  
 Schedule Time:

01:00  02:00  03:00  04:00  05:00  06:00  
 07:00  08:00  09:00  10:00  11:00  12:00  
 13:00  14:00  15:00  16:00  17:00  18:00  
 19:00  20:00  21:00  22:00  23:00  00:00

Interval: 0 s [Check] [Set]

[Clear] [Reload]

RTU Name	Modbus Slave ID	Start Address	Length
ES104	1	501	5

RTU -> RTU Parameters -> Enter Sensor Slave ID, Address , Length -> Add

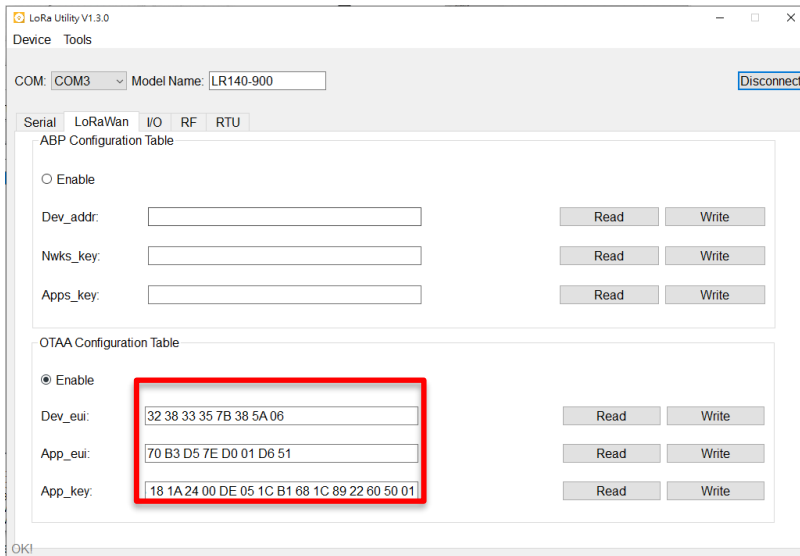
Example: ES104, Slave ID :1, Address(PLC):501 , Length: 5

Select Countdown Time -> 1 minutes -> set (set LR140 every 1 minute to send RTU data to LoRaWAN)

ES104 Modbus Register Information

Default slave ID	Parameters Function	Register Add. (HEX / DEC)	PLC Add. (Modbus Add.)	Note
	Device ID Storage Add.	07D0H / 2000	2001 (42001)	R/W , Range:1~254
	Serial Baud Rate Add.	07D1H / 2001	2002 (42002)	R/W , Default: 2 2(9600), 0(2400), 1(4800)
1	Humidity	01F4H / 500	501(40501)	R/O, Real Value = Read Value /10
1	Temperature	01F5H / 501	502(40502)	R/O, Real Value = Read Value /10
1	PM2.5	01F7H / 503	504(40504)	R/O, Real Value = Read Value
1	PM10	01F8H / 504	505 (40505)	R/O, Real Value = Read Value

# LoRaWAN settings on LR140



Get LoRaWAN node dev\_eui and appkey for OTAA join  
Select OTAA Enable

Join to LoRaWAN Server by OTAA (over the air)



Disconnect USB and repowering, the LR140 will auto join  
LoRaWAN gateway and auto send LoRaWAN packet in a period

# Add LR140 Device Info to WR322-EC

- <http://host:8088/index.asp>
- LoRaWAN Server-> Application-> Devices -> add Device
- Paste the DevEUI -> Submit
- LoRaWAN Server-> Application-> Devices -> Activation, Select Join mode to OTAA, paste Application key -> Submit

Home > LoRaWAN Server > Application

General Application

### Device List

Name	Device EUI	Join Mode	Description	Profile	Last Seen At	Actions
LR140	323833357b385a06	ABP	LR140	Default	null	<a href="#">Edit</a> <a href="#">Activation</a> <a href="#">Del</a>

**Add Device**

Home > LoRaWAN Server > Application

General Application

### Device Configuration

Device Name: LR140

Device Description: LR140

Device Profiles: Default

Device EUI: 32 38 33 35 7b 38 5a 06 **Generate**

**Submit**

Home > LoRaWAN Server > Application

General Application

### Device Activation

Device EUI: 32 38 33 35 7b 38 5a 06

Join Mode: **OTAA**

Device Address: **Generate**

Network Session Key: **Generate**

Application Session Key: **Generate**

Application Key: a8 18 1a 24 00 de 05 1c b1 68 1c 89 22 60 50 01 **Generate**

**Submit**

LoRa Utility V1.3.0

Device Tools

COM: COM3 Model Name: LR140-900 **Disconnect**

Serial LoRaWan I/O RF RTU

#### ABP Configuration Table

Enable

Dev\_addr:  **Read** **Write**

Nwks\_key:  **Read** **Write**

Apps\_key:  **Read** **Write**

#### OTAA Configuration Table

Enable

Dev\_eui: **32 38 33 35 7B 38 5A 06** **Read** **Write**

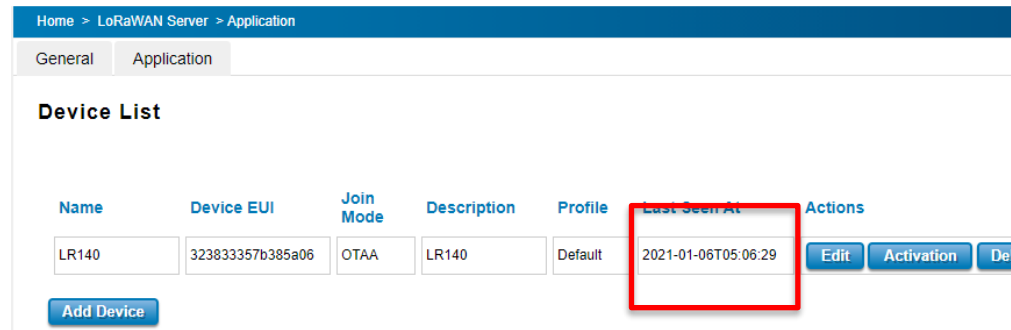
App\_eui: 70 B3 D5 7E D0 01 D6 51 **Read** **Write**

App\_key: **18 1A 24 00 DE 05 1C B1 68 1C 89 22 60 50 01** **Read** **Write**

OK!

# Display LR140 data on WR322-EC

- Connect to <http://host:8088/index.asp> -> Application/Device List/Last seen at
- Connect to <http://host:8080> (username/passwd: admin/admin)-> Gateway -> WR322-EC -> Live LoRaWAN Frames. You can see WR322-EC received data through LoraWan



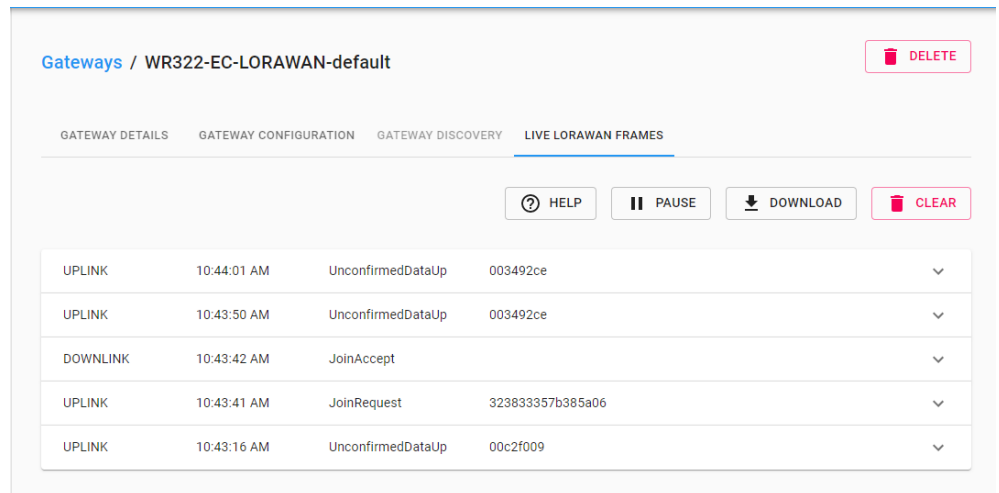
Home > LoRaWAN Server > Application

General Application

### Device List

Name	Device EUI	Join Mode	Description	Profile	Last Seen At	Actions
LR140	323833357b385a06	OTAA	LR140	Default	2021-01-06T05:06:29	Edit Activation Del

Add Device



Gateways / WR322-EC-LORAWAN-default

DELETE

GATEWAY DETAILS GATEWAY CONFIGURATION GATEWAY DISCOVERY LIVE LORAWAN FRAMES

HELP PAUSE DOWNLOAD CLEAR

UPLINK	10:44:01 AM	UnconfirmedDataUp	003492ce	▼
UPLINK	10:43:50 AM	UnconfirmedDataUp	003492ce	▼
DOWNLINK	10:43:42 AM	JoinAccept		▼
UPLINK	10:43:41 AM	JoinRequest	323833357b385a06	▼
UPLINK	10:43:16 AM	UnconfirmedDataUp	00c2f009	▼

# Open Node-RED on WR322-EC

- Open the node-red on your browser: `http://<EC Series IP Address>:1880`
- For example:  
`http://192.168.1.1:1880`
- And you will directly see the default flow for LoRaWAN LR140 and ES104/ES101-WS sensor

The screenshot displays the Node-RED web interface. The main workspace shows a flow for 'LoRaWAN LR140'. The flow starts with an 'MQTT Input' node (application/5/device/323833357b385a06/#) which is connected to an 'ES104 Sensor Decoder' node. This decoder node is connected to a 'msg payload' node. Below this, there is an 'ES101-WS Sensor Decoder' node connected to another 'msg payload' node. To the right, there is an 'OTA MQTTS Client' node (v1/devices/me/telemetry) which is connected to a 'msg payload' node. The interface includes a left sidebar with 'input' and 'output' node categories, a top search bar, and a right sidebar with 'info' and 'flow description' panels. The 'info' panel shows the flow name 'LoRaWAN LR140' and status 'Enabled'. The 'flow description' panel is currently empty.

# Edit MQTT input for LoRaWAN server

- Edit mqtt node
- Edit Topic  
"application/[ApplicationID]/device/[DevEUI]/#" to get device data
- Example:  
application/1/device/323833357b385a06/#
- Click "Deploy" to apply settings

## Application List

ID	Name	Description
1	WR322-EC-LoRaWAN	WR322-EC-LoRaWAN

Add Application

## Device List

Name	Device EUI	Join Mode	Description	Profile	Last Seen At
LR140	323833357b385a06	OTAA	LR140	Default	2021-01-06T05:15:29

Add Device

Flow 1 LoRaWAN LR140

MQTT Input

application/1/device/323833357b385a06/#

connected

msg

f

### Edit mqtt in node

Delete Cancel Done

node properties

Server localhost:1883

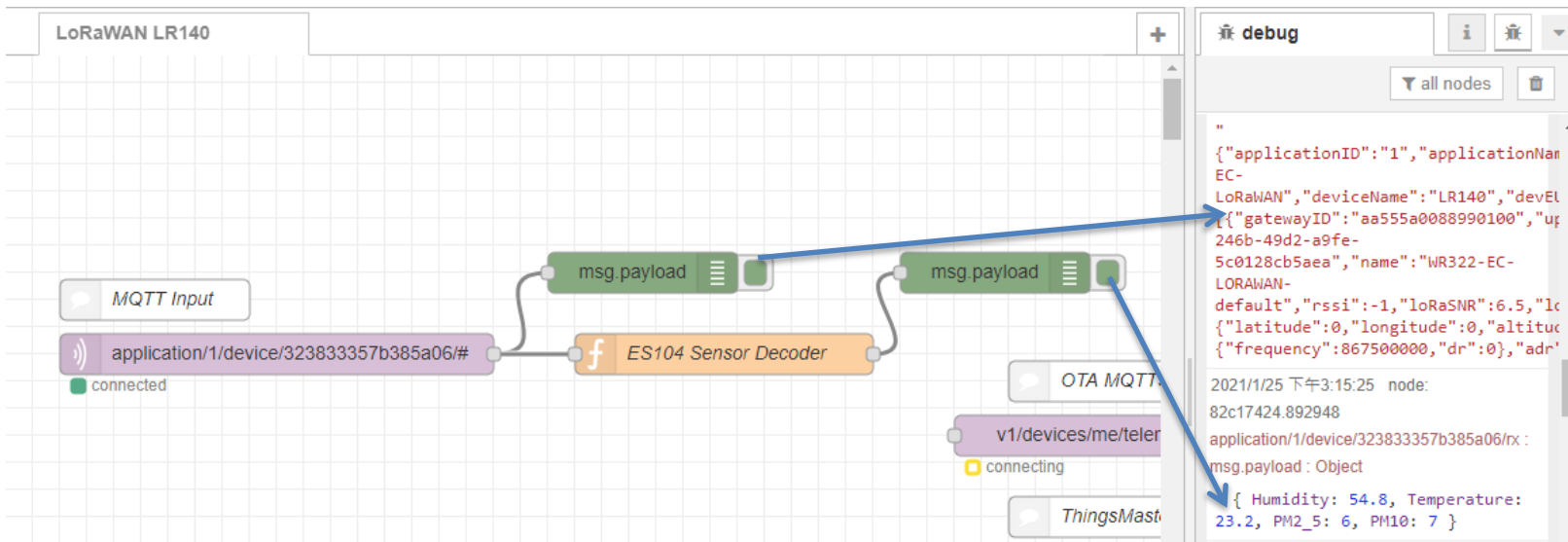
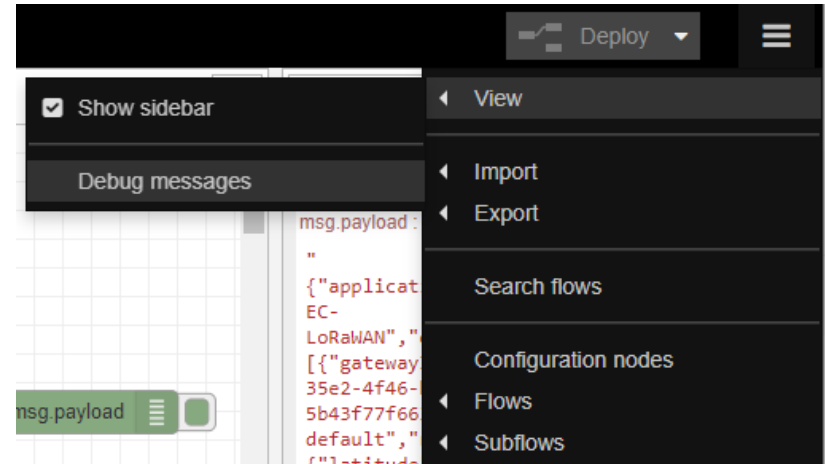
Topic application/1/device/323833357b385a06/#

QoS 2

Name Name

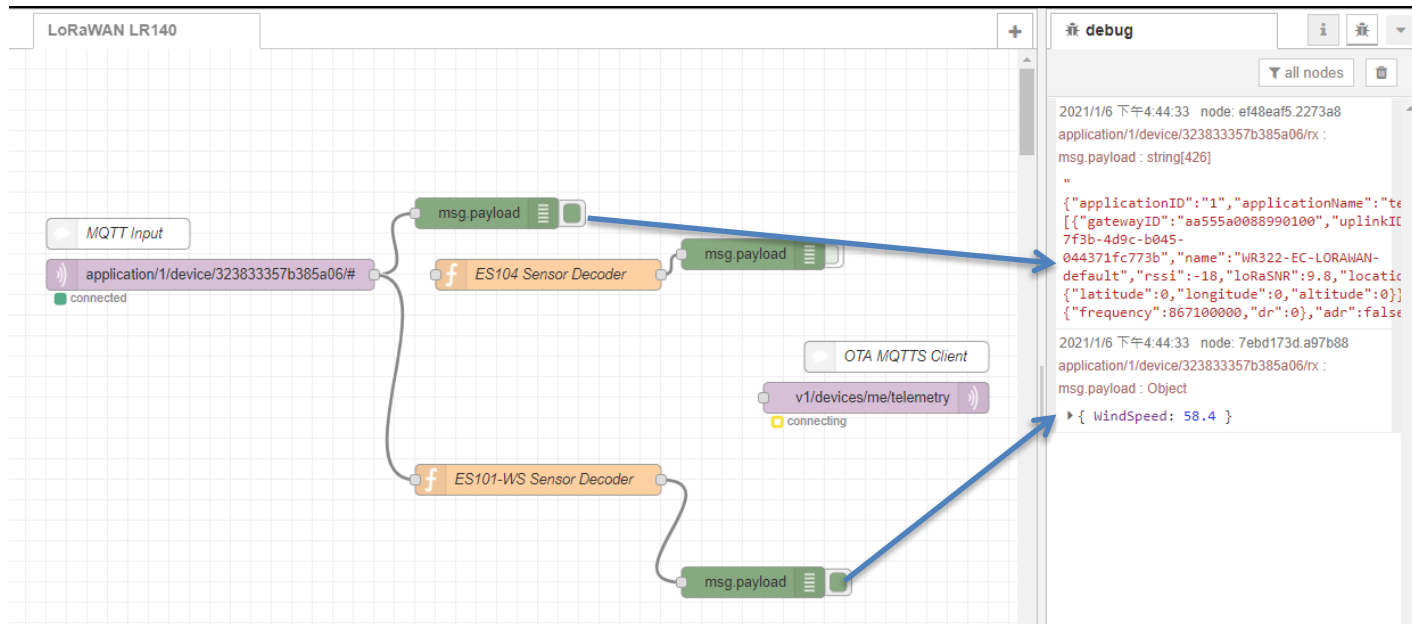
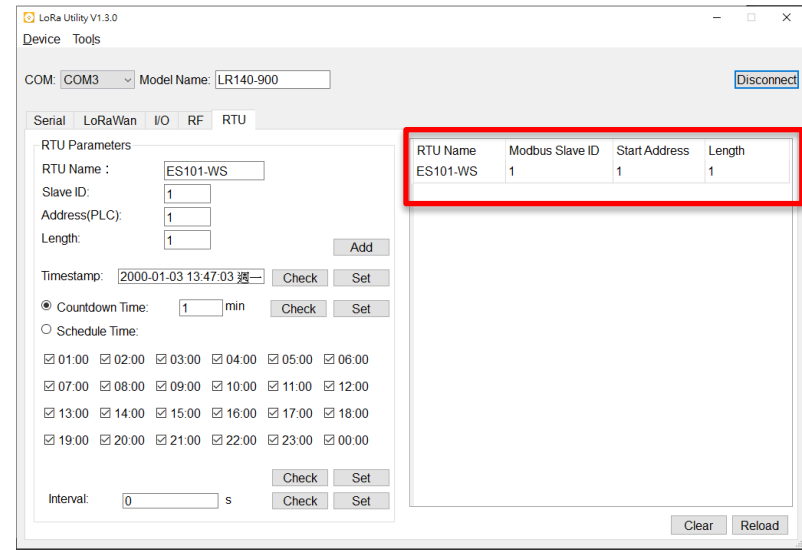
# Display Sensor ES104 data by Node-RED

- View -> Debug Message
- It will display LR140 mqtt data and ES104 decoded data
- The ES104 decoded data will be  
{ "Humidity":58.8,"Temperature":21,"PM2\_5":4,"PM10":4}



# Display Sensor ES101-WS data by Node-RED

- Replace Sensor ES140 to ES101-WS
- Disconnect MQTT Input and ES104 node
- Connect MQTT Input and ES101-WS node
- The ES101-WS decoded data will be {"WindSpeed":58.4}



# LR140 payload format and decoder

LR140 upstream payload Format:

<b>0x0485</b> (2 byte)	<b>0x0485</b> (2byte)	<b>Time</b> (7 byte)	<b>Sensor Name</b> (8 byte)
---------------------------	--------------------------	-------------------------	--------------------------------

<b>Modbus ID</b> (1 byte)	<b>Reg addr</b> (2 byte)	<b>Reg num</b> (1 byte)	<b>Reg val(Reg num*2 byte)</b>	....	<b>Reg val(Reg num*2 byte)</b>	<b>Oxaa</b> (1 byte)
------------------------------	-----------------------------	----------------------------	--------------------------------	------	--------------------------------	-------------------------

ES104 Modbus Register Information				
Default slave ID	Parameters Function	Register Add. (HEX / DEC)	PLC Add. (Modbus Add.)	Note
	Device ID Storage Add.	07D0H/2000	2001(42001)	R/W , Range:1~254
	Serial Baud Rate Add.	07D1H/2001	2002(42002)	R/W , Default 2 2(9600), 0(2400), 1(4800)
1	Humidity	01F4H/ 500	501(40501)	R/O, Real Value = Read Value /10
1	Temperature	01F5H/ 501	502(40502)	R/O, Real Value = Read Value /10
1	PM2.5	01F7H/ 503	504(40504)	R/O, Real Value = Read Value
1	PM10	01F8H/ 504	505(40505)	R/O, Real Value = Read Value

You will see LR140 data encode by base64

Example:

“BIUEhRQMhGmHMGbFUzEwNAAAAAH1AQUeAtIASwEFAAYAqg==”

Convert base64 to Hex:

04850485 140c1e 03 073200 4553313034000000 01 f501 05 1e02 d200 b301 0500 0600 aa

Sensor Name: “4553313034000000” -> “ES104”

Modbus ID: 01

Reg addr: “f501” -> 0x01f5 = 501

Reg num: 05

Reg 1 value: “1e02” -> 0x021e = 542 -> Humidity: 54.2

Reg 2 value: “d200” -> 0x00d2 = 210 -> Temperature: 21.0

Reg 4 value: “0500” -> 0x0005 = 5 -> PM2.5: 5

Reg 5 value: “0600” -> 0x0006 = 6 -> PM10: 6

You can add decoder for other sensors by copy the “ES104 Sensor Decoder” and modify it.

data": "BIUEhRQMhGmHMGbFUzEwNAAAAAH1AQUeAtIASwEFAAYAqg=="

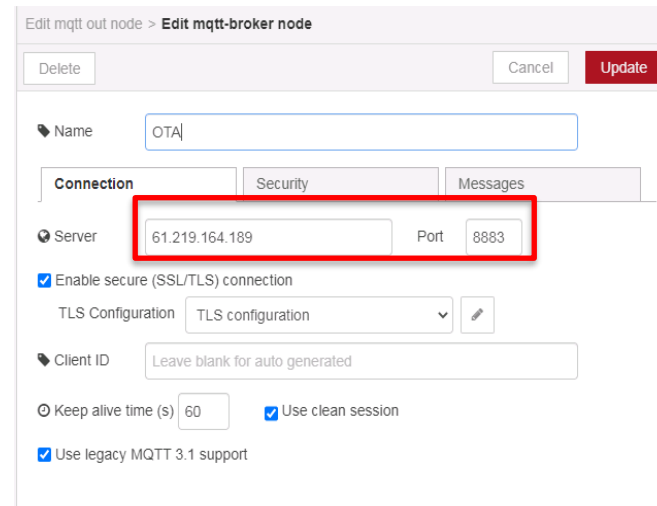
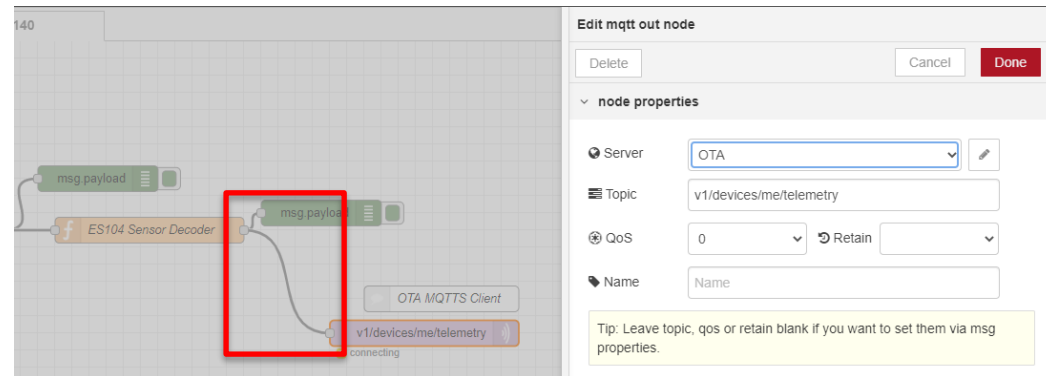
ES104 Sensor Decoder

```

1 // Decoder for ES104
2 function ES104_Decode(bytes) {
3   var t = 0, h = 0, pm25 = 0, pm10 = 0;
4   var mid = 0;
5   var reg;
6   var num;
7
8   if((bytes[0] == 0x04) && (bytes[1] == 0x85) &&
9     (bytes[2] == 0x04) && (bytes[3] == 0x85))
10  {
11    num = bytes[22]; // Reg num
12    mid = bytes[19]; // Modbus ID
13    reg = bytes[21] * 0x100 + bytes[20]; // Reg addr
14
15    if(num == 5)
16    {
17      if(reg == 0x01f5)
18      {
19        h = bytes[24] * 0x100 + bytes[23];
20        h = h/10;
21        t = bytes[26] * 0x100 + bytes[25];
22        t = t/10;
23        pm25 = bytes[30] * 0x100 + bytes[29];
24        pm10 = bytes[32] * 0x100 + bytes[31];
25        return {"Humidity":h, "Temperature":t, "PM2_5":pm25, "PM10":pm10}
26      }
27    }
28    else if(num == 1)
  
```

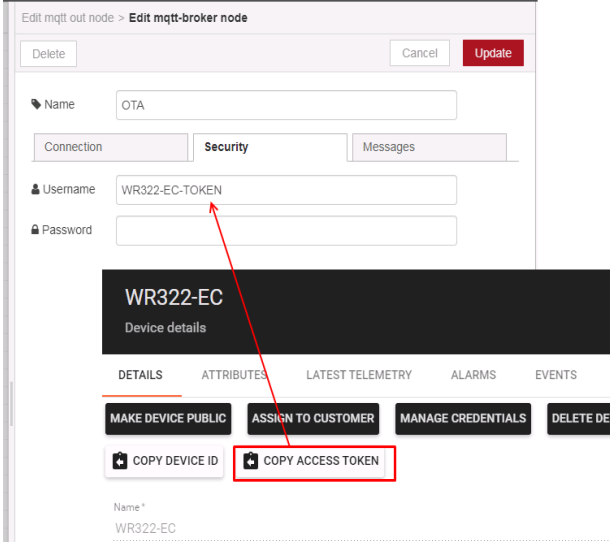
# OTA integration (Optional)

- Connect “ES104 Sensor Decoder” to OTA “mqtt out node”
- Edit the mqtt out node
- Edit the mqtt-broker node
- Fill in Server IP, Port, and enable the (SSL/TLS) connection



# OTA integration (cont.)

- The Username in the Security tab should be set to the ACCESS TOKEN copy from OTA device details
- Upload the MQTTS Certificate and press Update
- The MQTT client will show "connected" if it connect to the OTA server



Edit mqtt out node > Edit mqtt-broker node

Delete Cancel Update

Name OTA

Connection Security Messages

Username WR322-EC-TOKEN

Password

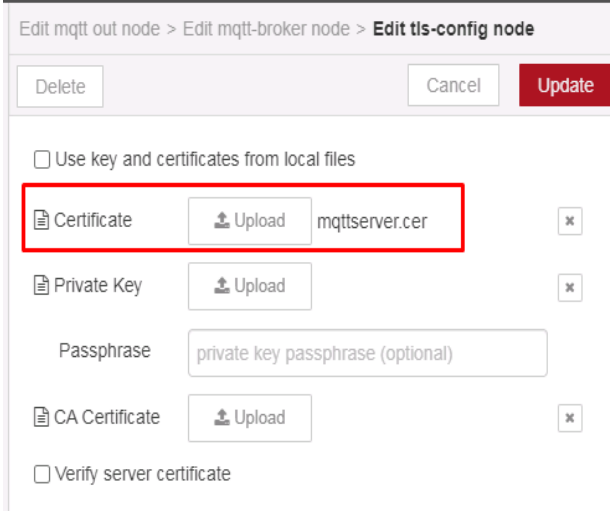
WR322-EC  
Device details

DETAILS ATTRIBUTES LATEST TELEMETRY ALARMS EVENTS

MAKE DEVICE PUBLIC ASSIGN TO CUSTOMER MANAGE CREDENTIALS DELETE DE

COPY DEVICE ID COPY ACCESS TOKEN

Name\*  
WR322-EC



Edit mqtt out node > Edit mqtt-broker node > Edit tls-config node

Delete Cancel Update

Use key and certificates from local files

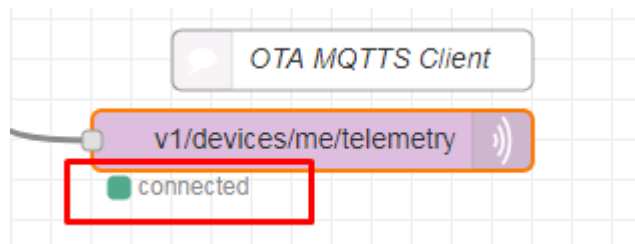
Certificate Upload mqttserver.cer X

Private Key Upload X

Passphrase private key passphrase (optional)

CA Certificate Upload X

Verify server certificate



# Display data from LR140 to OTA

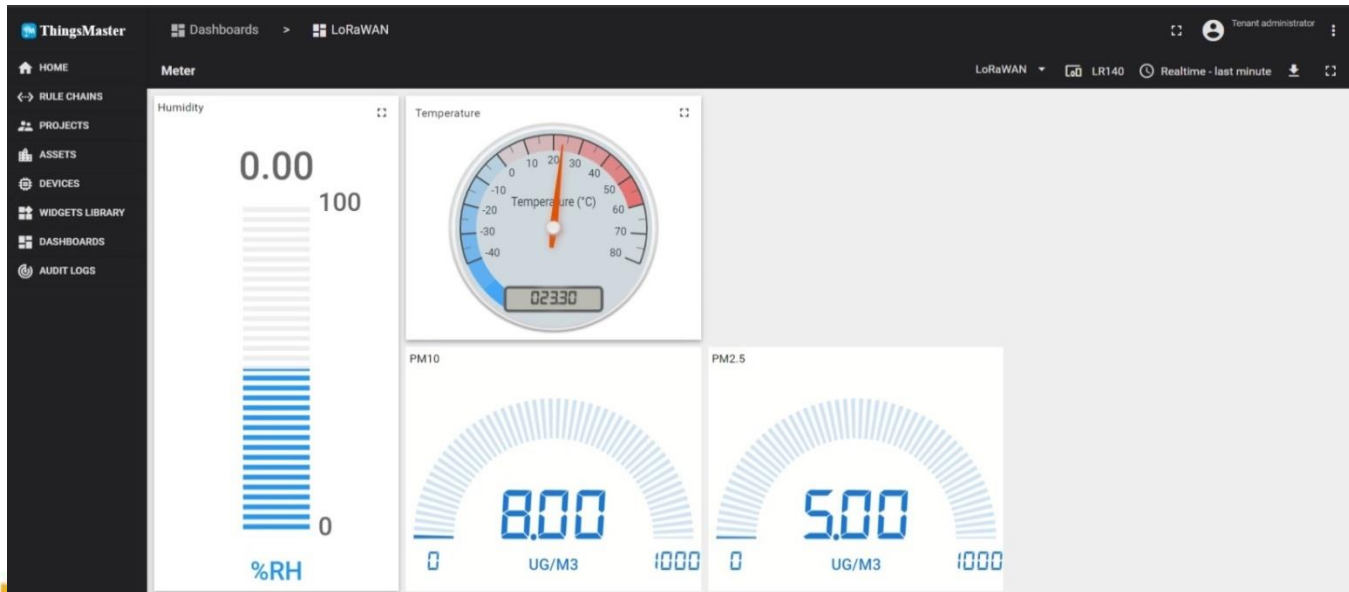
- OTA -> Device -> LASTEST TELEM
- OTA -> dashboard

WR322-EC  
Device details

DETAILS ATTRIBUTES LATEST TELEMETRY ALARMS EVENTS RELATIONS EXTENSIONS

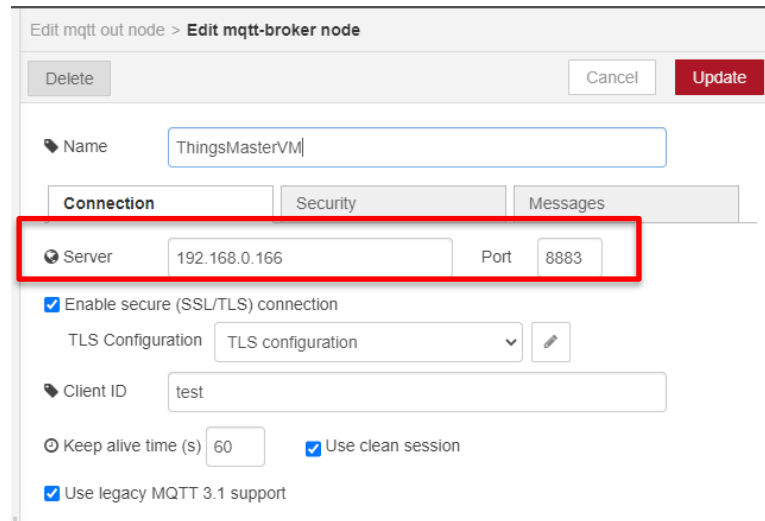
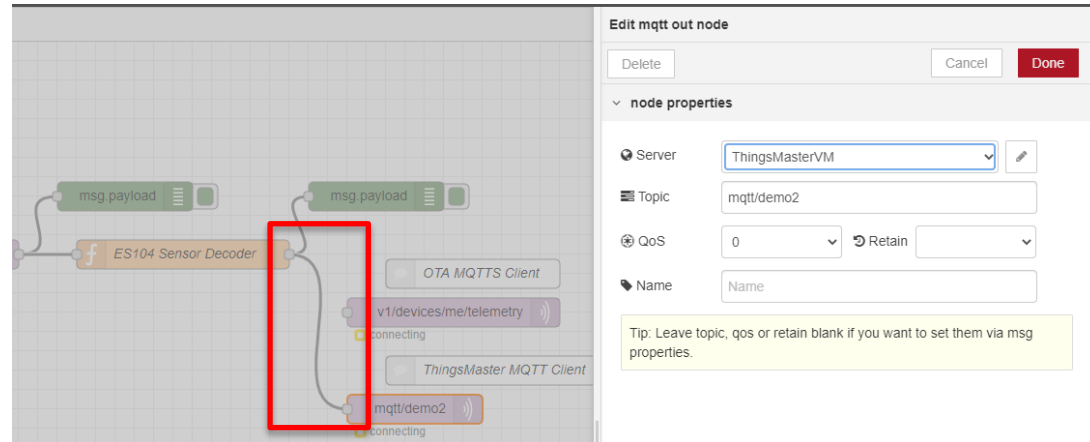
Latest telemetry

<input type="checkbox"/>	Last update time ↓	Key	Value
<input type="checkbox"/>	2021-01-06 13:56:47	Humidity	58.7
<input type="checkbox"/>	2021-01-06 13:56:47	PM10	3
<input type="checkbox"/>	2021-01-06 13:56:47	PM2.5	3
<input type="checkbox"/>	2021-01-06 13:56:47	Temperature	21



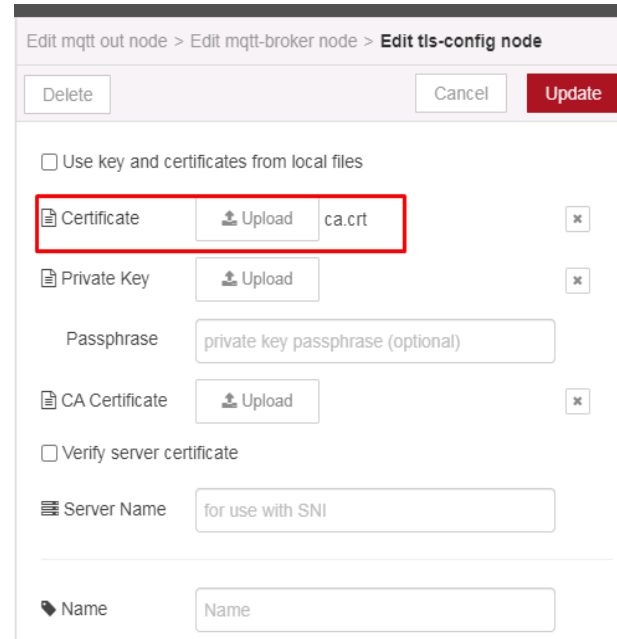
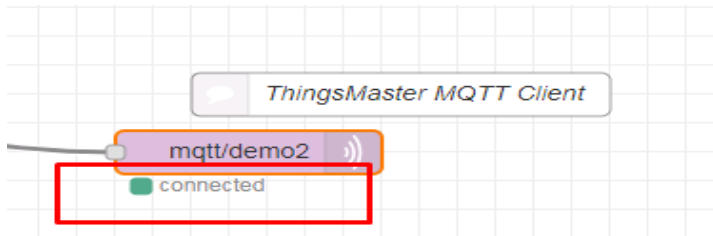
# ThingsMaster integration (Optional)

- Connect “ES104 Sensor Decoder” to ThingsMaster “mqtt out node”
- Edit the mqtt out node
- Edit the mqtt-broker node
- Fill in Server IP, Port, and enable the (SSL/TLS) connection



# ThingsMaster integration (cont.)

- Upload the MQTTS Certificate and press Update
- The MQTT client will show "connected" if it connect to the ThingsMasterserver



A screenshot of the ThingsMaster configuration page for the MQTT client. The page is titled "Edit mqtt out node > Edit mqtt-broker node > Edit tls-config node". It contains several fields and buttons:

- Use key and certificates from local files
- Certificate (ca.crt) with an Upload button and a close button (x).
- Private Key with an Upload button and a close button (x).
- Passphrase: private key passphrase (optional)
- CA Certificate with an Upload button and a close button (x).
- Verify server certificate
- Server Name: for use with SNI
- Name: Name

Buttons: Delete, Cancel, Update

# Display data from LR140 to ThingsMaster

