

Senselive X9000

4-way DI/AI IO Controller/4G Gateway



OVERVIEW

X9000 is a 4G data acquisition gateway with RS485 support for remote control. It features 4 DI/AI inputs, optical isolation, 12-bit ADC, and configurable AI (4-20mA, 0-5V, 0-10V). Supports TCP, HTTP, MQTT, and converts DLT645/Modbus RTU to JSON for seamless data integration.

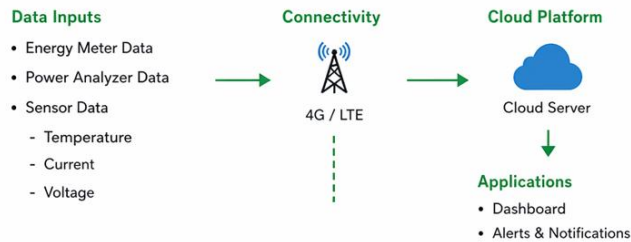
KEY HIGHLIGHTS

- ▶ **Cellular Connectivity:** 4G LTE (FDD/TDD) with 2G GSM fallback
- ▶ **Smart Data Processing:** Data alarms, scaling, change-based uploads & offline alerts
- ▶ **Protocol Conversion:** Modbus RTU to JSON.
- ▶ **I/O:** 4× Digital Inputs, 4× Analog Inputs (4–20 mA, 0–10 V, Resistance)
- ▶ **Communication Protocols:** Modbus RTU/TCP, MQTT, HTTP, TCP/UDP.
- ▶ **Modbus Cache:** Up to 10,000 registers for high-speed responses
- ▶ **Remote Management:** Web UI, SLVirCom, AT Commands & FOTA
- ▶ **Industrial Grade:** 9–24 VDC, –40°C to +85°C, metal enclosure, wall mount

APPLICATIONS

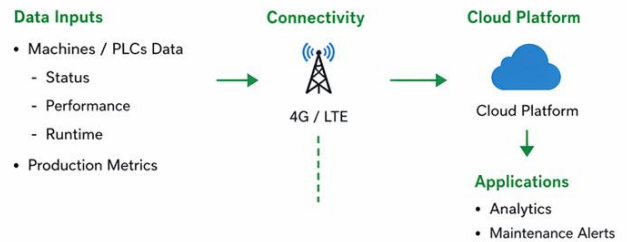
1. ENERGY MONITORING

Real-time monitoring of energy parameters and consumption.



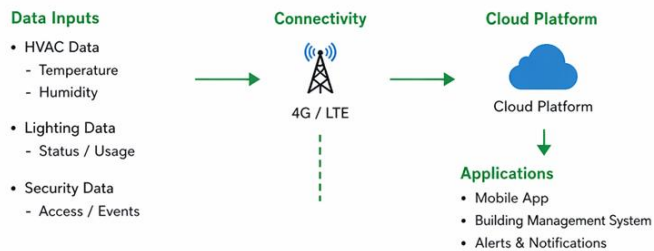
2. SMART MANUFACTURING

Improve productivity and reduce downtime with real-time data.



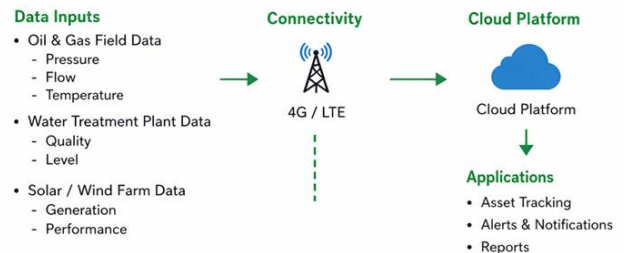
3. BUILDING AUTOMATION

Automate and monitor building systems for efficiency and comfort.

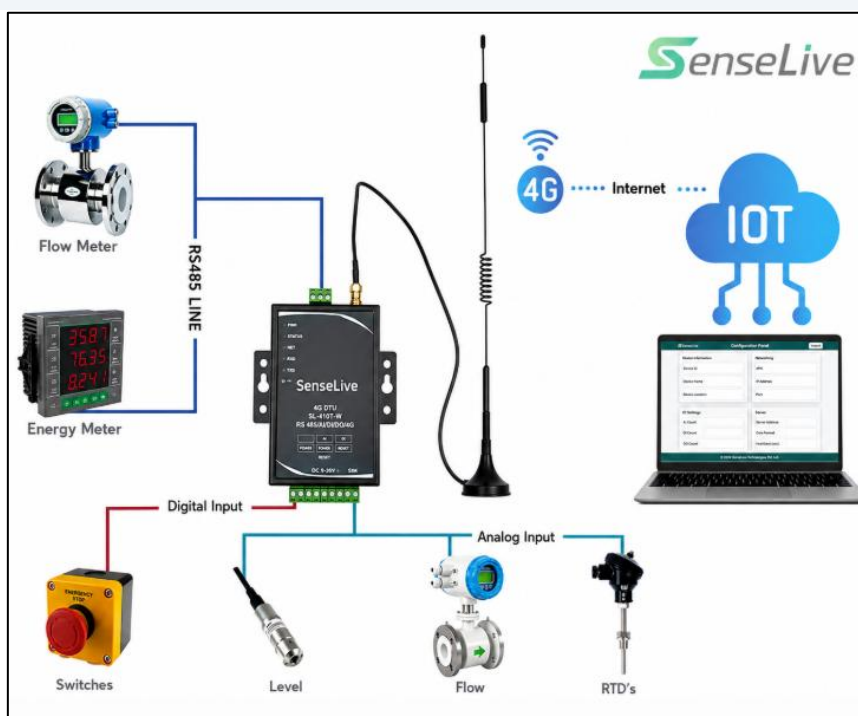


4. REMOTE ASSET MANAGEMENT

Monitor and manage remote assets from anywhere.



FUNCTIONAL BLOCK DIAGRAM

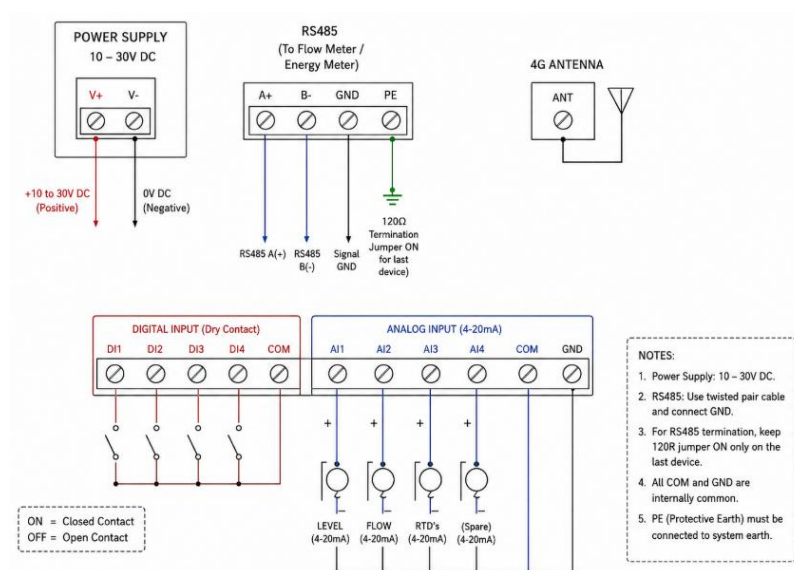


TECHNICAL SPECIFICATIONS

GENERAL	
Model Number	X9000
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
Wireless	4G LTE Cat-1
Serial Port Type	RS485*2: 485IO, 485-4G
Protocols	Modbus RTU, Modbus TCP, MQTT, HTTP/REST, JSON
Antenna	50 Ω SMA female (stick or suction-cup)
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)	9.4x 6.5 x 2.5mm

Weight	≈ 150 g
Indicators	Power, Network, Data LEDs
COMPLIANCE	
Certifications	CE, FCC, RoHS, BIS (where applicable)
Warranty	12 months from invoice date(Only on manufacturing defect)

Terminal Diagram



Parameter Configuration

1 Software Installation:-

To download Vicrom software just click on the below link.

<https://senselive.in/downloads/software/SenseLiveConfig2/SenseLiveConfig2.zip>

2 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
- +4G & -4G(To send data) and +IO & -IO(To set parameter of device).
- **AI port:** Connect the analog device to any AI port (AI1, AI2, AI3, or AI4).

3 IO controller configuration:-

1. After installing SenseConfig_Tool, connecting the hardware, running the software, and clicking on "Device Management," proceed with the necessary configurations.

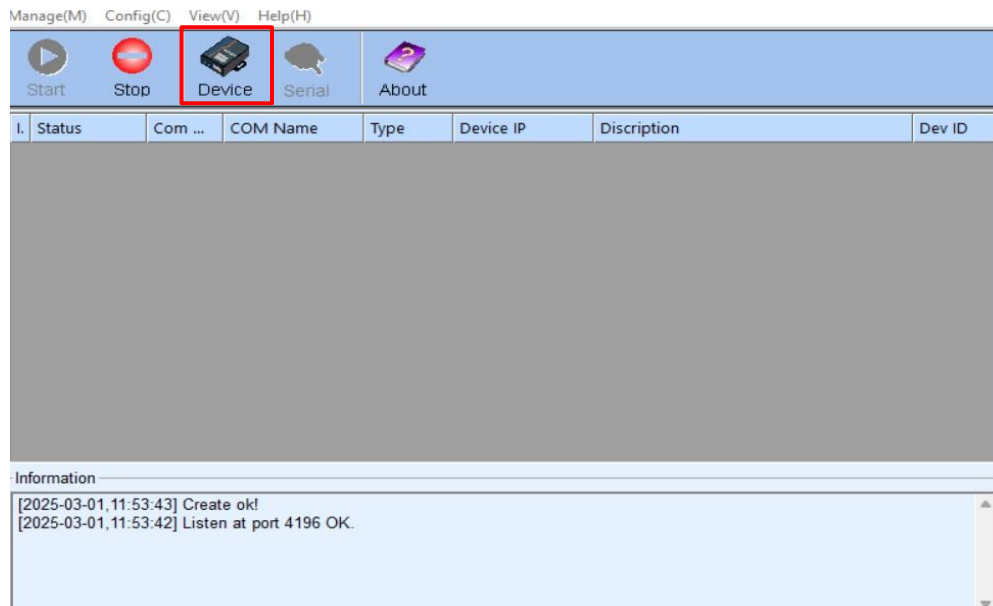


Fig.1 Device management

2. In the Device Management interface, click on "IO controller" as shown in the figure.

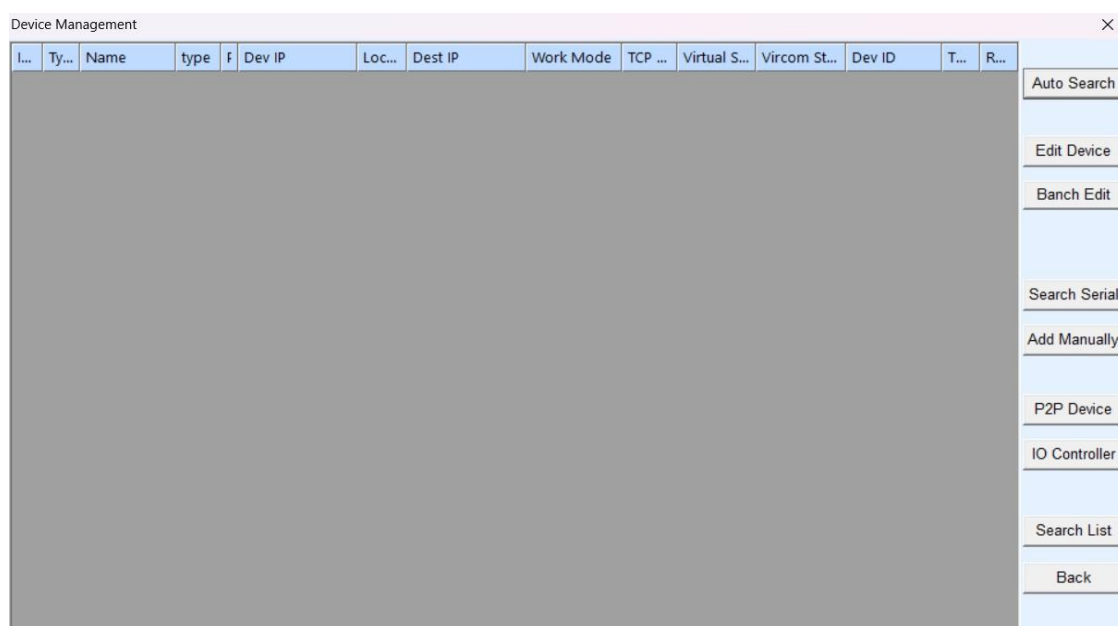


Fig.2 Device management

3. In the IO controller interface, click on "Open and Search." Then, the information box will

display “Found Device.”

The screenshot shows the 'remote digital IO control and analog acquisition' window. The 'Communication through RS485/RS232' section has 'COM7' selected, 'Baud rate' set to '115200', and 'Parity' set to 'None'. The 'Open and Search' button is highlighted with a red rectangle. The 'Information' panel on the right displays a log of found devices: '[12:30:57] Found device with 9600bps.' and '[11:33:12] Found device with 9600bps.'

Fig.3 IO controller

4. For the DI and AI upload simultaneously click on “DI auto report type” and set DI+AI as shown in fig 4.
5. A1 auto report should be 100.
6. If you want to edit Modbus parameter ,you can edit Baudrate ,Slave ID, Parity, stop bit.
7. For AI calculation

The specific usage of the data 01 82 is related to the type of AI. Converting 01 82 to decimal gives Vin=386. The calculation formula for different AI types is as follows:

- 5V : Actual voltage value = $(V_{in} / 1024) * 5 = 1.8848$;
- 10V : Actual voltage value = $(V_{in} / 1024) * 10 = 3.7695$;
- 4~20mA : Real current = $(A_{in} / 1024) * 5 / 200 * 1000 = 9.4238$;

The screenshot shows the 'Serial parameters' section. The 'DI auto report type' is set to 'DI+AI' and 'AI1 Auto-report' is set to '100', both highlighted with red rectangles. Other parameters include 'Device addr: 1', 'Baud rate: 115200', 'Parity: None', 'Firmware Ver.', 'DO PowerOn: 0x', '32bit DI count save: 0', 'DO hold time: 0', 'DO hold bit sel.', 'DI hold it for 2 seconds', and 'DI debounce for 50ms'.

Fig.4 serial parameter

4 4G Configuration:-

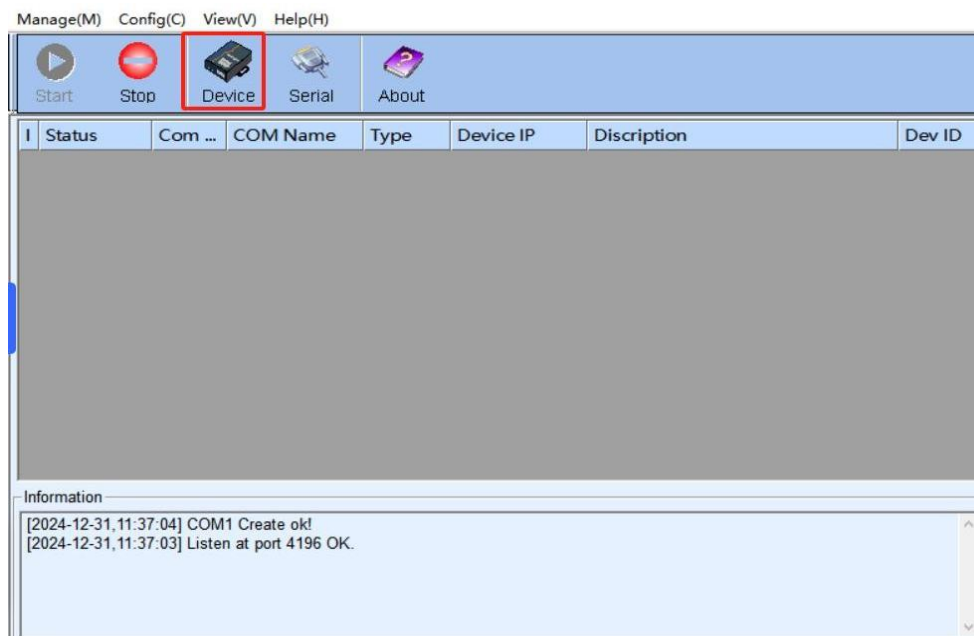


Fig.5 Device management

- In the Device Management interface, click on “Search Serial” as shown in the figure. Then, select a baud rate and COM port. After that, click on "Search," and the device will be connected.

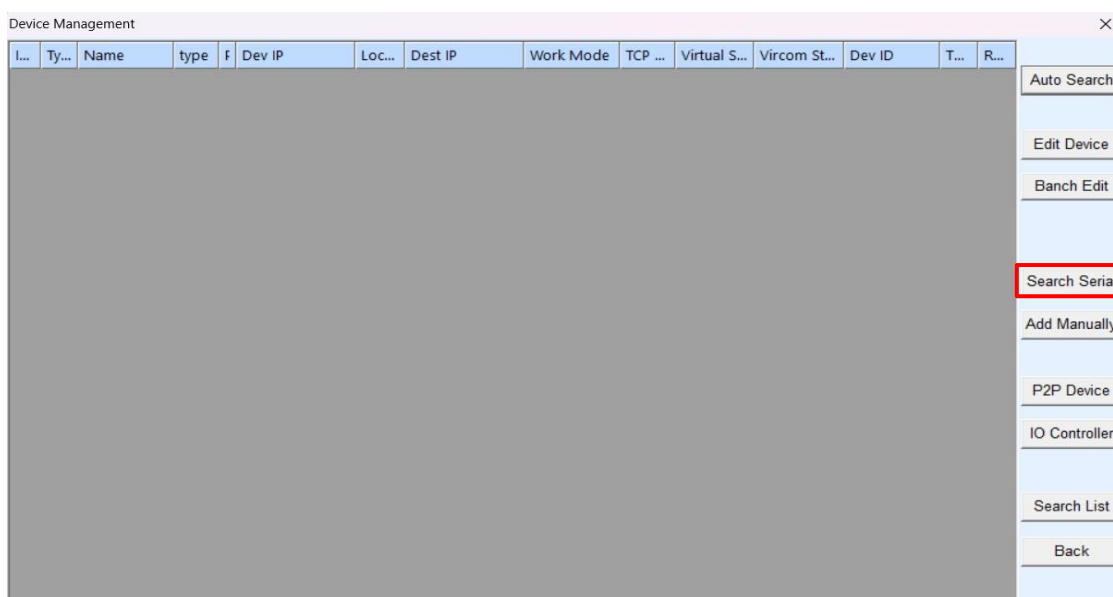


Fig.6 Device management

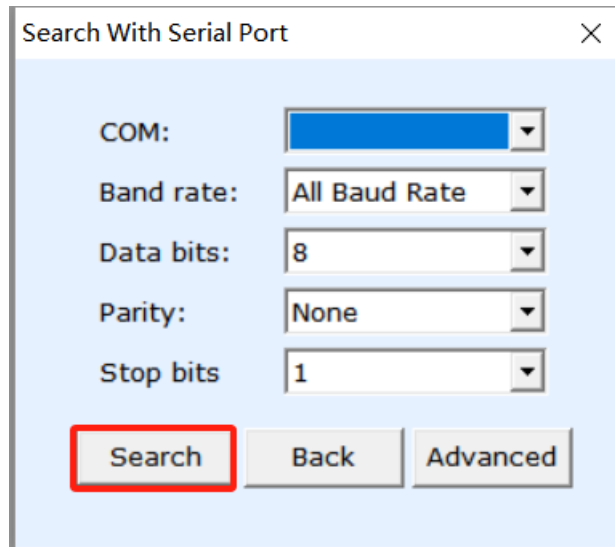


Fig.7 Search serial

- After the selection of com port and band rate then go back to device setting interface and click on search configuration tool interface open as you see in fig 8.

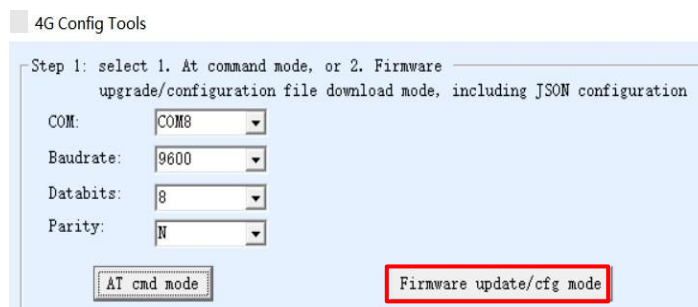


Fig. 8 configuration tool

4. MQTT Communication settings :-

- In fig 8 configuration tool, there is “firmware configuration” click on that.
- In 4G configuration tool interface click on firmware configuration as show in fig 8.
- In configuration save location select folder which is created on your PC, as show in fig 8.
- Then click on MQTT configuration, MQTT interface is open as you see in fig 10.

Webpage&code download tool

☒ Direct download mode

Configuration save location: C:\Users\haris\OneDrive\Desktop\SL6408

Special configs:

Config file source: Read from local directory

Modbus cfg. **MQTT cfg.** **JSON cfg.** Reg packet Cmd change HTTP cfg. Param file Clear local dir.

☐ Code file download mode

Select code file: C:\firmware.bin

☐ Download through the network

Device IP address or domain: 192.168.1.200

Download port (Don't modify): 1092

☒ Download through serial port

Serial port: COM7

Baud Rate: 9600

Flash size: 256 KB

DevID: Bind ID

Please close any other configuration window before downloading.

Download

Fig.9 firmware configuration

MQTT settings

Port for MQTT (only supported by XX12 series): 1

MQTT server IP:

MQTT server port: 1883

User name: mqttname

Key:

MQTT ID (Unique): mqttid18779 ☐ Add device ID at the end

Subscribe Topic1: mqttsub

Subscribe Topic2:

Subscribe Topic3:

Publish Topic: mqttsub ☐ Add device ID at the end

Fig.10 MQTT Setting

Configure the MQTT Broker and save it, then click on "Download" as you see in fig 9.

4. JSON Configuration:-

- After configuring MQTT, return to the firmware configuration and click on the JSON configuration, as shown in Figure 9, and Download JSON.

Fig.11 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 9.”
- 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 Other Data source: Current Time Format: 2025-02-22 16:05:10 Fixed String: No quotation ☐

Modbus RTU Settings

- Slave Address: 1 - IP: 0 . 0 . 0 . 0
 - Modbus Function Code: 3 - Port: 502
 - Register Address: 0

645/698 Protocol

- 645/698 Version: 97 Version - Read FE numbers: 0
 - Device ID(6B): 000000000001 - Write FE numbers: 0
 - Data type: 9410 - 698 Data type: Total positiv
 - Keep invalid 0 ☐ - 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 va , special value: 0 .Set data to 1 if online: ☐
 10. Enable overrun alarm: ☐ , minimum normal value: 0 maximum normal value: 0

Design and View

Enter Next Del and Next

Exit Design

Save and Exit Cancel and Exit

Fig.12 Add JSON node

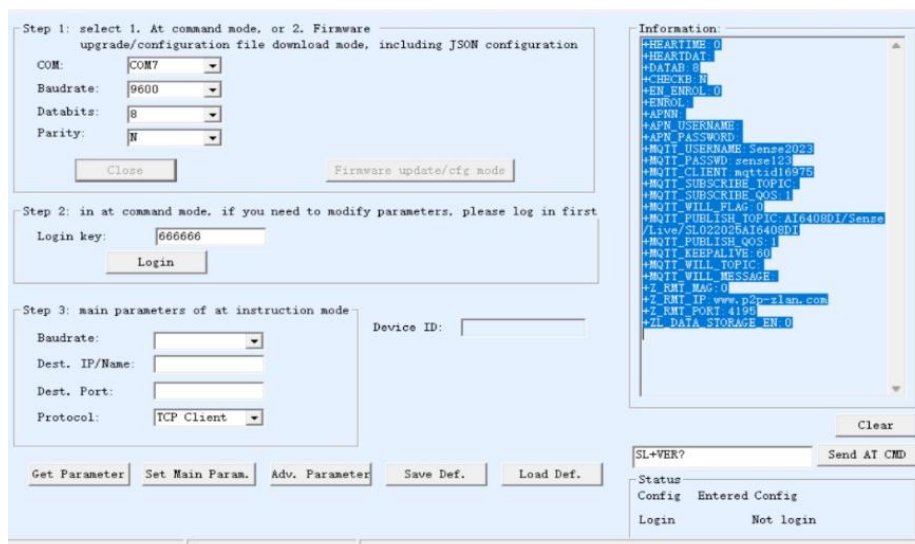
- Modbus address:-

Function code	Function	Address range
01/02	Read DI	0~3
04	Read AI	0~3
04	Read AI high precision value	32~35
03	Read the basic parameters.	63~67

03	Read extended parameters	68~162
03	Read DI 16-bit count	0~3
03	Read DI 32-bit counting	256~263
06	Set parameters	63~67
06	Set the extended parameters	68~162
06	Set DI 16-bit count	0~3

- Check parameter in configuration tool:-

1. Go back to the device management interface, click on Serial Search, and the Configuration Tool interface will open.
2. Click on "AT cmd mode." On right side information box show the data.
3. Click on "Login."
4. The default login key is 666666.
5. After logging in, you can change the baud rate and other parameters by clicking on "Advance parameters".



Step 1: select 1. At command mode, or 2. Firmware upgrade/configuration file download mode, including JSON configuration

COM: COM7
Baudrate: 9600
Databits: 8
Parity: N

Close Firmware update/cfg mode

Step 2: in at command mode, if you need to modify parameters, please log in first

Login key: 666666
Login

Step 3: main parameters of at instruction mode

Baudrate: [dropdown]
Dest. IP/Name: [text]
Dest. Port: [text]
Protocol: TCP Client [dropdown]

Device ID: [text]

Get Parameter Set Main Param. Adv. Parameter Save Def. Load Def.

Information:

```
+HEARTTIME 0
+HEARTDAT 0
+DATAS 0
+CHECK 0
+EN_ENROL 0
+ENROL 0
+APN 0
+APN_USERNAME
+APN_PASSWORD
+MQTT_USERNAME Sense2023
+MQTT_PASSWORD sense123
+MQTT_CLIENT mqttid16975
+MQTT_SUBSCRIBE_TOPIC
+MQTT_SUBSCRIBE_QOS 1
+MQTT_WILL_FLAG 0
+MQTT_PUBLISH_TOPIC AI640801/Sense
/Live/SLO22026AI640801
+MQTT_PUBLISH_QOS 1
+MQTT_KEEPAIVE 60
+MQTT_WILL_TOPIC
+MQTT_WILL_MESSAGE
+Z_RMT_MAC 0
+Z_RMT_IP www.p2p-zlan.com
+Z_RMT_PORT 4195
+Z_DATA_STORAGE_EN 0
```

Clear

SL+VER? Send AT CMD

Status
Config Entered Config
Login Not login

Fig.13 Main parameter

1. After logging in, you can change the baud rate and other parameters by clicking on “Advance parameters”.

2. Output on MQTT Explore:-

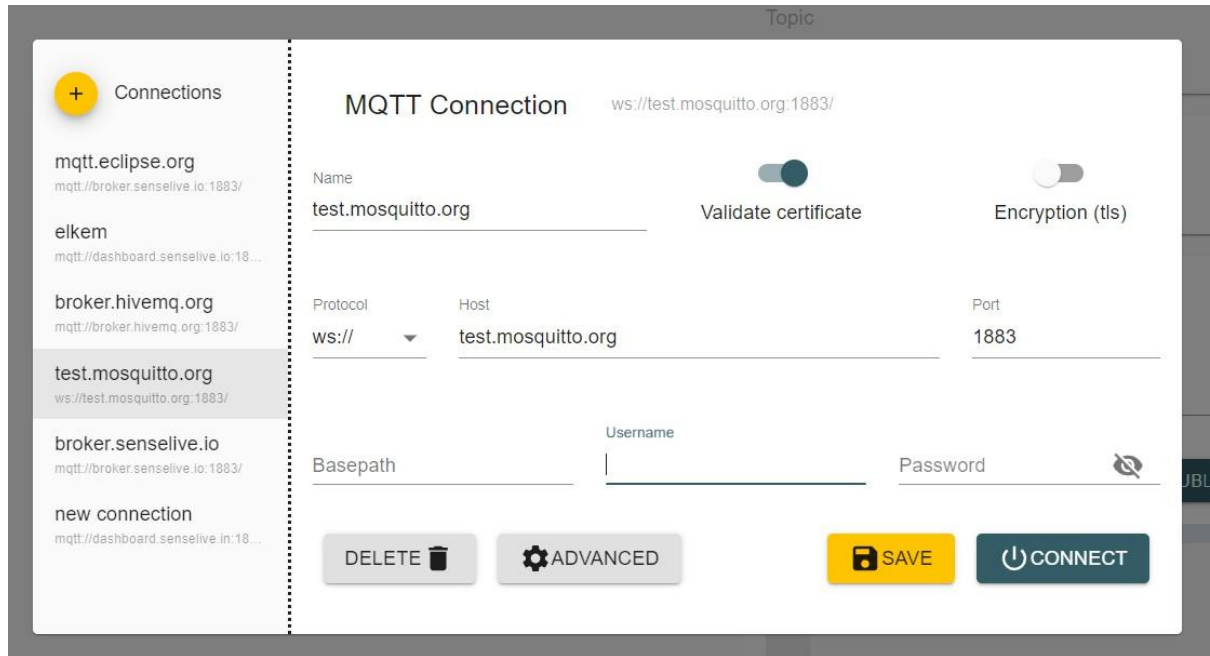
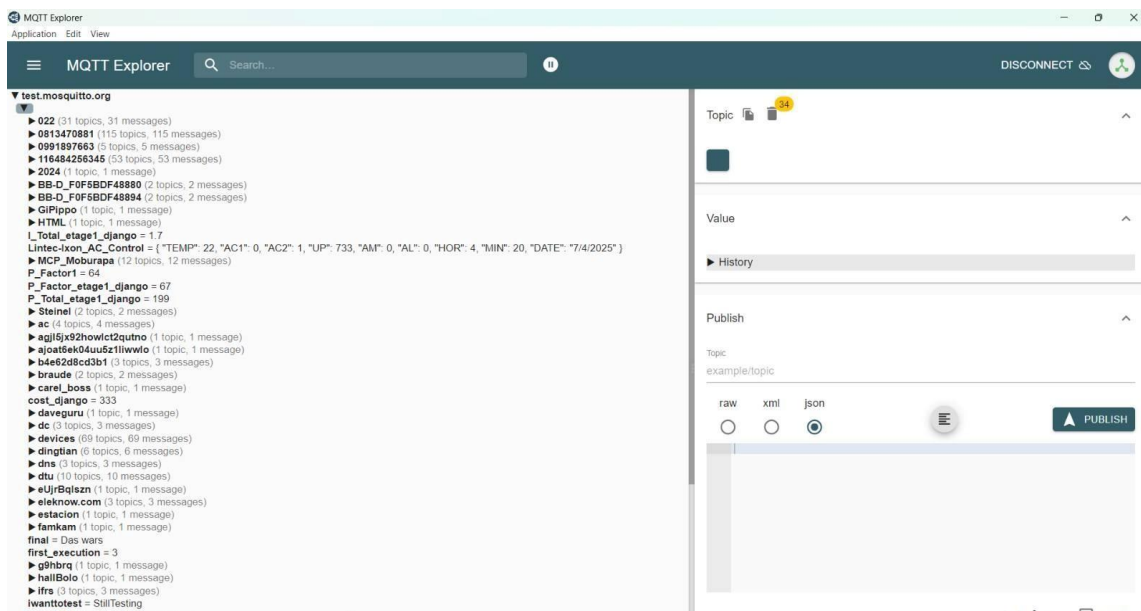


Fig.14 MQTT Explore Application

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



- HTTP Configuration:-
 - 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".

Network

IP Mode: Static

IP Address: 192 . 168 . 1 . 200

Port: 502

Work Mode: TCP Client

Net Mask: 255 . 255 . 255 . 0

Gateway: 192 . 168 . 1 . 1

Dest. IP/Domain: itform.senselive.io Local IP

Dest. Port: 8080 ☐ UDP Dynamic

Fig.14 device setting interface

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

JSON To Modbus RTU Settings

1. Data transmit interval to server: 3000 (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 HTTP time.

2. Select the cloud platform to access: None

3. The Uplayer Protocol of JSON: HTTP POST

GET/POST URL(not include the ahead "http://") senselive.io/api/v1/uwIaJQdobRwymKKFJSpg/telemetry

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

4. Add prefix to upload data(e.g. 01 02): Prefix format: HEX

5. After 1 times of upload, serial send data: Condition(Def. empty):

6. Add or Remove Modbus Registers: JSON Upload JSON Download Remove All

7. Click Save Setting and disply Save Setting ☐ Keyword name is Unicode

8. Export/Import config file. Upload Export Upload Import Download Export Download Import

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

Fig.13 JSON configuration interface

- 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 Uploader 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: JSON Upload JSON Download Remove All
- Click Save Setting and display Save Setting ☐ Keyword name is Unicode
- Export/Import config file. Upload Export Upload Import Download Export Download Import

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

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

Fig.13 JSON configuration interface

- As you can see in fig 13, the set parameters is visible on box.
 - The output is visible on your platform.

Created time	Name	Device profile
2025-04-16 12:13:53	TFRENZY	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

TFRENZY Device details								
Details	Attributes	Latest telemetry						
<table border="1"> <thead> <tr> <th>Last update time</th> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>2025-04-16 17:52:38</td> <td>data</td> <td>{"temperature":555555555.55,"temp":71.62,"temp3":...</td> </tr> </tbody> </table>			Last update time	Key	Value	2025-04-16 17:52:38	data	{"temperature":555555555.55,"temp":71.62,"temp3":...
Last update time	Key	Value						
2025-04-16 17:52:38	data	{"temperature":555555555.55,"temp":71.62,"temp3":...						

Fig.13 Output on platform

WHAT'S IN THE BOX

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

MECHANICAL DIMENSIONS 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.