PM1000H Meter with gateway

Software:

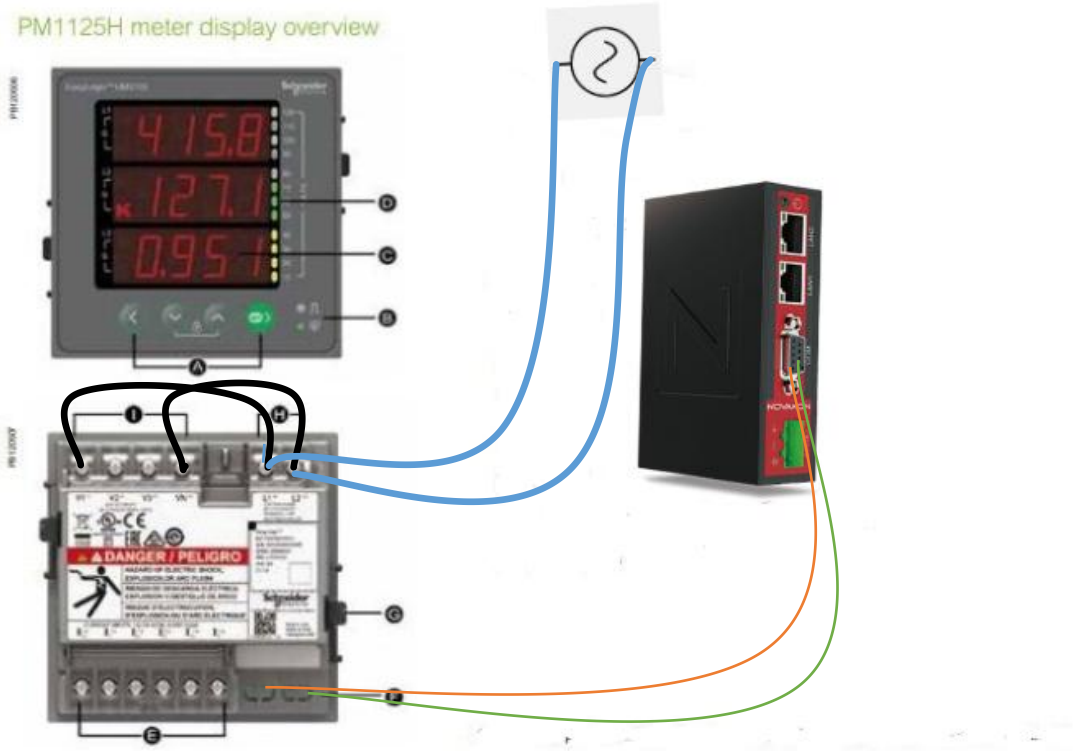
NUM	item	software	Notes
1	Novakon Gateway	Novakon web page configuration	Download : https://mosquitto.org/files/binary/win64/mosquitto-2.0.18-install-windows-x64.exe
2	mosquittolocalserver	link	

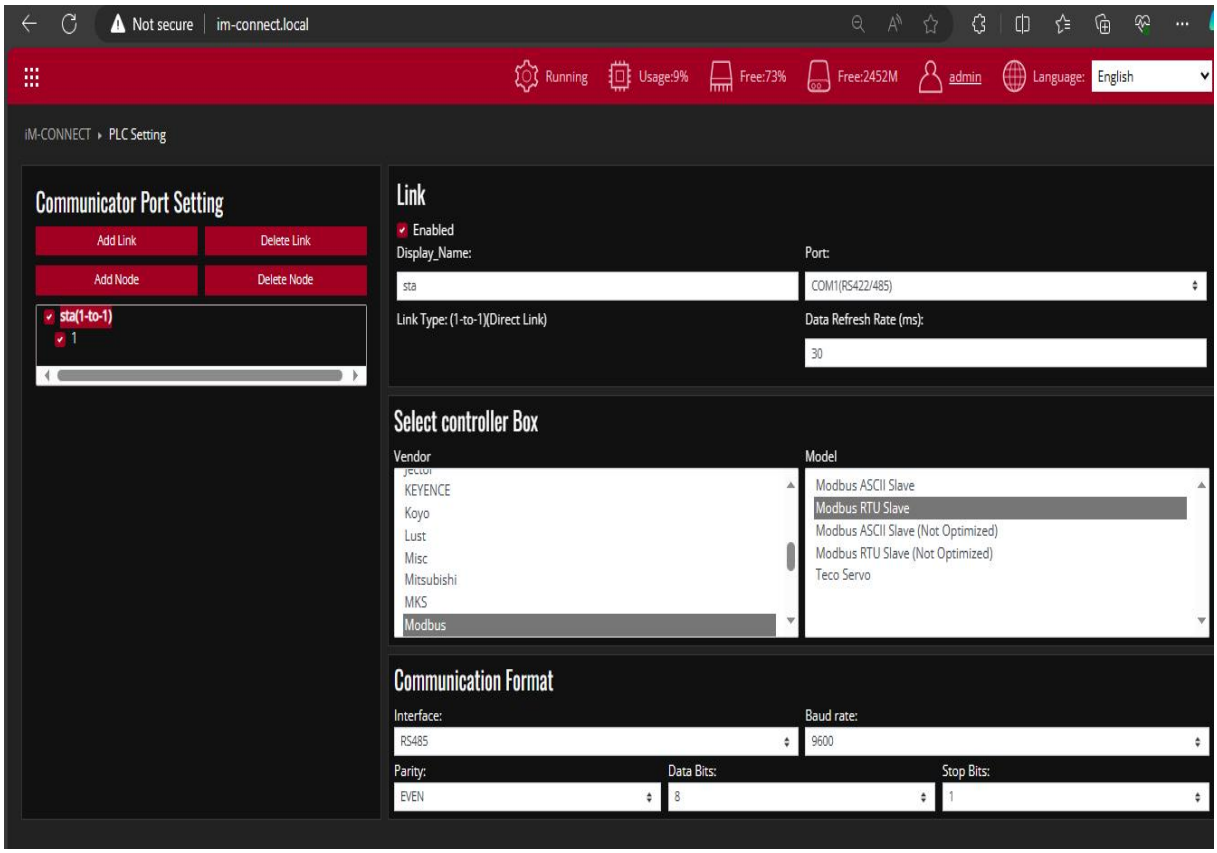
Hardware:

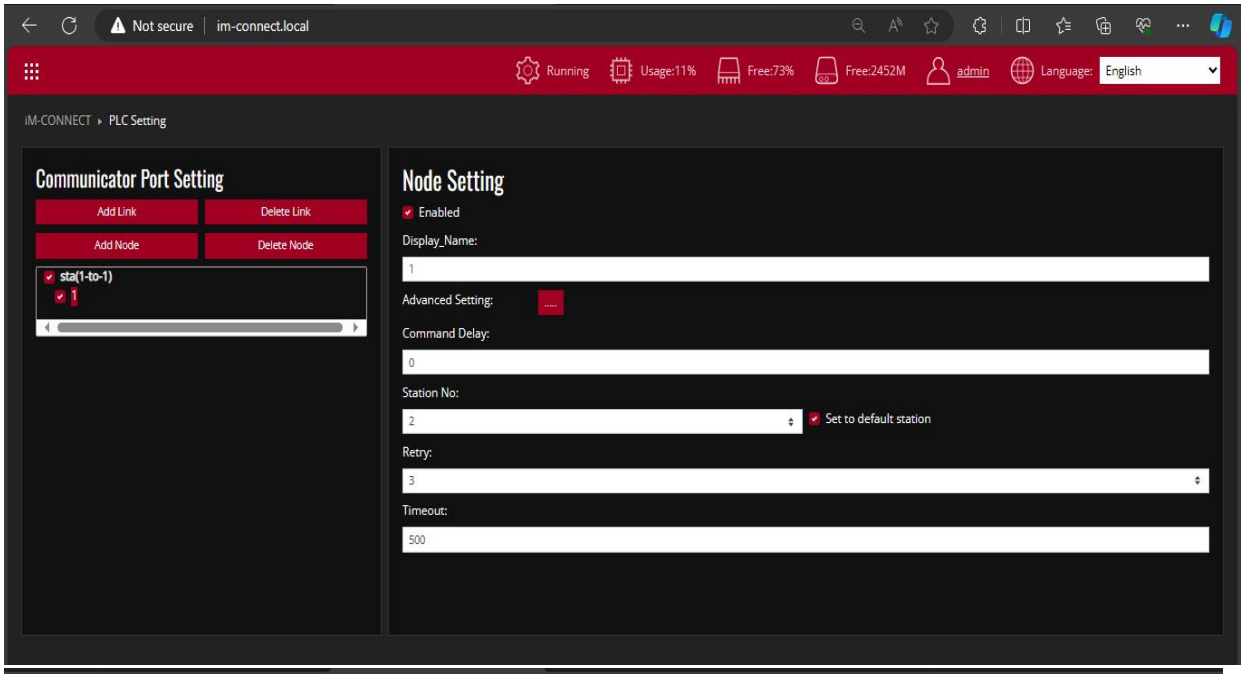
NUM	item	model
1	Novakon Gateway	GW-01

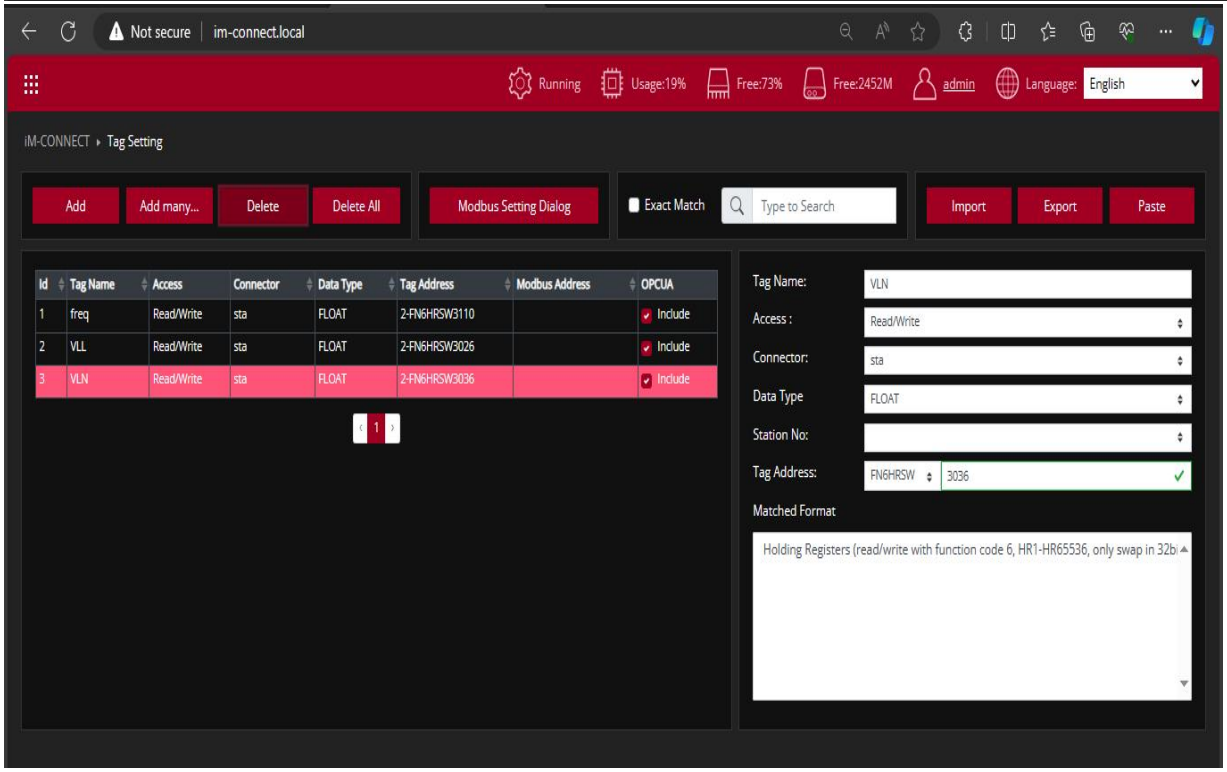
Commnuication parameter and wiring:

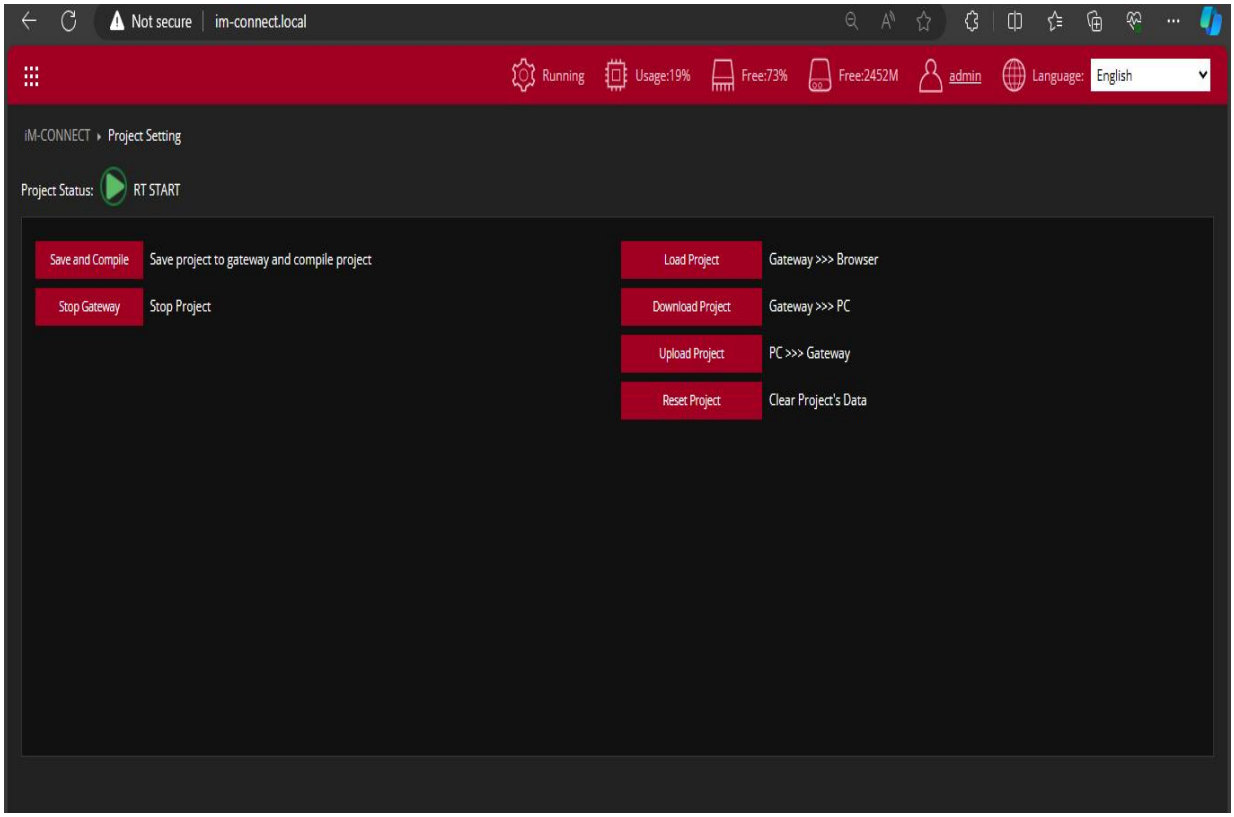
vendor	Modus
interface	RS485
Baud rate	9600
parity	even
Data bits	8
Stop bits	1
address	2


PM1125H meter display overview
415.8
127.1
0.95
DANGER / PELIGRO

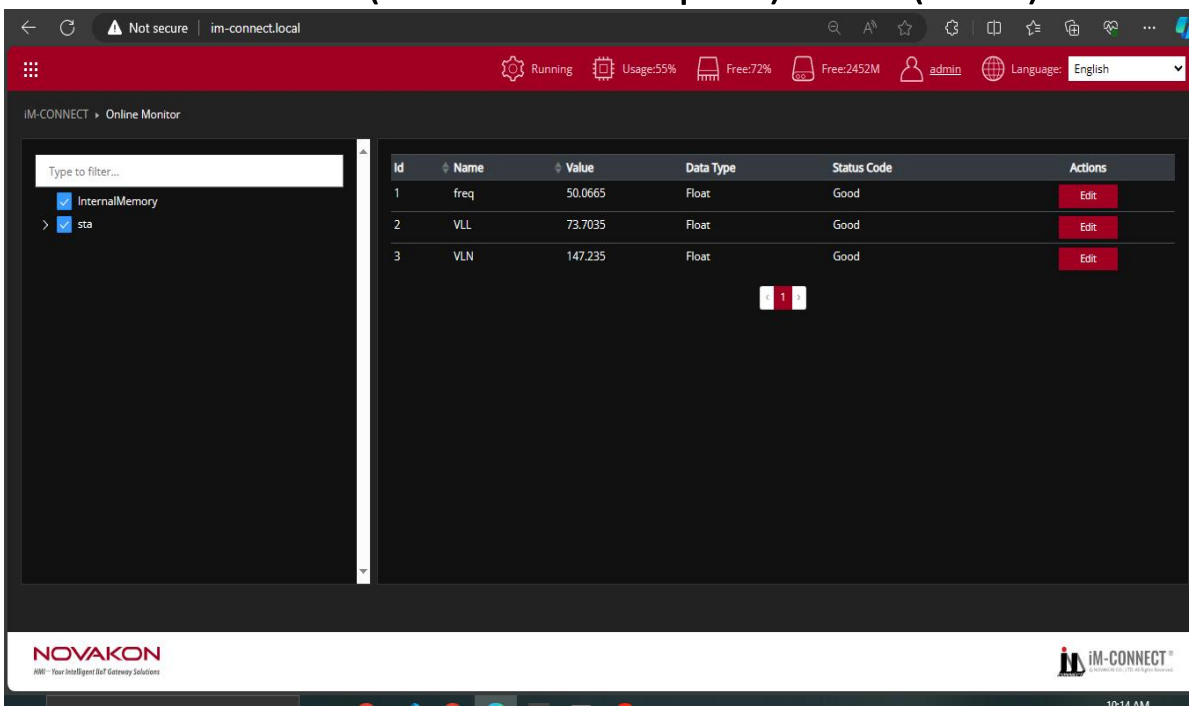

Not secure
im-connect.local
Running
Usage:9%
Free:73%
Free:2452M
admin
Language:
English
iM-CONNECT ▸ PLC Setting
Communicator Port Setting
Add Link
Delete Link
Add Node
Delete Node
sta(1-to-1)
1
Link
Enabled
Display_Name:
sta
Port:
COM1(RS422/485)
Link Type: (1-to-1)(Direct Link)
Data Refresh Rate (ms):
30
Select controller Box
Vendor
KEYENCE
Koyo
Lust
Misc
Mitsubishi
MKS
Modbus
Model
Modbus ASCII Slave
Modbus RTU Slave
Modbus ASCII Slave (Not Optimized)
Modbus RTU Slave (Not Optimized)
Teco Servo
Communication Format
Interface:
RS485
Baud rate:
9600
Parity:
EVEN
Data Bits:
8
Stop Bits:
1


Not secure
im-connect.local
Running
Usage:11%
Free:73%
Free:2452M
admin
Language:
English
iM-CONNECT ▸ PLC Setting
Communicator Port Setting
Add Link
Delete Link
Add Node
Delete Node
sta(1-to-1)
1
Node Setting
Enabled
Display_Name:
1
Advanced Setting:
Command Delay:
0
Station No:
2
Set to default station
Retry:
3
Timeout:
500


Not secure
im-connect.local
Running
Usage:19%
Free:73%
Free:2452M
admin
Language:
English
iM-CONNECT ▸ Tag Setting
Add
Add many...
Delete
Delete All
Modbus Setting Dialog
Exact Match
Type to Search
Import
Export
Paste
Id | Tag Name | Access | Connector | Data Type | Tag Address | Modbus Address | OPCUA
1 | freq | Read/Write | sta | FLOAT | 2-FN6HRSW3110 | | Include
2 | VLL | Read/Write | sta | FLOAT | 2-FN6HRSW3026 | | Include
3 | VLN | Read/Write | sta | FLOAT | 2-FN6HRSW3036 | | Include
1
Tag Name:
VLN
Access :
Read/Write
Connector:
sta
Data Type
FLOAT
Station No:
Tag Address:
FN6HRSW
3036
Matched Format
Holding Registers (read/write with function code 6, HR1-HR65536, only swap in 32b

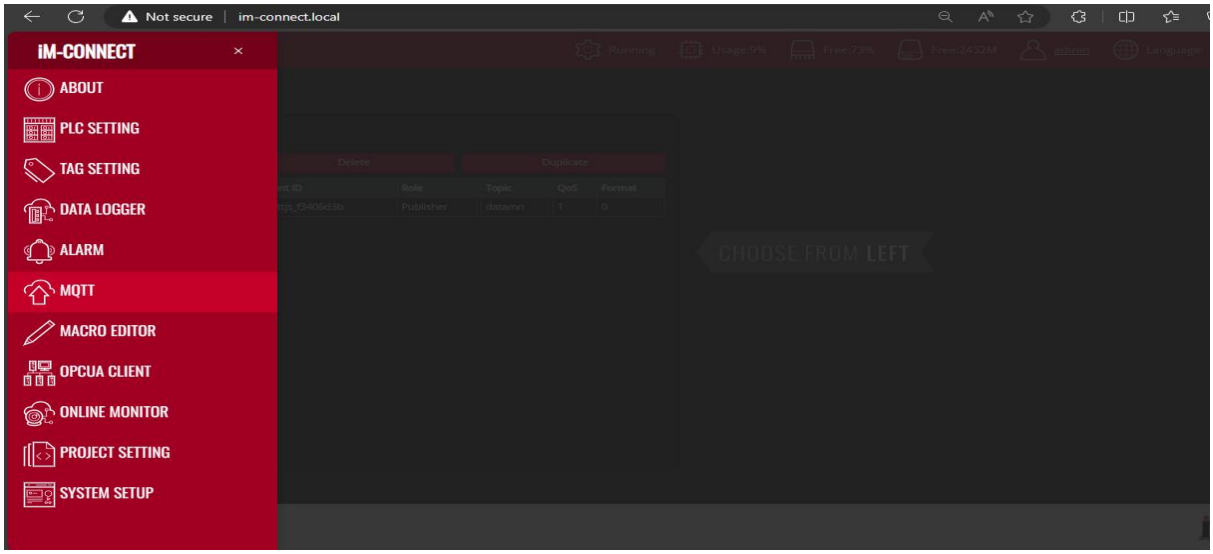


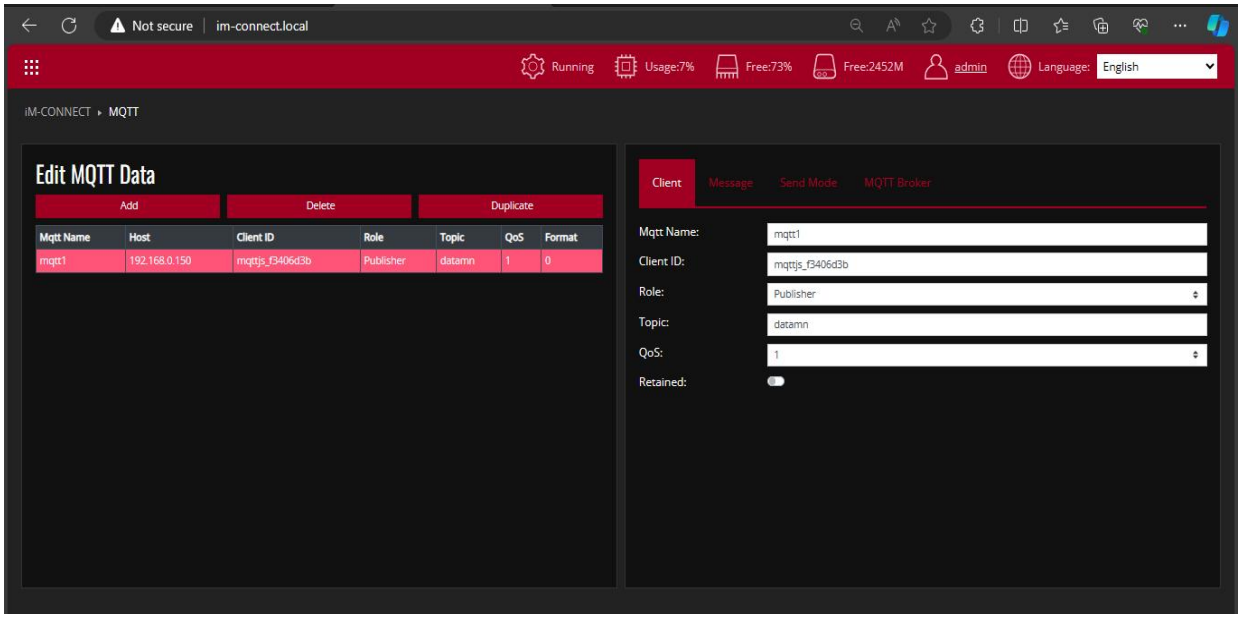
After enter data (save and compile) then (start)



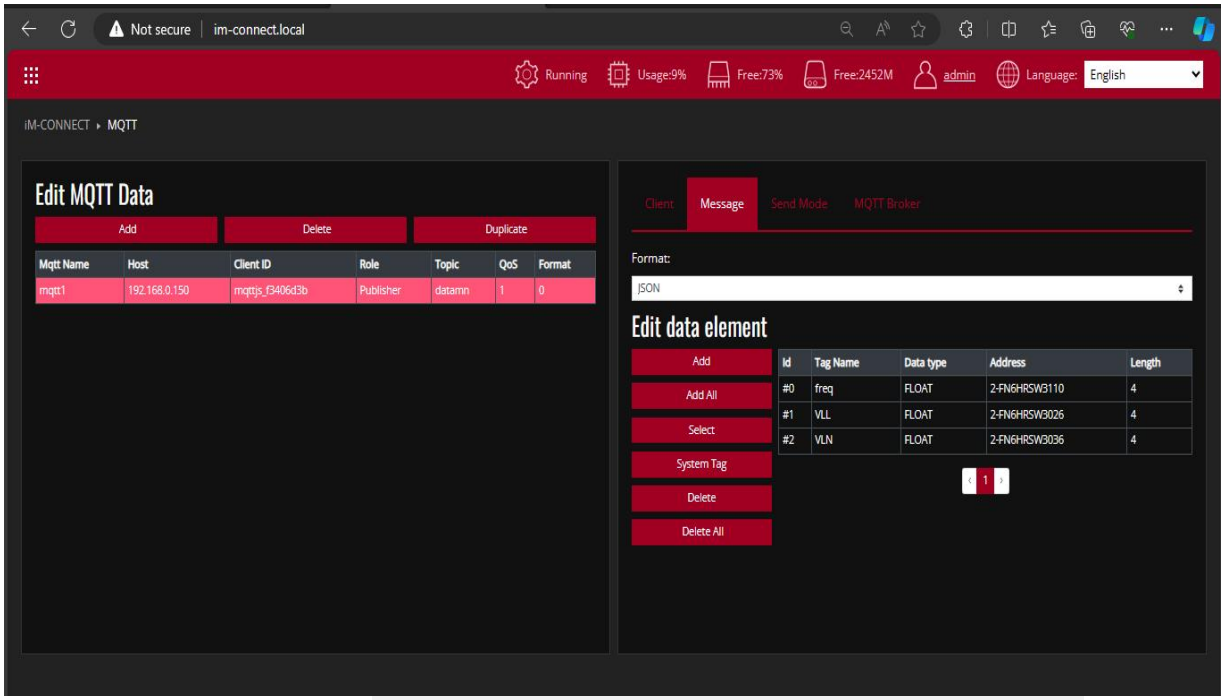
1-Go to online monitor to test the connection ..

2-If it good then go to mqqt

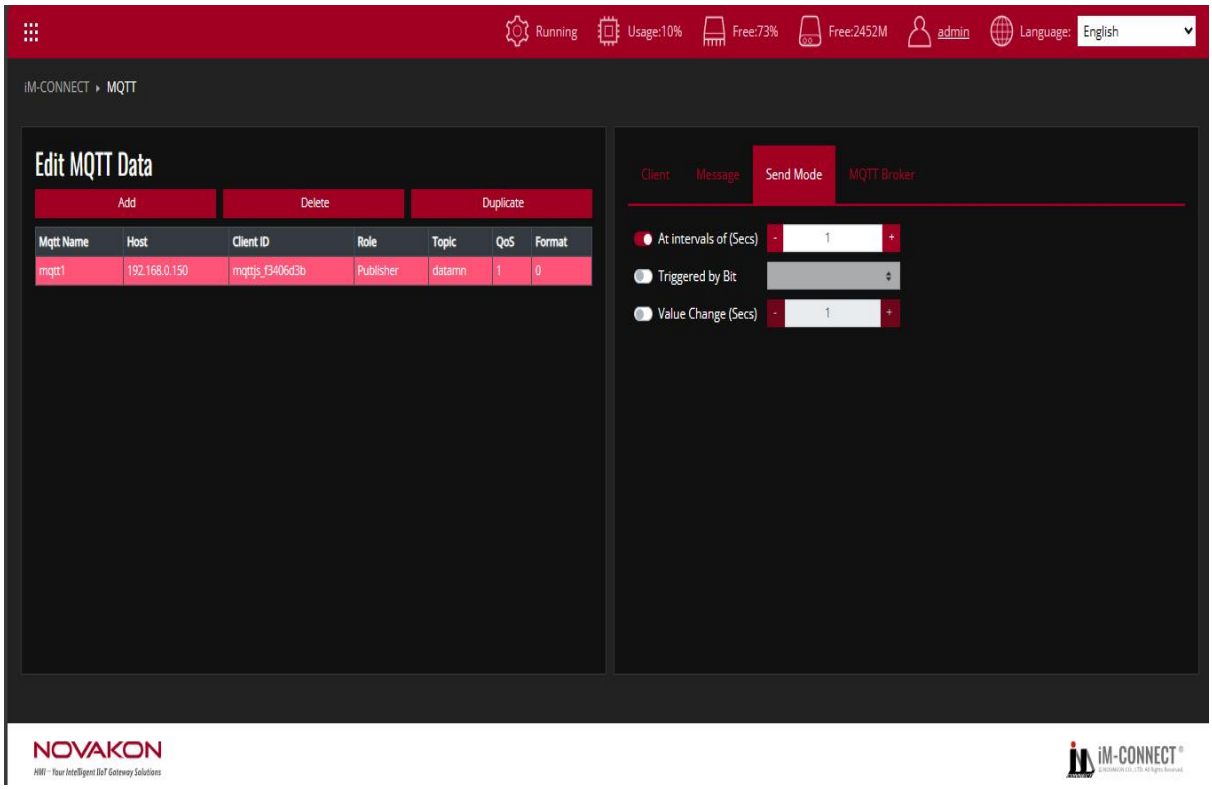


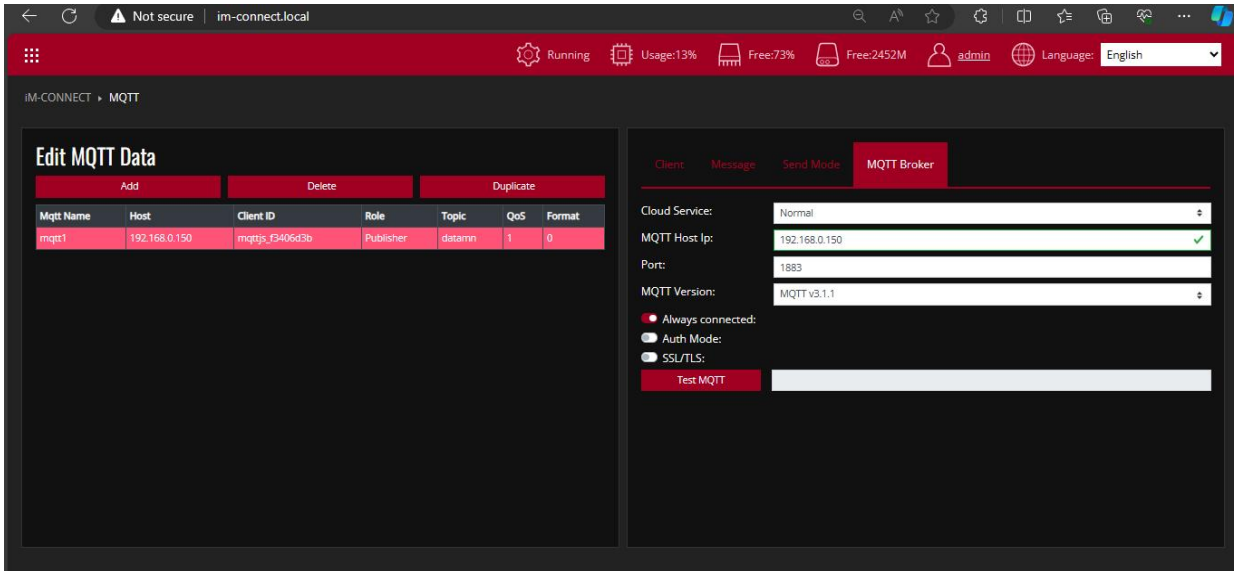


- client id is auto generated
- topic should be unique to avoid overlap

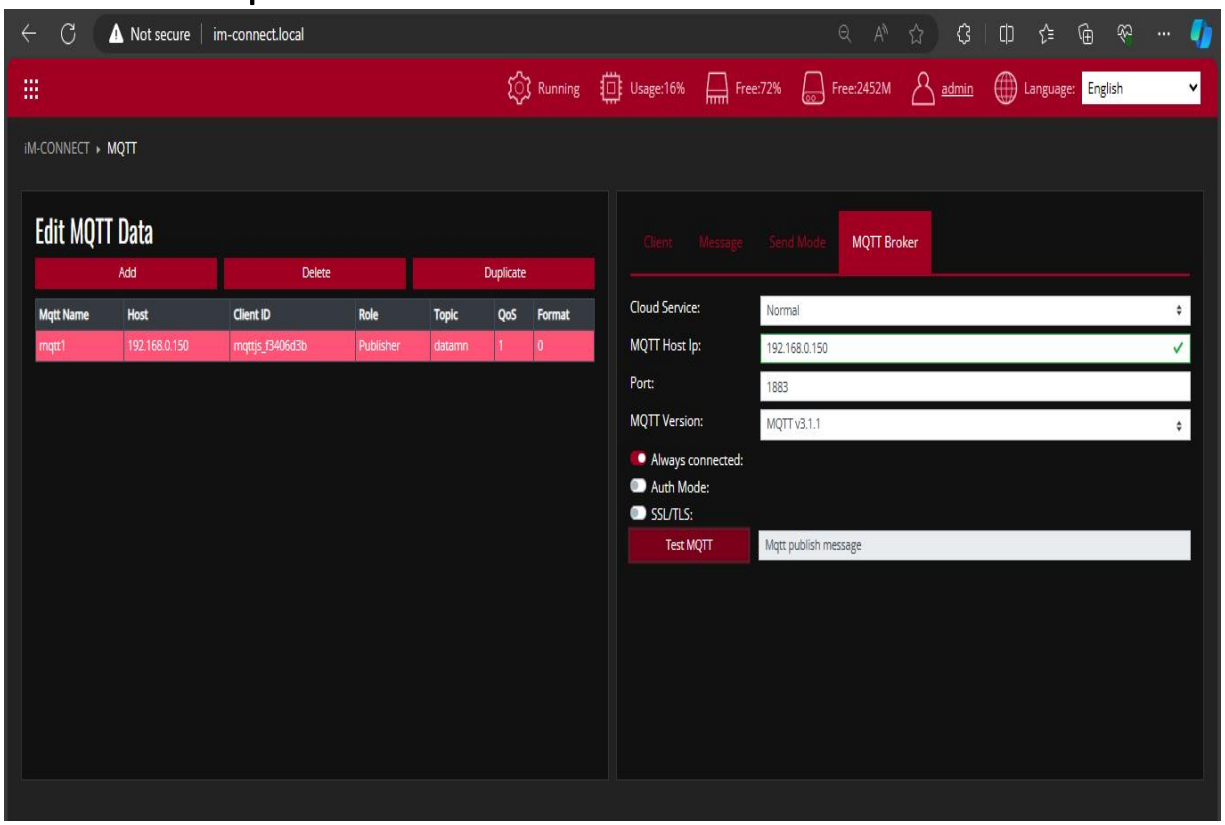


-format (json& customized) json: is a lightweight format for storing and transporting data also used when data is sent from a server to a web page
and easy to understand.
Cusomized : designed for specific application and send specific massage

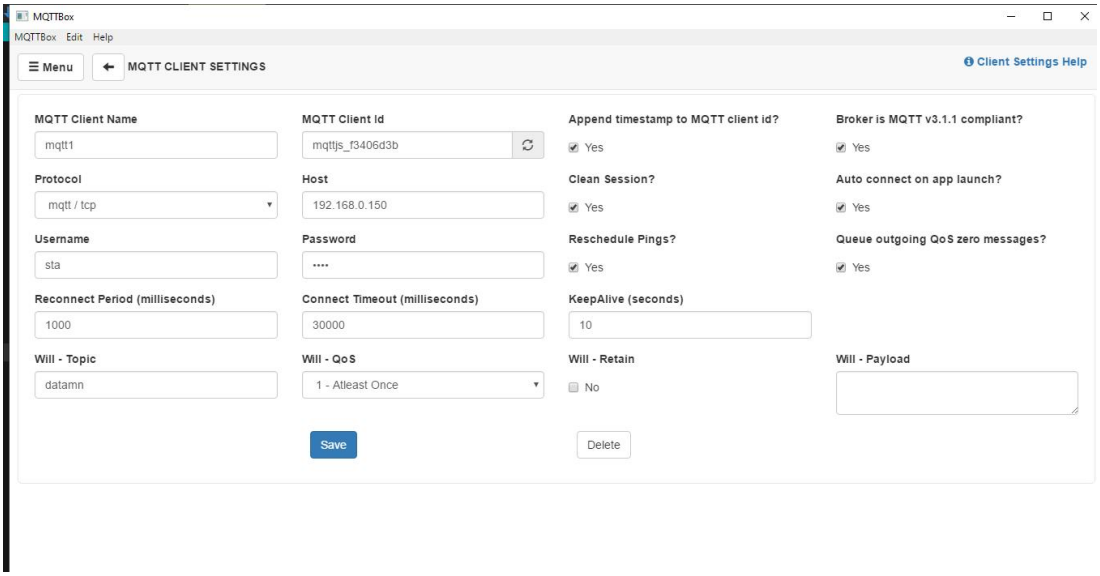




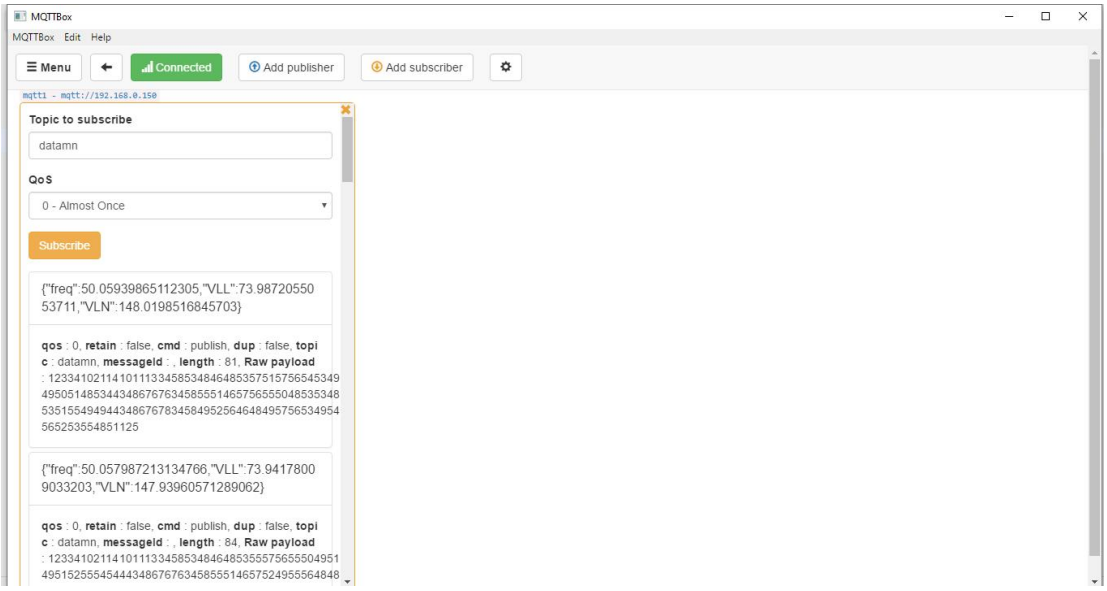
- Enter the ip address of pc
- Enter the port 1883



Open MQTTbox and fill data (host&topic&client id&username&password&qos)



Choose subscribe and



Then test mqqt and massage must be (Mqtt publish message) to confirm the connection

* if there is any probem follow this steps *

local mosquitto server setup:

1. download the local server app from the provided link before
2. install the app
3. after instillation check the mosquitto services is run correctly
4. the broker ip is your device ip (wifi - ethernet)

note: if you test broker and other devices not connected to it add these two lines in **mosquitto.conf** file in setup file of mosquitto app in c drive

listener 1883

allow_anonymous true

and then save the file

5-Check service and make sure that mosquitto is running

6-Check firewall setting by Open Windows Defender Firewall Settings:

*Press Win + S to open the Windows search bar.
*Type "Windows Defender Firewall" and select the corresponding result.
*Choose "Advanced Settings" (Windows Firewall with Advanced Security):
On the left-hand side, you will see "Advanced settings." Click on it.
*Create a New Inbound Rule:
*In the left-hand pane, select "Inbound Rules."
*In the right-hand pane, click on "New Rule..." to open the New Inbound Rule Wizard.
*Select "Port" and Click "Next":
*Choose "Port" and click "Next."
*Specify TCP or UDP and the Port Number:
*Choose either "TCP" or "UDP," depending on the protocol used by your MQTT broker (most likely TCP).
*Select "Specific local ports" and enter the MQTT broker port number (e.g., 1883 for MQTT/TCP).
*Click "Next."
*Choose "Allow the Connection" and Click "Next":
*
*Select "Allow the connection" and click "Next."
*Specify When to Apply the Rule and Click "Next":

*Choose when to apply the rule. Generally, keeping all options selected is fine.
*Click "Next."
*Name the Rule and Add a Description (Optional):

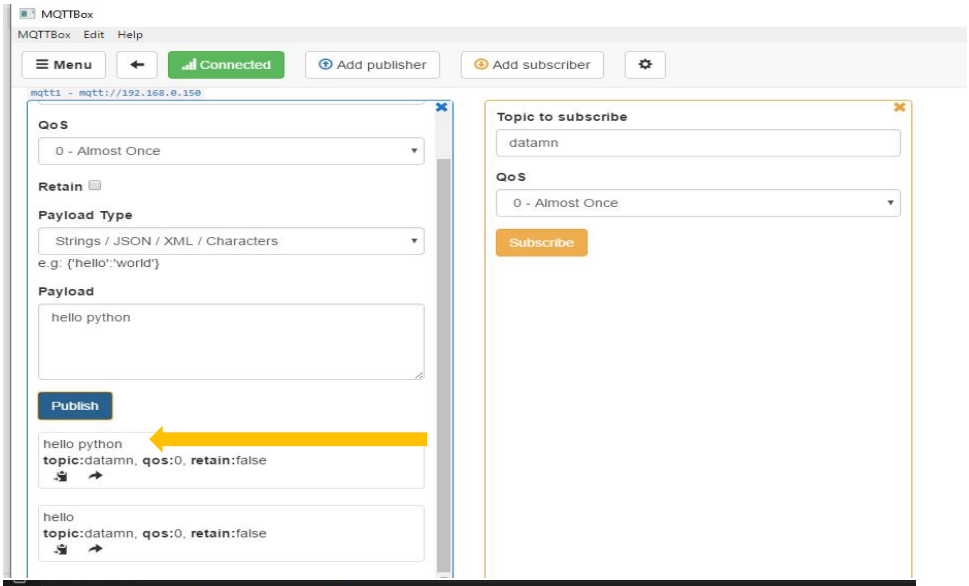
*Provide a name and, if desired, a description for the rule.
*Click "Finish."

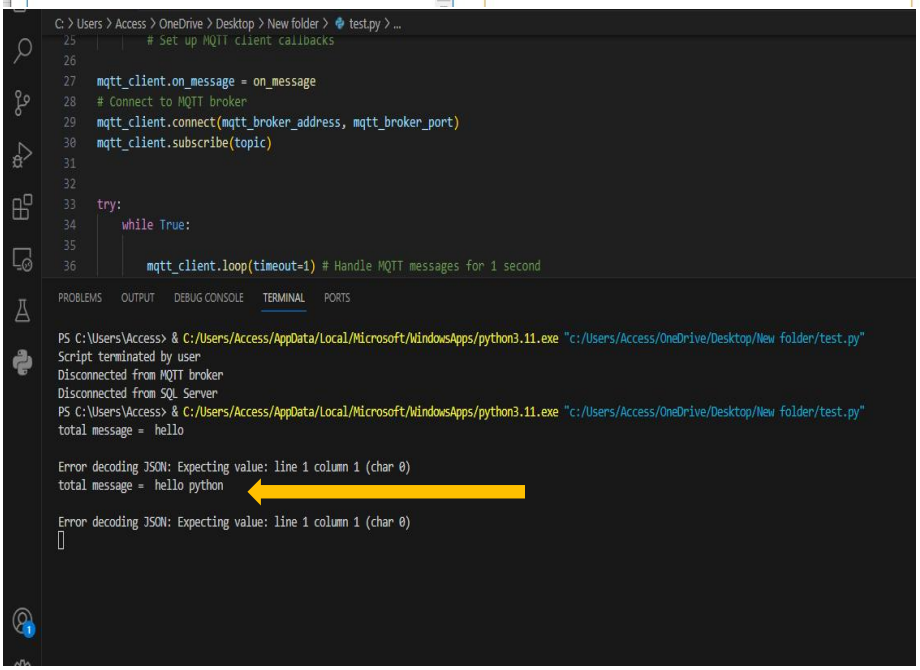
Connect with python

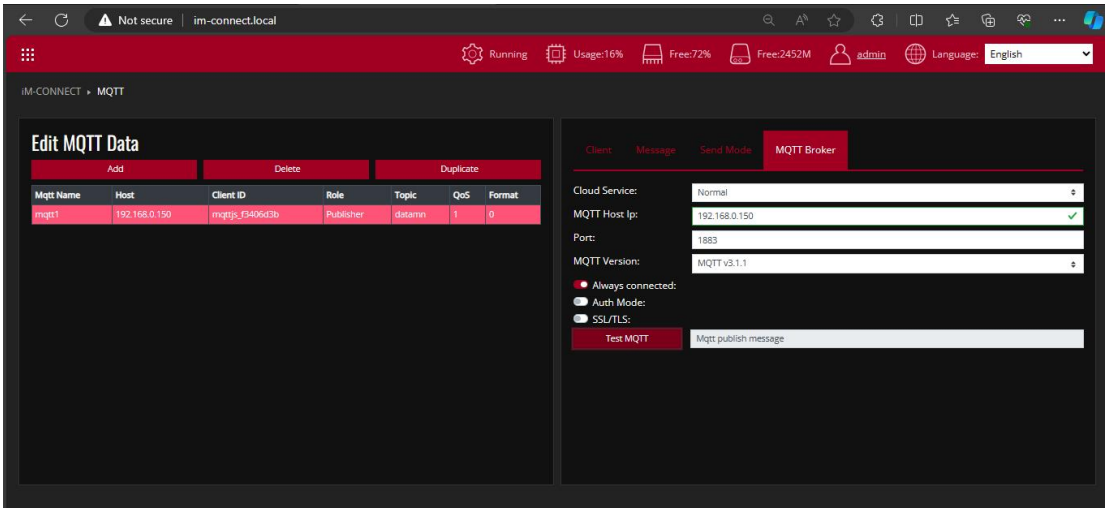
C: > Users > Access > OneDrive > Desktop > New folder > test.py > ...

```python
import paho.mqtt.client as mqtt
import json

# MQTT broker connection details
mqtt_broker_address = "192.168.0.150"
mqtt_broker_port = 1883
topic = "datamn" # Replace with the actual MQTT topic


# Connect to MQTT broker
mqtt_client = mqtt.Client()


def on_message(client, userdata, message):
    try:
        # Decode the JSON message
        valuet = message.payload.decode("utf-8")
        print("total message = ",valuet)

        payload = json.loads(valuet)
        print("payload = ",payload)

    except json.JSONDecodeError as e:
        print(f"Error decoding JSON: {e}")
        # Set up MQTT client callbacks

mqtt_client.on_message = on_message
# Connect to MQTT broker
mqtt_client.connect(mqtt_broker_address, mqtt_broker_port)
mqtt_client.subscribe(topic)


try:
    while True:

        mqtt_client.loop(timeout=1) # Handle MQTT messages for 1 second

except KeyboardInterrupt:
    print("Script terminated by user")
finally:
    # Disconnect from MQTT broker
    mqtt_client.disconnect()
    print("Disconnected from MQTT broker")
    # Disconnect from SQL Server
    print("Disconnected from SQL Server")
```

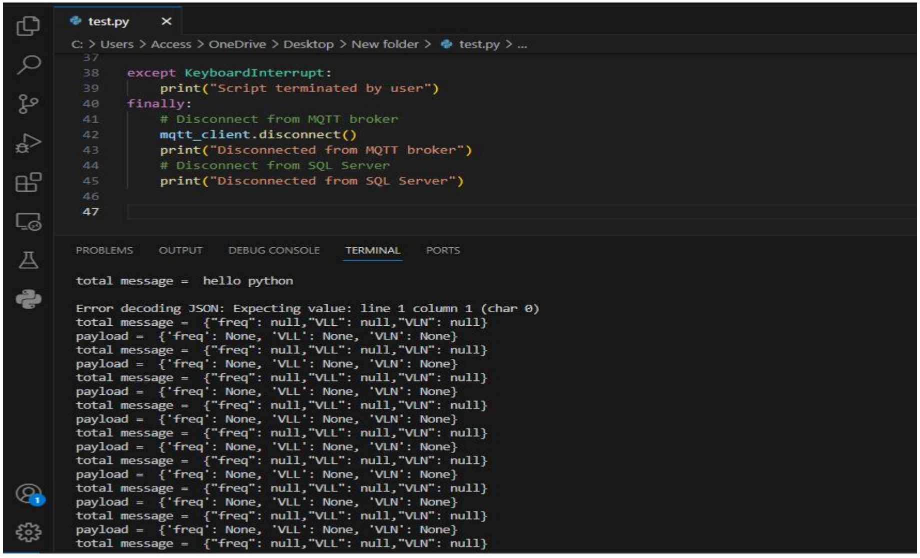
test from mqttbox


MQTTBox
MQTTBox Edit Help
Menu
Connected
Add publisher
Add subscriber
mqtt1 - mqtt://192.168.0.150
QoS
0 - Almost Once
Retain
Payload Type
Strings / JSON / XML / Characters
e.g: {'hello':'world'}
Payload
hello python
Publish
hello python
topic:datamn, qos:0, retain:false
hello
topic:datamn, qos:0, retain:false
Topic to subscribe
datamn
QoS
0 - Almost Once
Subscribe


C: > Users > Access > OneDrive > Desktop > New folder > test.py > ...
# Set up MQTT client callbacks
mqtt_client.on_message = on_message
# Connect to MQTT broker
mqtt_client.connect(mqtt_broker_address, mqtt_broker_port)
mqtt_client.subscribe(topic)
try:
while True:
mqtt_client.loop(timeout=1) # Handle MQTT messages for 1 second
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Access> & C:/Users/Access/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Access/OneDrive/Desktop/New folder/test.py"
Script terminated by user
Disconnected from MQTT broker
Disconnected from SQL Server
PS C:\Users\Access> & C:/Users/Access/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Access/OneDrive/Desktop/New folder/test.py"
total message = hello
Error decoding JSON: Expecting value: line 1 column 1 (char 0)
total message = hello python
Error decoding JSON: Expecting value: line 1 column 1 (char 0)


Not secure | im-connect.local
Running
Usage:16%
Free:72%
Free:2452M
admin
Language:
English
iM-CONNECT ▸ MQTT
Edit MQTT Data
Add
Delete
Duplicate
Mqtt Name
Host
Client ID
Role
Topic
QoS
Format
mqtt1
192.168.0.150
mqttjs_f3406d3b
Publisher
datamn
1
0
Client
Message
Send Mode
MQTT Broker
Cloud Service:
Normal
MQTT Host Ip:
192.168.0.150
Port:
1883
MQTT Version:
MQTT v3.1.1
Always connected:
Auth Mode:
SSL/TLS:
Test MQTT
Mqtt publish message

The value of the parameter (Freq , VLL , VLN) will received on python

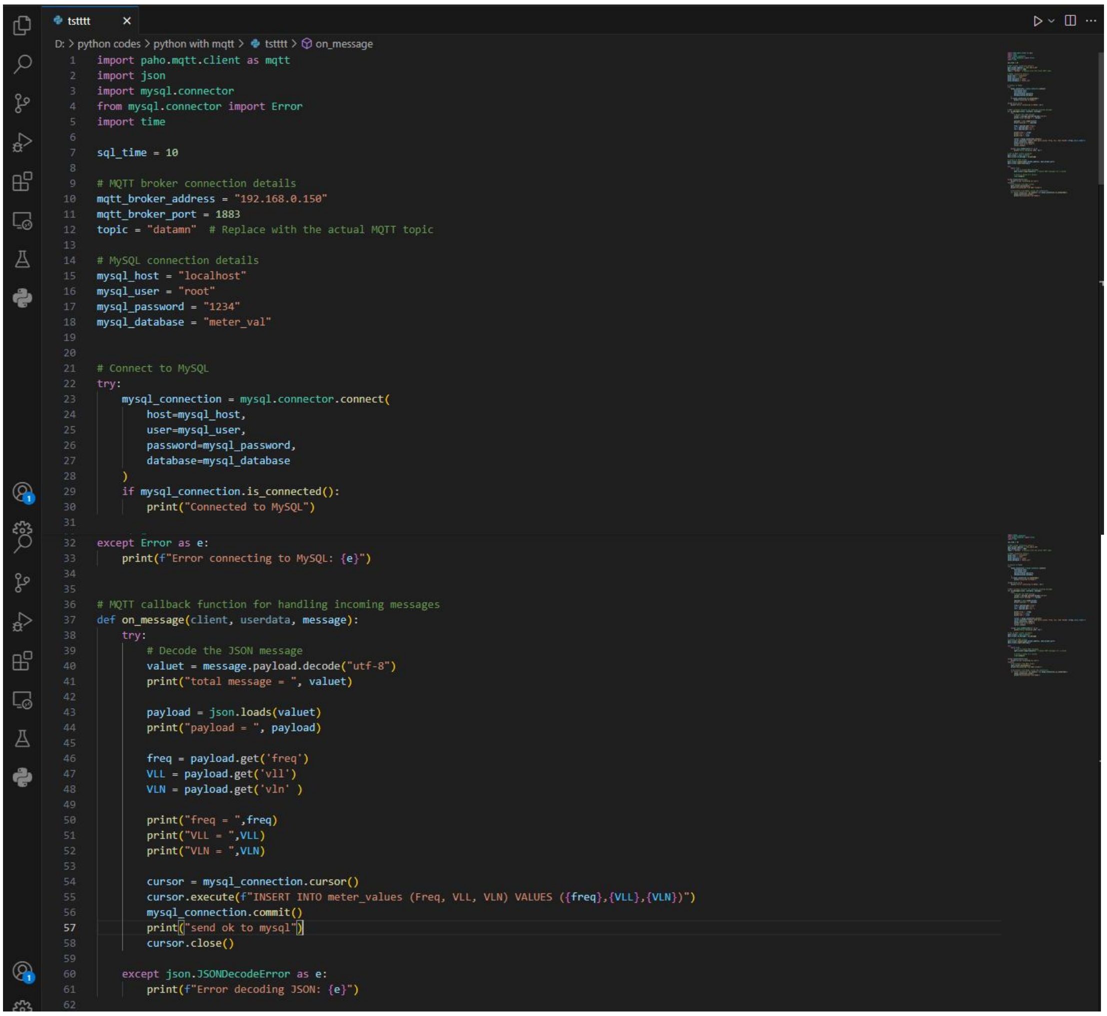


```python

except KeyboardInterrupt:
    print("Script terminated by user")
finally:
    # Disconnect from MQTT broker
    mqtt_client.disconnect()
    print("Disconnected from MQTT broker")
    # Disconnect from SQL Server
    print("Disconnected from SQL Server")


```

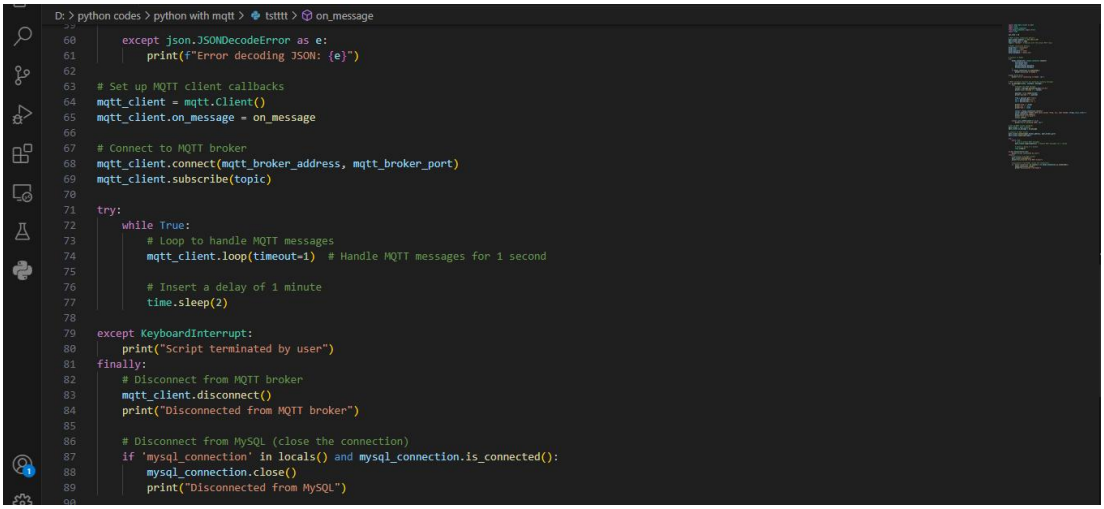
```
total message =  hello python

Error decoding JSON: Expecting value: line 1 column 1 (char 0)
total message =  {"freq": null,"VLL": null,"VLN": null}
payload =  {'freq': None, 'VLL': None, 'VLN': None}
total message =  {"freq": null,"VLL": null,"VLN": null}
payload =  {'freq': None, 'VLL': None, 'VLN': None}
total message =  {"freq": null,"VLL": null,"VLN": null}
payload =  {'freq': None, 'VLL': None, 'VLN': None}
total message =  {"freq": null,"VLL": null,"VLN": null}
payload =  {'freq': None, 'VLL': None, 'VLN': None}
total message =  {"freq": null,"VLL": null,"VLN": null}
payload =  {'freq': None, 'VLL': None, 'VLN': None}
total message =  {"freq": null,"VLL": null,"VLN": null}
payload =  {'freq': None, 'VLL': None, 'VLN': None}
total message =  {"freq": null,"VLL": null,"VLN": null}
payload =  {'freq': None, 'VLL': None, 'VLN': None}
total message =  {"freq": null,"VLL": null,"VLN": null}
payload =  {'freq': None, 'VLL': None, 'VLN': None}
total message =  {"freq": null,"VLL": null,"VLN": null}
```

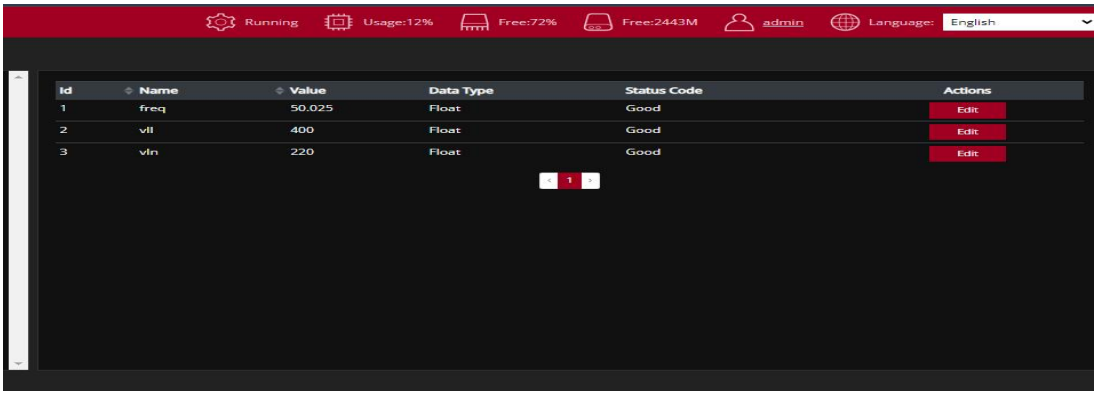
note the value is null because of desconnecting the meter

Connect with database(get data from gateway)



```python
import paho.mqtt.client as mqtt
import json
import mysql.connector
from mysql.connector import Error
import time

sql_time = 10

# MQTT broker connection details
mqtt_broker_address = "192.168.0.150"
mqtt_broker_port = 1883
topic = "datamn"  # Replace with the actual MQTT topic

# MySQL connection details
mysql_host = "localhost"
mysql_user = "root"
mysql_password = "1234"
mysql_database = "meter_val"


# Connect to MySQL
try:
    mysql_connection = mysql.connector.connect(
        host=mysql_host,
        user=mysql_user,
        password=mysql_password,
        database=mysql_database
    )
    if mysql_connection.is_connected():
        print("Connected to MySQL")

except Error as e:
    print(f"Error connecting to MySQL: {e}")


# MQTT callback function for handling incoming messages
def on_message(client, userdata, message):
    try:
        # Decode the JSON message
        valuet = message.payload.decode("utf-8")
        print("total message = ", valuet)

        payload = json.loads(valuet)
        print("payload = ", payload)

        freq = payload.get('freq')
        VLL = payload.get('vll')
        VLN = payload.get('vln' )

        print("freq = ",freq)
        print("VLL = ",VLL)
        print("VLN = ",VLN)

        cursor = mysql_connection.cursor()
        cursor.execute(f"INSERT INTO meter_values (Freq, VLL, VLN) VALUES ({freq},{VLL},{VLN})")
        mysql_connection.commit()
        print("send ok to mysql")
        cursor.close()

    except json.JSONDecodeError as e:
        print(f"Error decoding JSON: {e}")

```





Read from database (mysql)