General Configure/Commands to Connect to IoT server for -NB & -NS NB-IoT models

last modified by Mengting Qiu

on 2024/08/28 14:44

Table of Contents

Table of Contents:

1. The use of this guideline

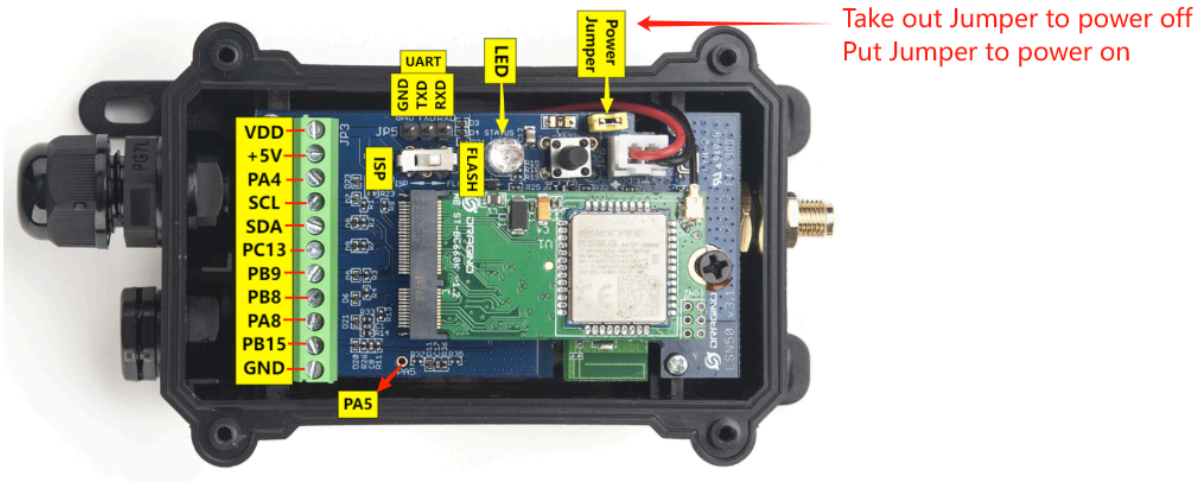
This configure instruction is for Dragino NB-IoT models with -NB or -NS suffix, for example DDS75-NB. These models use the same NB-IoT Module **BC660K-GL** and has the same software structure. The have the same configure instruction to different IoT servers. Use can follow the instruction here to see how to configure to connect to those servers.

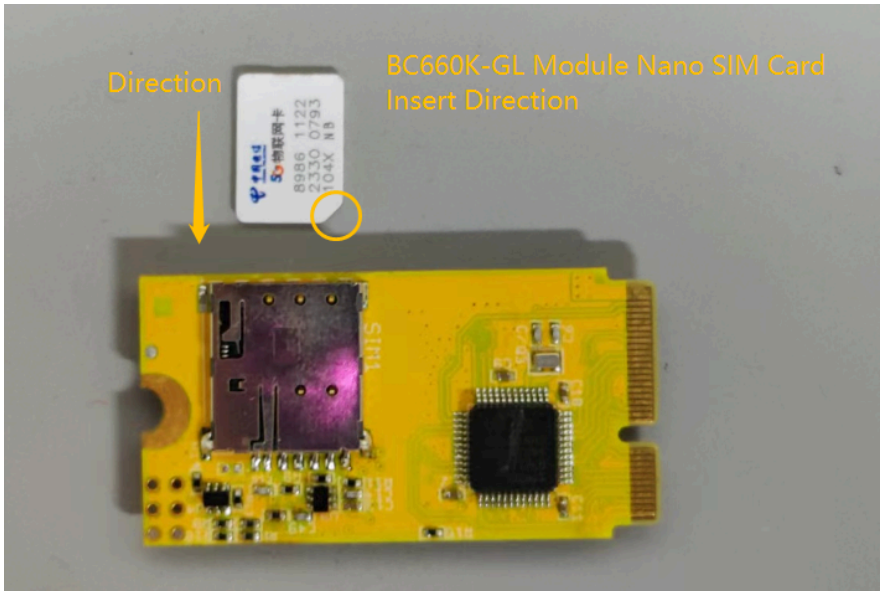
2. Attach Network

2.1 General Configure to attach network

To attache NB-IoT sensors to NB-IoT Network, You need to:

1. Get a NB-IoT SIM card from Service Provider. (Not the same as the SIM card we use in mobile phone)
2. Power Off End Node (See below for the power off/on position)
3. Insert the SIM card to Sensor. (See below for direction)
4. Power On End Node
5. Configure APN in the sensor (AT+APN=<APN>), example AT+APN=iot.1nce.net





After doing above, the NB-IoT Sensors should be able to attach to NB-IoT network .

The -NB and -NS models support **LTE Cat NB2**, with below frequency band: multiple frequency bands of **B1/B2/B3/B4/B5/B8/B12/B13/B14/B17/B18/B19/B20/B25/B28/B66/B70/B85** . Make sure you use a the NB-IoT SIM card.

SIM Provider	AT+APN=	NB-IoT Coverage
1NCE	iot.1nce.net	Coverage Reference Link Austria, Belgium, Bulgaria, Croatia, Czech Republic, Denmark, Finlan Latvia, Malta, Netherlands, Norway, Puerto Rico, Russia, Slovak , Repu UK, US Virgin Islands
China Mobile	No need configure	China Mainland, HongKong
China Telecom	ctnb	China Mainland

2.2 Speed Up Network Attach time

BC660K-GL supports multi bands **B1/B2/B3/B4/B5/B8/B12/B13/B14/B17/B18/B19/B20/B25/B28/B66/B70/B85.** It will search one by one and try to attach, this will take a lot of time and even cause attach fail and show **Signal Strenght:99**. User can lock the band to specify band for its operator to make this faster.

AT+QBAND? // Check what is the current used frequency band
AT+QBAND=1,4 // Set to use 1 frequency band. Band4
Europe General **AT+QBAND=2,8,20** // Set to use 2 frequency bands. Band 8 and Band 20
Global General : **AT+QBAND=10,8,20,28,2,4,12,13,66,85,5**

Verizon AT+QBAND=1,13
AT&T AT+QBAND=3,12,4,2
Telstra AT+QBAND=1,28
Softband AT+QBAND=2,3,8

After connection is successful, user can use **AT+QENG=0** to check which band is actually in used.

By default, device will search network for 5 minutes. User can set the time to 10 minutes by **AT+CSQTIME=10** so it can search longer.

See bands used for different provider: **NB-IoT Deployment , Bands, Operator list**

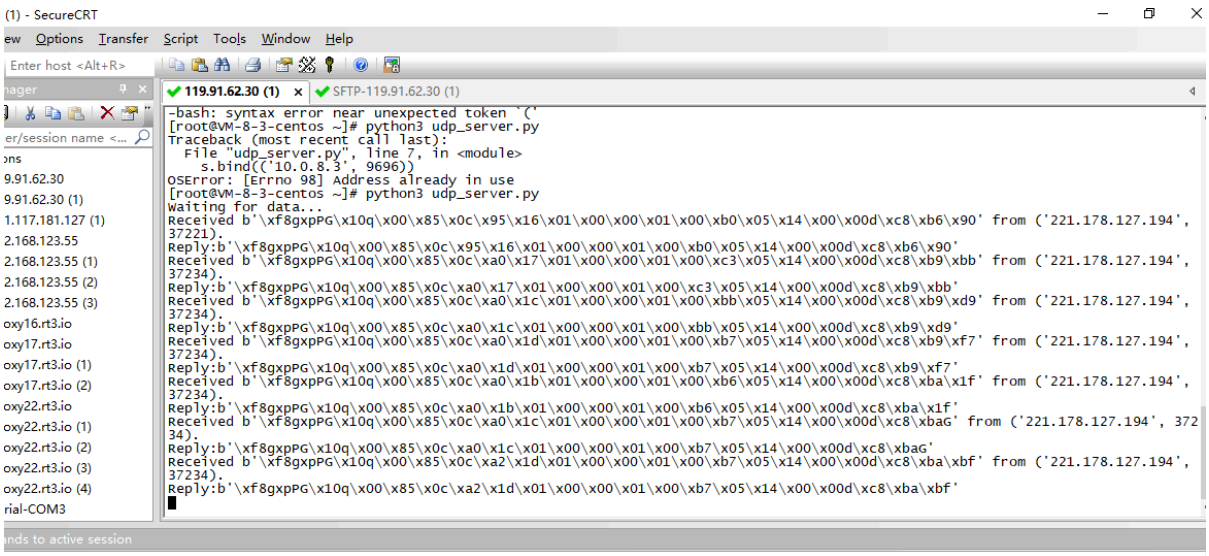
3. Configure to connect to different servers

3.1 General UDP Connection

The NB-IoT Sensor can send packet to server use UDP protocol.

3.1.1 Simulate UDP Connection by PC tool

We can use PC tool to simulate UDP connection to make sure server works ok.



3.1.2 Configure NB-IoT Sensor

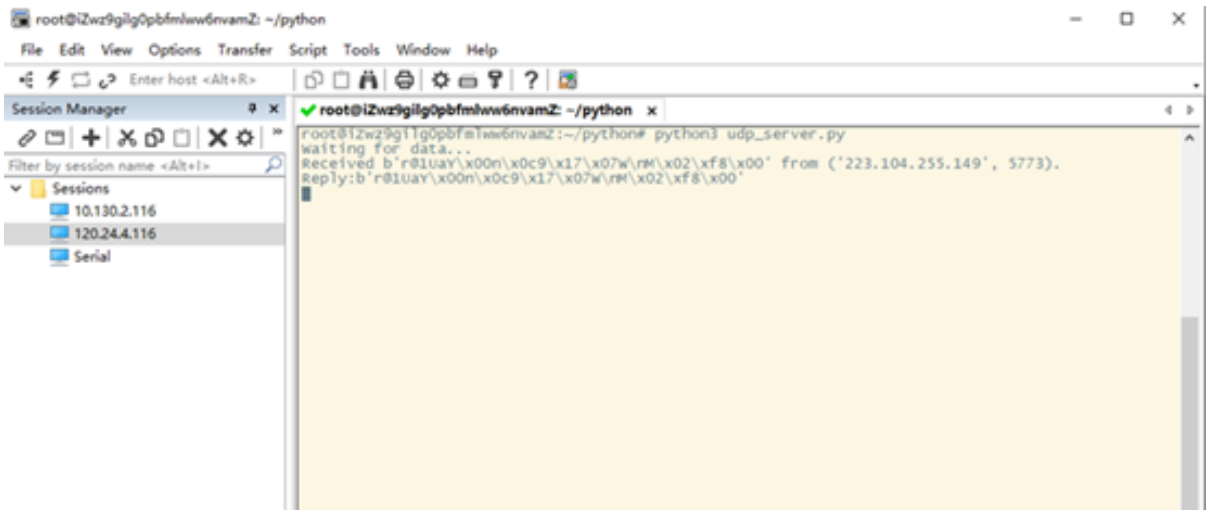
3.1.2.1 AT Commands

AT Commands:

- **AT+PRO=2,0** // Set to use UDP protocol to uplink ,Payload Type select Hex payload
- **AT+SERVADDR=120.24.4.116,5601** // Set UDP server address and port

```
AT+NAME866207053462762
[816]reboot error:NRST!
DRAGINO SN50V3-NB NB-IoT Sensor Node
Image Version: v1.0.0
NB-IoT Stack : D-BC660K-001
Protocol in Used: UDP
[7669]NBIOT has responded.
[12001]Echo mode turned off successfully.
[13347]Disable the reporting of deep sleep event URC.
[14706]Model information:BC660K-GL.
[16046]The IMEI number is:866207053462762.
[17394]The IMSI number is:460083823106206.
[19442]Set the data format for sending and receiving.
Currently set frequency band: 1,2,3,4,5,8,12,13,17,18,19,20,25,28,66,70,85
[26484]Signal Strength:24
[31518]PSM mode configured
[34151]DNS configuration is successful
[40195]No DNS resolution required
[41233]*****Upload start:0*****
[41769]remaining battery =3245 mv
[42563]DS18B20(1) temp is -0.06
[42666]adc_mV(1):53.00
[42693]No I2C device detected
[46230]Open a Socket Service successfully
[53283]Datagram is sent by RF
[54317]Send complete
[55341]*****End of upload*****
AT+PWRM2
```

3.1.2.2 Uplink Example



3.2 General MQTT Connection

The NB-IoT Sensor can send packet to server use MQTT protocol.

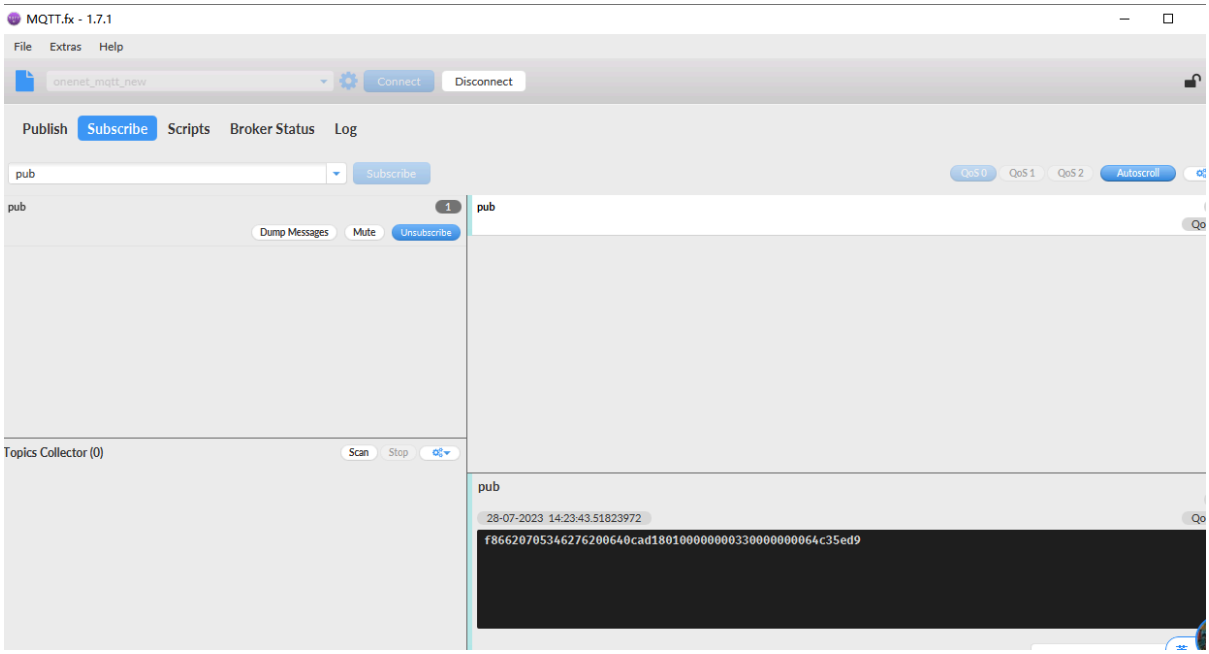
Below are the commands.

AT Commands:

- **AT+PRO=3,0** // Set to use MQTT protocol to uplink, Payload Type select Hex payload.
- **AT+SERVADDR=120.24.4.116,1883** // Set MQTT server address and port

- **AT+CLIENT=CLIENT** // Set up the CLIENT of MQTT
- **AT+UNAME=UNAME** // Set the username of MQTT
- **AT+PWD=PWD** // Set the password of MQTT
- **AT+PUBTOPIC=NSE01_PUB** // Set the sending topic of MQTT
- **AT+SUBTOPIC=NSE01_SUB** // Set the subscription topic of MQTT

```
DRAGINO SN50V3-NB NB-IoT Sensor Node
Image Version: v1.0.0
NB-IoT Stack : D-BC660K-001
Protocol in Used: MQTT
[7370]NBIOT has responded.
[11701]Echo mode turned off successfully.
[13047]Disable the reporting of deep sleep event URC.
[14406]Model information:BC660K-GL.
[15746]The IMEI number is:866207053462762.
[17094]The IMSI number is:460083823106206.
[19142]Set the data format for sending and receiving.
Currently set frequency band: 1,2,3,4,5,8,12,13,17,18,19,20,25,28,66,70,85
[26184]Signal Strength:25
[31218]PSM mode configured
[33851]DNS configuration is successful
[39895]No DNS resolution required
[40933]*****Upload start:0*****
[41469]remaining battery =3245 mv
[42263]DS18B20(1) temp is -0.06
[42366]adc_mV(1):56.00
[43471]No I2C device detected
[48909]Opened the MQTT client network successfully
[52467]Successfully connected to the server
[57522]Subscribe to topic successfully
[61068]Close the port successfully
AT+PWRM2
[62269]Send complete
[63293]*****End of upload*****
```

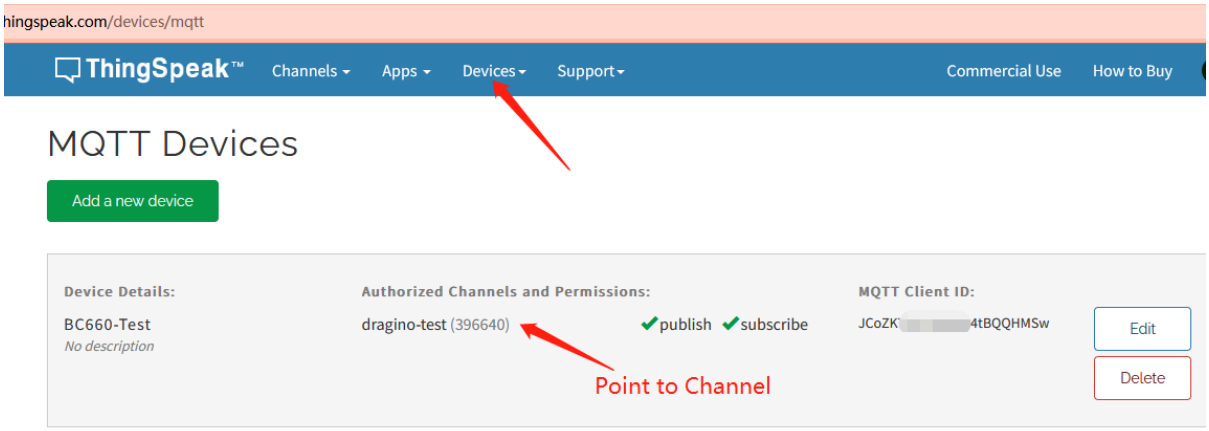


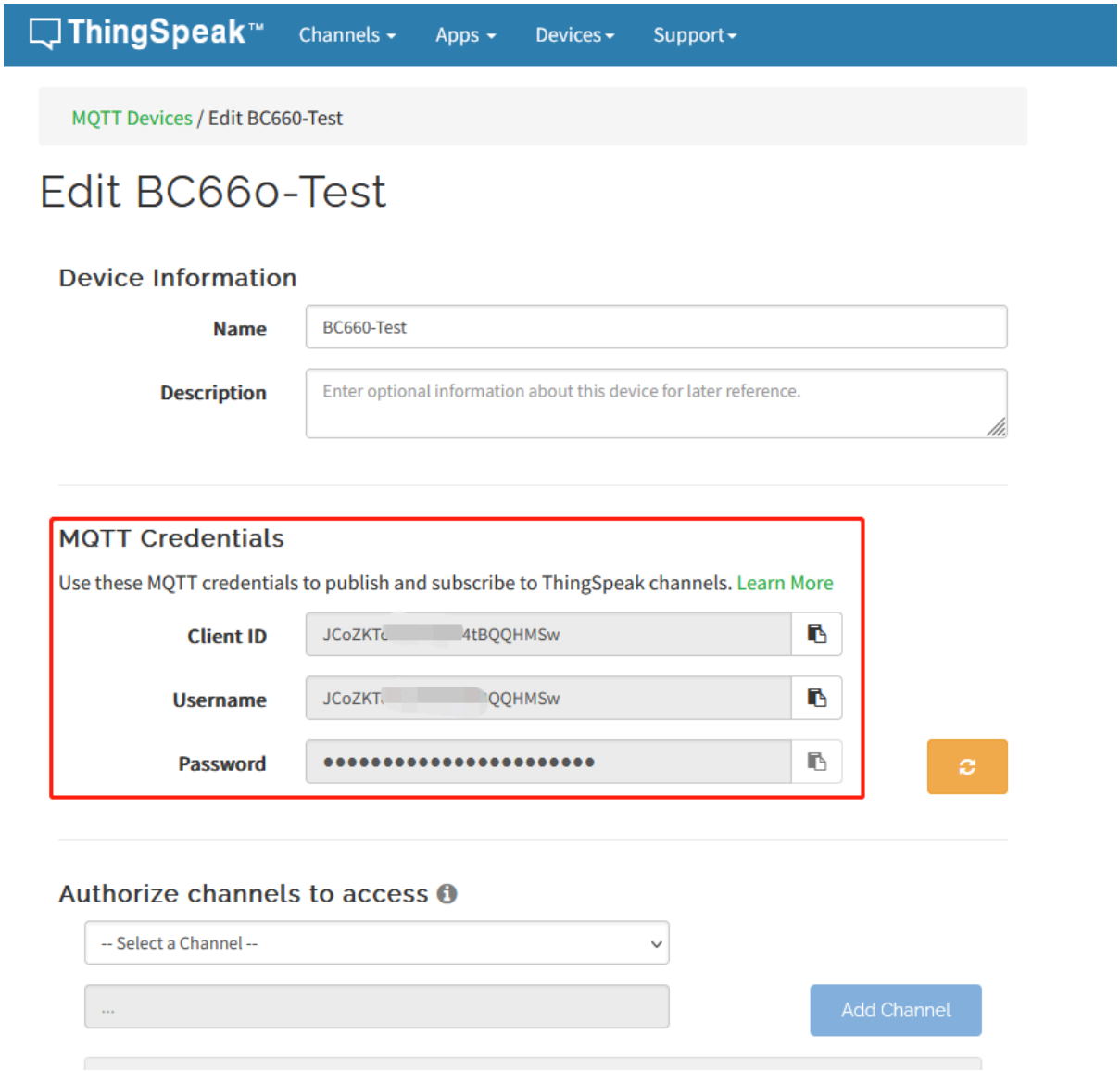
Notice: MQTT protocol has a much higher power consumption compare with UDP/ CoAP protocol. Please check the power analyze document and adjust the uplink period to a suitable interval.

3.3 ThingSpeak (via MQTT)

3.3.1 Get MQTT Credentials

ThingSpeak connection uses MQTT Connection. So we need to get MQTT Credentials first. You need to point MQTT Devices to Th

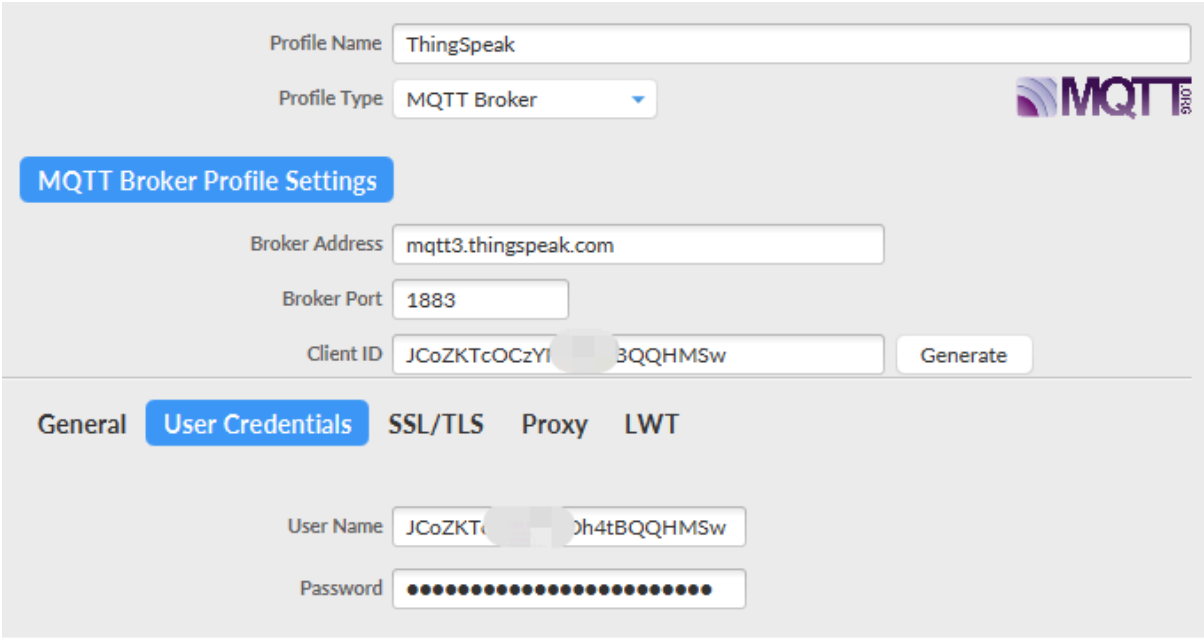

hingspeak.com/devices/mqtt
ThingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
MQTT Devices
Add a new device
Device Details:
BC660-Test
No description
Authorized Channels and Permissions:
dragino-test (396640)
publish
subscribe
MQTT Client ID:
JCoZK
4tBQQHMSw
Edit
Delete
Point to Channel



3.3.2 Simulate with MQTT.fx

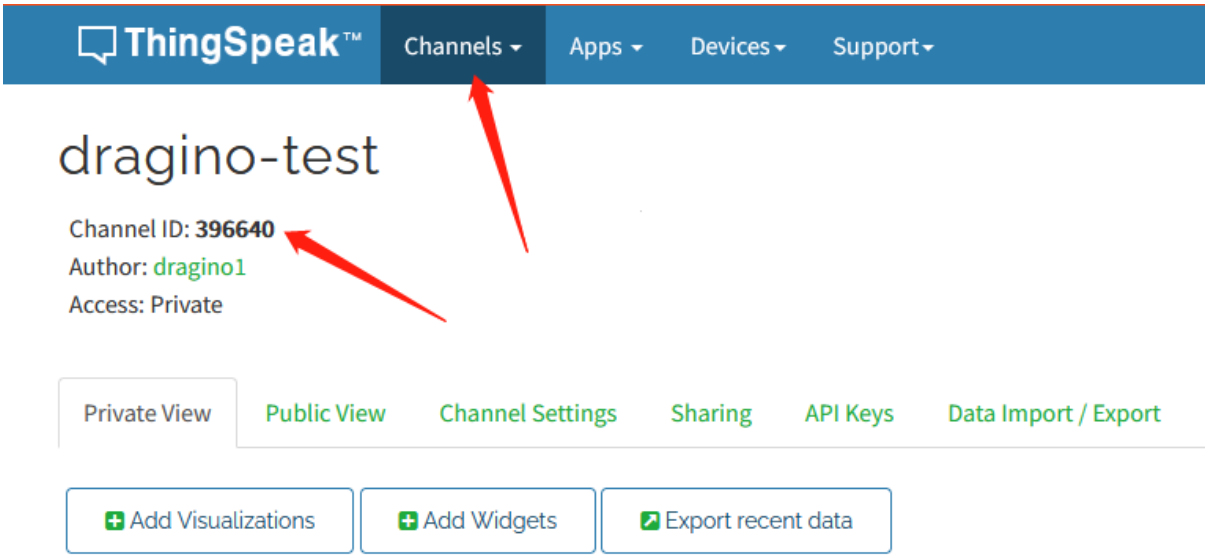
3.3.2.1 Establish MQTT Connection

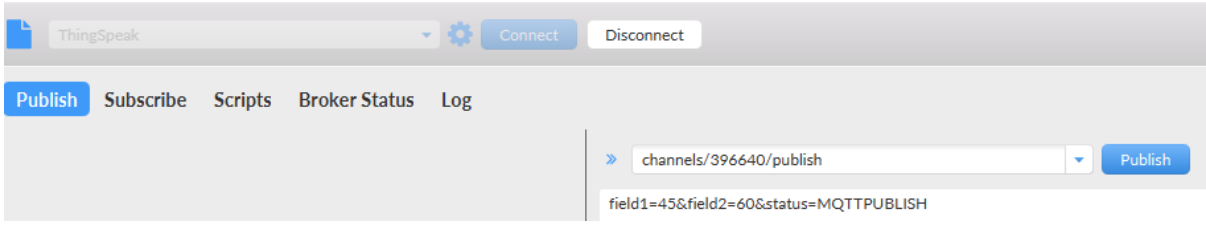
After we got MQTT Credentials, we can first simulate with PC tool MQTT.fx tool to see if the Credentials and settings are fine.



- **Broker Address:** mqtt3.thingspeak.com
- **Broker Port:** 1883
- **Client ID:** <Your ThingSpeak MQTT ClientID>
- **User Name:** <Your ThingSpeak MQTT User Name>
- **Password:** <Your ThingSpeak MQTT Password>

3.3.2.2 Publish Data to ThingSpeak Channel



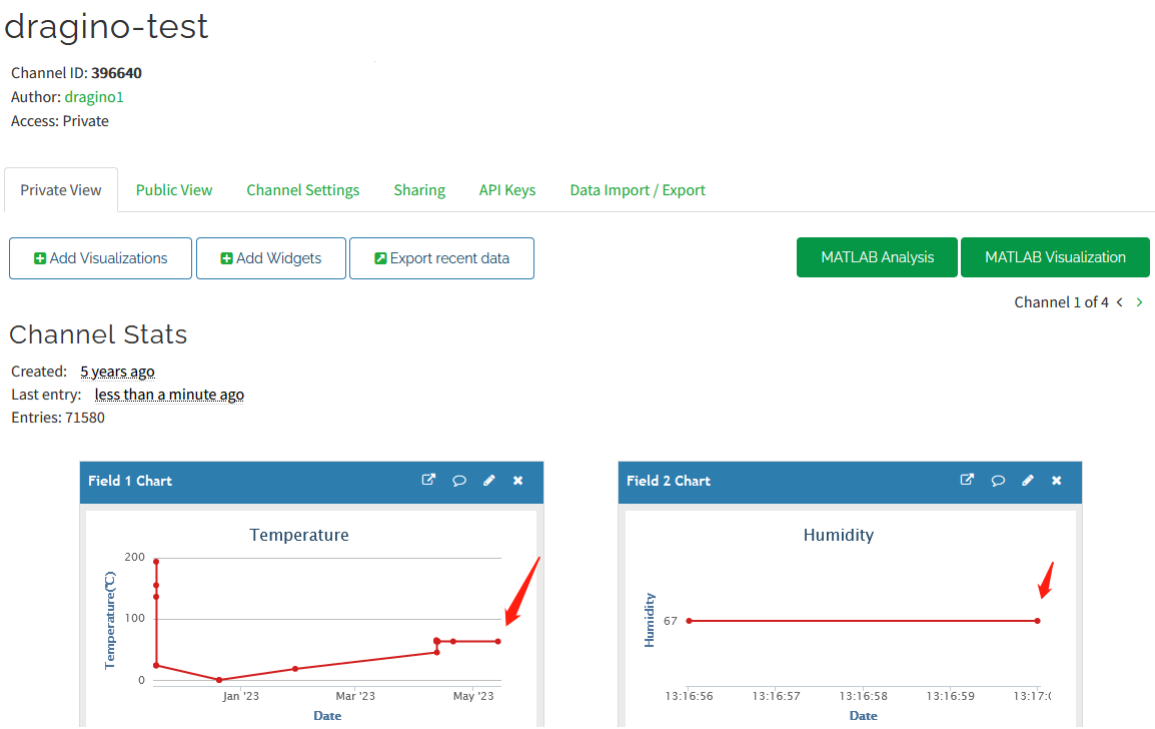


In MQTT.fx, we can publish below info:

- **Topic:** channels/YOUR_CHANNEL_ID/publish
- **Payload:** field1=63&field2=67&status=MQTTPUBLISH

Where 63 and 67 are the value to be published to field1 & field2.

Result:



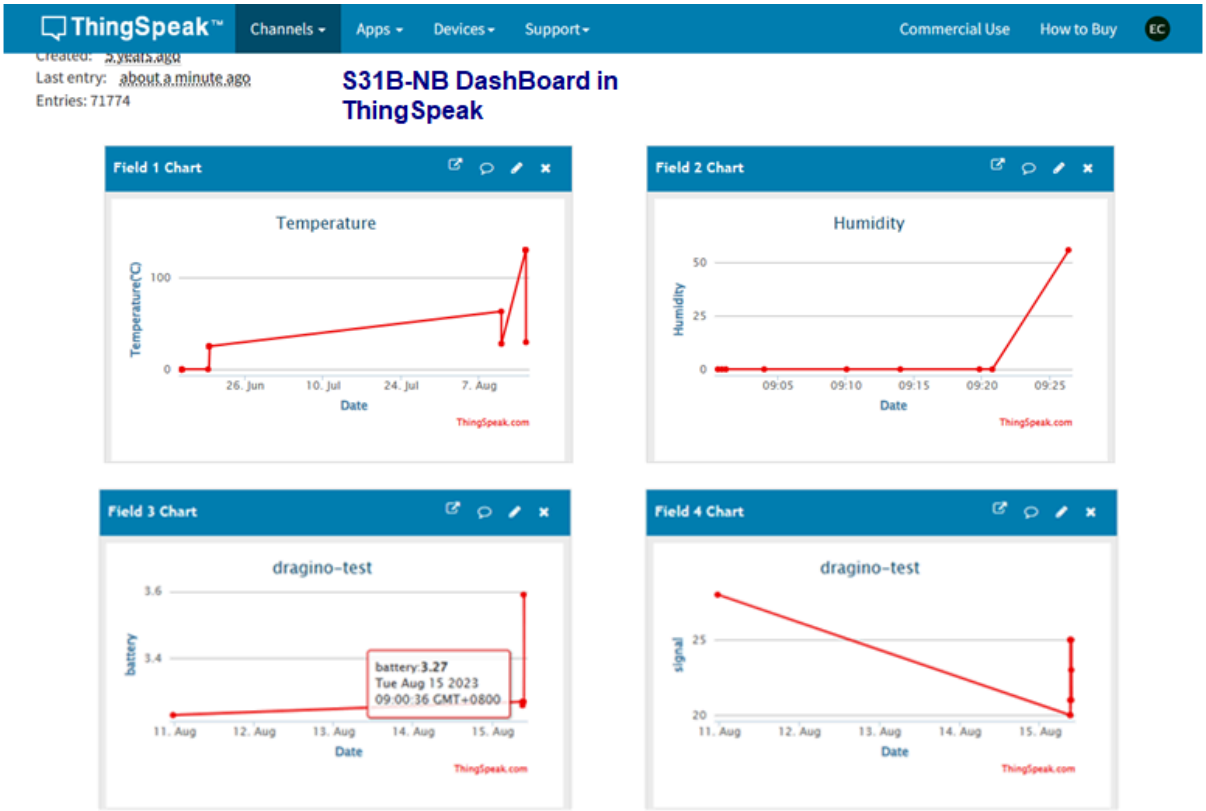
3.3.3 Configure NB-IoT Sensor for connection

3.3.3.1 AT Commands:

In the NB-IoT, we can run below commands so to publish the channels like MQTT.fx

- **AT+PRO=3,1** // Set to use ThingSpeak Server and Related Payload
- **AT+CLIENT=<Your ThingSpeak MQTT ClientID>**
- **AT+UNAME=<Your ThingSpeak MQTT User Name>**
- **AT+PWD=<Your ThingSpeak MQTT Password>**
- **AT+PUBTOPIC=<YOUR_CHANNEL_ID>**
- **AT+SUBTOPIC=<YOUR_CHANNEL_ID>**

3.3.3.2 Uplink Examples



For SE01-NB

For DDS20-NB

For DDS45-NB

For DDS75-NB

For NMDS120-NB

For SPH01-NB

For NLM01-NB

For NMDS200-NB

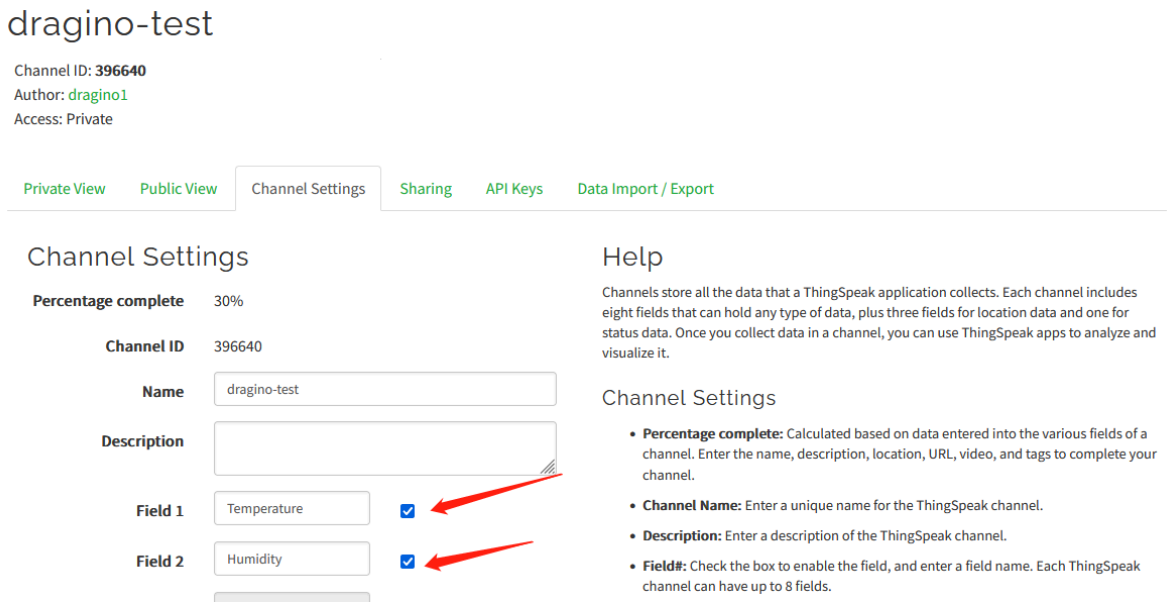
For CPN01-NB

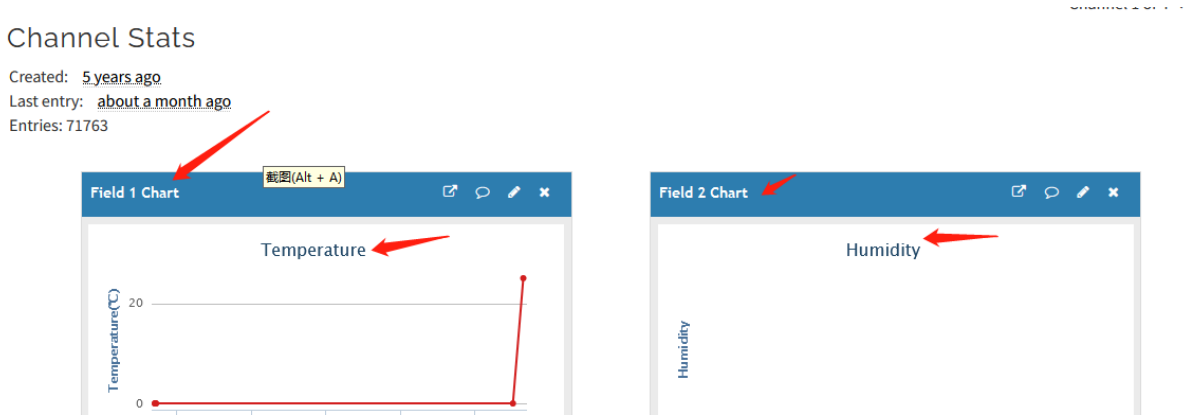
For DS03A-NB

For SN50V3-NB

3.3.3.3 Map fields to sensor value

When NB-IoT sensor upload to ThingSpeak. The payload already specify which fileds related to which sensor value. Use need to create fileds in





Below is the NB-IoT Product Table show the mapping.

	Field1	Field2	Field3	Field
S31x-NB	Temperature	Humidity	Battery	RSS
SE01-NB	Temperature	Humidity	conduct	diele
DDS20-NB	distance	Battery	RSSI	
DDS45-NB	distance	Battery	RSSI	
DDS75-NB	distance	Battery	RSSI	
NMDS120-NB	distance	Battery	RSSI	
SPH01-NB	ph	Temperature	Battery	RSS
NLM01-NB	Humidity	Temperature	Battery	RSS
NMDS200-NB	distance1	distance2	Battery	RSS
CPN01-NB	alarm	count	door open duration	calc
DS03A-NB	level	alarm	pb14door open num	pb1
SN50V3-NB mod1	mod	Battery	RSSI	DS1
SN50V3-NB mod2	mod	Battery	RSSI	DS1
SN50V3-NB mod3	mod	Battery	RSSI	adc
SN50V3-NB mod4	mod	Battery	RSSI	DS1
SN50V3-NB mod5	mod	Battery	RSSI	DS1
SN50V3-NB mod6	mod	Battery	RSSI	cour

3.4 Datacake

Dragino NB-IoT sensors has its template in **Datacake** Platform. There are two version for NB Sensor,

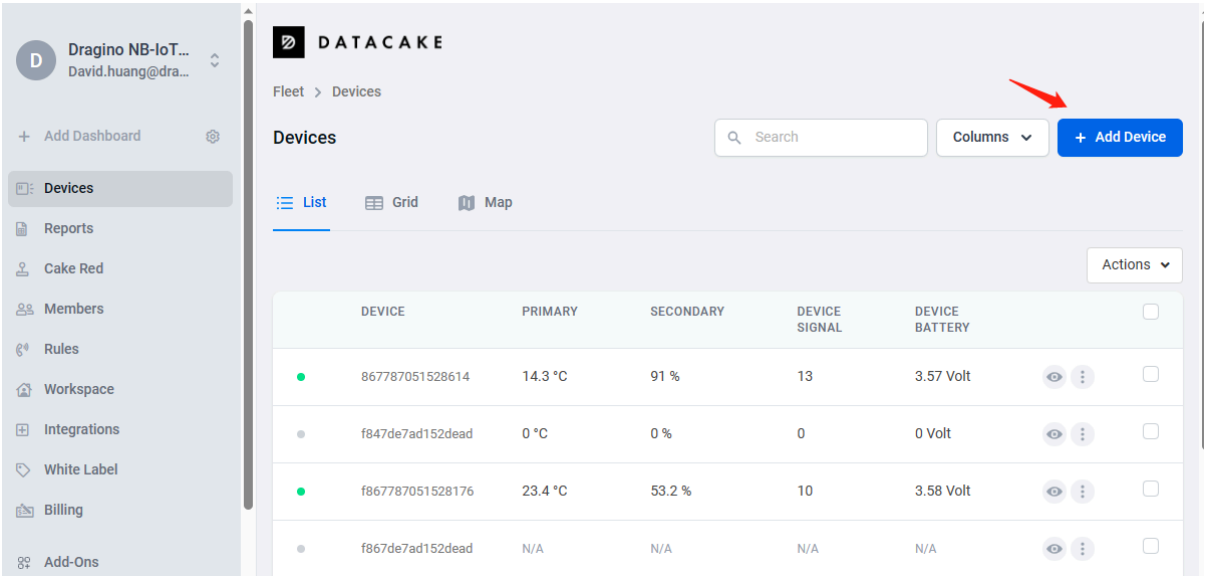
As example for S31B-NB. there are two versions: **S31B-NB-1D and S31B-NB-GE.**

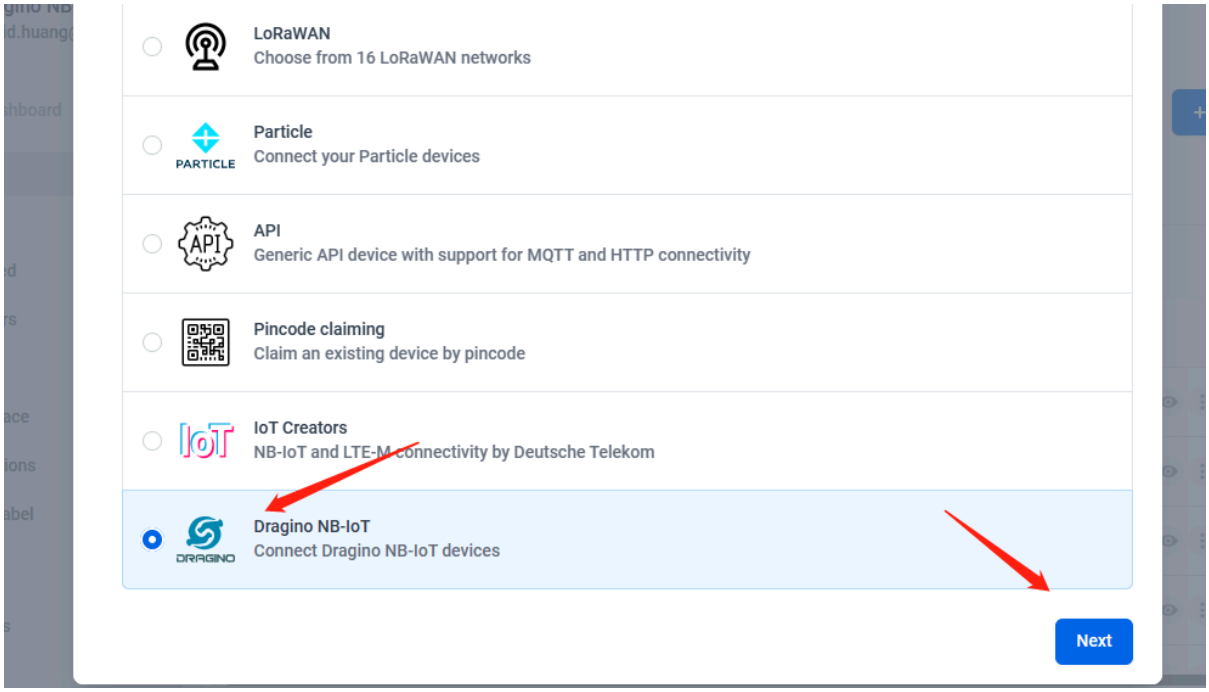
- **S31B-NB-1D**: This version have pre-configure DataCake connection. User just need to Power on this device, it will auto connect send data to DataCake Server.
- **S31B-NB-GE**: This verson doesn't have pre-configure Datacake connection. User need to enter the AT Commands to connect to Datacake. See below for instruction.

3.4.1 For device Already has template

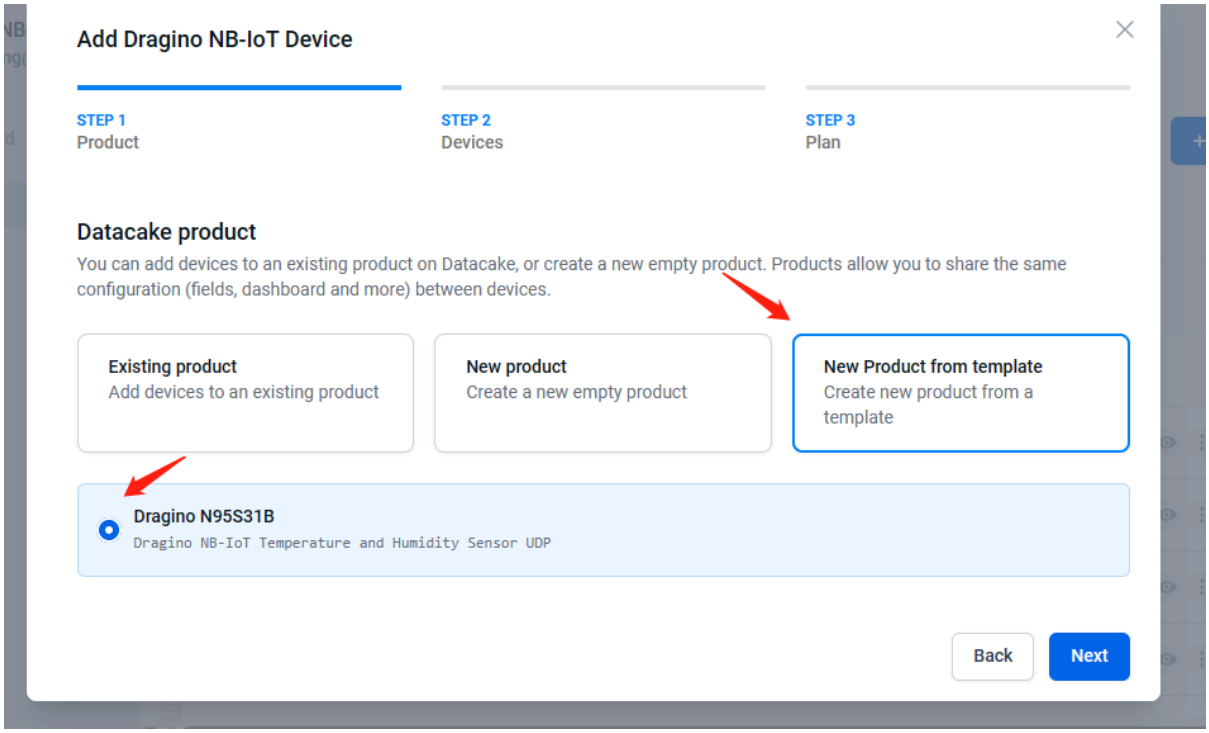
3.4.1.1 Create Device

Add Device in DataCake.

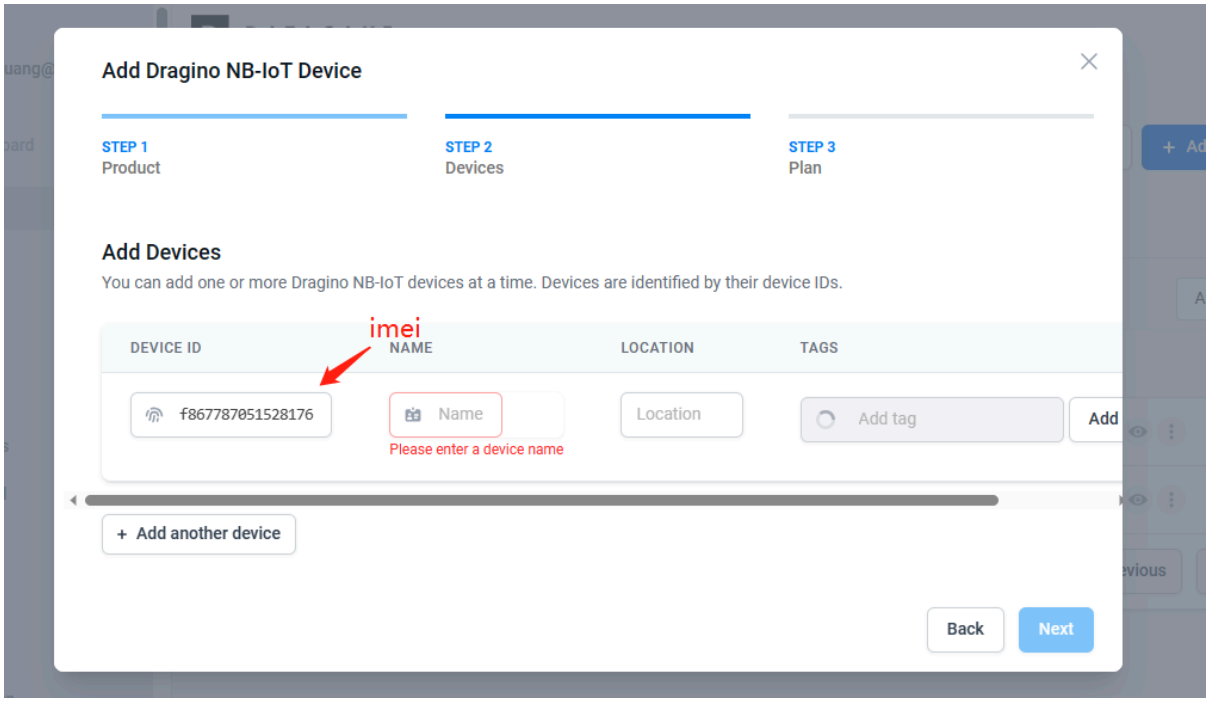


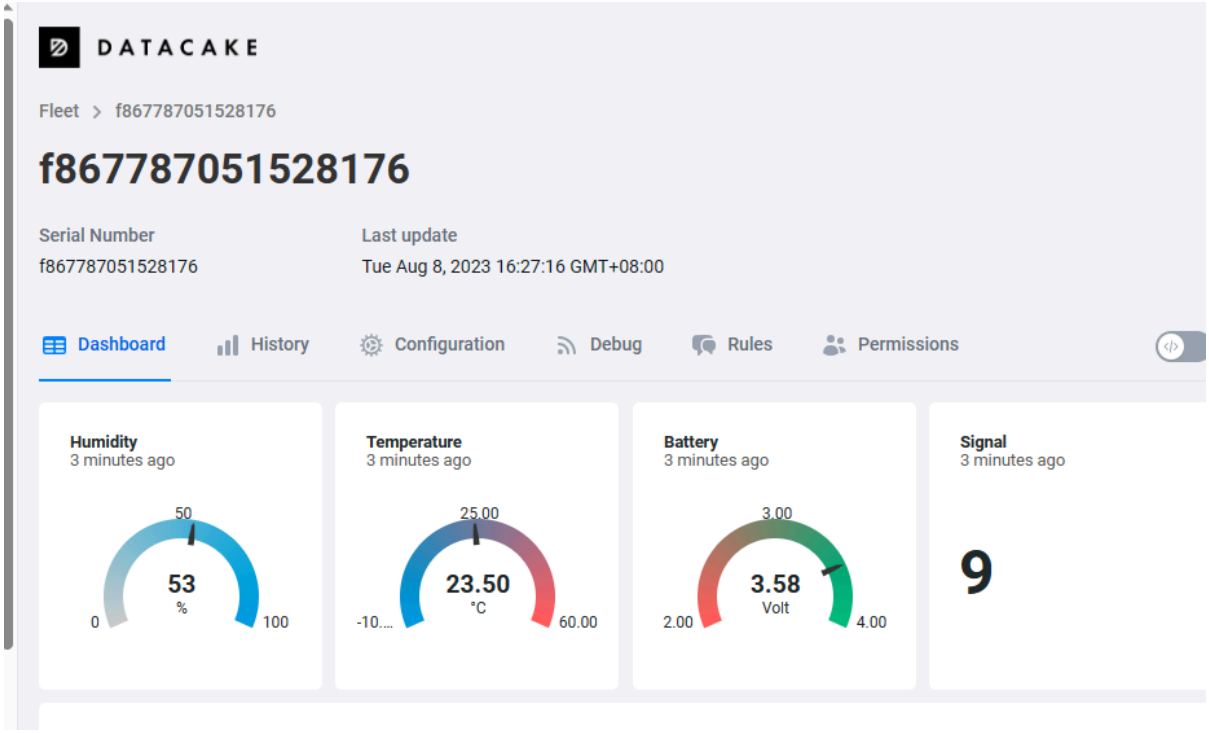


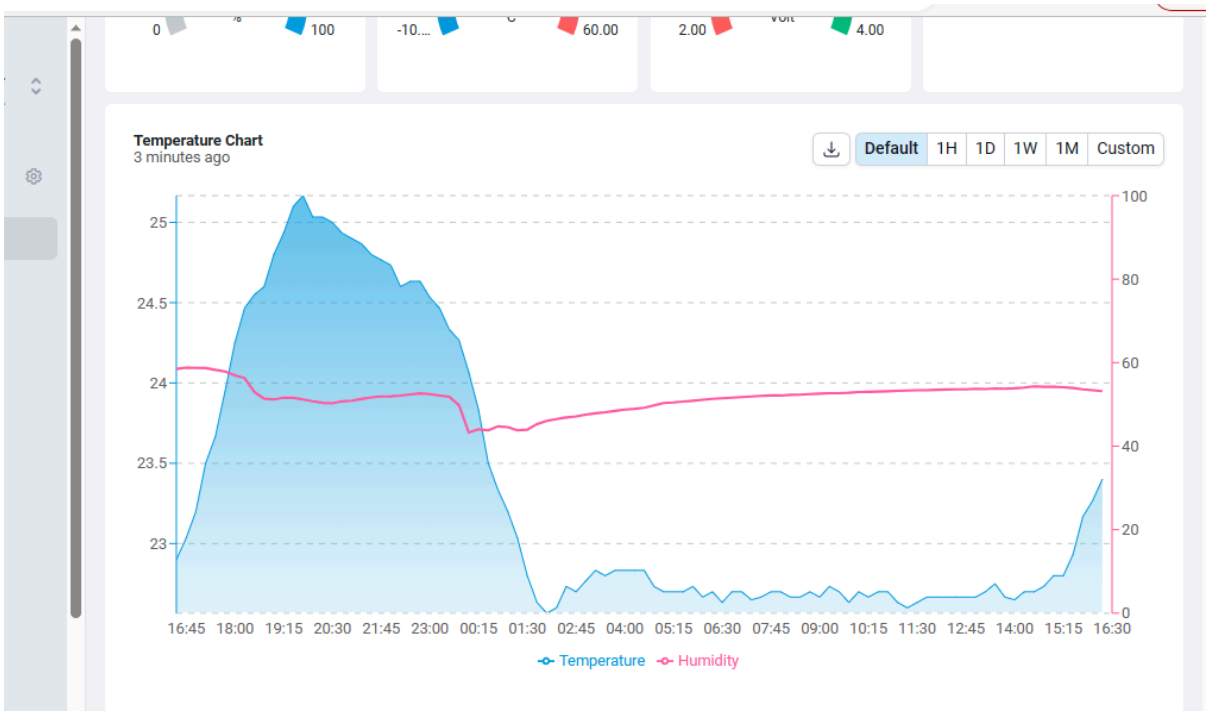
Choose the correct model from template.



Fill Device ID. The device ID needs to be filled in with IMEI, and a prefix of **'f'** needs to be added.


Add Dragino NB-IoT Device
STEP 1
Product
STEP 2
Devices
STEP 3
Plan
Add Devices
You can add one or more Dragino NB-IoT devices at a time. Devices are identified by their device IDs.
imei
DEVICE ID
NAME
LOCATION
TAGS
f867787051528176
Name
Please enter a device name
Location
Add tag
Add
+ Add another device
Back
Next

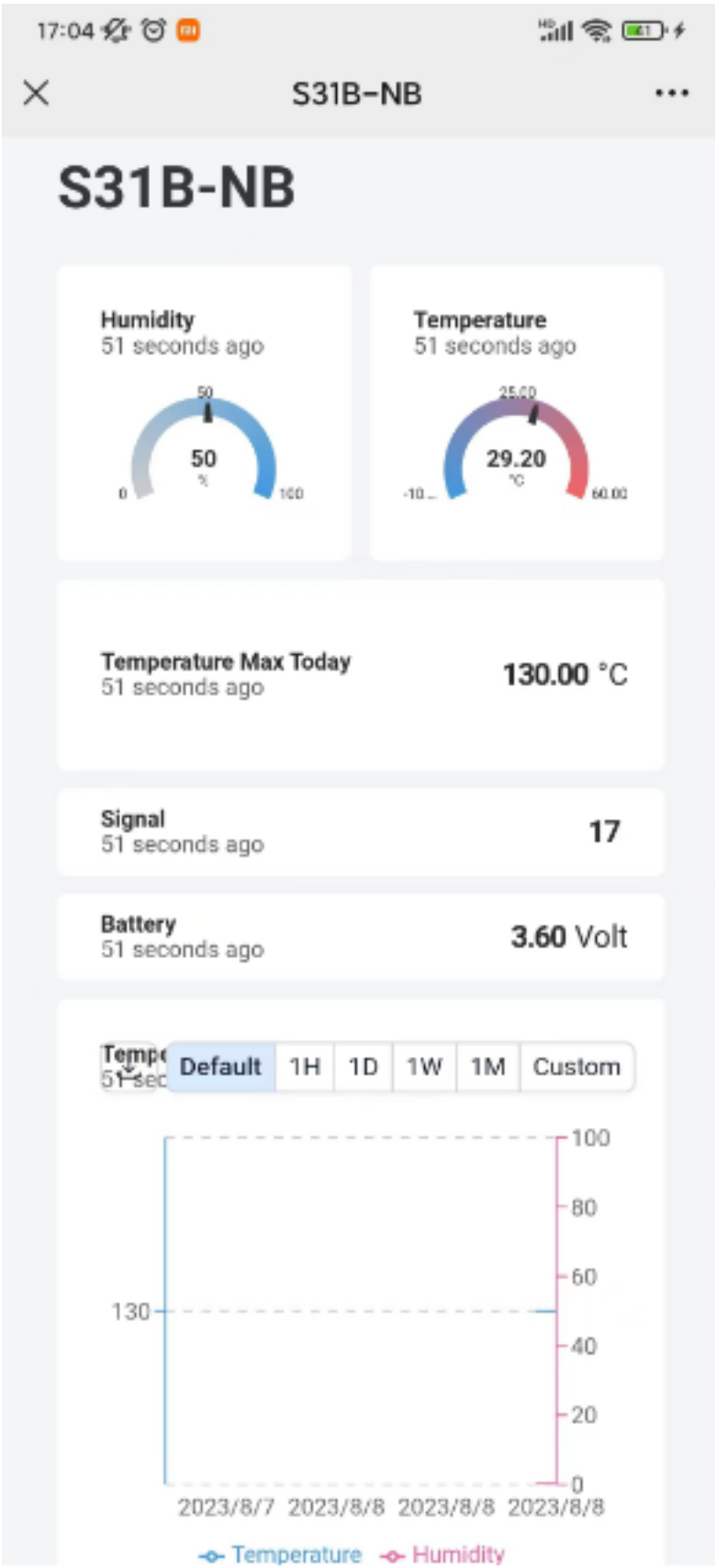

DATACAKE
Fleet > f867787051528176
f867787051528176
Serial Number
f867787051528176
Last update
Tue Aug 8, 2023 16:27:16 GMT+08:00
Dashboard
History
Configuration
Debug
Rules
Permissions
Humidity
3 minutes ago
50
53
%
0
100
Temperature
3 minutes ago
25.00
23.50
°C
-10....
60.00
Battery
3 minutes ago
3.00
3.58
Volt
2.00
4.00
Signal
3 minutes ago
9



3.4.2 For Device already registered in DataCake before shipped

3.4.2.1 Scan QR Code to get the device info

Users can use their phones or computers to scan QR codes to obtain device data information.

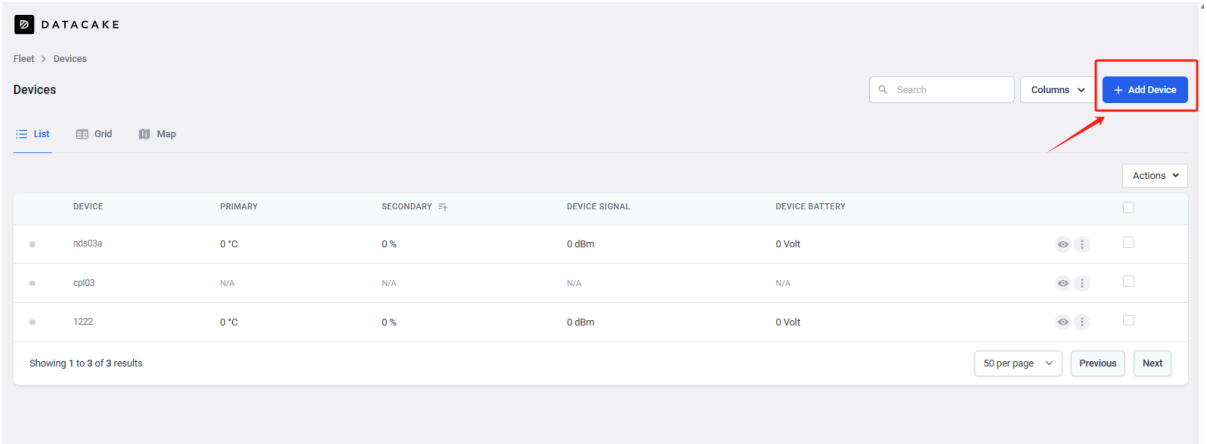

17:04
S31B-NB
S31B-NB
Humidity
51 seconds ago
50
%
0
100
Temperature
51 seconds ago
25.00
29.20
°C
-10
60.00
Temperature Max Today
51 seconds ago
130.00 °C
Signal
51 seconds ago
17
Battery
51 seconds ago
3.60 Volt
Default
1H
1D
1W
1M
Custom
100
80
60
40
20
0
130
2023/8/7
2023/8/8
2023/8/8
2023/8/8
Temperature
Humidity

3.4.2.2 Claim Device to User Account

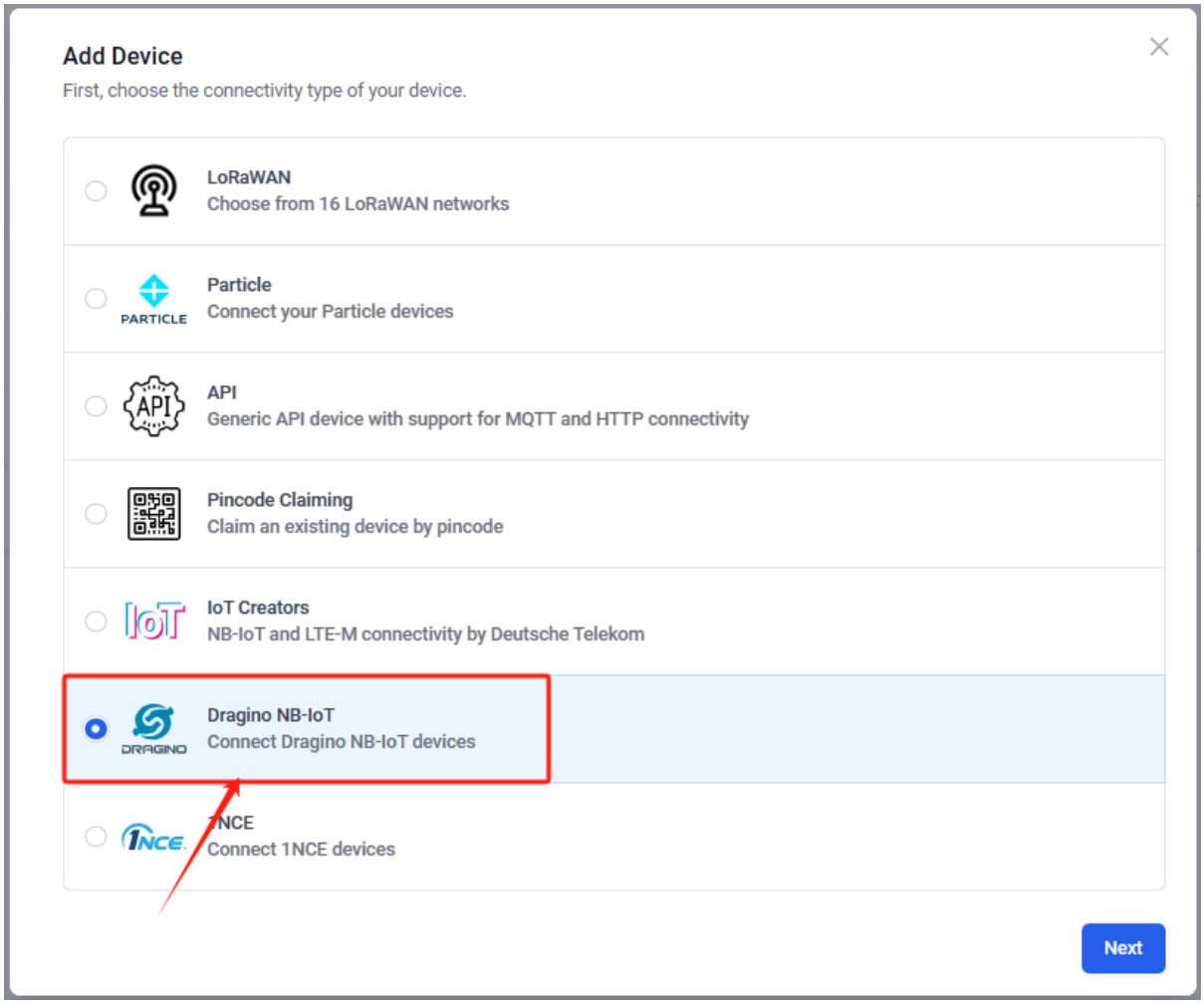
By Default, the device is registered in Dragino's DataCake Account. User can Claim it to his account.

3.4.3 Manual Add Decoder in DataCake (don't use the template in DataCake)

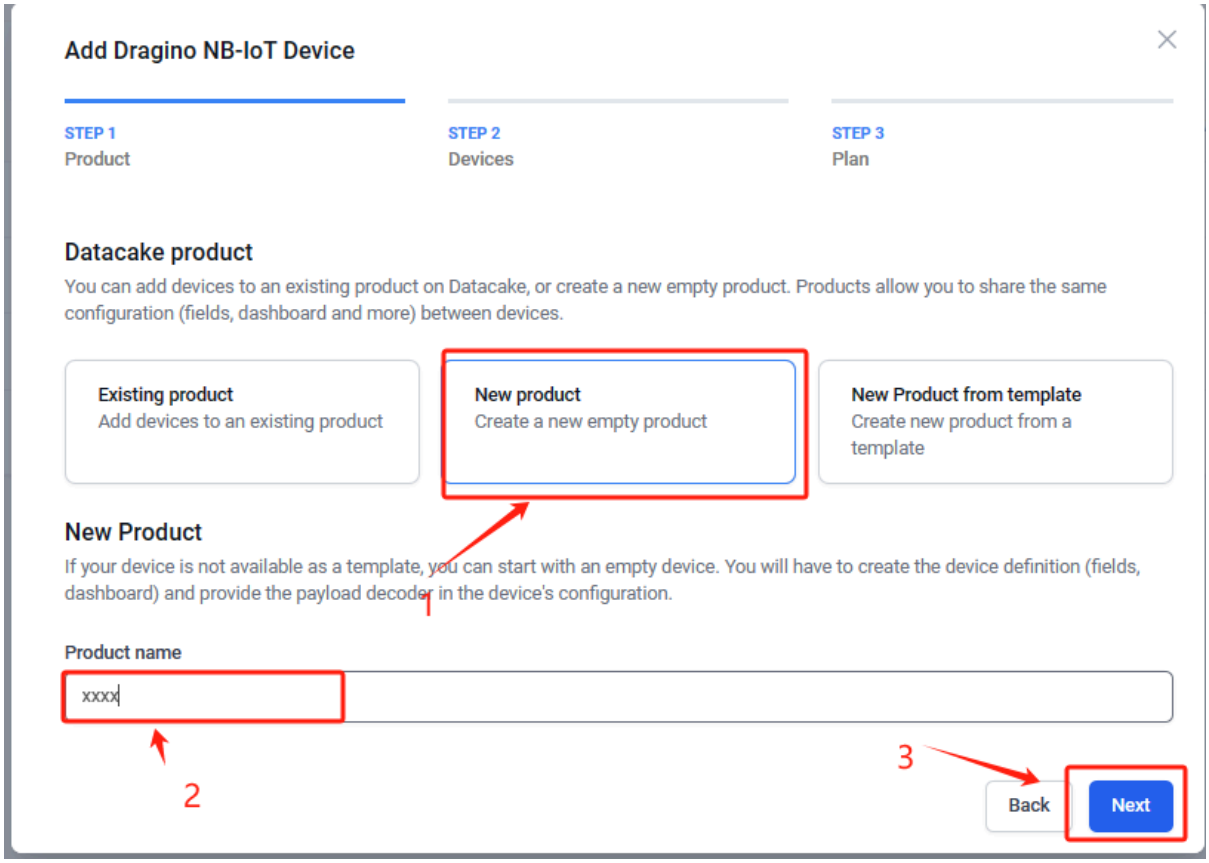
Step1: Add a device



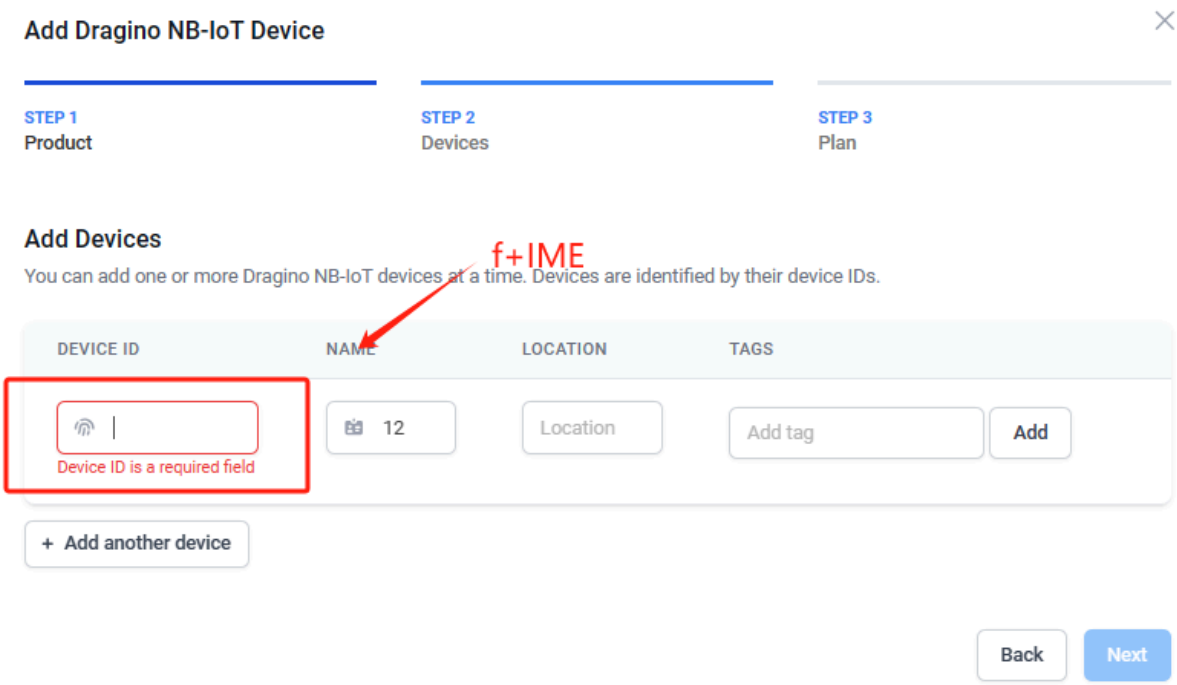
Step2: Choose your device type,please select dragino NB-IOT device



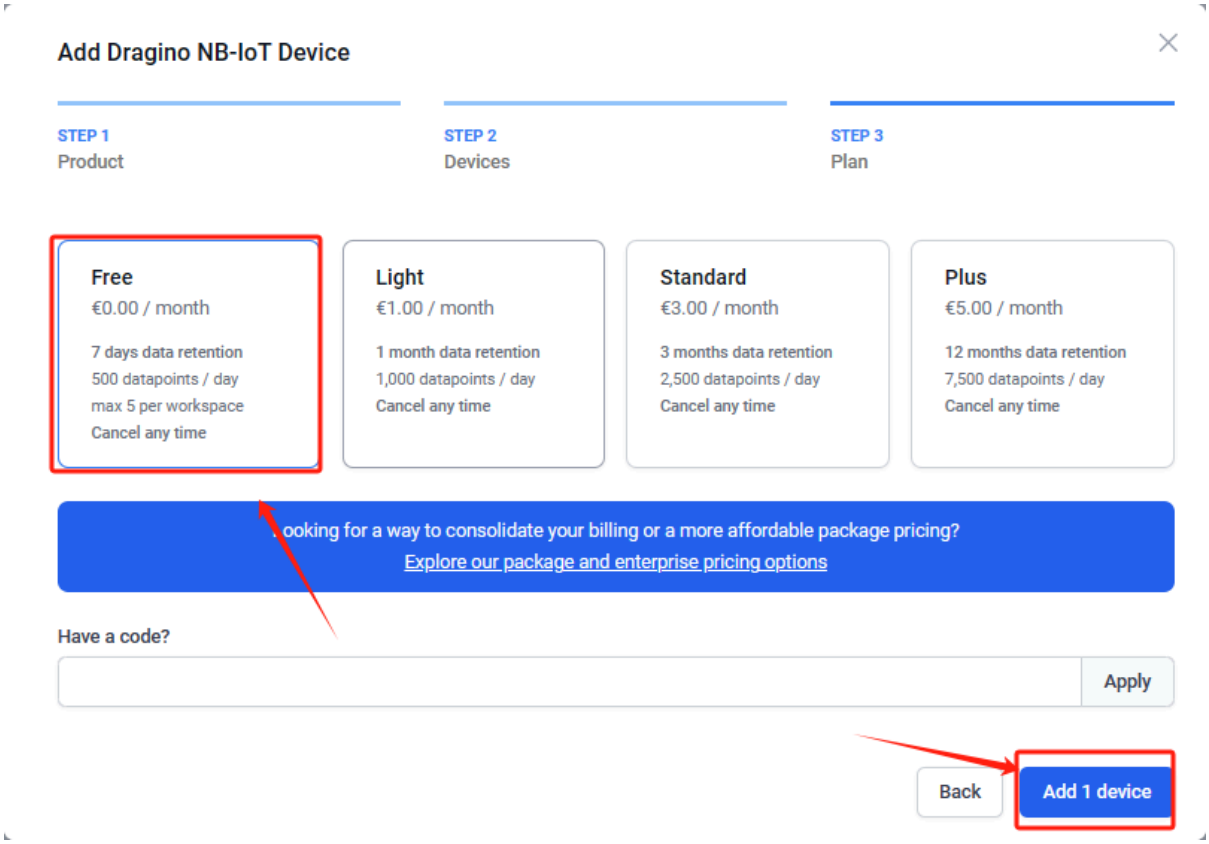
Step3: Choose to create a new device



Step4: Fill in the device ID of your NB device



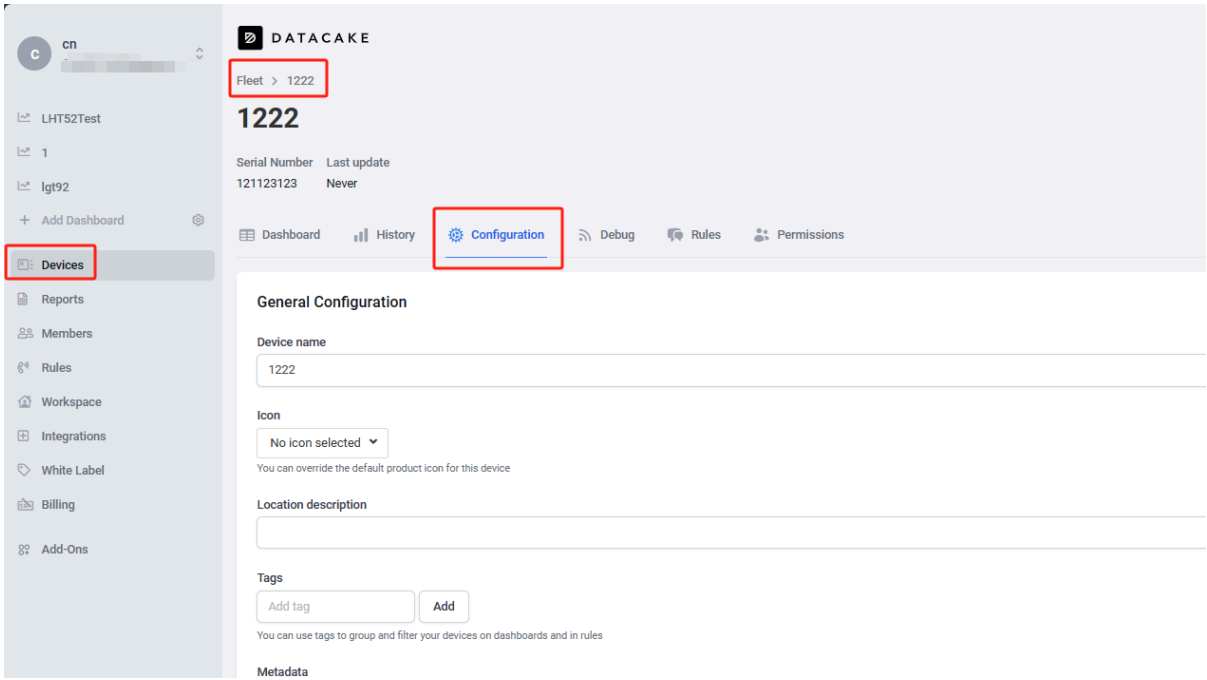
Step5: Please select your device plan according to your needs and complete the creation of the device

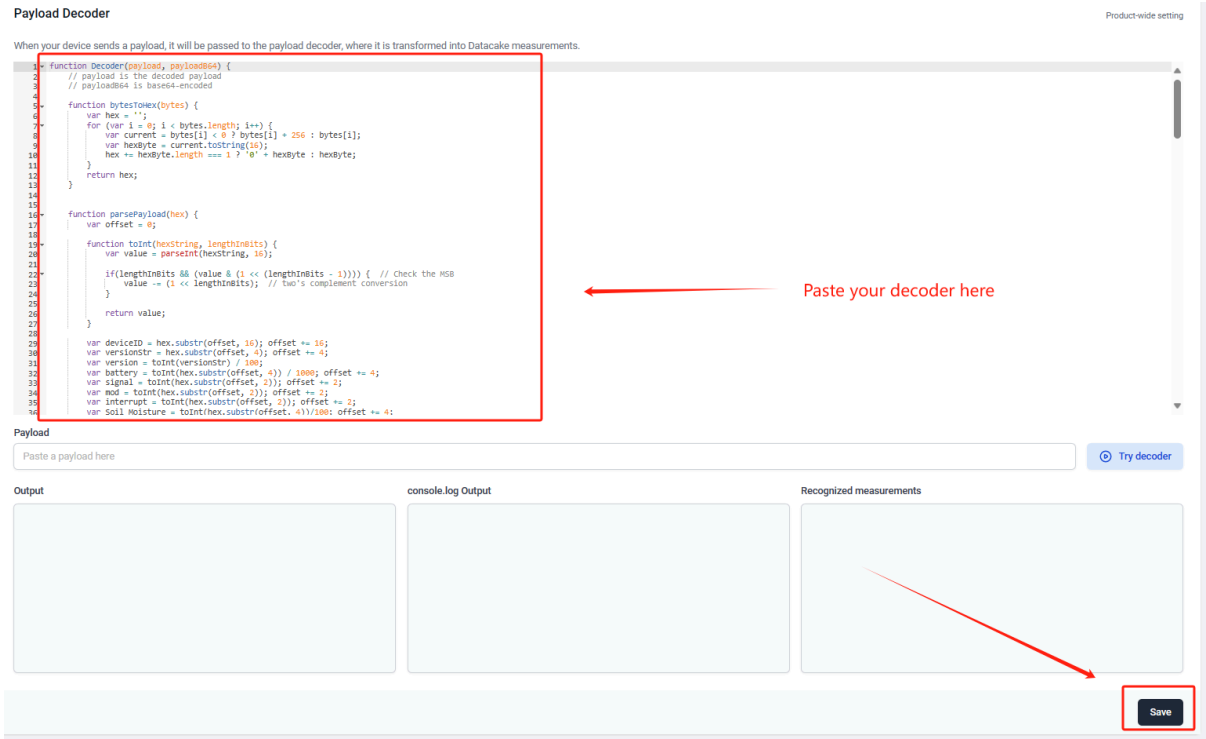


Step6: Please add the decoder at the payload decoder of the device configuration.

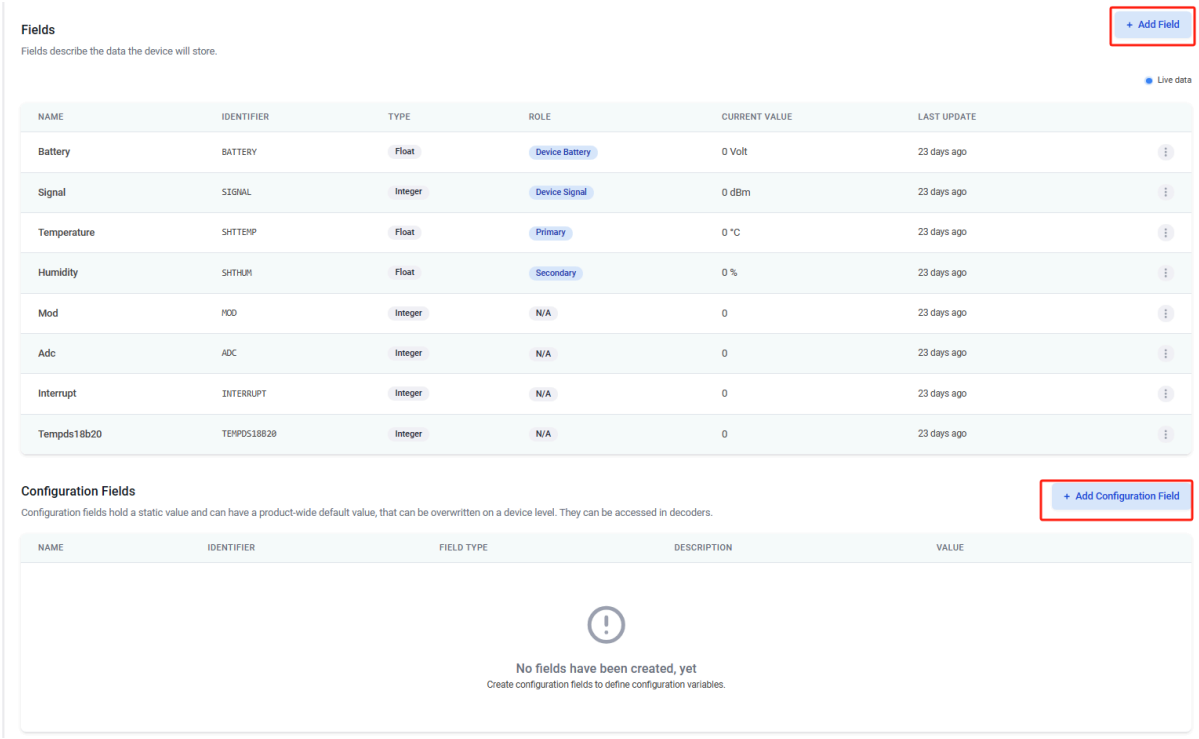
Decoder location: dragino-end-node-decoder/Datacake-Dragino_NB at main · dragino/dragino-end-node-decoder (github.com)

Due to version update, please use the following decoder for the new version firmware: dragino-end-node-decoder/Datacake-Dragino_NB_New_Version at main · dragino/dragino-end-node-decoder (github.com)

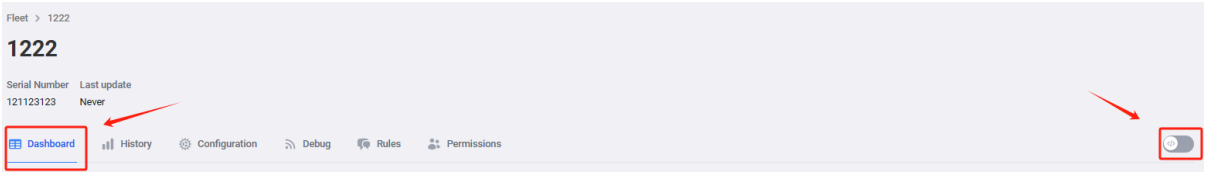

cn
LHT52Test
1
lgt92
Add Dashboard
Devices
Reports
Members
Rules
Workspace
Integrations
White Label
Billing
Add-Ons
DATACAKE
Fleet > 1222
1222
Serial Number
121123123
Last update
Never
Dashboard
History
Configuration
Debug
Rules
Permissions
General Configuration
Device name
1222
Icon
No icon selected
You can override the default product icon for this device
Location description
Tags
Add tag
Add
You can use tags to group and filter your devices on dashboards and in rules
Metadata

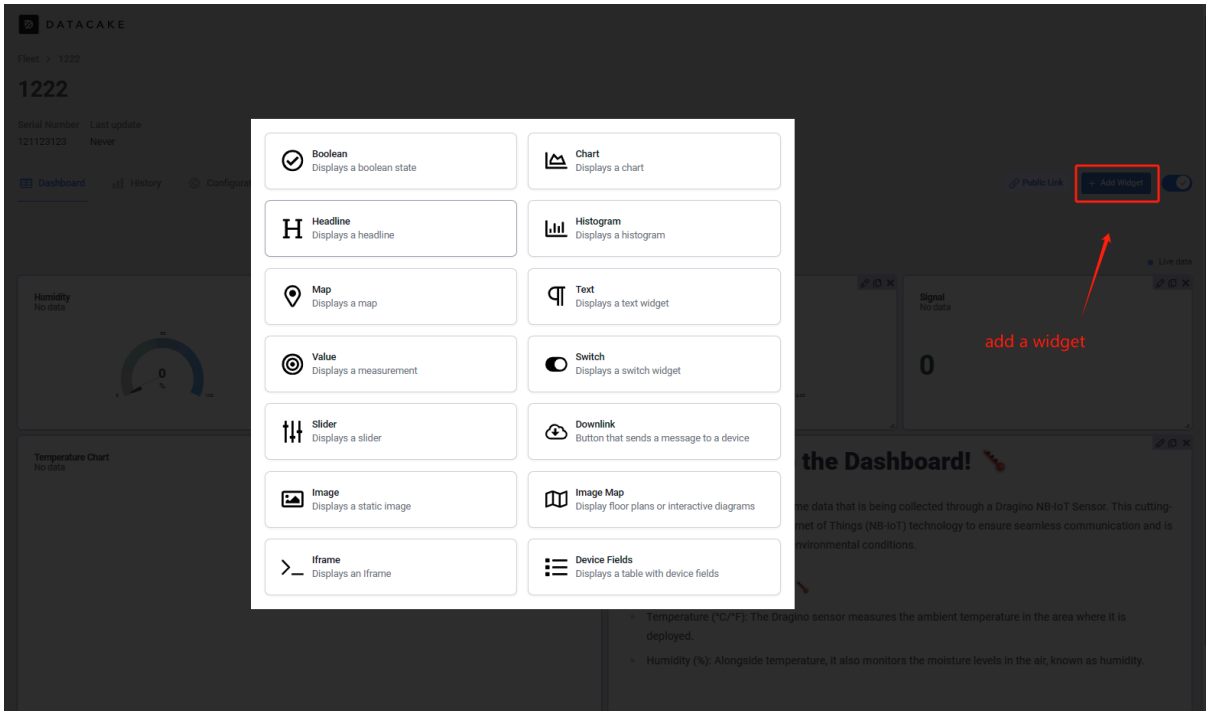


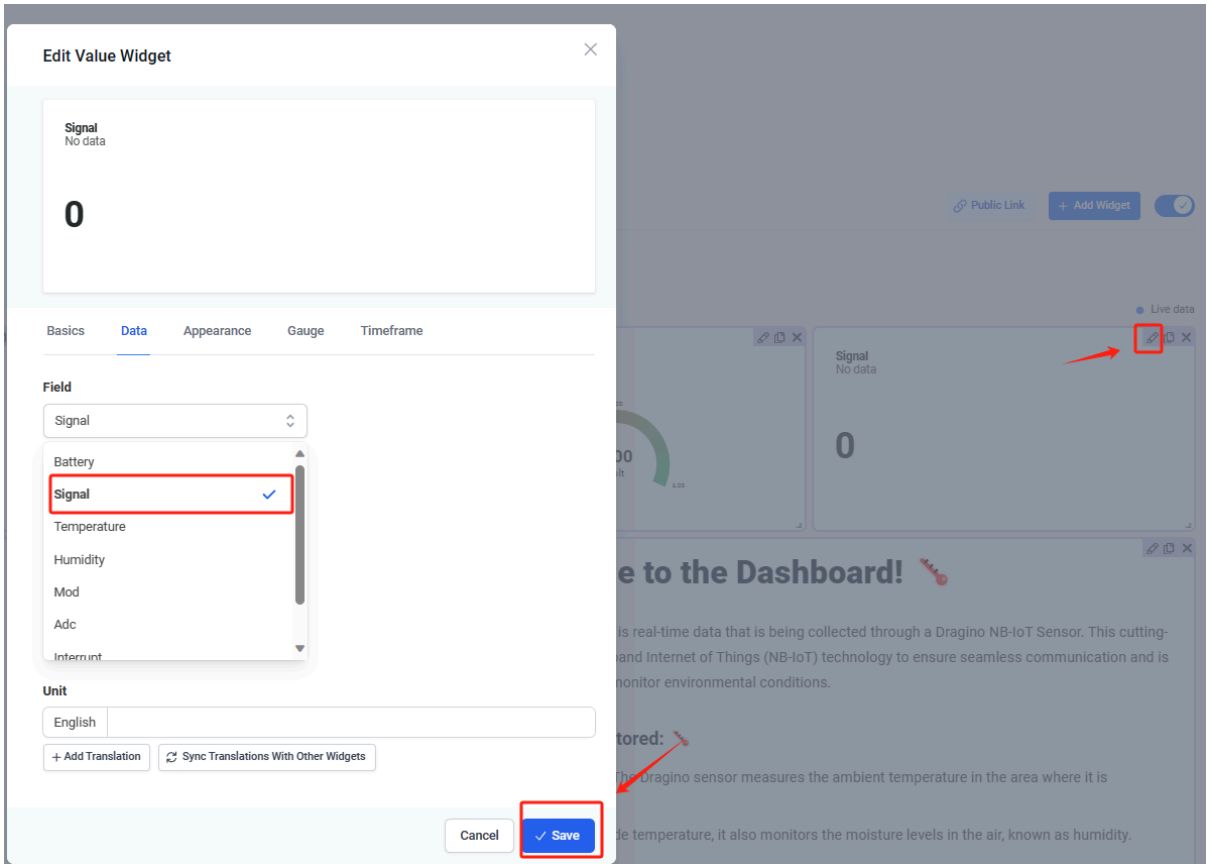
Step7: Add the output of the decoder as a field



Step8: Customize the dashboard and use fields as parameters of the dashboard




DATACAKE
1222
Boolean
Displays a boolean state
Chart
Displays a chart
Headline
Displays a headline
Histogram
Displays a histogram
Map
Displays a map
Text
Displays a text widget
Value
Displays a measurement
Switch
Displays a switch widget
Slider
Displays a slider
Downlink
Button that sends a message to a device
Image
Displays a static image
Image Map
Display floor plans or interactive diagrams
Iframe
Displays an Iframe
Device Fields
Displays a table with device fields
add a widget
the Dashboard!



3.4.4 For device have not configured to connect to DataCake

Use AT command for connecting to DataCake

AT+PRO=2,0

AT+SERVADDR=67.207.76.90,4445

3.5 Node-Red (via MQTT)

3.5.1 Configure Node-Red

Take S31-NB UDP protocol as an example.

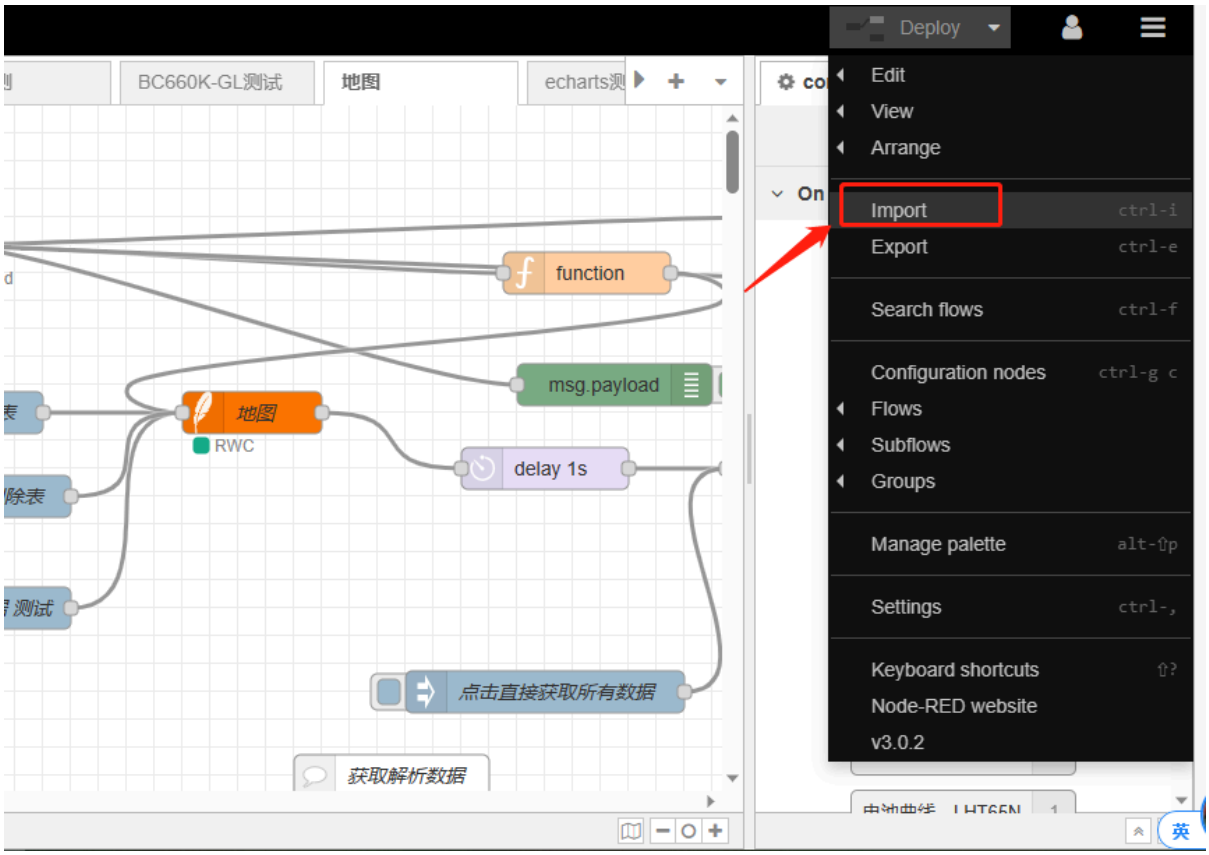
Dragino provides input flow examples for the sensors.

User can download the required JSON file through Dragino Node-RED input flow template.

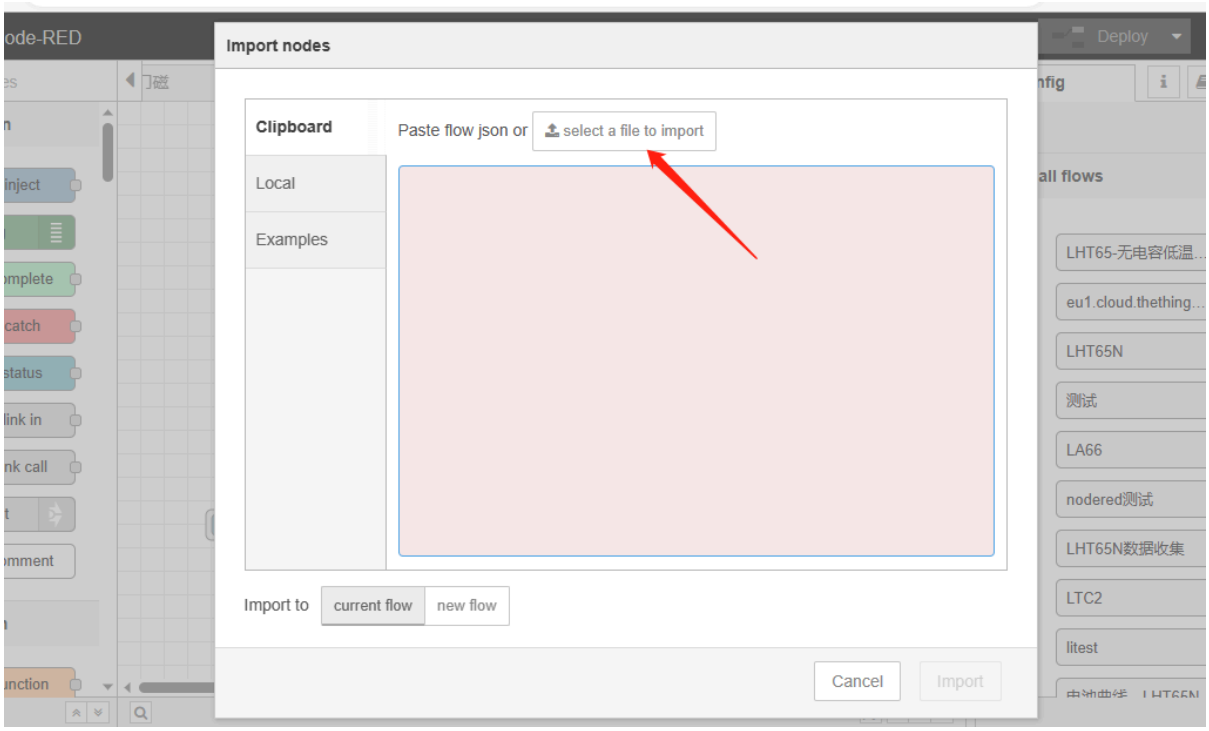
Download sample JSON file link: https://www.dropbox.com/sh/mduw85jcuwsua22/AAAvwPhg9z6dLjJhmZjqBf_ma?dl=0

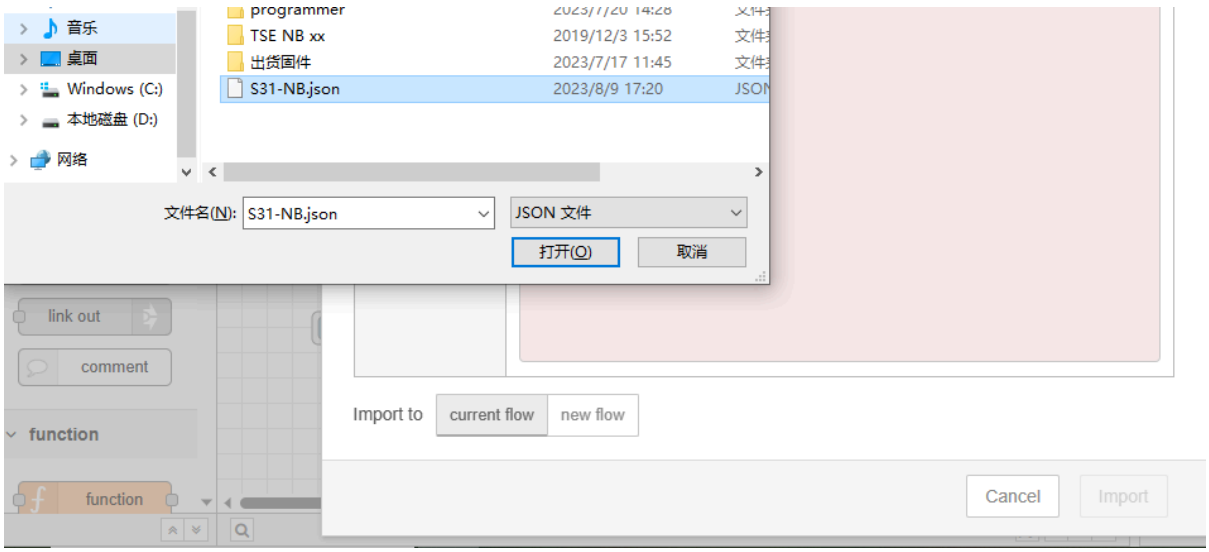
We can directly import the template.

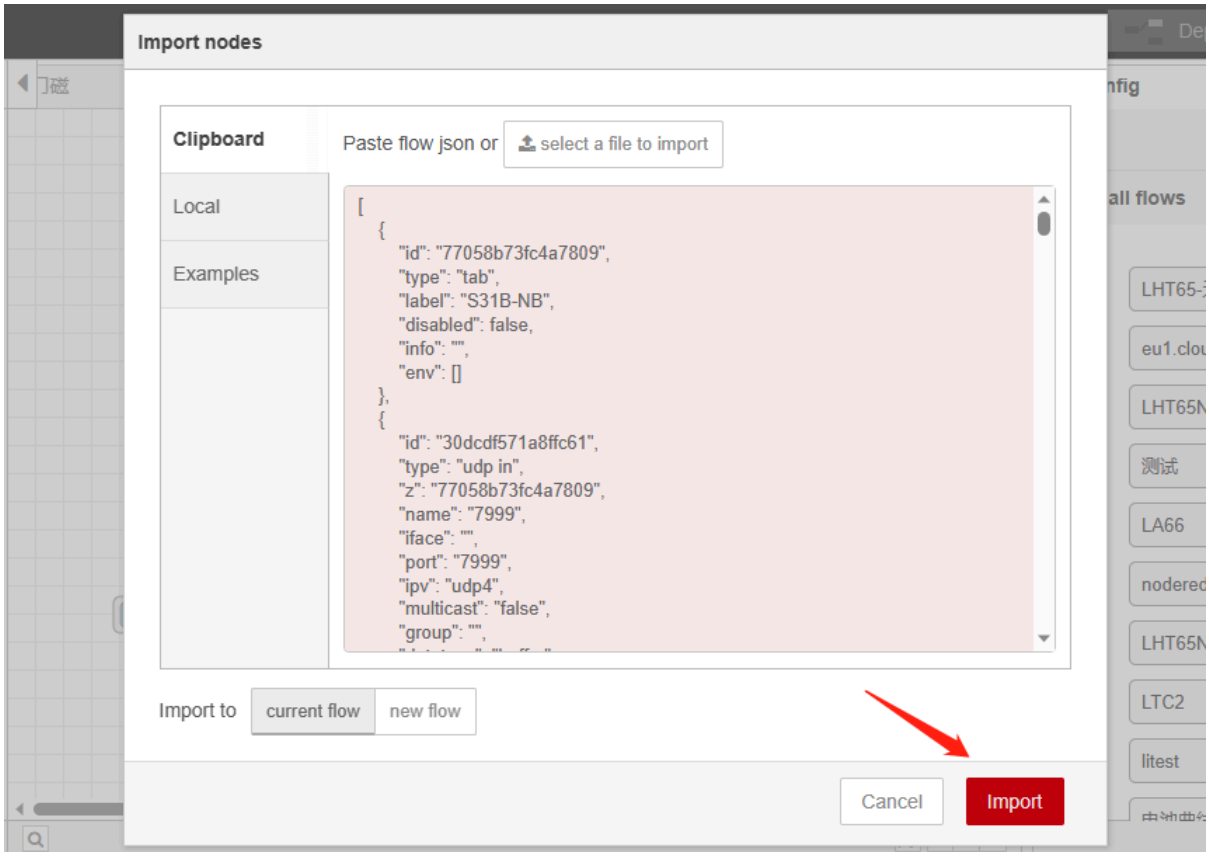
The templates for S31-NB and NB95S31B are the same.



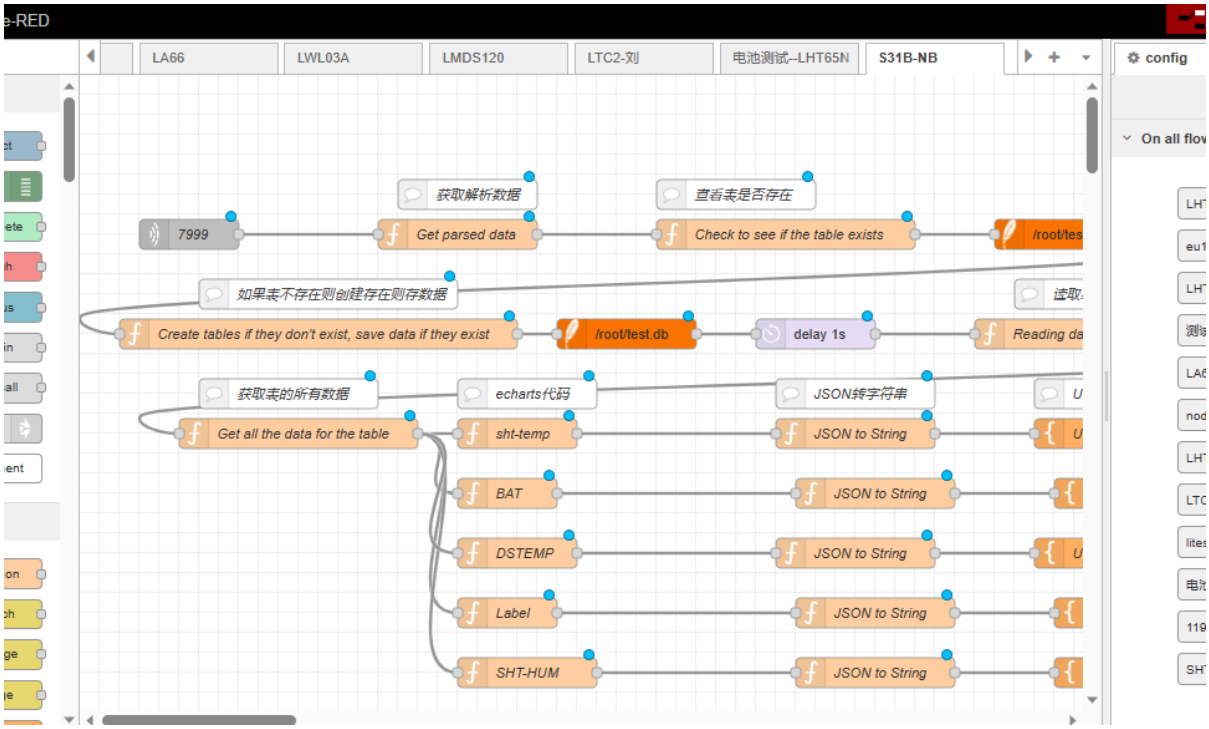
Please select the NB95S31B template.


Import nodes
Clipboard
Paste flow json or
select a file to import
Local
Examples
Import to
current flow
new flow
Cancel
Import
Deploy
all flows
LHT65-无电容低温...
eu1.cloud.thething...
LHT65N
测试
LA66
nodered测试
LHT65N数据收集
LTC2
litest

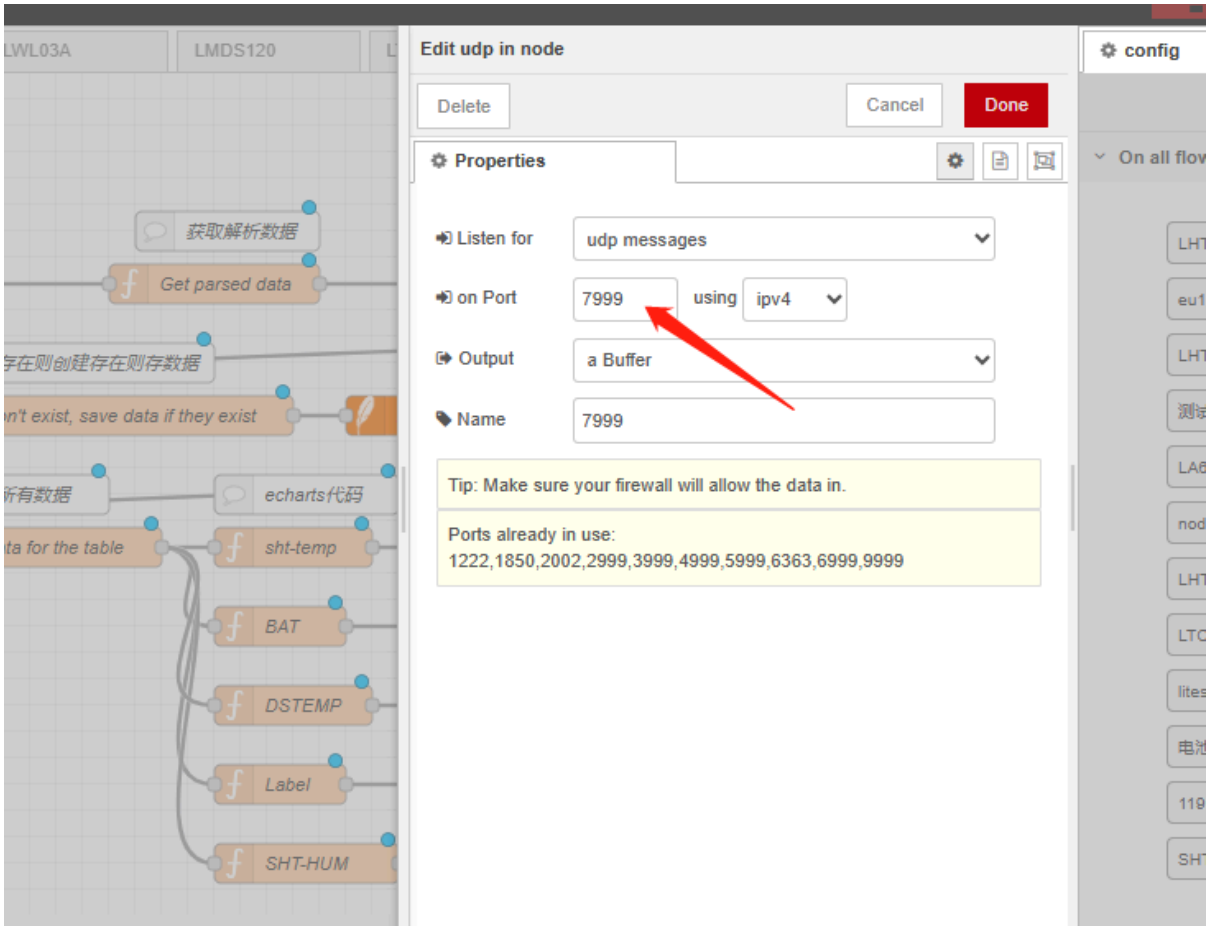

音乐
桌面
Windows (C:)
本地磁盘 (D:)
网络
programmer
TSE NB xx
2019/12/3 15:52
出货固件
2023/7/17 11:45
S31-NB.json
2023/8/9 17:20
文件名(N): S31-NB.json
JSON 文件
打开(O)
取消
link out
comment
function
function
Import to
current flow
new flow
Cancel
Import



Successfully imported template.

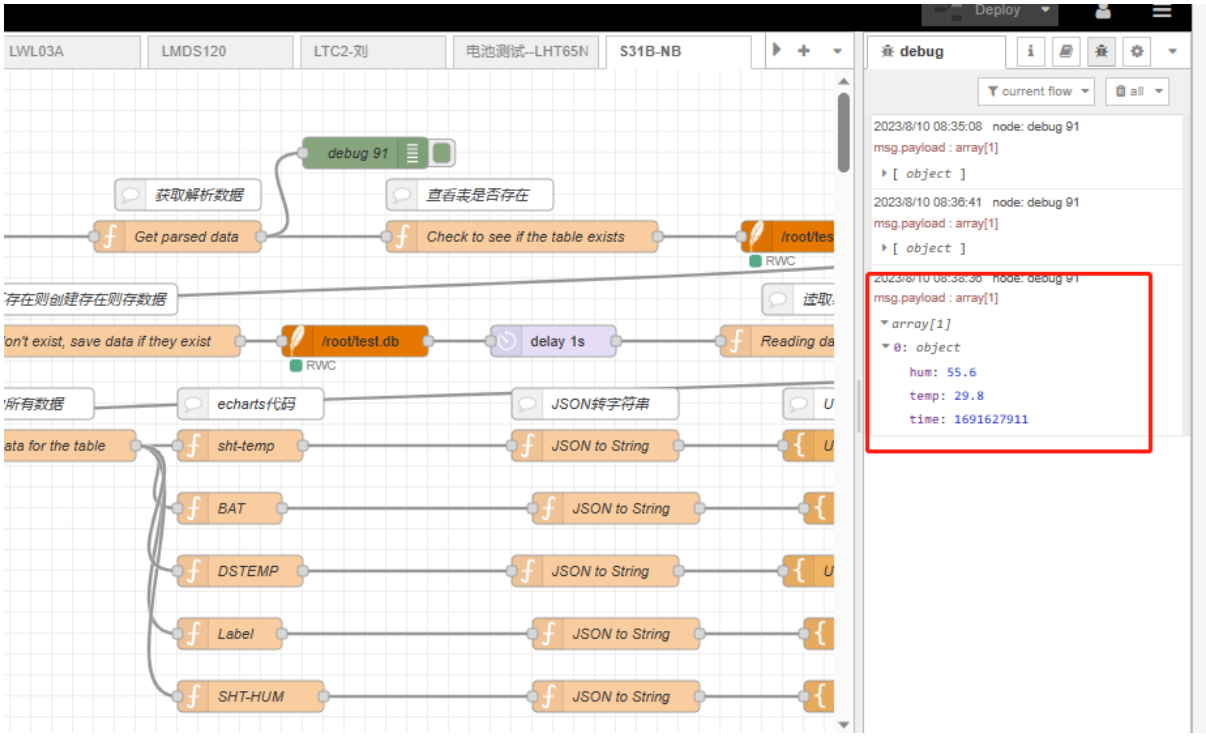


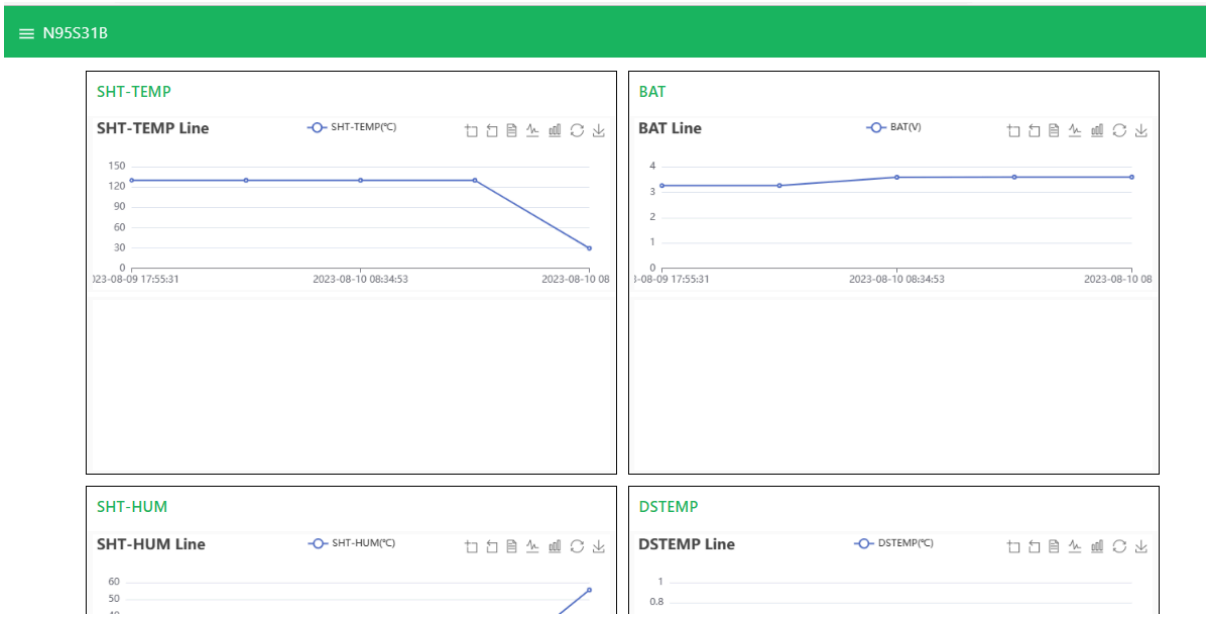
Users can set UDP port.



3.5.2 Simulate Connection

We have completed the configuration of UDP. We can try sending packets to node red.


LWL03A
LMDS120
LTC2-刘
电池测试--LHT65N
S31B-NB
debug 91
获取解析数据
查看表是否存在
Get parsed data
Check to see if the table exists
/root/test.db
RWC
delay 1s
Reading da
echarts代码
JSON转字符串
sht-temp
JSON to String
BAT
DSTEMP
Label
SHT-HUM
debug
current flow
all
2023/8/10 08:35:08 node: debug 91
msg.payload : array[1]
[object]
2023/8/10 08:36:41 node: debug 91
msg.payload : array[1]
[object]
msg.payload : array[1]
array[1]
0: object
hum: 55.6
temp: 29.8
time: 1691627911



3.5.3 Configure NB-IoT Sensors

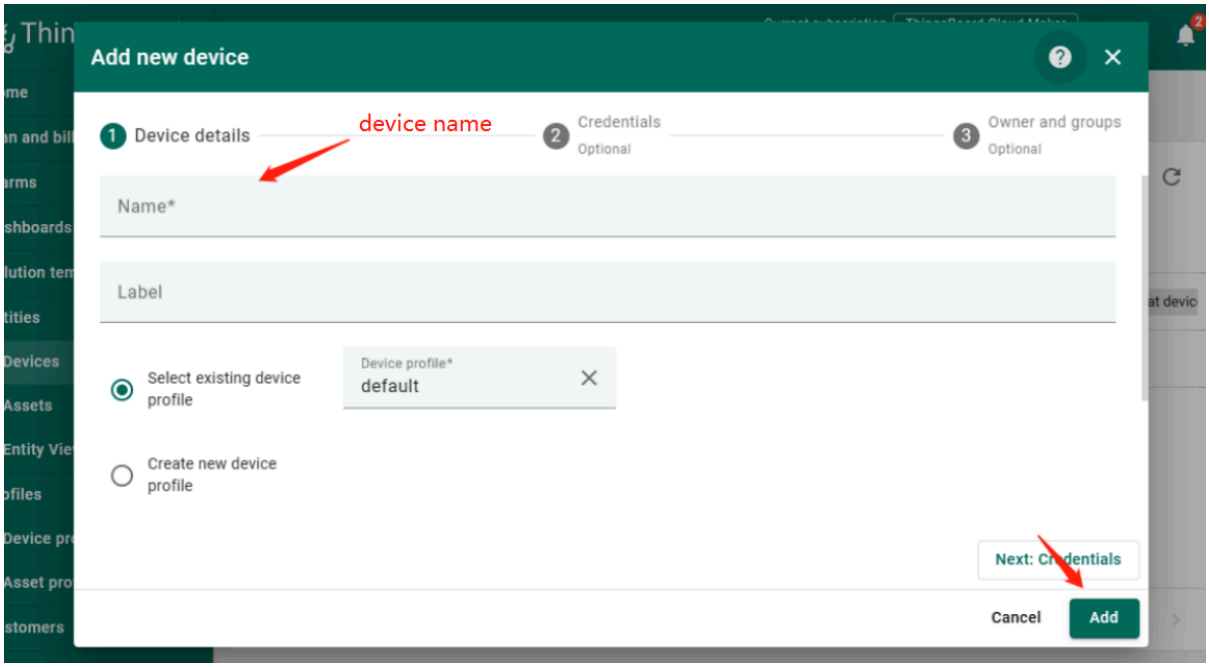
- **AT+PRO=3,0 or 3,5** // **hex format or json format**
- **AT+SUBTOPIC=<device name>or User Defined**
- **AT+PUBTOPIC=<device name>or User Defined**
- **AT+CLIENT=<device name> or User Defined**
- **AT+UNAME=<device name> or User Defined**
- **AT+PWD="Your device token"**

3.6 ThingsBoard.Cloud (via MQTT)

3.6.1 Configure ThingsBoard

3.6.1.1 Create Device

Create a New Device in ThingsBoard. Record Device Name which is used for MQTT connection.

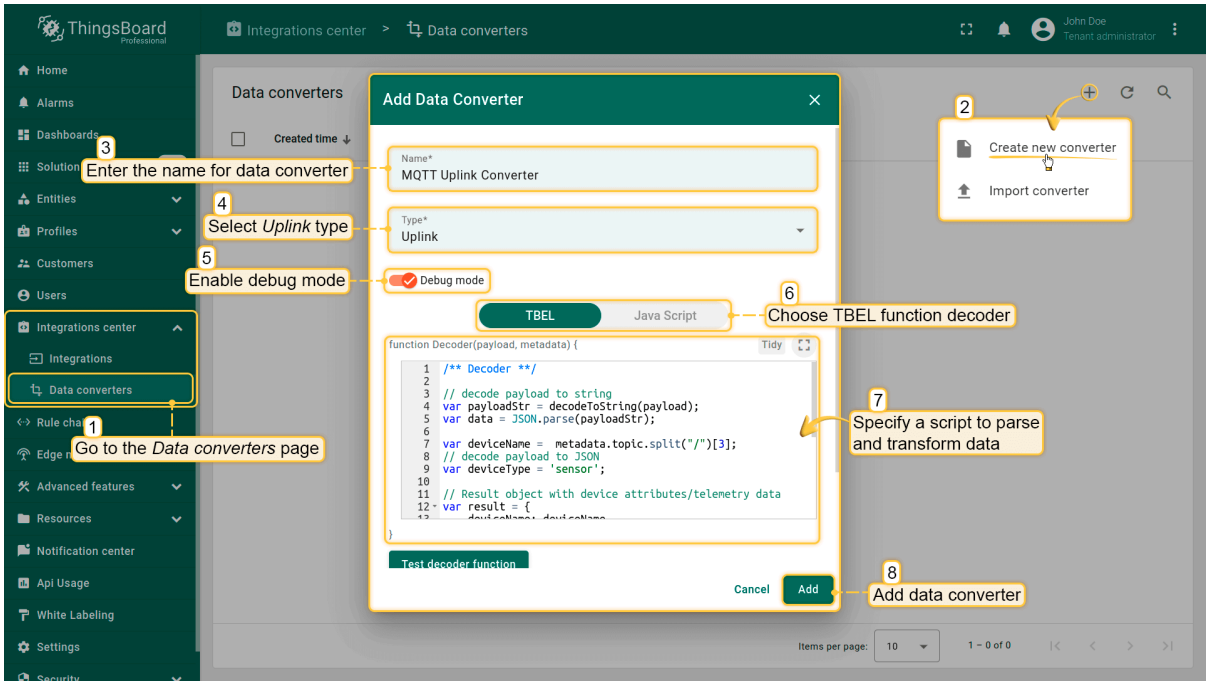


3.6.1.2 Create Uplink & Downlink Converter

Uplink Converter

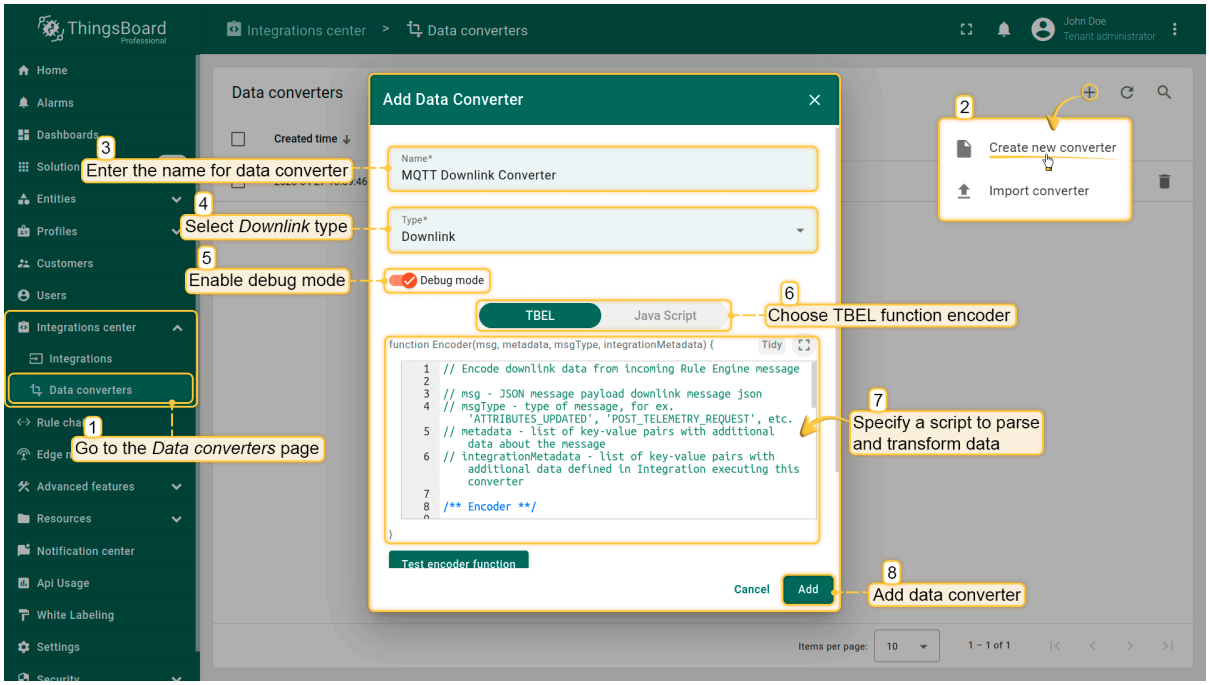
The purpose of the decoder function is to parse the incoming data and metadata to a format that ThingsBoard can consume. device value objects. Nested objects are not supported.

To create an uplink converter go to the**Integrations center** -
> **Data converters** page and click **“plus”** button. Name it **“MQTT Uplink Converter”** and select type **"Uplink"**. Use debug mode for now.



Downlink Converter

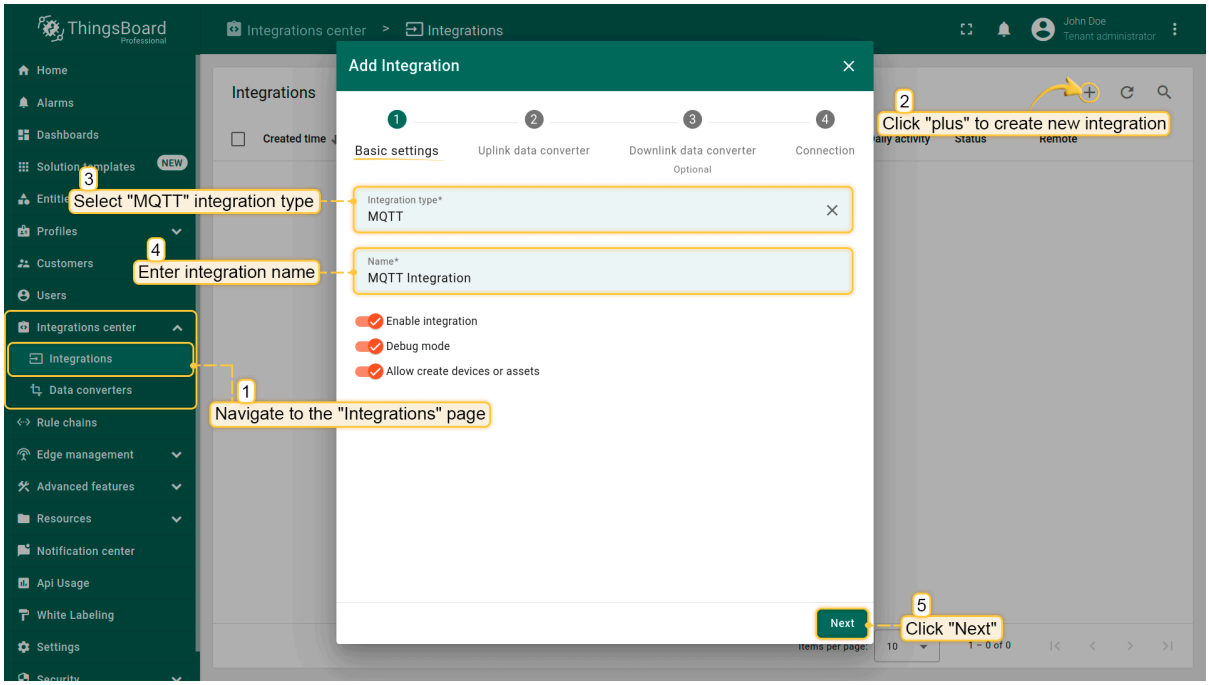
The Downlink converter transforming outgoing RPC message and then the Integration sends it to external MQTT broke



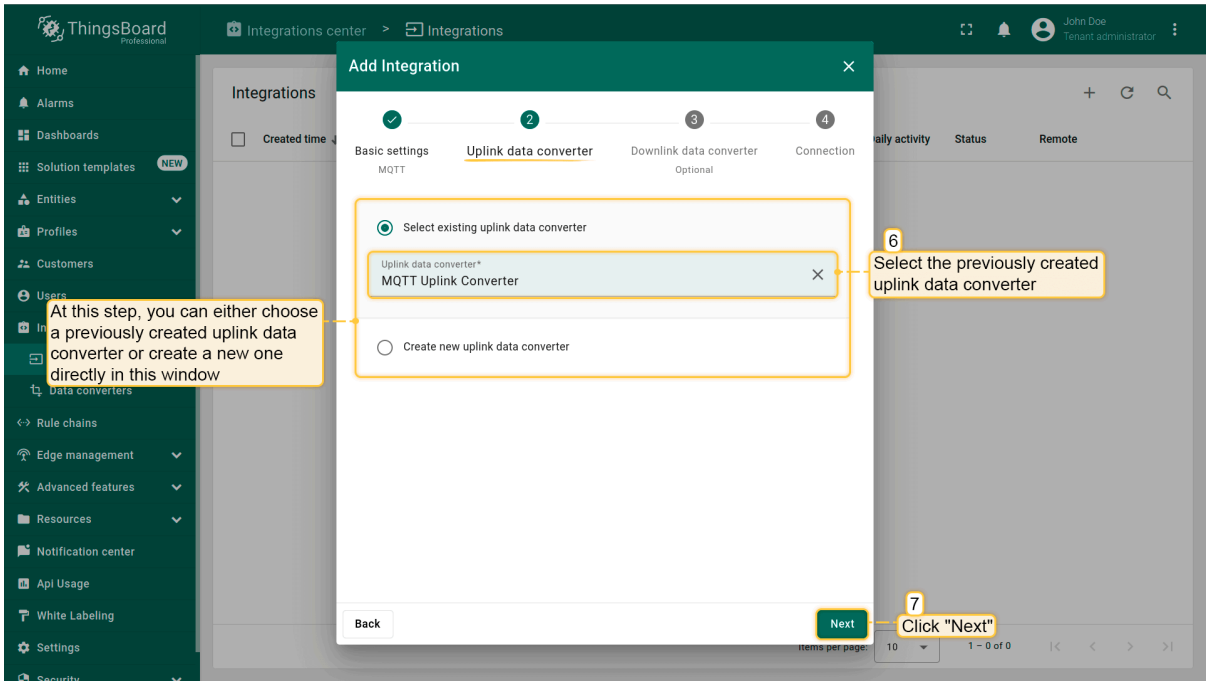
Note:
Our device payload is already human readable data. Therefore, users do not need to write decoders. Simply create by def

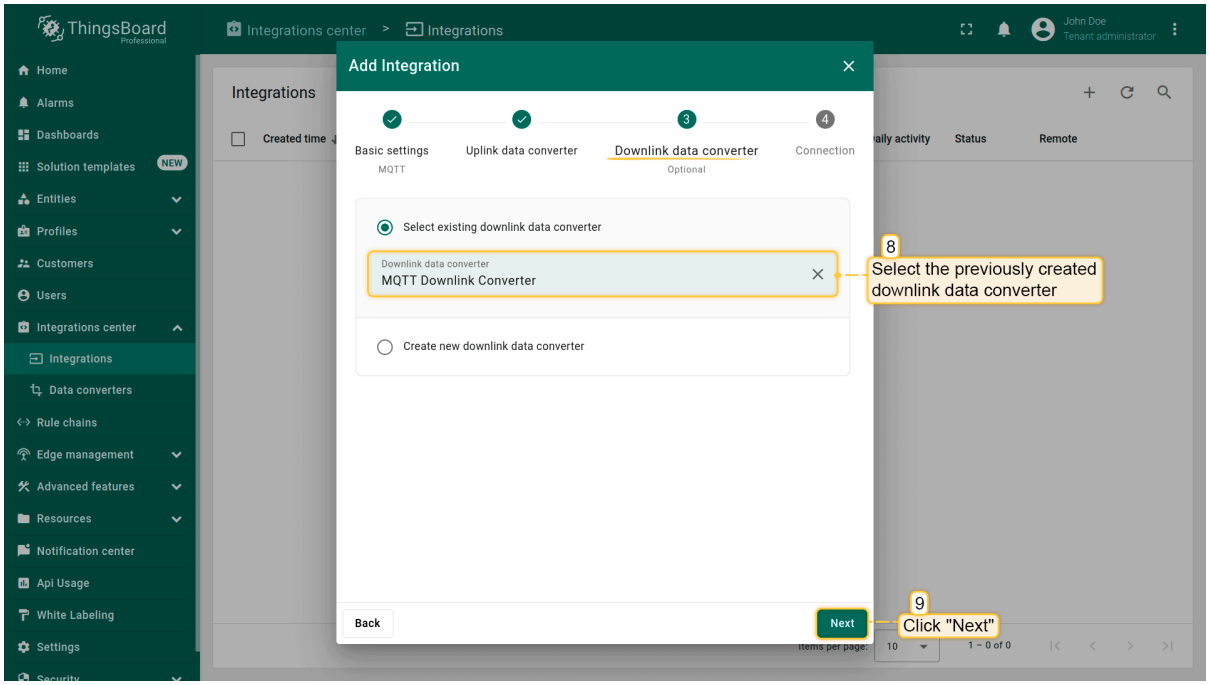
3.6.1.3 MQTT Integration Setup

Go to the **Integrations center -> Integrations page** and click "**plus**" icon to add a new integration. Name it"**MQTT Integration**", select type **MQTT**;



- The next steps is to add the recently created uplink and downlink converters;

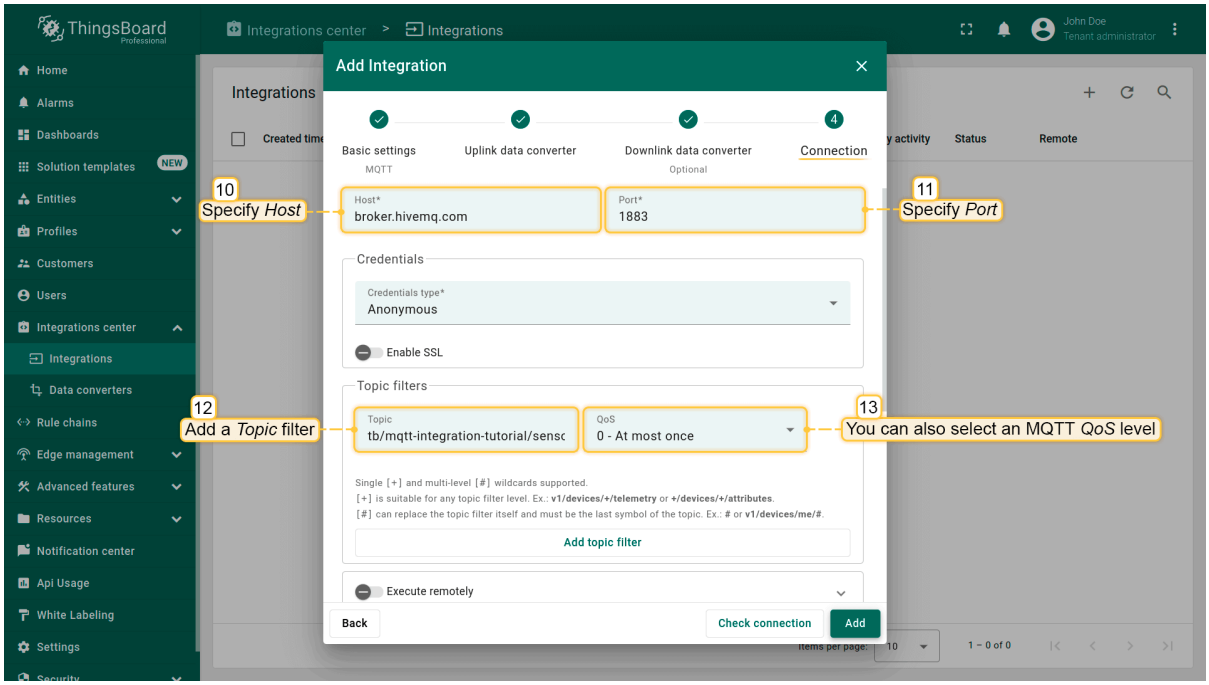

ThingsBoard
Professional
Integrations center > Integrations
John Doe
Tenant administrator
Home
Alarms
Dashboards
Solution templates
NEW
Entities
Profiles
Customers
Users
Data converters
Rule chains
Edge management
Advanced features
Resources
Notification center
Api Usage
White Labeling
Settings
Integrations
Created time
Status
Remote
Add Integration
Basic settings
MQTT
Uplink data converter
Downlink data converter
Optional
Connection
Select existing uplink data converter
Uplink data converter*
MQTT Uplink Converter
Create new uplink data converter
Back
Next
At this step, you can either choose a previously created uplink data converter or create a new one directly in this window
6
Select the previously created uplink data converter
7
Click "Next"



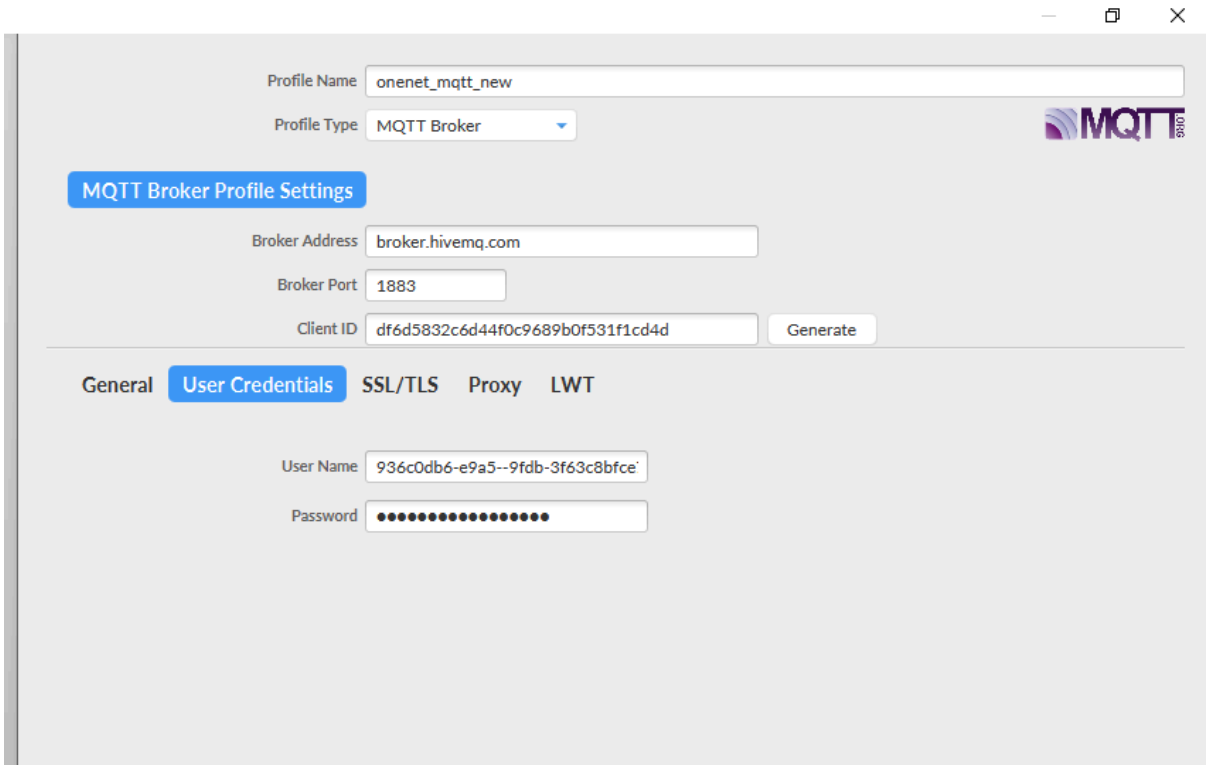
Add a topic filter:

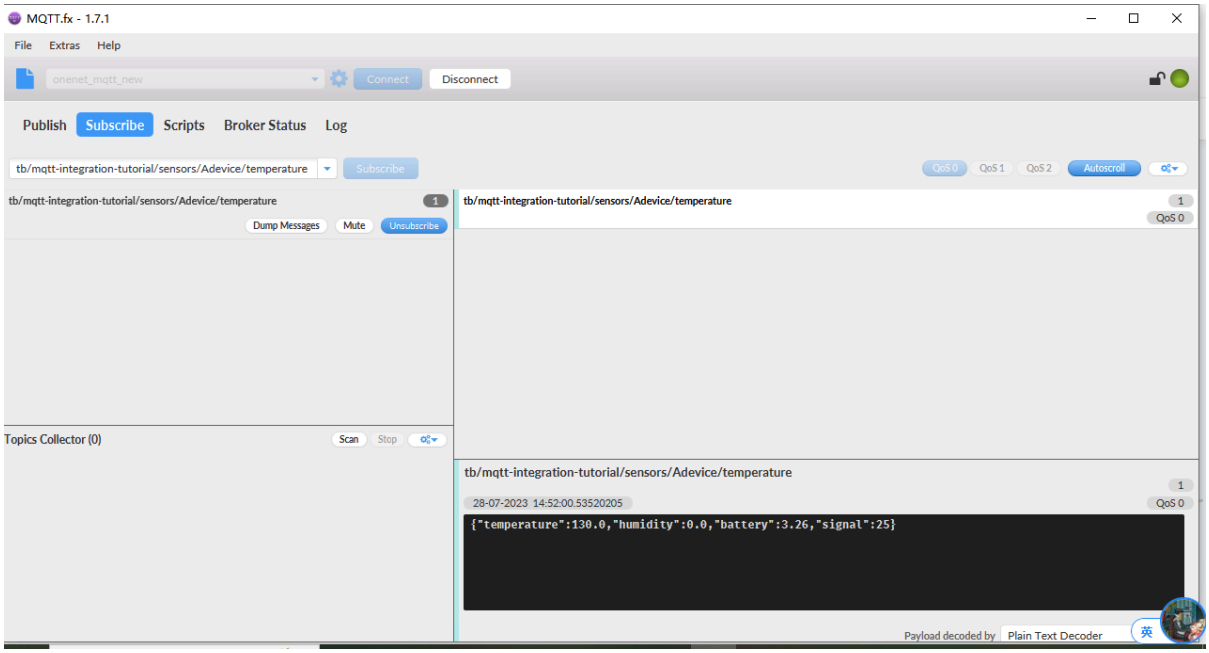
Consistent with the theme of the node setting.

You can also select an MQTT QoS level. We use MQTT QoS level 0 (At most once) by default;



3.6.2 Simulate with MQTT.fx


Profile Name
onenet_mqtt_new
Profile Type
MQTT Broker
MQTT Broker Profile Settings
Broker Address
broker.hivemq.com
Broker Port
1883
Client ID
df6d5832c6d44f0c9689b0f531f1cd4d
Generate
General
User Credentials
SSL/TLS
Proxy
LWT
User Name
936c0db6-e9a5--9fdb-3f63c8bfce
Password

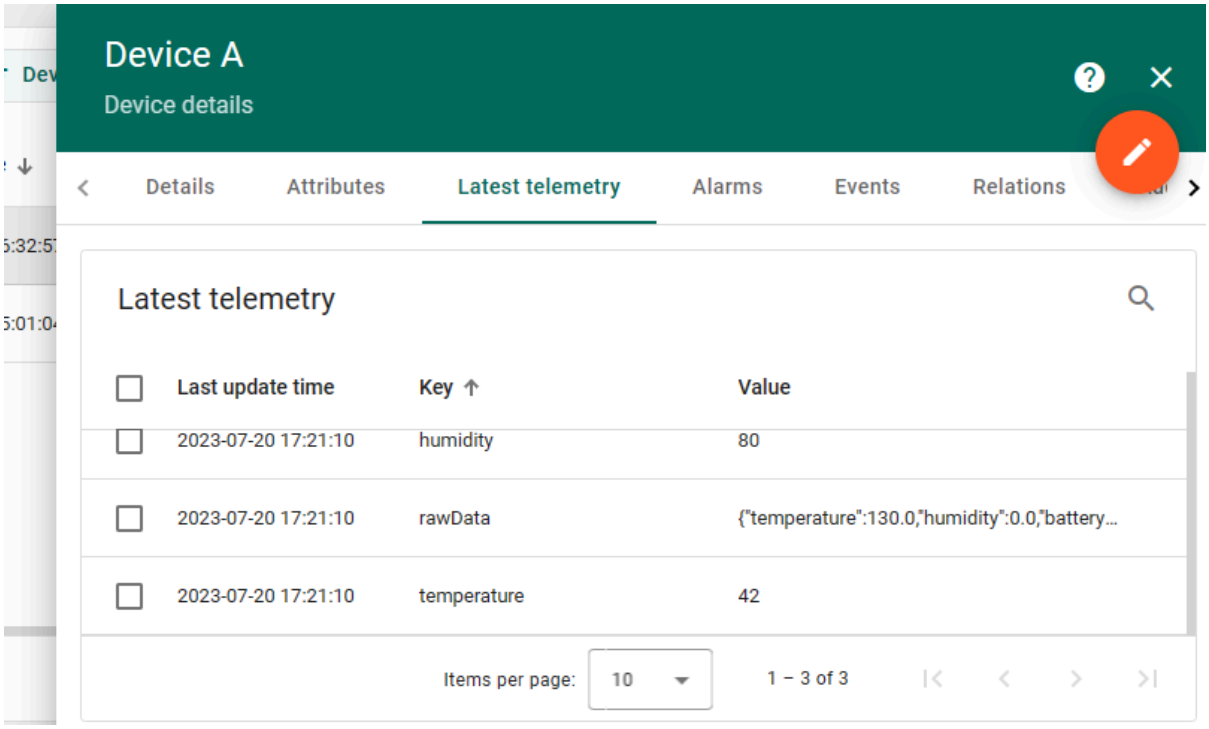


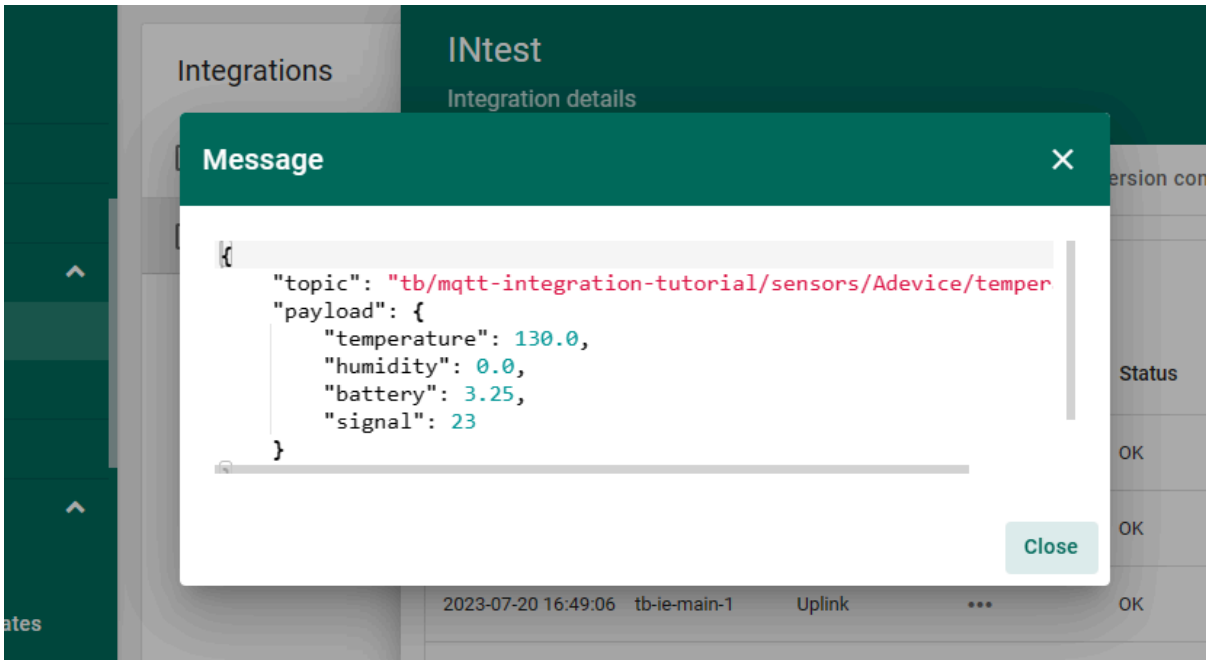
3.6.3 Configure NB-IoT Sensor

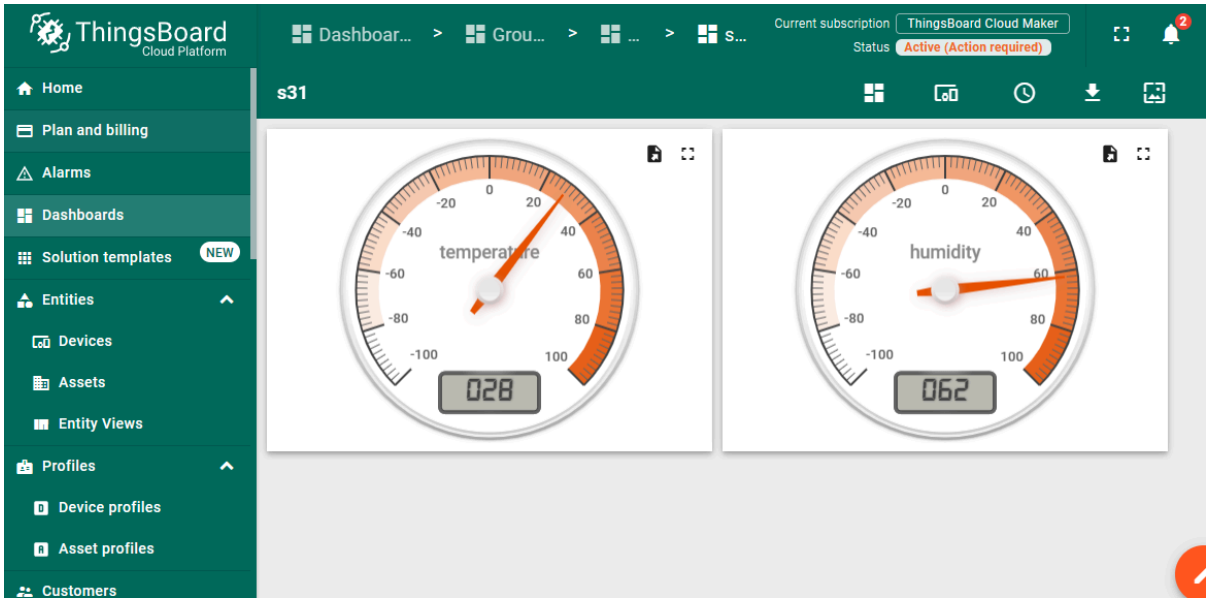
AT Commands

- **AT+PRO=3,3** // Use MQTT to connect to ThingsBoard. Payload Type set to 3.
- **AT+SUBTOPIC=<device name>**
- **AT+PUBTOPIC=<device name>**
- **AT+CLIENT=<device name> or User Defined**
- **AT+UNAME=<device name> or User Defined**
- **AT+PWD=<device name> or User Defined**

Test Uplink by click the button for 1 second


Device A
Device details
Details
Attributes
Latest telemetry
Alarms
Events
Relations
Latest telemetry
Last update time
Key
Value
2023-07-20 17:21:10
humidity
80
2023-07-20 17:21:10
rawData
{"temperature":130.0,"humidity":0.0,"battery...
2023-07-20 17:21:10
temperature
42
Items per page:
10
1 – 3 of 3


Integrations
INtest
Integration details
Message
{
"topic": "tb/mqtt-integration-tutorial/sensors/Adevice/temper
"payload": {
"temperature": 130.0,
"humidity": 0.0,
"battery": 3.25,
"signal": 23
}
Close
Status
OK
OK
2023-07-20 16:49:06
tb-ie-main-1
Uplink
OK



3.7 ThingsBoard.Cloud (via COAP)

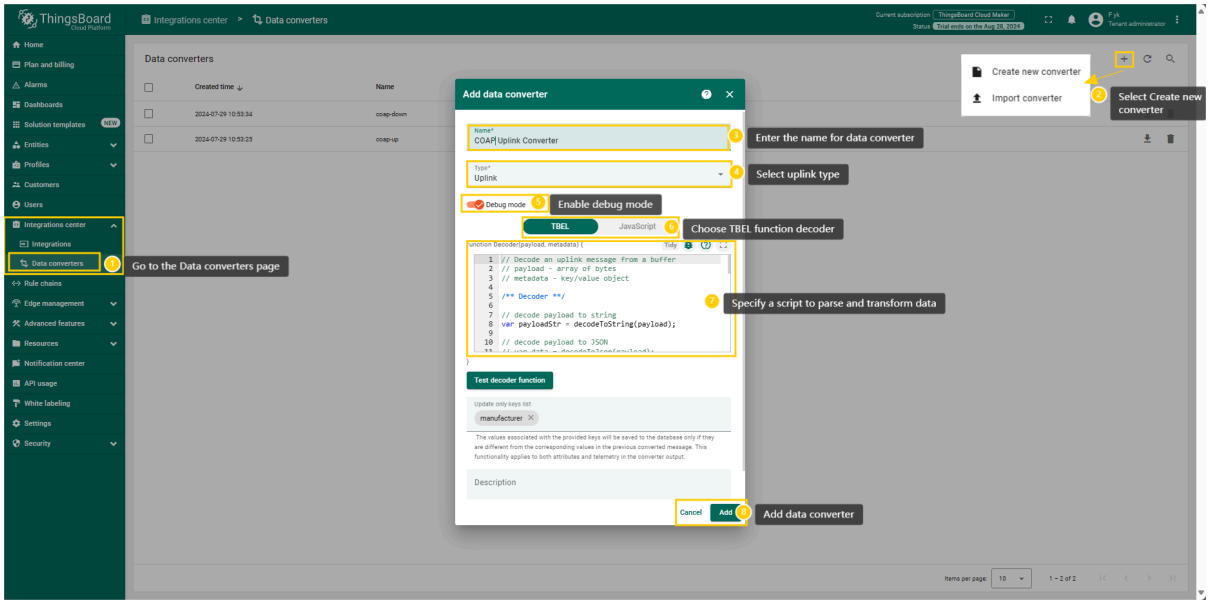
3.7.1 Configure ThingsBoard

3.7.1.1 Create Uplink & Downlink Converter

Uplink Converter

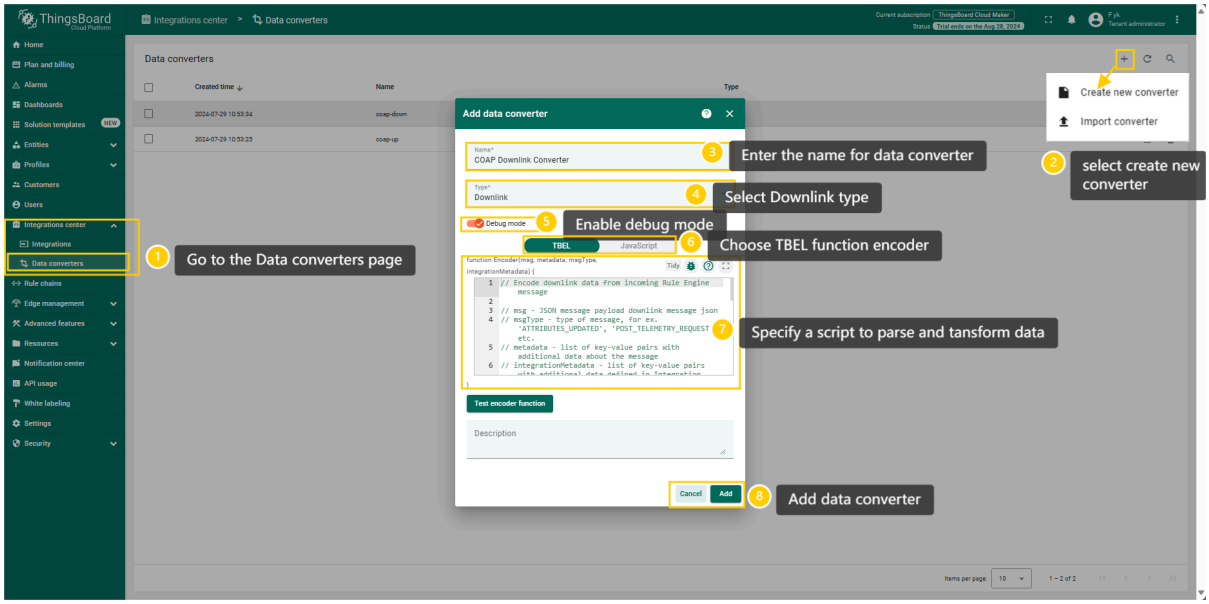
The purpose of the decoder function is to parse the incoming data and metadata to a format that ThingsBoard can consume. device value objects. Nested objects are not supported.

To create an uplink converter go to the**Integrations center** -
> **Data converters** page and click **“plus”** button. Name it **“COAP Uplink Converter”** and select type **"Uplink"**. Use debug mode for now.



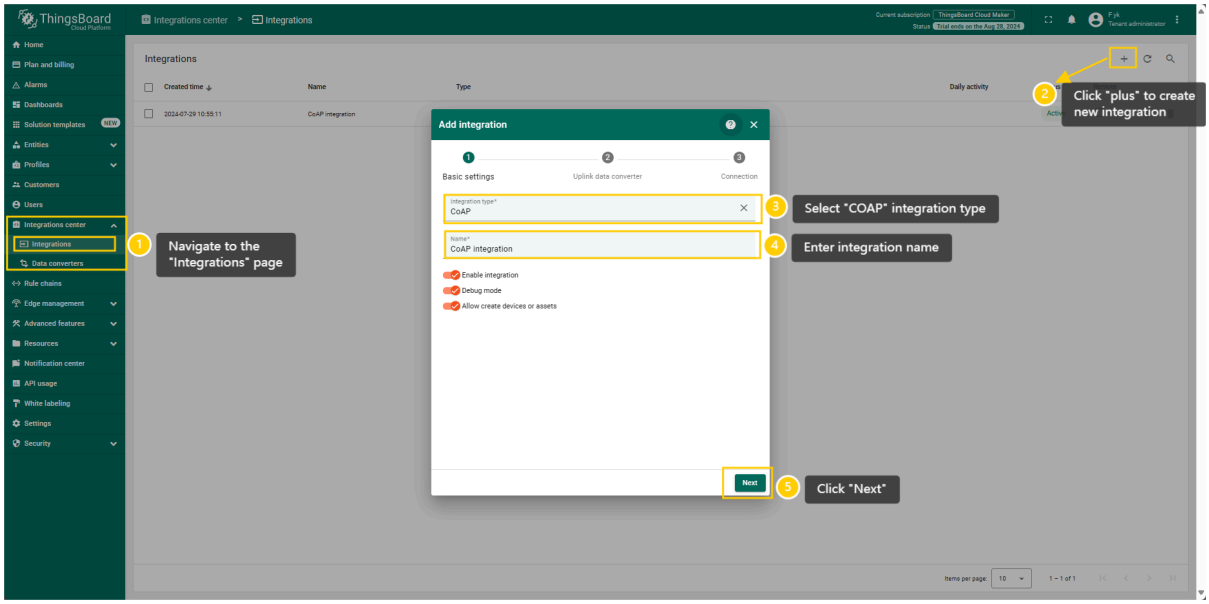
Downlink Converter

The Downlink converter transforming outgoing RPC message and then the Integration sends it to external COAP broker.

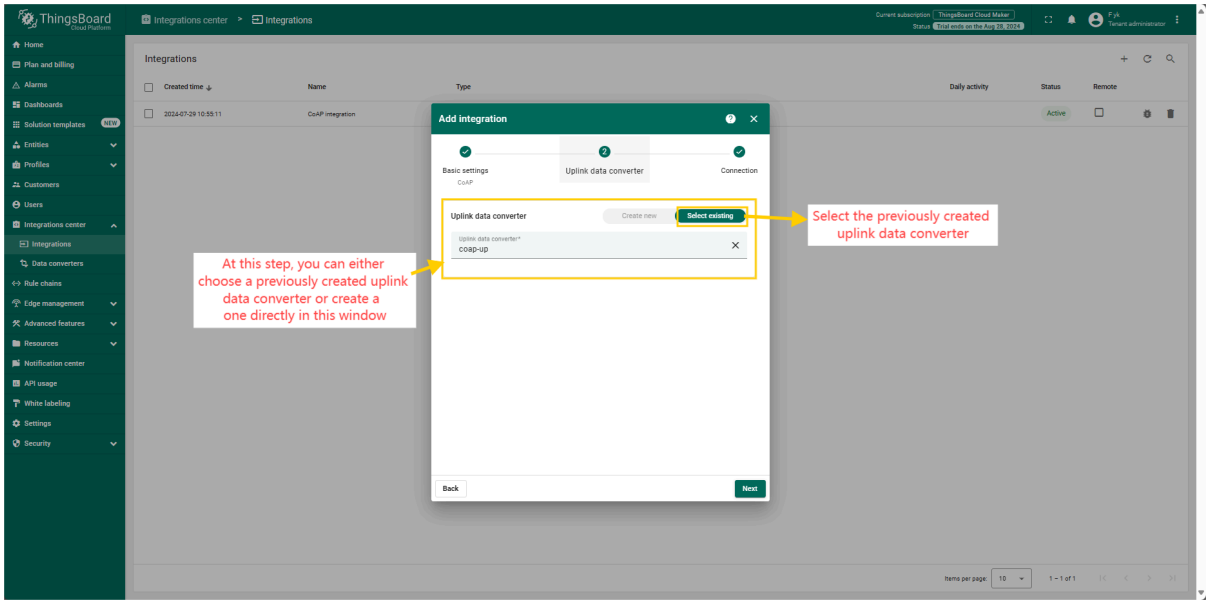


3.7.1.2 COAP Integration Setup

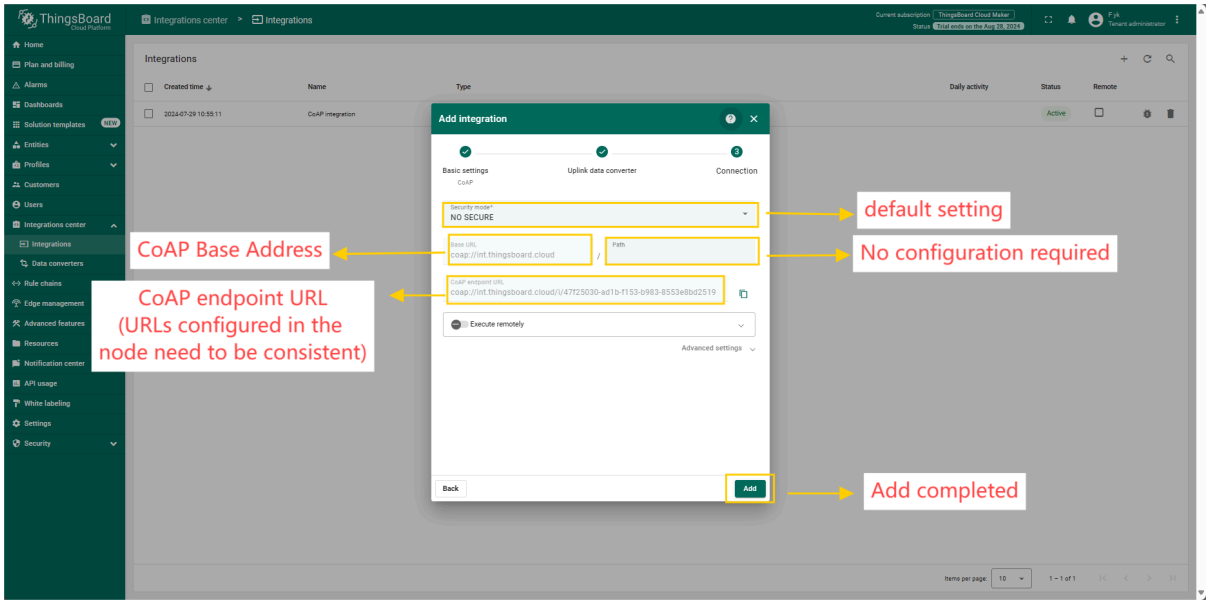
Go to the **Integrations center -> Integrations page** and click "**plus**" icon to add a new integration. Name it**"CoAP Integration"**, select type **COAP** ;



The next steps is to add the recently created uplink converters;



3.7.1.3 Add COAP Integration

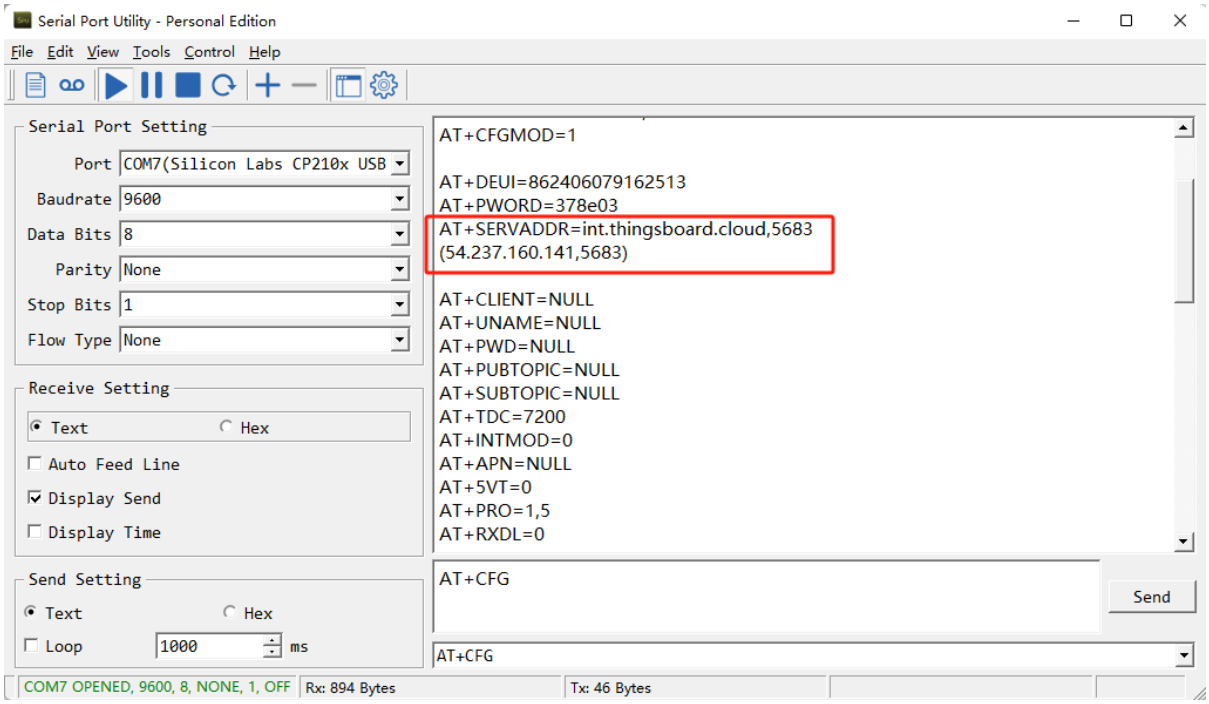


3.7.2 Node Configuration(Example: Connecting to the Thingsboard platform)

3.7.2.1 Instruction Description

- AT+PRO=1,0(HEX format uplink) &AT+PRO=1,5(JSON format uplink)
- AT+SERVADDR=COAP Server Address,5683

Example: AT+SERVADDR=int.thingsboard.cloud,5683(The address is automatically generated when the COAP integration is created)



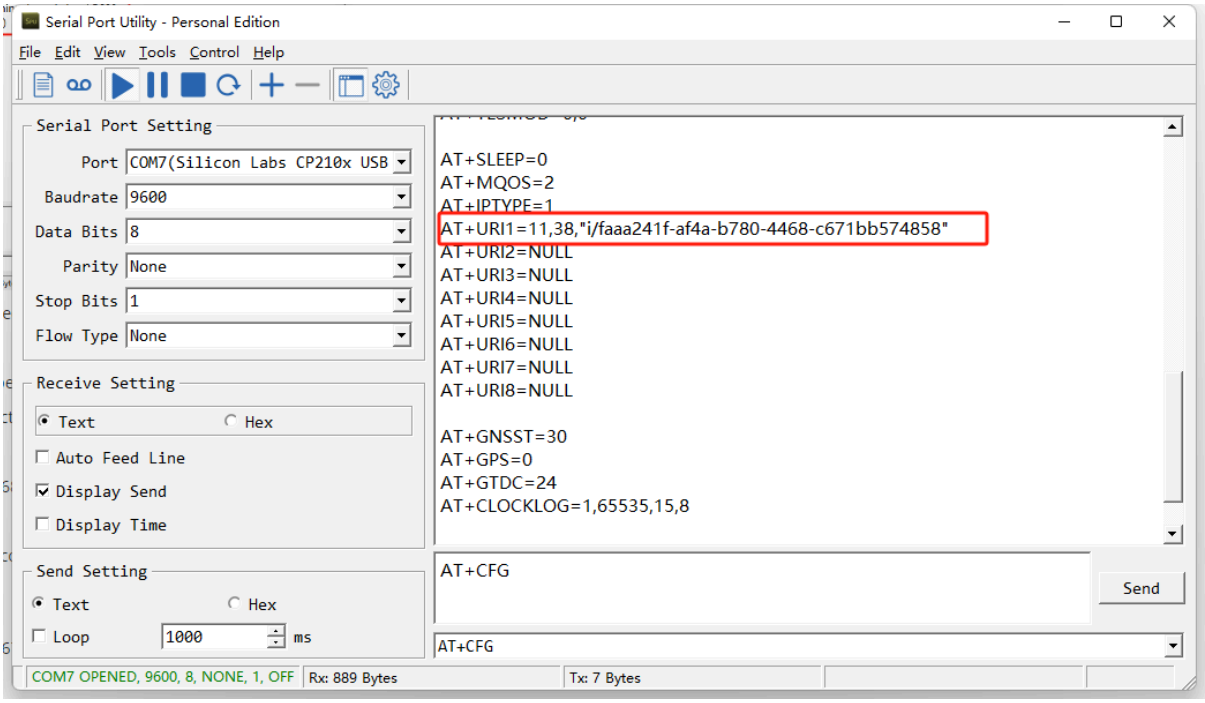
Note:The port for the COAP protocol has been fixed to 5683

- AT+URL1=11,**character length**,"Needs to be consistent with the CoAP endpoint URL in the platform"

If the module used is **BC660K, only one** URL directive needs to be configured,

e.g.

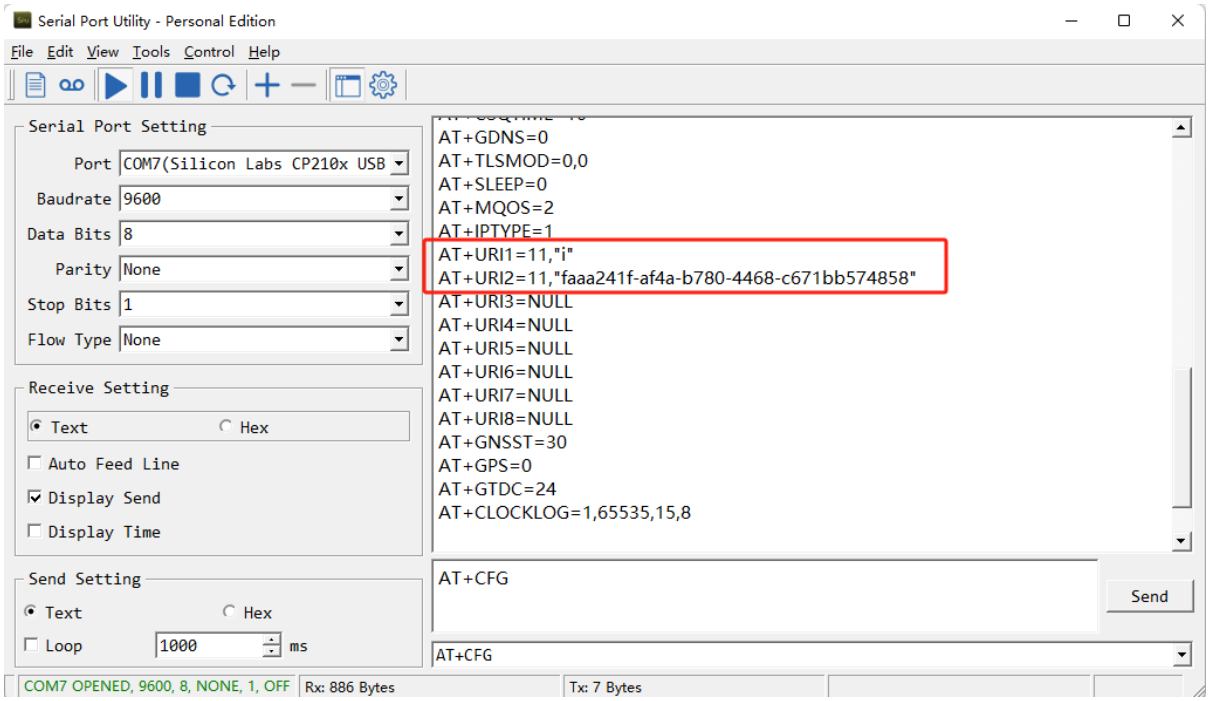
- AT+URL1=11,38, "i/faaaa241f-af4a-b780-4468-c671bb574858"



If you are using a **BG95-M2** module, you need to configure **TWO** URL commands,

e.g.

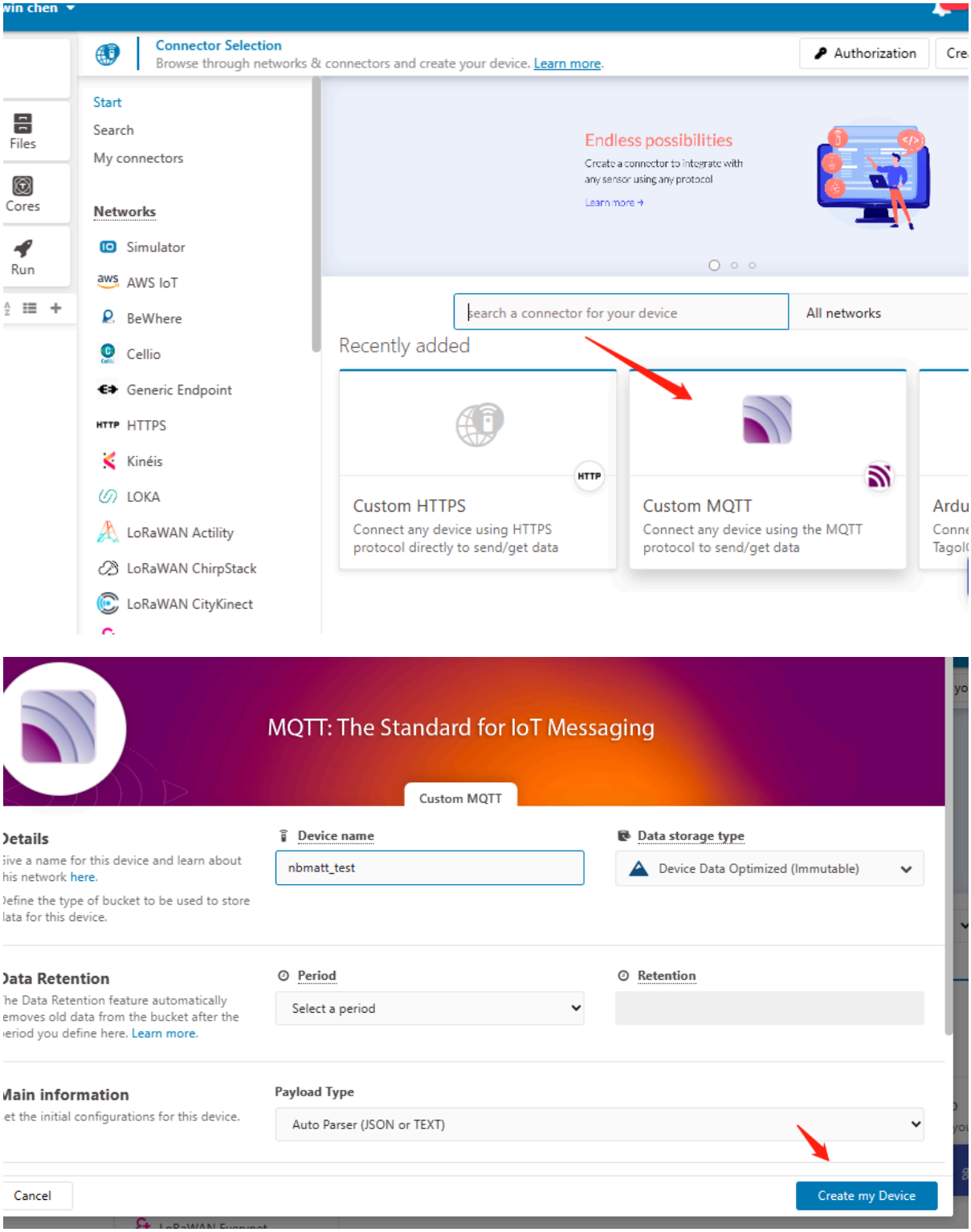
- AT+URL1=11, "i";
- AT+URL2=11,"/faaaa241f-af4a-b780-4468-c671bb574858"



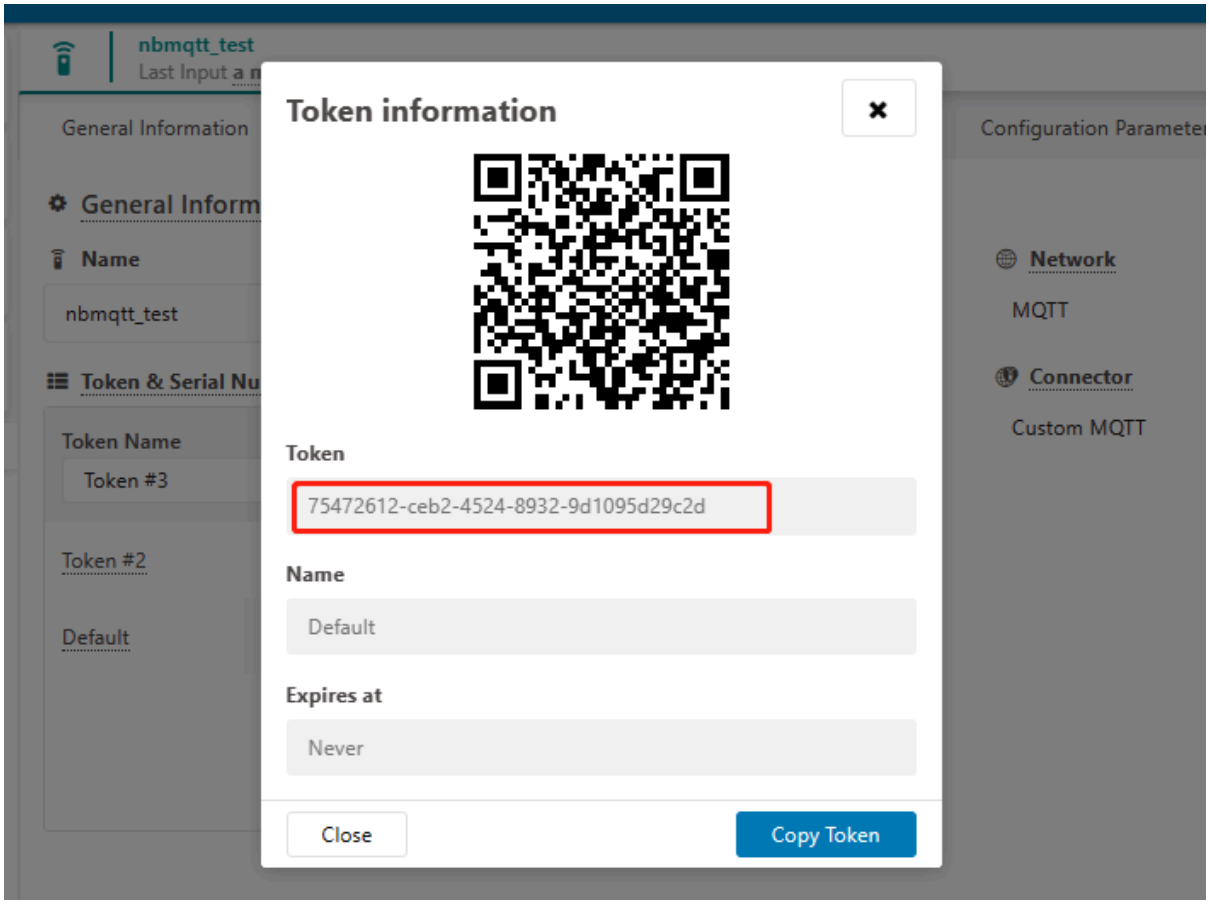
3.8 Tago.io (via MQTT)

3.8.1 Create device & Get Credentials

We use MQTT Connection to send data toTago.io. We need to Create Device and Get MQTT Credentials first.



Go to the Device section and create a device. Then, go to the section tokens and copy your device-token.



The device needs to enable the TLS mode and set the**AT+TLSMOD=1,0** command.

On the Connection Profile window, set the following information:

- **Profile Name: “Any name”**
- **Broker Address: mqtt.tago.io**
- **Broker Port: 8883**
- **Client ID: “Any value”**

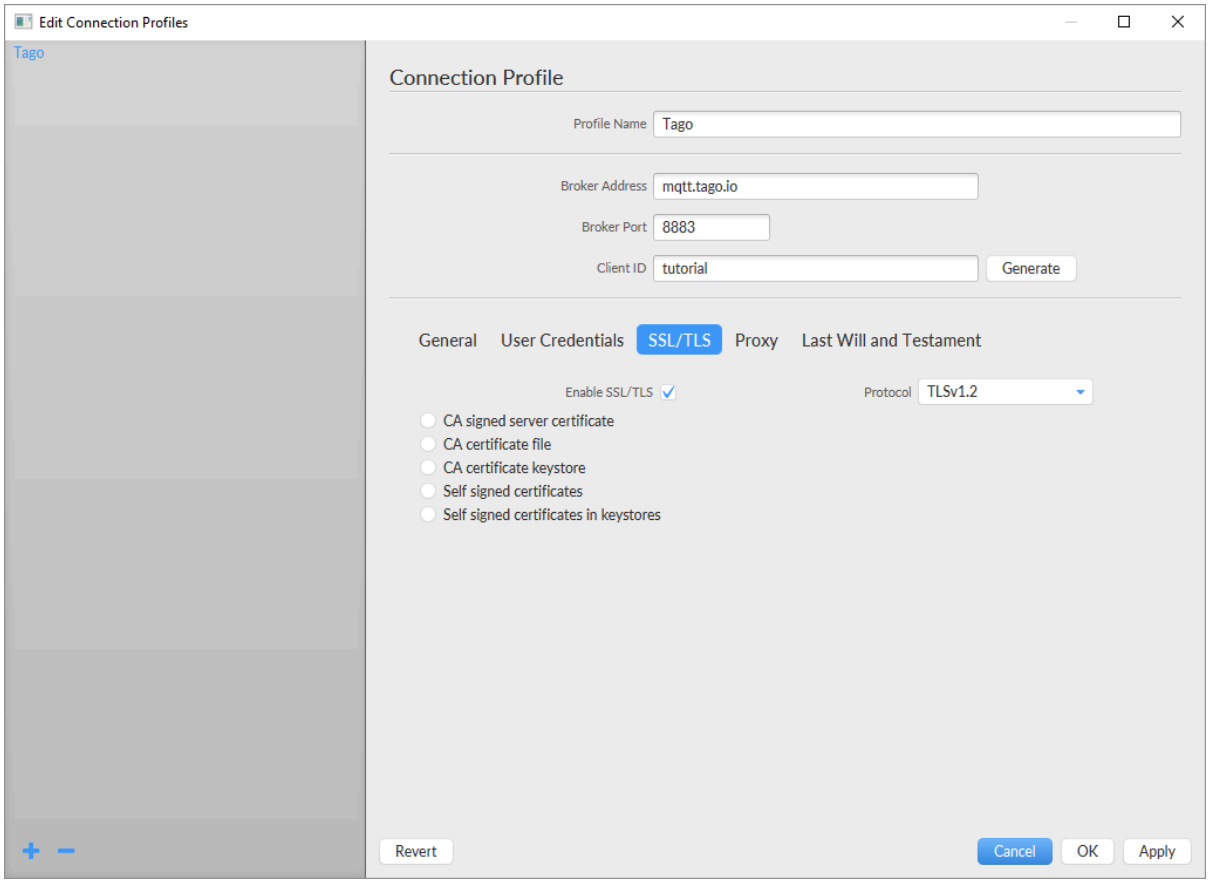
On the section User credentials, set the following information:

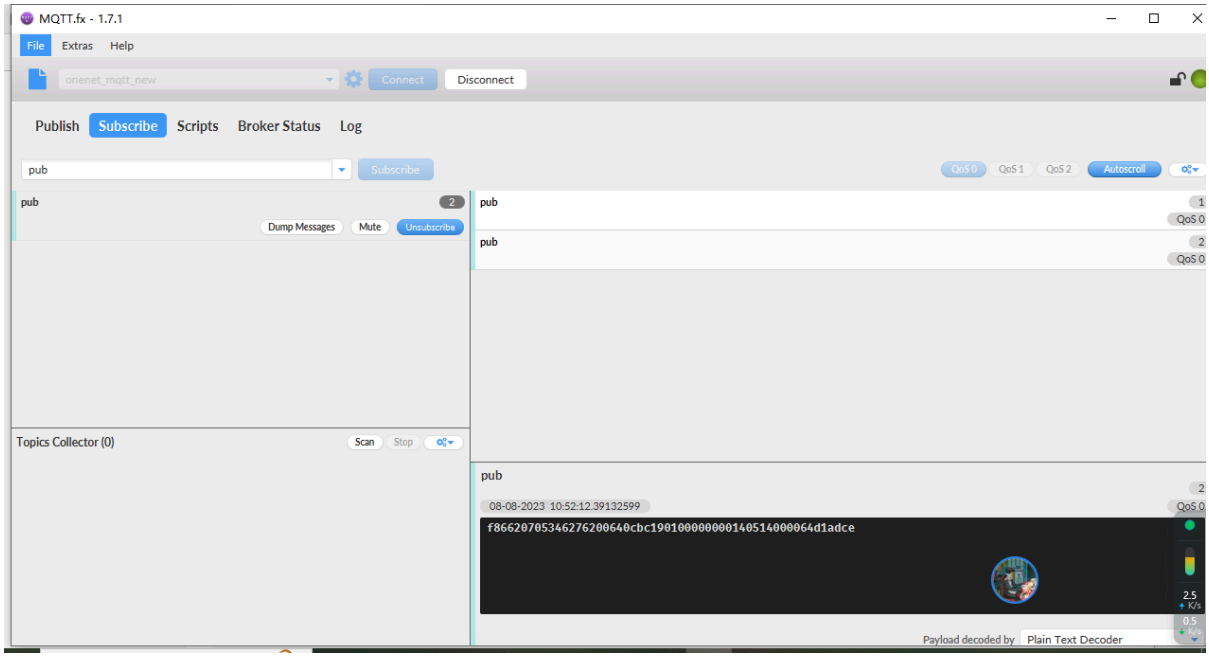
- **User Name: “Any value”** // **Tago validates your user by the token only**
- **Password: “Your device token”**
- **PUBTOPIC: “Any value”**
- **SUBTOPIC: “Any value”**

AT command:

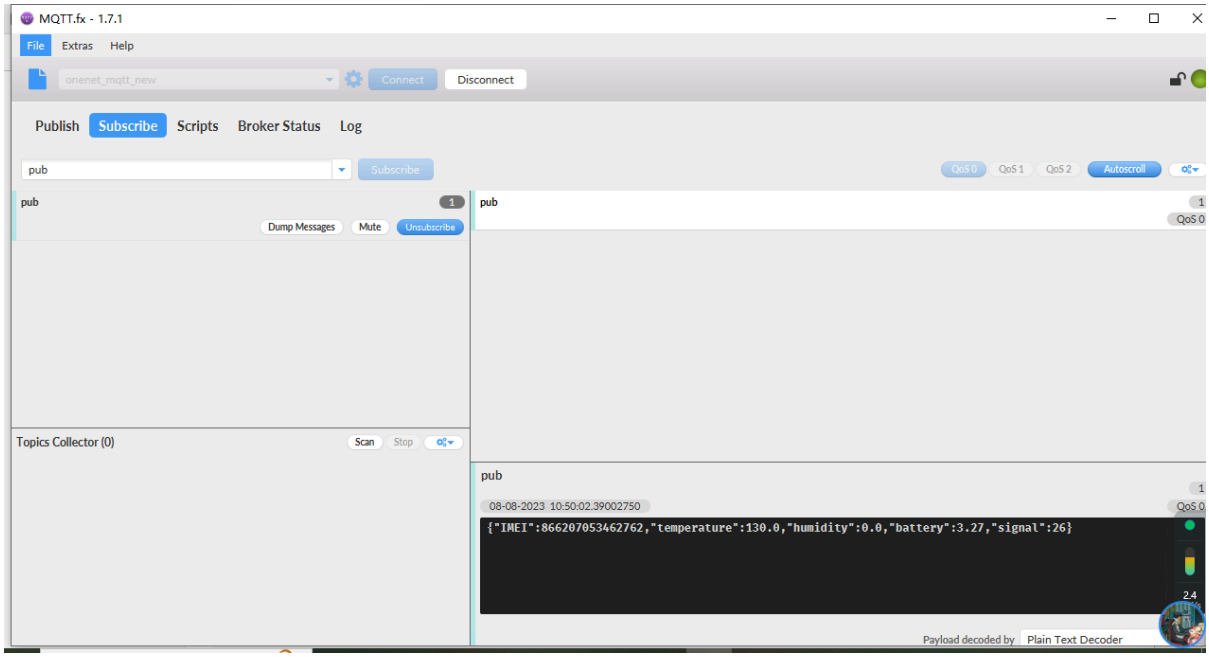
- **AT+PRO=3,0 or 3,5** // **hex format or json format**
- **AT+SUBTOPIC=<device name>or User Defined**
- **AT+PUBTOPIC=<device name>or User Defined**
- **AT+CLIENT=<device name> or User Defined**
- **AT+UNAME=<device name> or User Defined**
- **AT+PWD=“Your device token”**

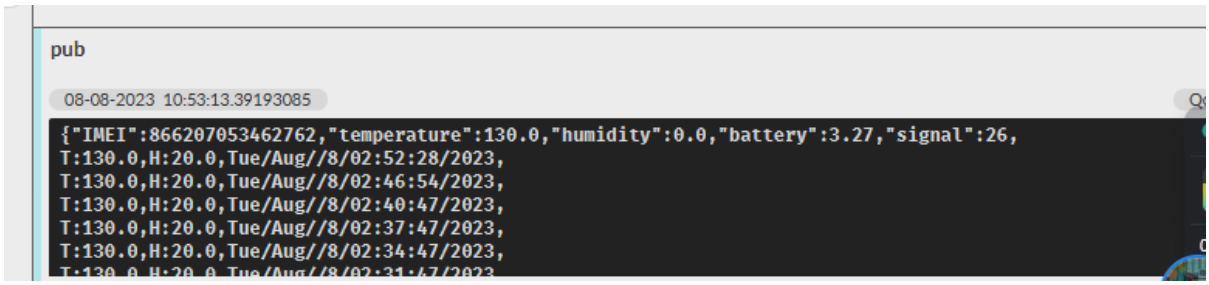
3.8.2 Simulate with MQTT.fx



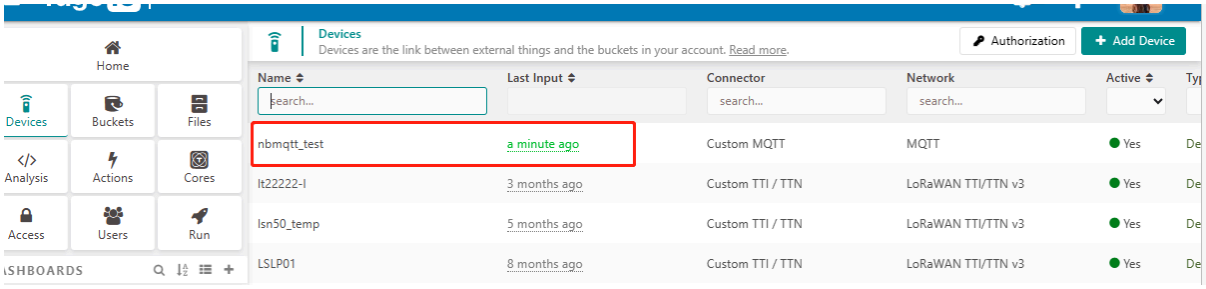


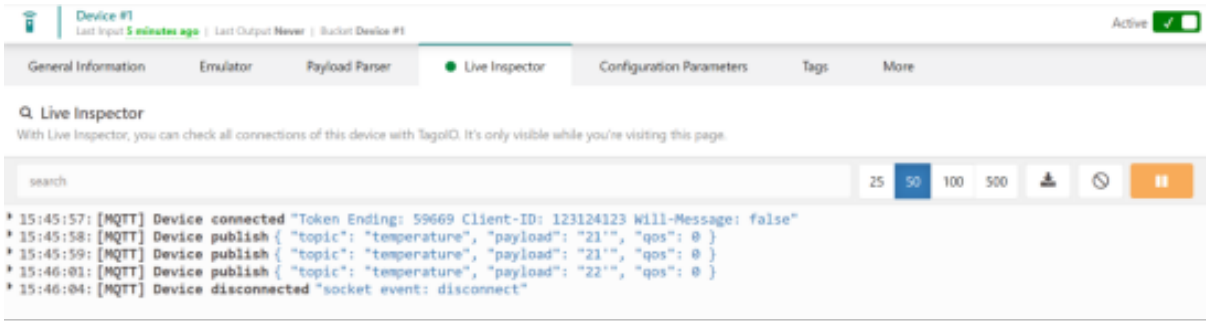
Users can run the **AT+PRO=3,5** command, and the payload will be converted to **JSON format**.





3.8.3 tago data



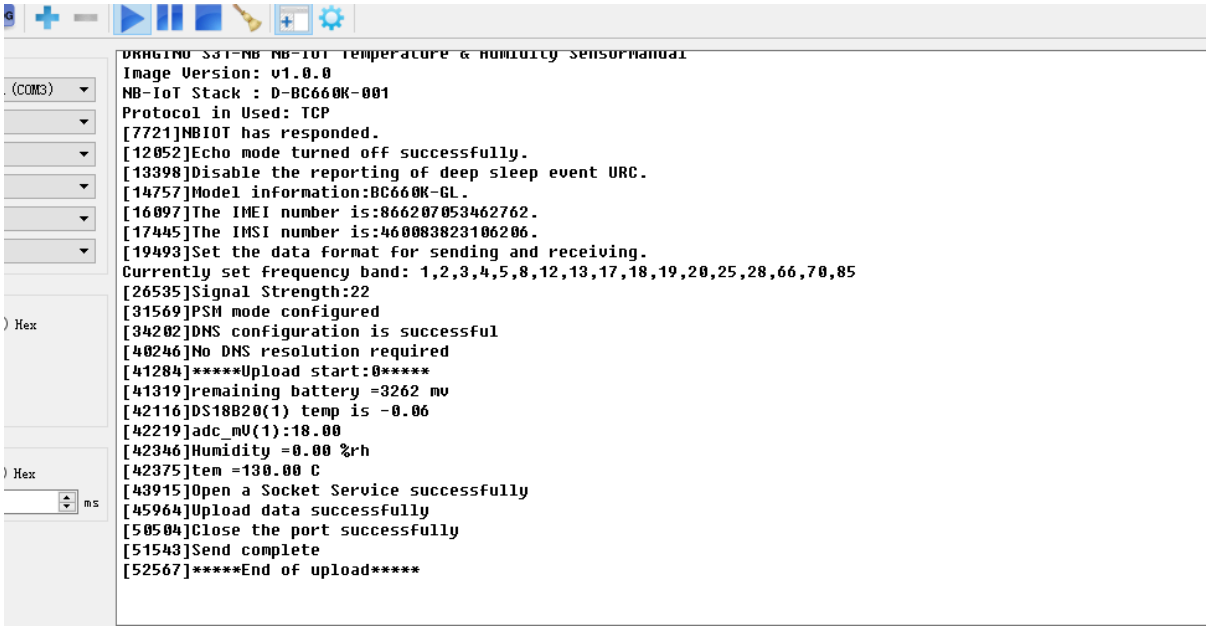


3.9 TCP Connection

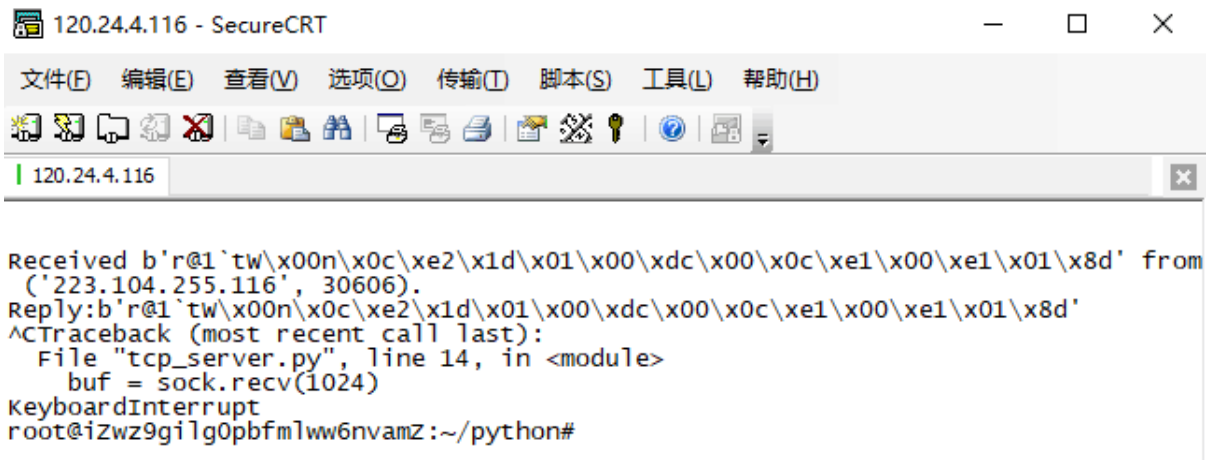
AT command:

- **AT+PRO=4,0** // Set to use TCP protocol to uplink(HEX format)
- **AT+PRO=4,1** // Set to use TCP protocol to uplink(JSON format)
- **AT+SERVADDR=120.24.4.116,5600** // to set TCP server address and port

Sensor Console Output when Uplink:



See result in TCP Server:



3.10 AWS Connection

Users can refer to Dragino NB device connection to AWS platform instructions

4. MQTT/UDP/TCP downlink

4.1 MQTT (via MQTT.fx)

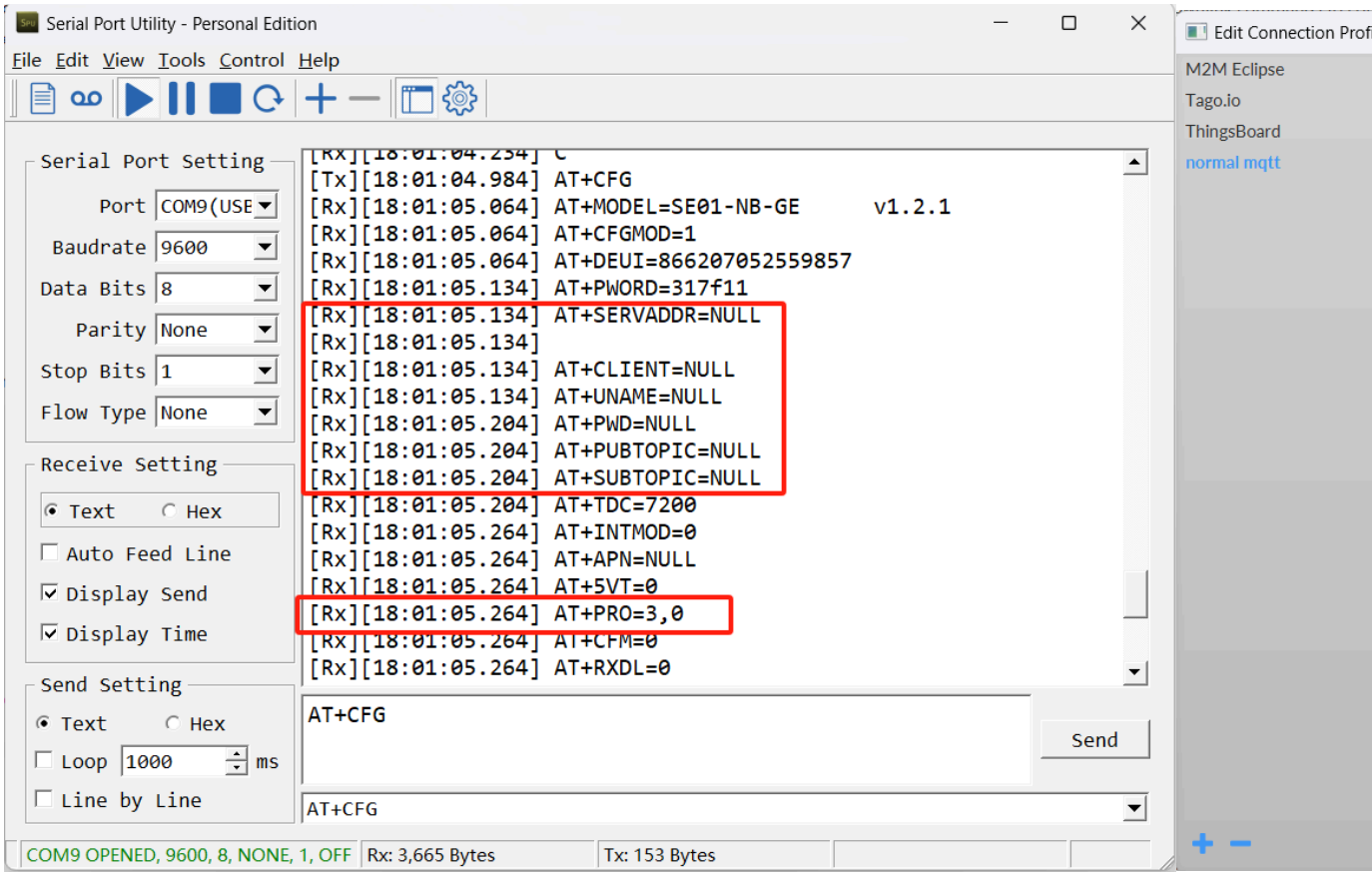
Configure MQTT connections properly and send downlink commands to configure nodes through the Publish function of MQTT.fx.

1. Configure node MQTT connection (via MQTT.fx):

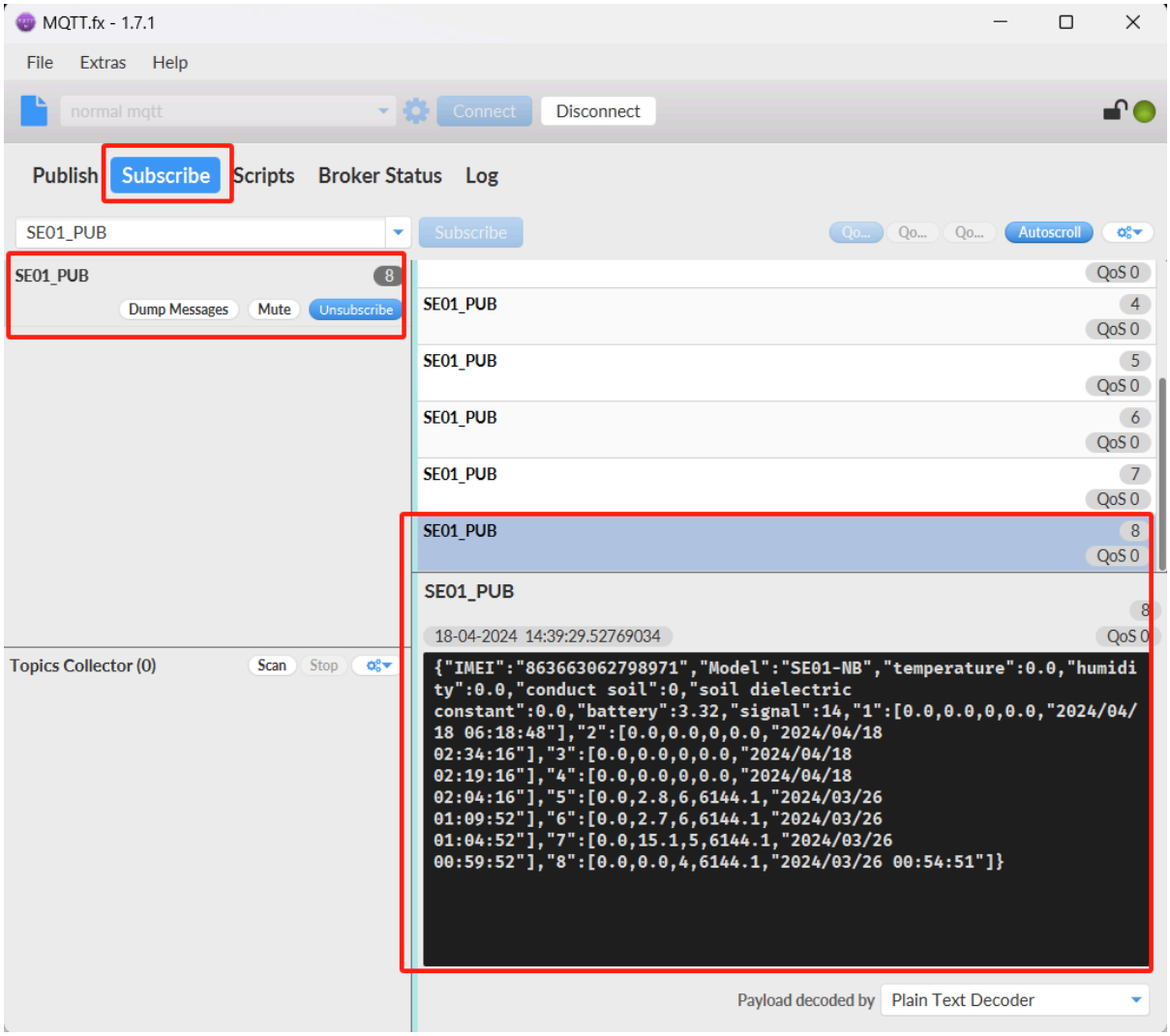
AT command:

- **AT+PRO=3,0 or 3,5** // hex format or json format
- **AT+SUBTOPIC=User Defined**
- **AT+PUBTOPIC=User Defined**
- **AT+UNAME=<device name> or User Defined**
- **AT+PWD=<device name> or User Defined**
- **AT+SERVADDR=8.217.91.207,1883** // to set MQTT server address and port

Note: To uplink and downlink via MQTT.fx, we need set the publish topic and subscribe topic different, for example: AT+SUBTOPIC=SE01_SUB & AT+PUBTOPIC=SE01_PUB.



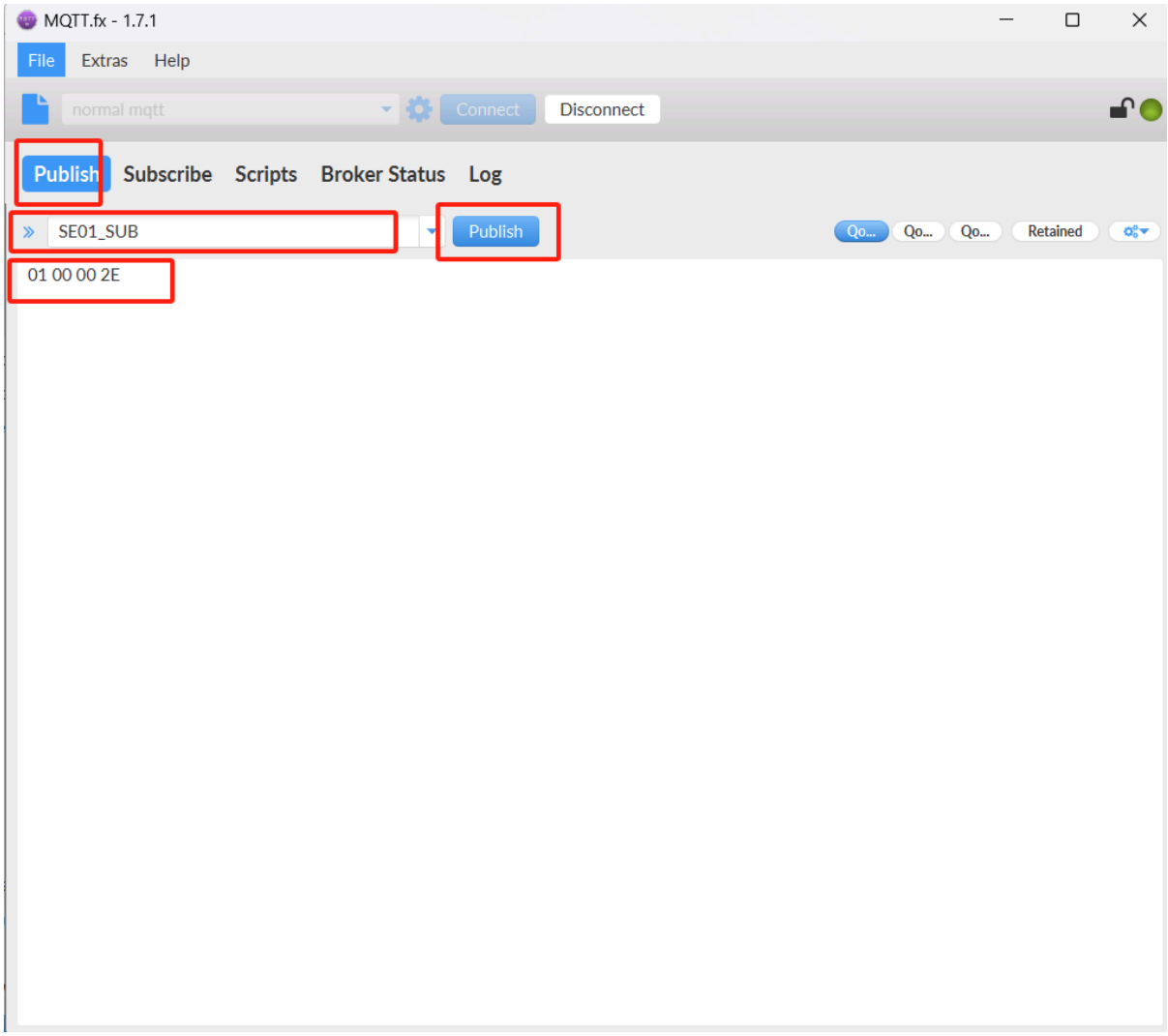
2. When the node uplink packets, we can observe the data in MQTT.fx.

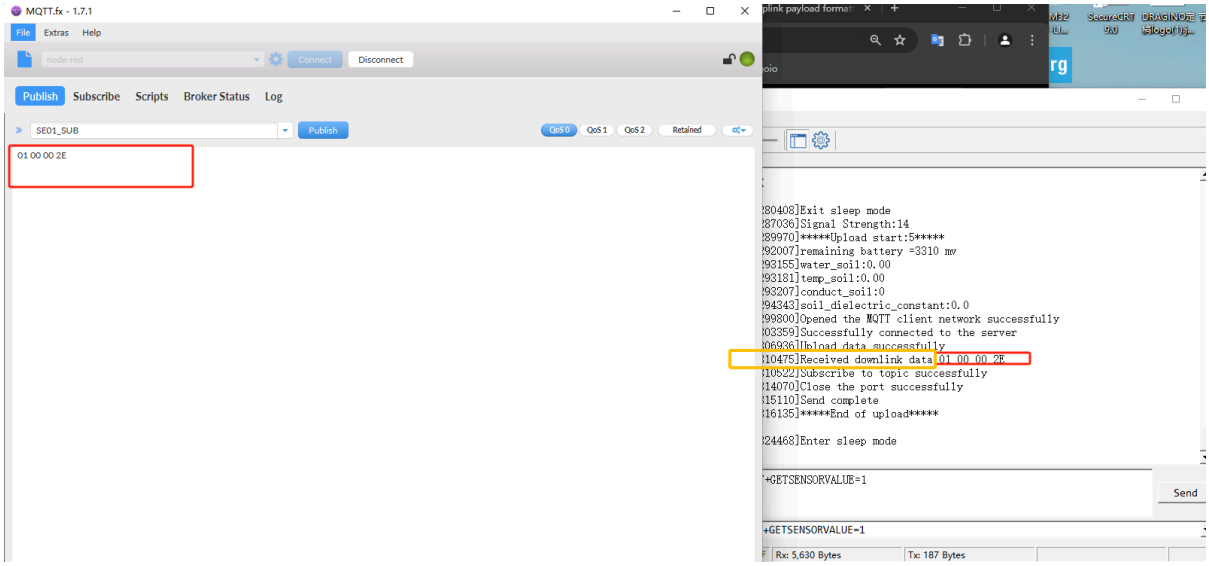


3. The downlink command can be successfully sent only when the downlink port is open.

The downlink port is opened for about 3 seconds after uplink packets are sent.

Therefore, when we see the node uplink packets in the **Subscribe** window, we need to immediately switch to the **publish** window to publish the **hex format** command.


MQTT.fx - 1.7.1
File
Extras
Help
normal mqtt
Connect
Disconnect
Publish
Subscribe
Scripts
Broker Status
Log
SE01_SUB
Publish
Qo...
Qo...
Qo...
Retained
01 00 00 2E



Note: Users can edit the hex command in advance. When the node uplink, directly click the publish button several times to increase the success rate of command configuration.

5. FAQ

5.1 What is the usage of Multi Sampling and One Uplink?

The NB series has the feature for Multi Sampling and one uplink. See one of them

http://wiki.dragino.com/xwiki/bin/view/Main/User%20Manual%20for%20LoRaWAN%20End%20Nodes/SN50v3-NB_BN-IoT_Sensor_Node_User_Manual/#H2.5Multi-SamplingsandOneuplink

User can use this feature for below purpose:

1. **Reduce power consumption**. The NB-IoT transmit power is much more higher than the sensor sampling power. To save battery life, we can sampling often and send in one uplink.
2. Give more sampling data points.

3. Increase reliable in transmission. For example. If user set
   - **AT+TR=1800** // The unit is seconds, and the default is to record data once every 1800 seconds (30 minutes, the minimum can be set to 180 seconds)
   - **AT+NOUD=24** // The device uploads 24 sets of recorded data by default. Up to 32 sets of record data can be uploaded.
   - **AT+TDC=7200** // Uplink every 2 hours.
   - this will mean each uplink will actually include the 6 uplink data (24 set data which cover 12 hours). So if device doesn't lost 6 continue data. There will not data lost.

5.2 Why the uplink JSON format is not standard?

The json format in uplink packet is not standard Json format. Below is the example. This is to make the payload as short as possible, due to NB-IoT transmit limition, a standard Json is not able to include 32 sets of sensors data with timestamp.

The firmware version released after 2024, Mar will use change back to use Json format. Detail please check changelog.

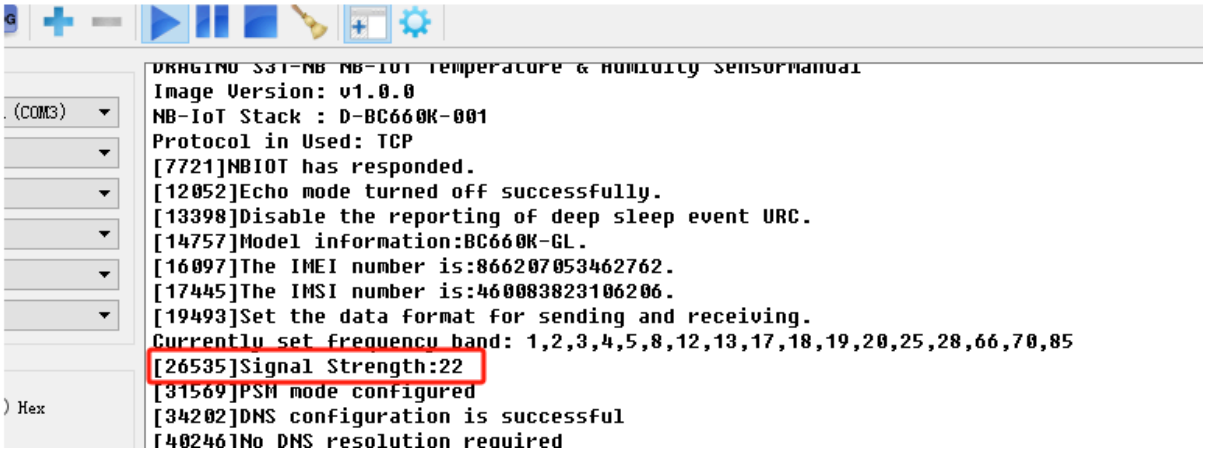
```
{
  "IMEI": "8[redacted]3",
  "Model": "D23-NB",
  "temperature1": -18.6,
  "temperature2": -409.5,
  "temperature3": -409.5,
  "battery": 3.33,
  "signal": 28,
  "1": {
    -18.1,
    -409.5,
    -409.5,
    2024/02/28 09: 01: 38
  },
  "2": {
    -17.8,
    -409.5,
    -409.5,
    2024/02/28 07: 51: 18
  },
```

6. Trouble Shooting:

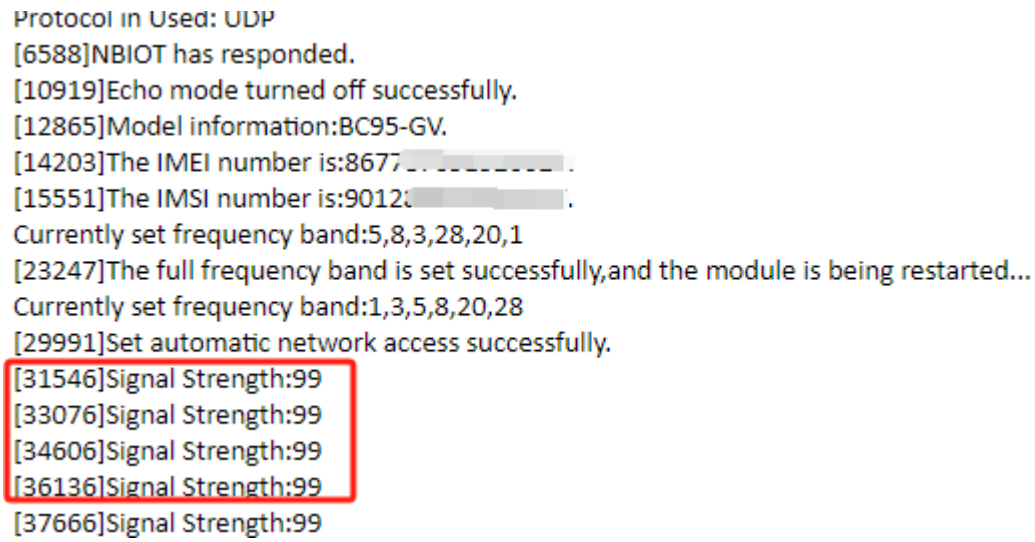
6.1 Checklist for debuging Network Connection issue. Signal Strenght:99 issue.

There are many different providers provide NB-IoT service in the world. They might use different band, different APN & different operator configuration. Which makes connection to NB-IoT network is complicate.

If end device successfully attached NB-IoT Network, User can normally see the signal strengh as below (between 0~31)



If fail to attach network, it will shows signal 99. as below:



When see this issue, below are the checklist:

- Does your SIM card support NB-IoT network? If SIM card doesn't not specify support NB-IoT clearly, normally it doesn't support. You need to confirm with your operator.
- Do you configure the correct APN? Check here for APN settings.
- Do you lock the frequency band? This is the most case we see. Explain and Instruction.
- Check if the device is attached to Carrier network but reject. (need to check with operator).
- Check if the antenna is connected firmly.

If you have check all above and still fail. please send console log files (as many as possible) to support@dragino.com so we can check.

6.2 Issue: "NBIOT did not respond"

11:24:22.397 [44596]NBIOT did not respond.
11:24:24.315 [46530]NBIOT did not respond.
11:24:26.256 [48464]NBIOT did not respond.
11:24:28.196 [50398]NBIOT did not respond.
11:24:30.115 [52332]NBIOT did not respond.
11:24:32.127 [54266]NBIOT did not respond.

11:24:32.127 [54299]Restart the module...
11:24:39.181 [61332]No response when shutting down

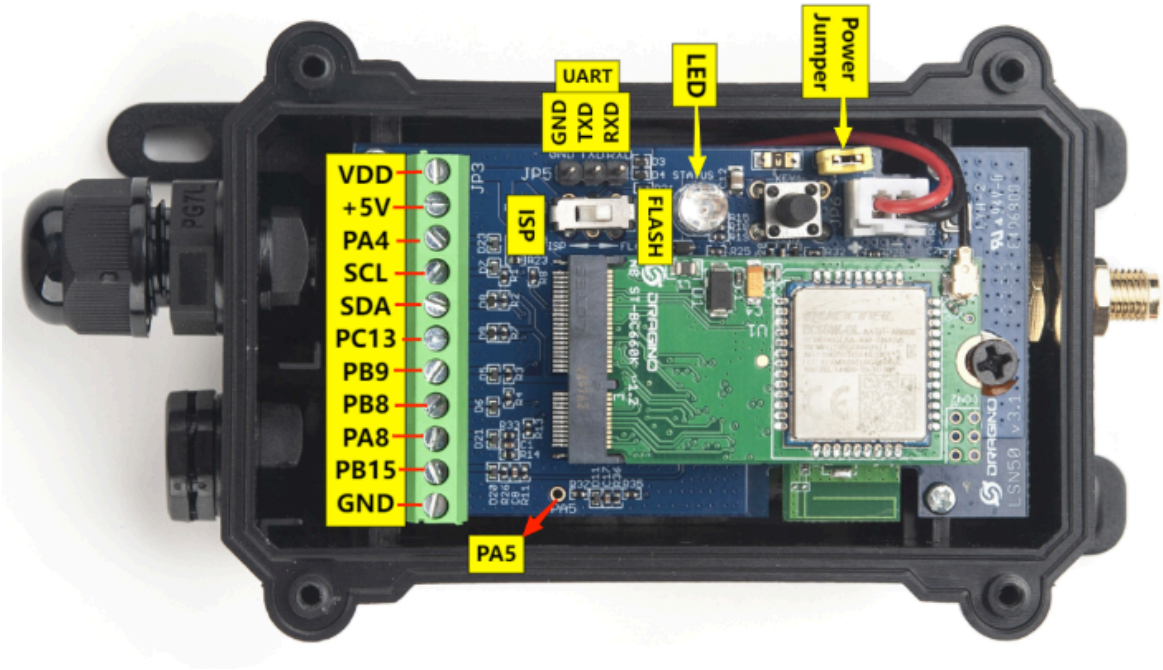
This issue might due to initiate issue for NB-IoT module. In this case, please try:

1) Open Enclosure

2) Power off device by pull out the power on Jumper

3) Power on device by connect back the power jumper.

4) push reset button.



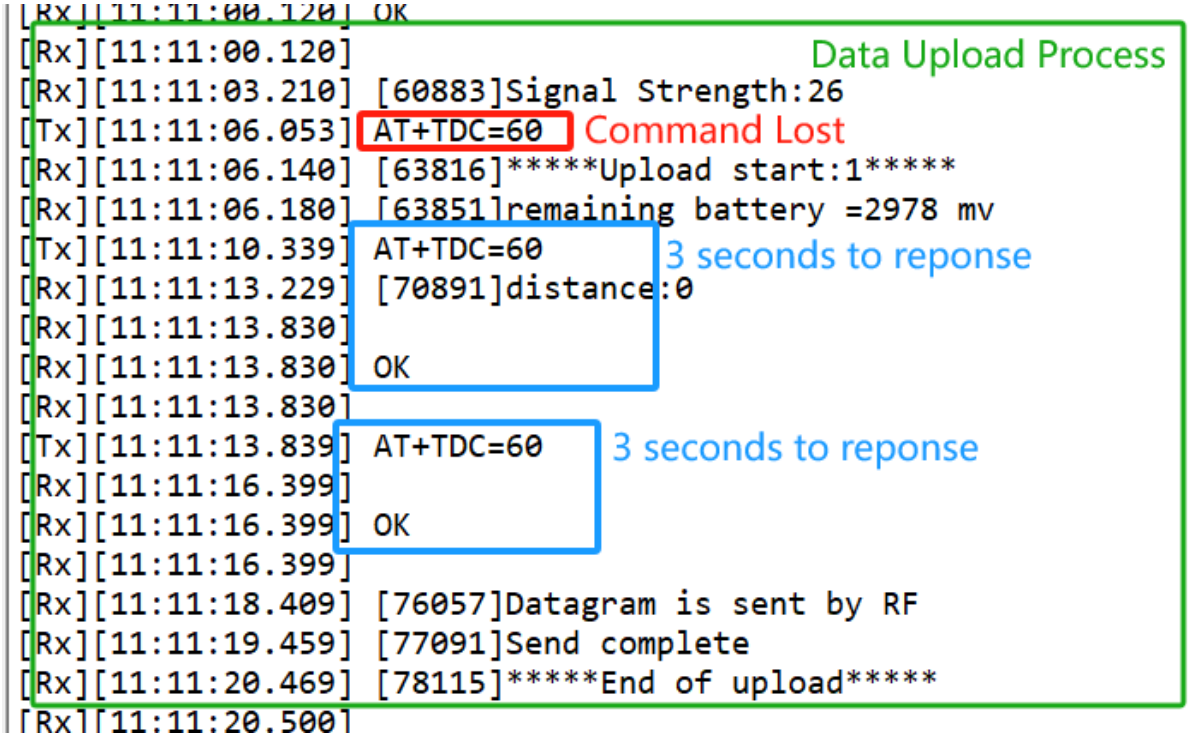
6.3 Issue: "Failed to readl MSI number"

[18170]Failed to read IMSI:1umber.
[20109]Failed to read IMSI numoer.
[22048]Failed to read IMSI number.
[29842lRestart the module...

Make sure that the SIM card is insert in correct direction and device is power off/on during insert. Here is reference link: Insert SIM Card.

6.4 Why sometime the AT Command is slow in reponse?

When the MCU is communicating with the NB-IoT module, the MCU response of AT Command will become slower, it might takes several seconds to response.



6.5 What is the Downlink Command by the NB device?

UDP:

Its downlink command is the same as the AT command, but brackets are required.
Example:

{AT+TDC=300}

MQTT:

Json:

The Json format in MQTT mode needs to be configured with all commands.
If you have configurations that need to be changed, please change them in the template below.
Template:

{
"AT+SERVADDR":"119.91.62.30,1882",
"AT+CLIENT":"JwcXKjQBNhQ2JykDDAA5Ahs",
"AT+UNAME":"usenamedragino",
"AT+PWD":"passworddragino",
"AT+PUBTOPIC":"123",
"AT+SUBTOPIC":"321",
"AT+TDC":"7200",
"AT+INTMOD":"0",
"AT+APN":"NULL",
"AT+5VT":"0",
"AT+PRO":"3,5",
"AT+TR":"900",
"AT+NOUD":"0",
"AT+CSQTIME":"5",
"AT+DNSTIMER":"0",

"AT+TLSMOD":"0,0",
"AT+MQOS":"0",
"AT+TEMPALARM1":"0",
"AT+TEMPALARM2":"10",
"AT+TEMPALARM3":"0"
}

Hex:

MQTT's hex format. Since many commands need to support strings, only a few commands are supported.

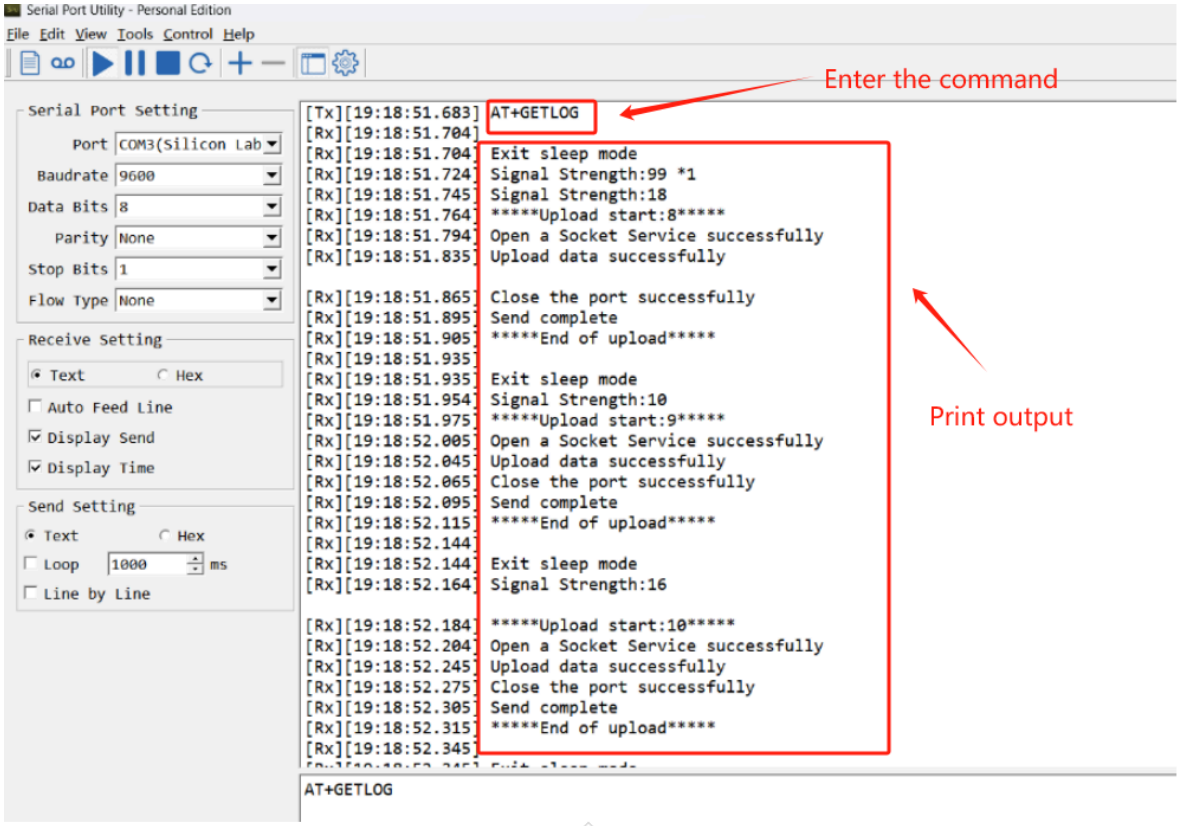
The supported commands are consistent with LoRaWAN's hex commands.
Please refer to the following link to obtain the hex format:

http://wiki.dragino.com/xwiki/bin/view/Main/End%20Device%20AT%20Commands%20and%20Downlink%20Command/

6.6 How to obtain device logs?

- **AT Command: AT +GETLOG**

This command can be used to query upstream logs of data packets.



6.7 How to find the AT Command Password if lost?

Why can't the password access AT command after upgrade(-NB)?

Because the new version of -NB firmware has updated the factory reset function, users can choose to restore all parameters to factory Settings, or keep the password to restore the rest of the parameters to factory Settings.

This update changes the password address of the firmware, so the password will be invalid after the customer upgrades from the old version of firmware (without FDR1 function) to the new version of firmware (with FDR1 function).

Two different restore factory Settings configurations.

AT command:

- **AT+FDR** // Reset Parameters to Factory Default.
- **AT+FDR1** // Reset parameters to factory default values except for passwords.(new)

Version Confirmation

We are now dividing the **old firmware**(without FDR1 function) with the **new firmware**(with FDR1 function) by whether it contains FDR1 functionality. Please refer to the table:

General Model	Firmware version (without FDR1 function)	Firmware version (with FDR1 function)
CPL03-NB, S31-NB, SN50V3-NB, TS01-NB, D20-NB, DS03A-NB, DDS04-NB, DDS45-NB, DDS20-NB, DDS75-NB, LDS12-NB, LDS40-NB, LMS01-NB, MDS120-NB, MDS200-NB, SE01-NB, SPH01-NB;	Before V1.2.1	After V1.2.1 (including V1.2.1)
WL03A-NB, SDI-12-NB;	Before V1.0.2	After V1.0.8 (including V1.0.2)
SW3L-NB, PS-NB;	Before V1.0.5	After V1.0.5 (including V1.0.5)
RS485-NB	Before V1.0.8	After V1.0.8 (including V1.0.8)

UART connection and firmware update methods

Users can query passwords only using the UART interface via the STM32CubeProgrammer.

See **UART Connection**.

update firmware through UART TTL interface : **Instruction**.

query the password via STM32CubeProgrammer

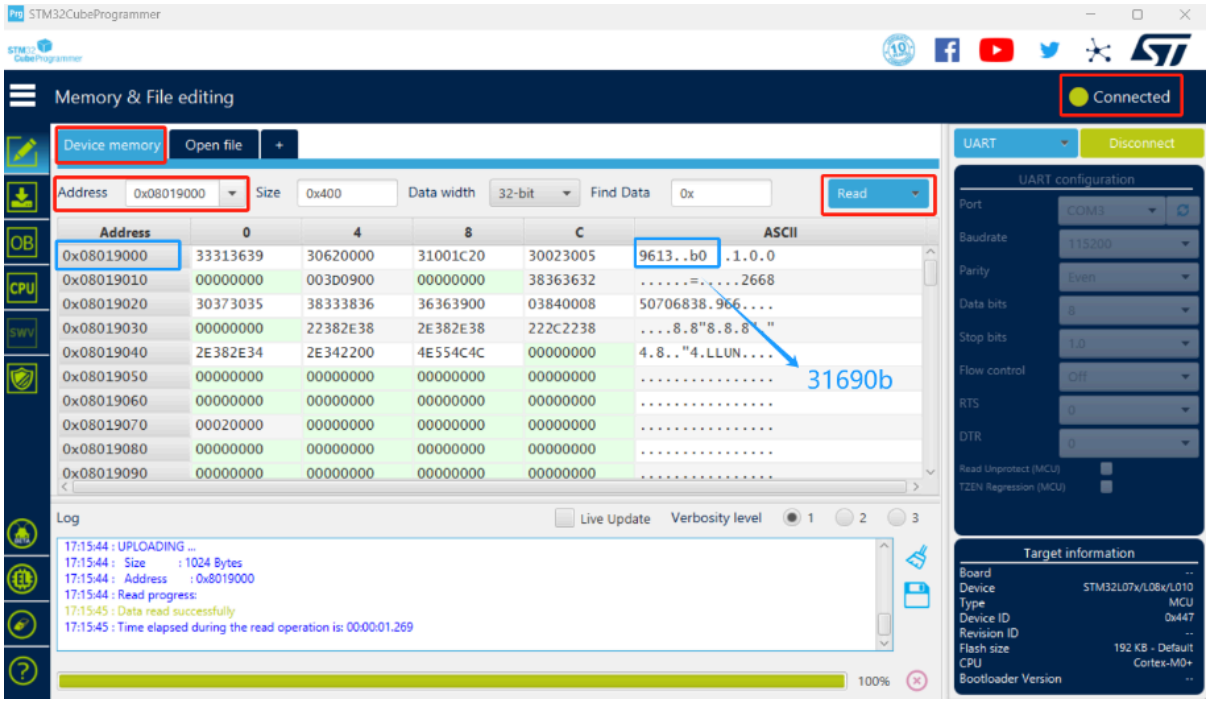
Users can use the password address to query the password through STM32CubeProgrammer.

- The password address for old firmware(without FDR1 function) : **0x08019000**
- The password address for new firmware(with FDR1 function) : **0x08025D00**

Notice: The password can only be queried after the firmware is run once.

Procedure for querying the password(old firmware):

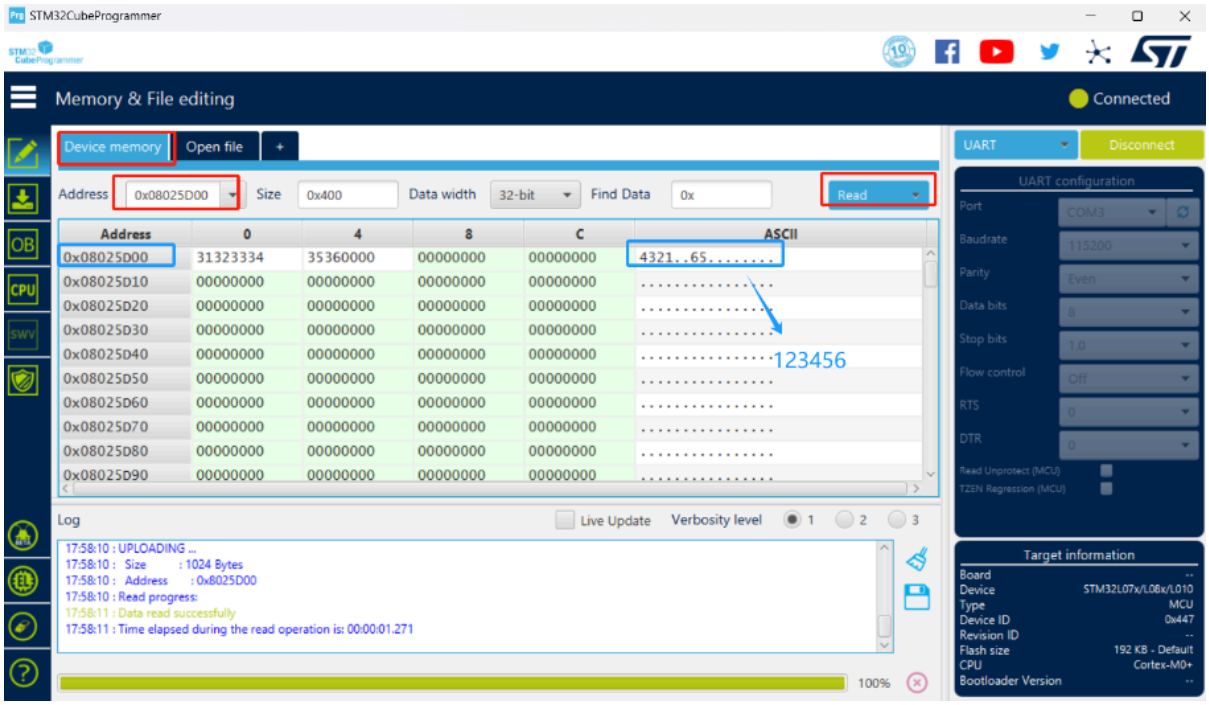
- After the firmware upgrade is complete, switch back to the **FLASH** and reset the node to **run the firmware once**.
- Then place the switch at the **ISP** and connect to the STM32CubeProgrammer (same as when burning the firmware).
- Click "Device memory", enter **0x08019000** in "Address", and click "Read"
- Find the 0x08019000 address field and then read the current password as shown in the screenshot below.



Procedure for querying the password(new firmware):

Refer to the old and new firmware division above, and run the firmware first after updating the firmware.

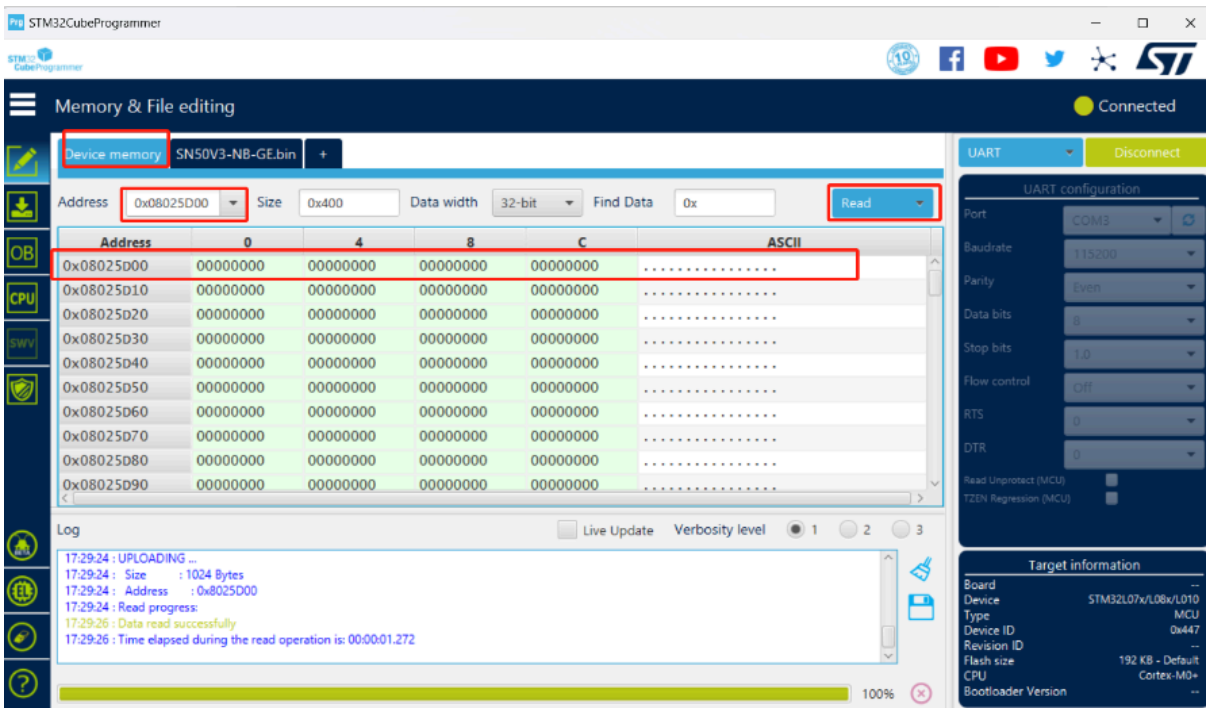
- After the firmware upgrade is complete, switch back to the **FLASH** and reset the node to **run the new firmware once**.
- Then place the switch at the **ISP** and connect to the STM32CubeProgrammer (same as when burning the firmware).
- Click "Device memory", enter **0x08025D00** in "Address", and click "Read"
- Find the 0x08025D00 address field and then read the current password as shown in the screenshot below.

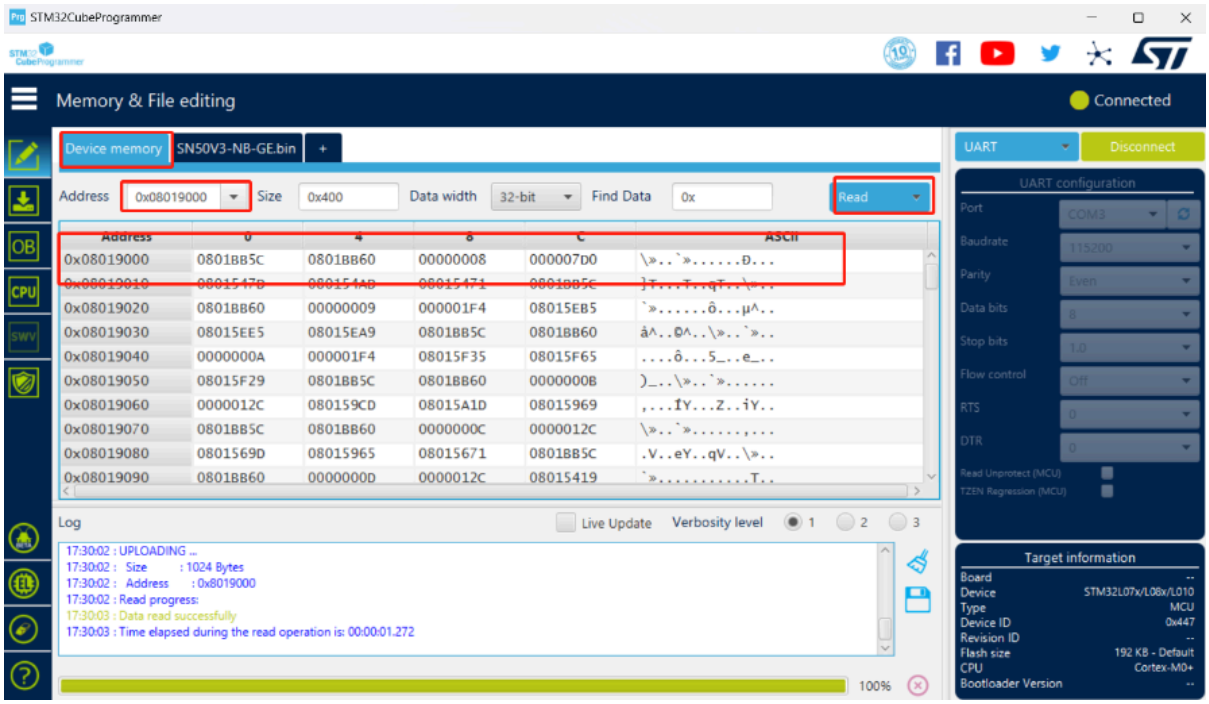


Special case

If the user has never changed the password manually, the user cannot find the valid password through the above two password addresses. In this case, the valid password is still the original password on the node box label (**AT +PIN**).

Invalid query screenshot example:




STM32CubeProgrammer
Memory & File editing
Connected
Device memory
SN50V3-NB-GE.bin
Address
0x08019000
Size
0x400
Data width
32-bit
Find Data
0x
Read
Address 0 4 8 C ASCII
0x08019000 0801BB5C 0801BB60 00000008 000007D0
0x08019020 0801BB60 00000009 000001F4 08015EB5
0x08019030 08015EE5 08015EA9 0801BB5C 0801BB60
0x08019040 0000000A 000001F4 08015F35 08015F65
0x08019050 08015F29 0801BB5C 0801BB60 0000000B
0x08019060 0000012C 080159CD 08015A1D 08015969
0x08019070 0801BB5C 0801BB60 0000000C 0000012C
0x08019080 0801569D 08015965 08015671 0801BB5C
0x08019090 0801BB60 0000000D 0000012C 08015419
Log
Live Update
Verbosity level
17:30:02 : UPLOADING ...
17:30:02 : Size : 1024 Bytes
17:30:02 : Address : 0x8019000
17:30:02 : Read progress:
17:30:03 : Data read successfully
17:30:03 : Time elapsed during the read operation is: 00:00:01.272
100%
UART
Disconnect
UART configuration
Port
Baudrate
Parity
Data bits
Stop bits
Flow control
RTS
DTR
Target information
Board
Device
Type
Device ID
Revision ID
Flash size
CPU
Bootloader Version
STM32L07x/L08x/L010
MCU
0x447
192 KB - Default
Cortex-M0+