

SenseLive X5050

SenseLive Ethernet Industrial Telemetry and Data Acquisition Unit

One-line product tagline – the headline benefit your customer cares about most.



OVERVIEW

The **SenseLive X5050** is a **compact industrial IoT gateway / serial device server** that connects **RS-485 devices to Ethernet and cloud systems**. It enables seamless data conversion (Modbus RTU → TCP/MQTT/HTTP) and supports **remote monitoring, configuration, and edge processing**—making it ideal for Industry 4.0 and automation setups.

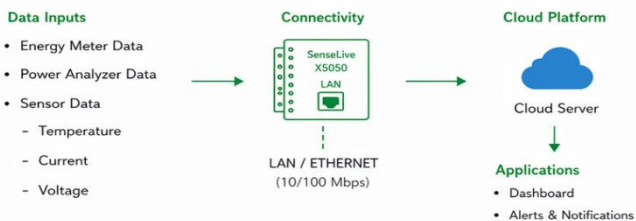
KEY HIGHLIGHTS

- ▶ Compact DIN-rail mount design for industrial panels
- ▶ Multi-protocol support: Modbus RTU/TCP, MQTT, HTTP/JSON
- ▶ Wide power input (9–24V DC) with protection
- ▶ Remote configuration via web interface & desktop tool
- ▶ RS-485 to Ethernet data conversion

APPLICATIONS

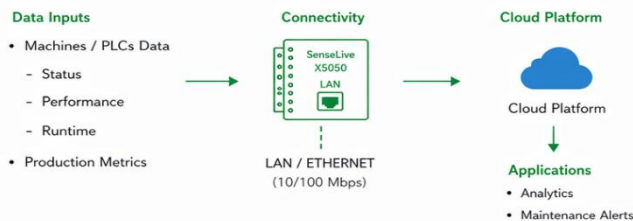
1. ENERGY MONITORING

Real-time monitoring of energy parameters and consumption.



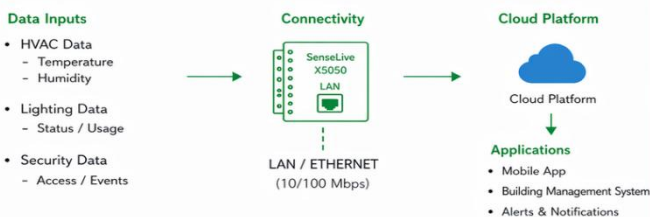
2. SMART MANUFACTURING

Improve productivity and reduce downtime with real-time data.



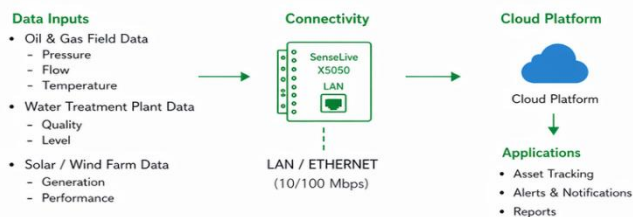
3. BUILDING AUTOMATION

Automate and monitor building systems for efficiency and comfort.

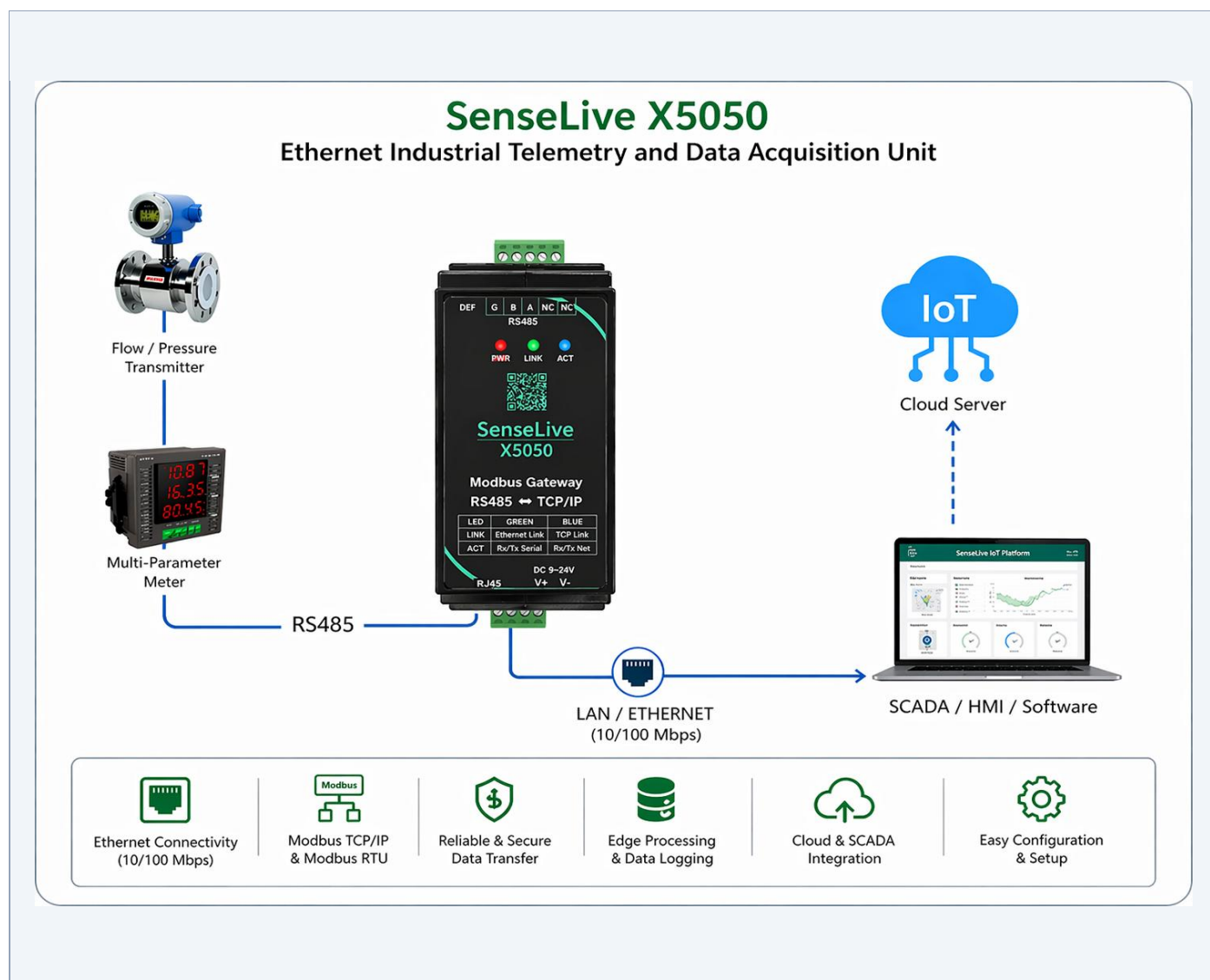


4. REMOTE ASSET MANAGEMENT

Monitor and manage remote assets from anywhere.



FUNCTIONAL BLOCK DIAGRAM



TECHNICAL SPECIFICATIONS

GENERAL	
Model Number	X5050
Product Type	IoT Gateway
Enclosure	ABS / Polycarbonate, IP20 (IP65 optional)
Mounting	DIN-Rail
POWER	
Input Voltage	9~24V DC
Power Consumption	≤ 3 W (typical)
Protection	Reverse polarity, over-voltage, surge
COMMUNICATION	

Wired Interfaces	RS-485, RS-232, Ethernet 10/100 Mbps
Wireless	Wi-Fi 802.11 b/g/n,
Protocols	Modbus RTU, Modbus TCP, MQTT, HTTP/REST, JSON
Antenna	No antenna
ENVIRONMENTAL	
Operating Temp.	–40 °C to +85 °C
Storage Temp.	–45 °C to +100 °C
Humidity	5 % – 95 % RH (non-condensing)
MECHANICAL	
Dimensions (L×W×H)	37.6 x 83.6 x 89.2mm
Weight	≈ 150 g
Indicators	Power, Network, Data LEDs
COMPLIANCE	
Certifications	CE, FCC, RoHS, BIS (where applicable)
Warranty	12 months from invoice date

ORDERING INFORMATION

PART NO.	DESCRIPTION
SL-XXXX-S	Standard variant – Wi-Fi + Ethernet, 9–36 V DC
SL-XXXX-L	LTE variant – 4G Cat-1 + Ethernet, with antenna
SL-XXXX-P	Pro variant – LTE + Wi-Fi + BLE, IP65 enclosure
SL-ACC-PSU	Optional 12 V / 1 A DIN-rail power supply

DEVICE CONFIGURATION

- Software Installation:-
To download software just click on the below link.

<https://senselive.in/downloads/software/SenseLiveConfig2/SenseLiveConfig2.zip>
- Hardware Connection:-
 - **Power Supply:** Connect on-site 2-wire power to positive and negative terminals.
 - **Serial Port:** Connect based on user device. For the first 485 port, connect 485+ to 1A and 485- to 1B.
 - **Network:** Use a standard network cable to connect directly to a computer or through a switch.
- Parameter Configuration:-
 - After installing Vircom and connecting the hardware, run the software and click on "Device Management."

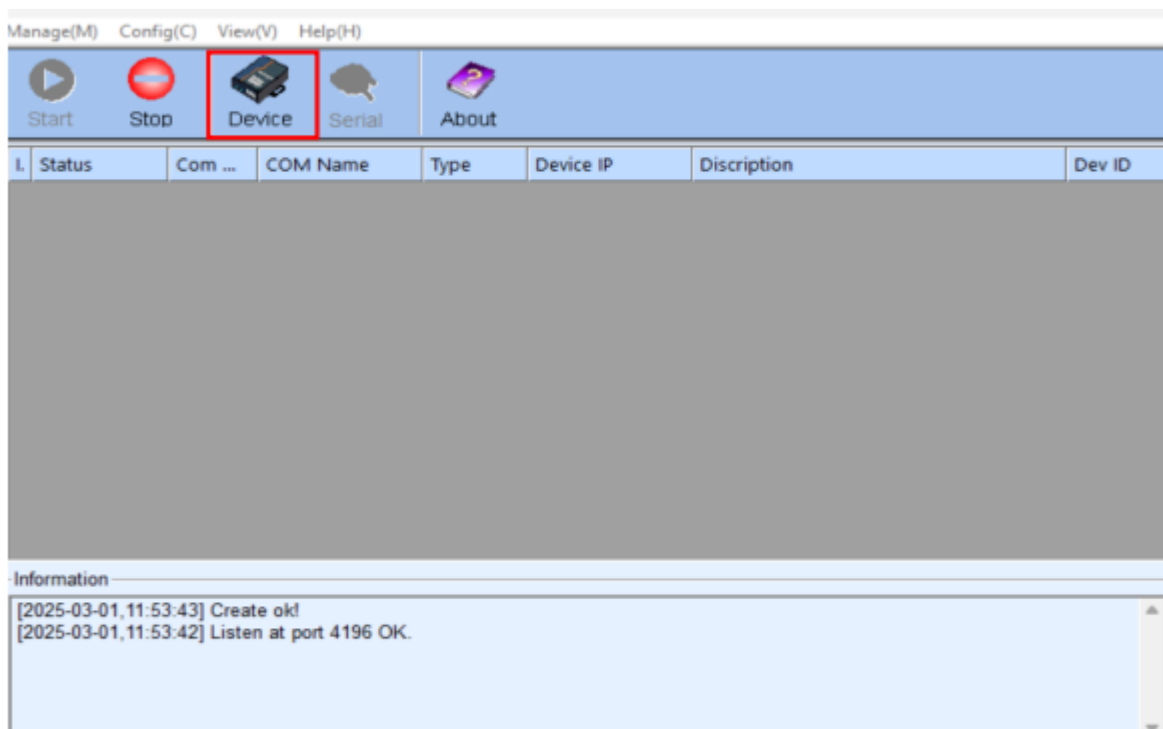


Fig.1 Main Interface

- In device interface click on “Auto search”. As you can see, the connected device is visible in the Device Manager, as shown in Figure 2.
- Click “Edit Device” to set the parameters.

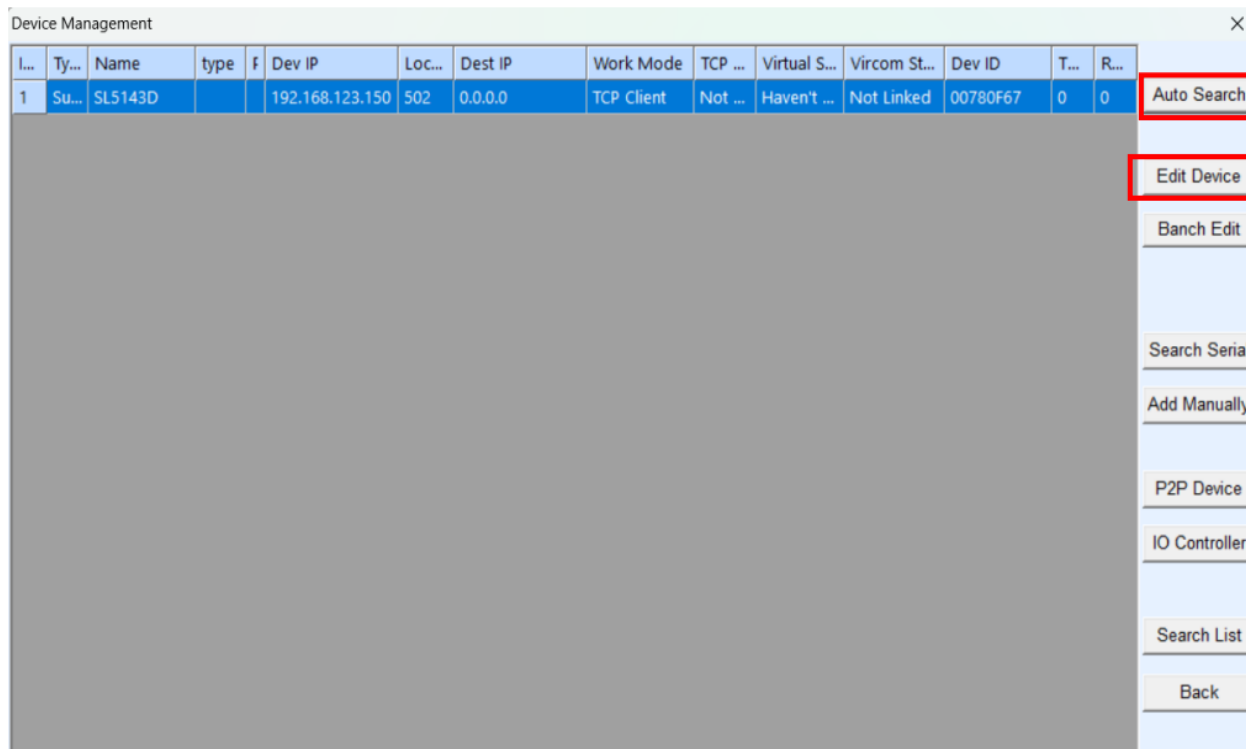


Fig. 2 Device List

- IN network setting you have to change IP address, port and baud rate and then click “modify setting”

The screenshot shows the 'Device Settings' window with the following settings:

- Device Info:** Virtual Serial (Not Use), Dev Type, Dev Name (SL5143D), Dev ID (28788B19AA78), MAC Addr (04EEE819AA90), Firmware Ver (V1.470).
- Function of the device:** Web Download, DNS System, REAL_COM Protocol, Modbus TCP To RTU, Serial Commnad, DHCP Support, Storage Extend, Multi-TCP Connection.
- Network:** IP Mode (Static), IP Address (192.168.1.200), Port (501), Work Mode (TCP Server), Net Mask (255.255.255.0), Gateway (192.168.1.1), Dest. IP/Domain (192.168.1.3), Local IP, Dest. Port (1883), UDP Dynamic.
- Serial:** Baud Rate (9600), Data Bits (8), Parity (None), Stop Bits (1), Flow Control (None).
- Advanced Settings:** DNS Server IP (8.8.4.4), Dest. Mode (Dynamic), Transfer Protocol (None), Keep Alive Time (60), Reconnect Time (12), Http Port (80), UDP Group IP (230.90.76.1), Register Pkt, Restart If No Data, Enable Parameter Send, Framing Rule, Max Frame Length (1300), Max Interval (3).

Buttons at the bottom: Get Default, Save As Default, Load Default, Modify Key, Firmware/Config, Restart Dev, **Modify Setting**, Cancel.

Fig. 3 Device setting

4. Web Portal:-

The screenshot shows the 'Configuration Portal' web interface with the following sections:

- Device Information:** Device Name, Device MAC, Firmware Version: V1.
- Serial Settings:** Baudrate, Data Bits, Parity, Stop Bits, Flow Control.
- Network Settings:** IP Addressing, IP Address, Local Port, Mode, Subnet Mask, Gateway, Destination IP/DNS, Destination Port, Web Port.
- Multi-Master Settings:** Protocol, Response Timeout, Multi-Master, Transmit Delay.
- Advanced Settings:** Watchdog Reset, Watchdog Reset Time, Re-connect Time.
- Modify Web Password:** New Password, Confirm Password.

Buttons: Save Configuration.

➤ To use the web portal, follow the steps below to set a static IP address, and then access it using that IP address.

5. Control Panel Setting:-

Open the control panel → Click Network and Internet → Click Network and Sharing Center → Click Change adapter settings → Open IPv4 Properties, Rightclick on your active network connection (Ethernet/Wi-Fi) → Click Properties.
To Connect to a Network (LAN or Internet)

Add the network credential, it need to be same as your device but last two digit should be different.

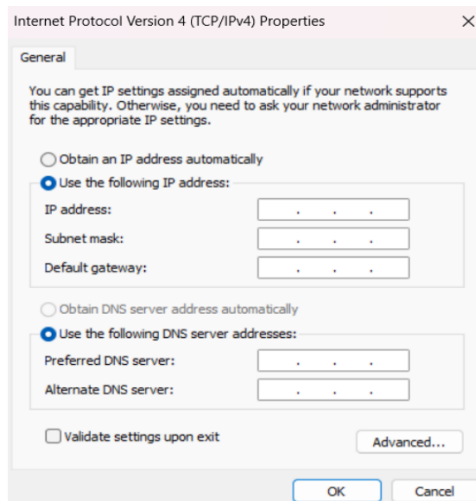


Fig.4 To Connect to a Network (LAN or Internet)

6. Modbus Communication Settings :-

6.1. In advance setting, set transfer protocol as Modbus_TCP protocol.

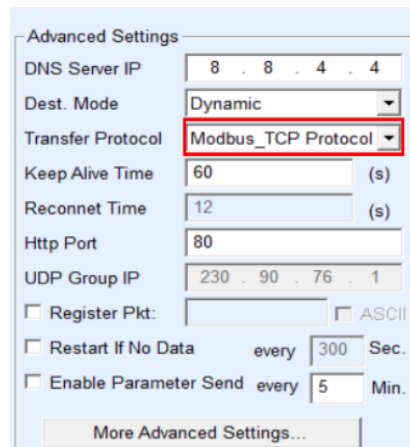


Fig.5 Enable Modbus TCP Functionv

6.2. In network setting, set work mode as TCP server and then click on “modify setting”.

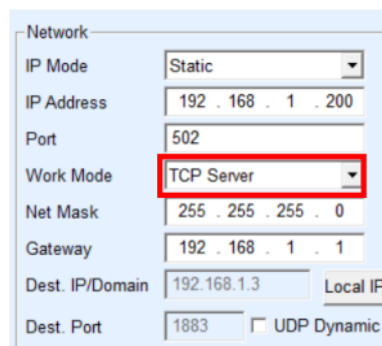


Fig.6 Modbus TCP as client.

7. MQTT Communication setting :-

- 7.1. For MQTT communication, the transfer protocol should be set to "None," and the work mode should be "TCP Client," as shown in Fig.7.

Fig.7 Device Setting

- 7.2. In device setting interface click on firmware configuration as show in fig 7
7.3. In configuration save location select folder which is created on your PC, as show in fig 8.
7.4. Then click on MQTT configuration

Fig.8 firmware configuration

MQTT settings

Port for MQTT (only supported by XX12 series):

MQTT server IP:

MQTT server port:

User name:

Key:

MQTT ID(Unique):

Subscribe Topic1:

Subscribe Topic2:

Subscribe Topic3:

Publish Topic:

Fig.9 MQTT Setting

7.5. Configure the MQTT Broker, MQTT server IP, port, username, password, subscribe topic, publish topic and save it, then click on "Download" as you see in fig 8 .

8. JSON Configuration:

8.1. After configuring MQTT, return to the firmware configuration interface and click on the JSON configuration, as shown in Figure 8, Download JSON

JSON To Modbus RTU Settings

Config and Options

Select port (only supported by XX12 series): ☐ Time sharing collection for each port

Time zone: ☐ The keyword name is Unicode encoding

1. Data transmit interval to (ms, range: 100 - 31718940, max 8.8hours, 0 is no send)

☐ Enable short link, when time come start link, then wait ms for establish TCP connection

Then send data, then after is close connection. ☐ Upload according to NTP time.

2. Select the cloud platform to access:

3. The Uplayer Protocol of JSON:

GET/POST URL(not include the ahead "http://")

The Variable Name of the POST(No need for pure json):

4. Add prefix to upload data(e. g. 01 02): Format:

Reg packet (sent when connecting to server):

5. After times of upload, serial send data: Condition(Def. empty):

Design timing send serial command table(support transparent transmission when NO JSON):

6. Add or Remove Modbus Registers:

7. Click to save JSON settings and display the results:

8. Export/Import config file.

Fig.10 MQTT Setting

- To set (water, energy) meters parameter, Click on the "JSON upload".
- Add slave address.
- Add the corresponding JSON keyword to store multiple readings of the energy meter. This keyword can be a number or a character, depending on the energy meter
- Add the Modbus function code so that you know which number corresponds to which function
- Add register address as per energy meter
- Then, click on "Enter Next." The register address will increase by 1, and you must assign the corresponding JSON keyword one by one after every click
- Click on "Save and Exit." The saved JSON parameters will be visible. Then, go back to firmware configuration interface and click on "Download as show in fig 8."
- If You have no of parameter's 100.
- You can import and export JSON files and make the required changes directly, making it easier to edit and manage the JSON configuration.
- Go to JSON to RTU interface at 8 no. there Is Upload Import and Upload export if you want to download existing JSON just click on Upload Export or If you want to import just click on Upload import.
- There is a standard CSV format that the system supports. Therefore, it would be better to

create a simple JSON structure first, export it to CSV, and then make the required changes directly in the CSV file.

Following is the 1. th design of register. It has been added: ☒

JSON node data type: ☒ Object data(Default value, including this node and later ones with { }, need Input JSON keyword)
☐ Array data(including data by [], without JSON keyword)

Corresponding JSON Keyword: 1 Data source: Modbus RTU

Modbus RTU Settings

- Slave Address: 0 - IP: 0 . 0 . 0 . 0
- Modbus Function Code: 3 - Port: 502
- Register Address: 0

645/698 Protocol

- 645/698 Version: 97 Version
- Device ID(6B): 000000000001
- Data type: 9410
- Keep invalid 0 ☐
- Read FE numbers: 0
- Write FE numbers: 0
- 698 Data type: Total positiv
- 698 Client Addr(CA): 0

1. Data length: 2 Bytes. 4 Bytes order: Big Endian (AI) (big-endin 4 bytes: Data ABCD, low address store 2 bytes AB)

2. Decimal point places: 4 digit. After get as intenger left shift the decimal point.

3. Enable shift and scale: ☐ Subtract integer: 0 then divide float: 1 Register is float ☐

4. Data format: Unsigned int Bool value at postion bit: 1

5. Add unit name to rear:

6. Add quotation to data: ☐

7. The Period between two RTU cmd: 100 (ms) minimum 10. 100ms for 9600bps, and 500ms for 2400bps.
 If timeout wait more: 0 (ms), before send next command. Set 0 to disable this function.

8. Transmit data to server when data changes: ☐

9. If RS485 device offline, set special value: ☐ Special value type: Special vs , special value: 0 .Set data to 1 if online: ☐

10. Enable overrun alarm: ☐ , minimum normal value: 0 maximum normal value: 0

Embedded JSON Related

Enter Embedded Exit Embedded

Design and View

Enter Next Del and Next

Exit Design

Save and Exit Cancel and Exit

Fig.11 Add JSON node

9. Output on MQTT Explore:-

Topic

MQTT Connection ws://test.mosquitto.org:1883/

Name test.mosquitto.org Validate certificate Encryption (tls)

Protocol Host Port

ws:// test.mosquitto.org 1883

Basepath Username Password

DELETE ADVANCED SAVE CONNECT

Fig.12 MQTT Explore Application

➤ You can search the topic which is configure in device.

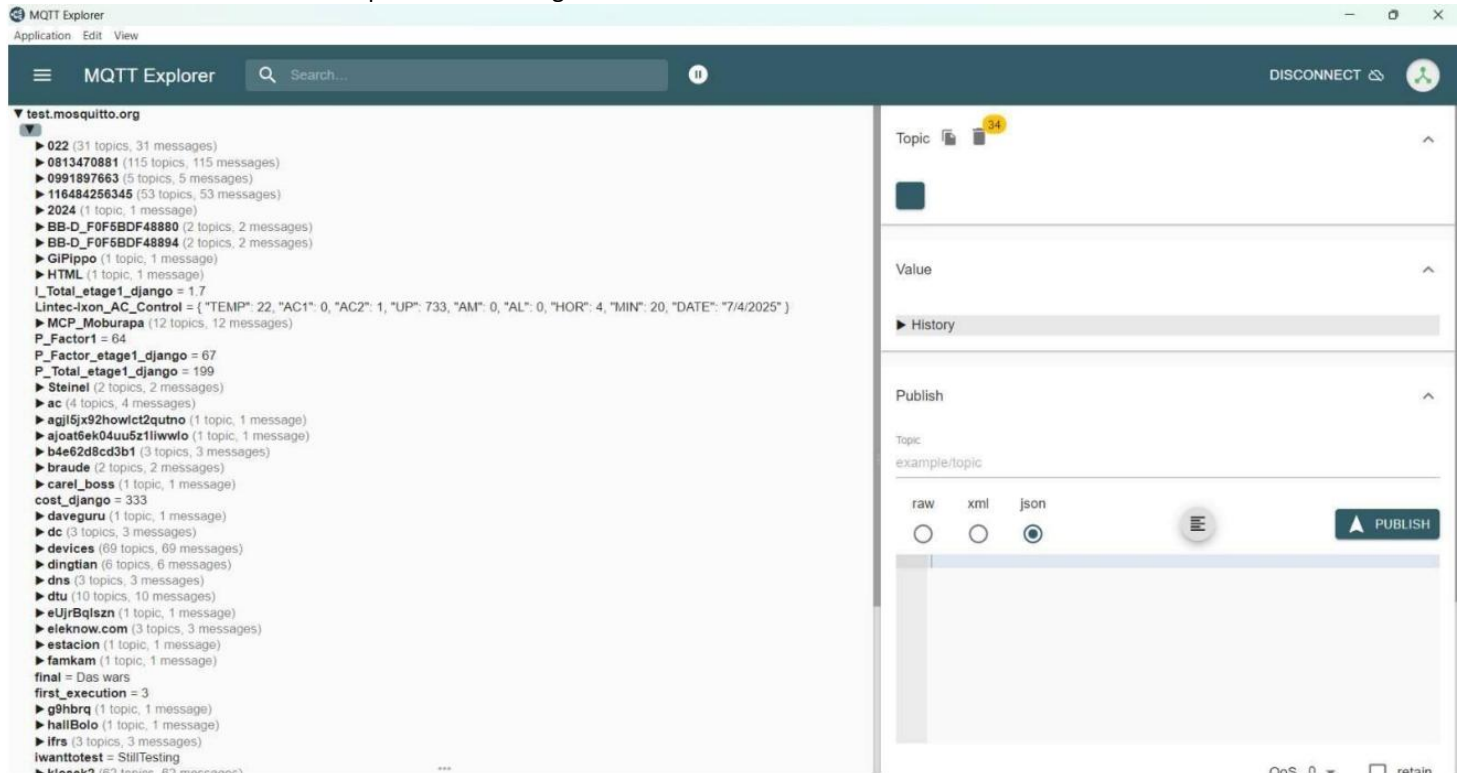


Fig.13 Broker interface

10. HTTP Configuration:-

10.1. Go to the device interface, then select work mode as TCP client. Enter your Platform ID in dest. IP/domain, set the port to 8080, and click "modify settings".

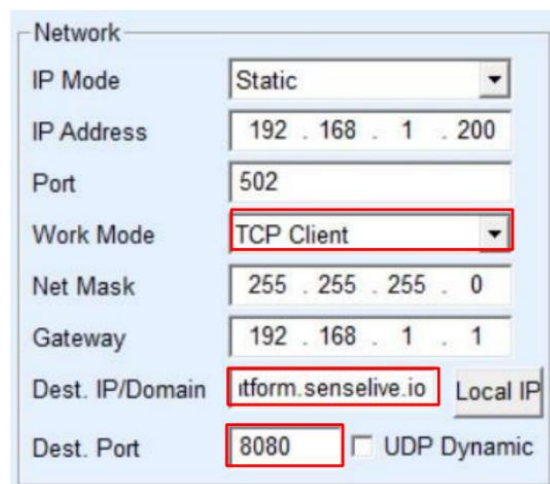


Fig.14 device setting interface

10.2. Next, click on the "JSON" tab, set the Data Transmit Interval to 3000, and select "HTTP POST" as the Uplayer Protocol. Enter the URL of your platform, then click "Save Settings".

JSON To Modbus RTU Settings

- Data transmit interval to server: (ms, range: 100 - 31718940, max 8.8hours)
☐ Enable short link, when time come start link, then wait ms for establish TCP connection
 Then send data, then after 1s close connection. ☐ Upload according to NTP time.
- Select the cloud platform to access:
- The Uplayer Protocol of JSON:
 GET/POST URL(not include the ahead "http://")
 The Variable Name of the POST(No need for pure json):
- Add prefix to upload data(e.g. 01 02): Prefix format:
- After times of upload, serial send data: Condition(Def. empty):
- Add or Remove Modbus Registers:
- Click Save Setting and dispaly ☐ Keyword name is Unicode
- Export/Import config file.

```
POST /api/v1/uwIaJQdobRwymKKFJSpg/telemetry HTTP/1.1
Content-Type: application/json
Content-Length:
Host: platform.senselive.io
```

10.3. After the HTTP configuration, proceed with the JSON configuration by following the steps given above to complete the process

JSON To Modbus RTU Settings

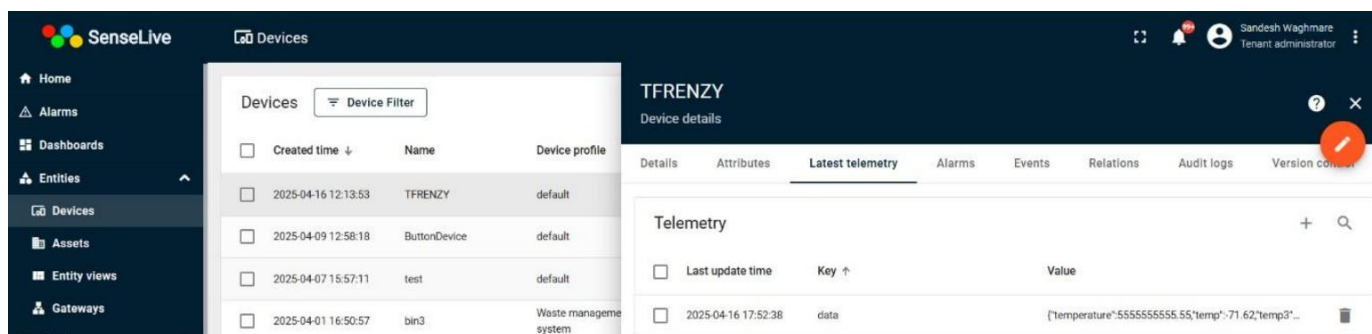
- Data transmit interval to server: (ms, range: 100 - 31718940, max 8.8hours)
☐ Enable short link, when time come start link, then wait ms for establish TCP connection
 Then send data, then after 1s close connection. ☐ Upload according to NTP time.
- Select the cloud platform to access:
- The Uplayer Protocol of JSON:
 GET/POST URL(not include the ahead "http://")
 The Variable Name of the POST(No need for pure json):
- Add prefix to upload data(e.g. 01 02): Prefix format:
- After times of upload, serial send data: Condition(Def. empty):
- Add or Remove Modbus Registers:
- Click Save Setting and dispaly ☐ Keyword name is Unicode
- Export/Import config file.

```
POST /api/v1/uwIaJQdobRwymKKFJSpg/telemetry HTTP/1.1
Content-Type: application/json
Content-Length:
Host: platform.senselive.io

{
  "data": 0
  {
    "temperature": 0,
    "temp": 0,
    "temp3": 0,
    "DeviceID": "123456789"
  }
}
```

10.4. As you can see in fig 13, the set parameters is visible on box.

10.5. The output is visible on your platform.



The screenshot displays the SenseLive web application interface. On the left is a navigation sidebar with options: Home, Alarms, Dashboards, Entities, Devices (selected), Assets, Entity views, and Gateways. The main content area is titled 'Devices' and includes a 'Device Filter' dropdown. Below this is a table listing devices:

Created time	Name	Device profile
2025-04-16 12:13:53	TFRENY	default
2025-04-09 12:58:18	ButtonDevice	default
2025-04-07 15:57:11	test	default
2025-04-01 16:50:57	bin3	Waste management system

To the right of the table is a 'Device details' panel for the selected device 'TFRENY'. It features tabs for Details, Attributes, Latest telemetry (active), Alarms, Events, Relations, Audit logs, and Version control. The 'Latest telemetry' tab shows a 'Telemetry' section with a table of data points:

Last update time	Key	Value
2025-04-16 17:52:38	data	{ "temperature": 55.55, "temp": -71.62, "temp3": ... }

WHAT'S IN THE BOX

- ▶ 1 × Product unit
- ▶ 1 × Quick start guide
- ▶ 1 × Mounting kit (DIN-rail clip)
- ▶ 1 × Antenna (where applicable)

NOTES & DISCLAIMERS

Specifications are subject to change without prior notice. All trademarks are the property of their respective owners. For the latest revision of this datasheet and complete user documentation, visit www.SenseLive.io.