

# MQTT-Subscriber

## MQTT-Subscriber

[Routine explanation](#)

[Introduction](#)

[Code Explanation](#)

## Routine explanation

### Introduction

In this section, we will use K230 as an MQTT subscriber.

We open the routine code and modify the WIFI information

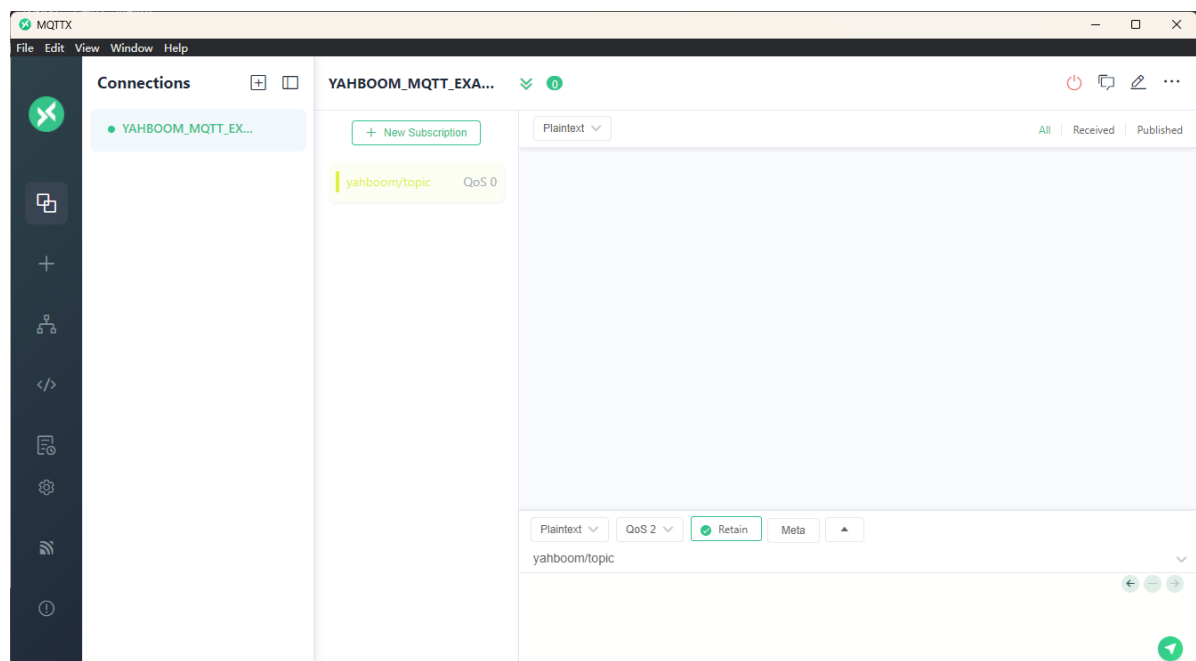
```
# WiFi配置参数 / WiFi configuration parameters
WIFI_SSID = "WIFI SSID"          # WiFi名称 / WiFi name
WIFI_PASSWORD = "WIFI ASSWORD" # WiFi密码 / WiFi password
```

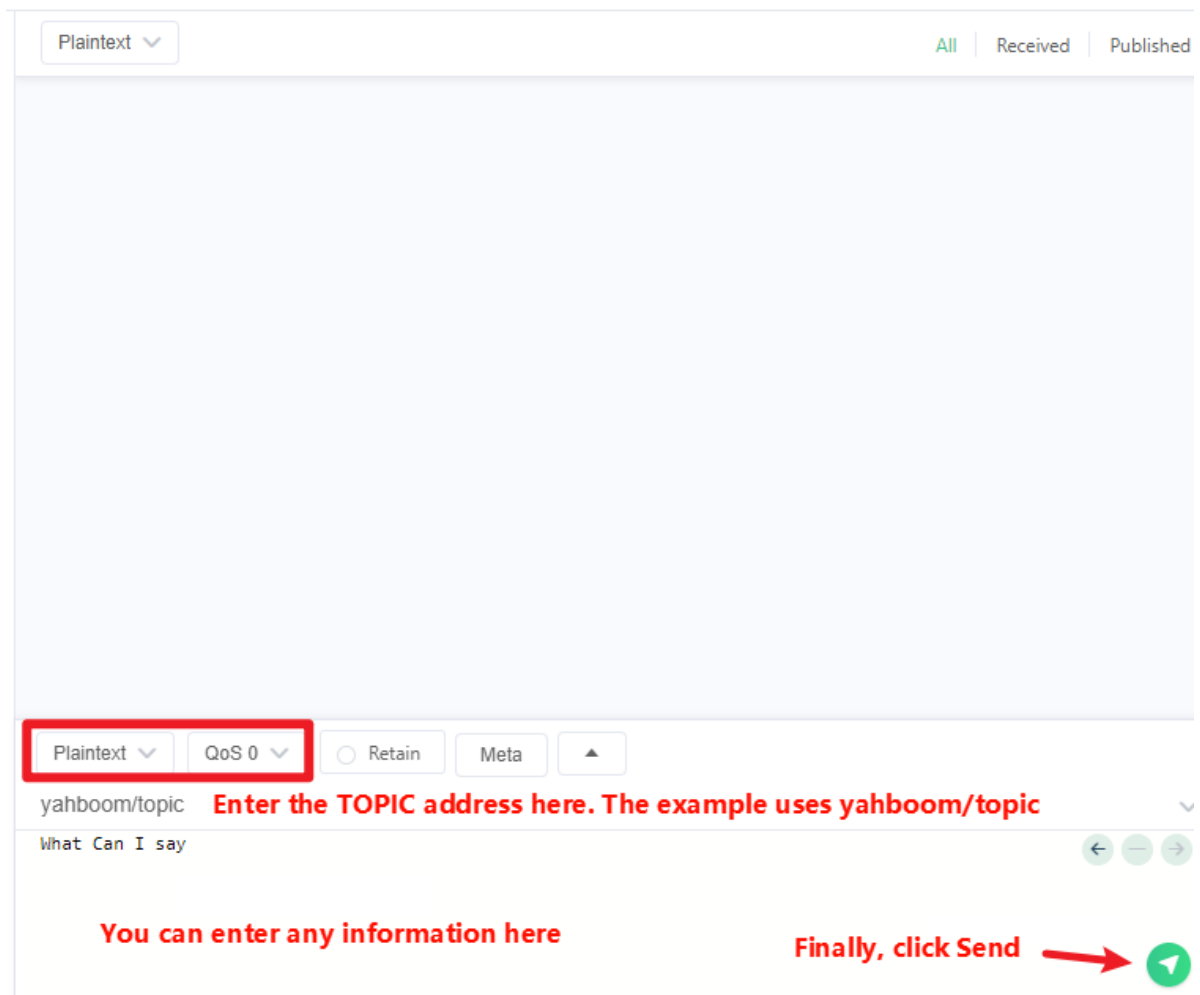
Click the Run button in the lower left corner to start K230. Wait for WIFI connection

Then we need to open MQTTX. Please refer to the previous section MQTT-Publisher for the installation and Chinese tutorial of the software.

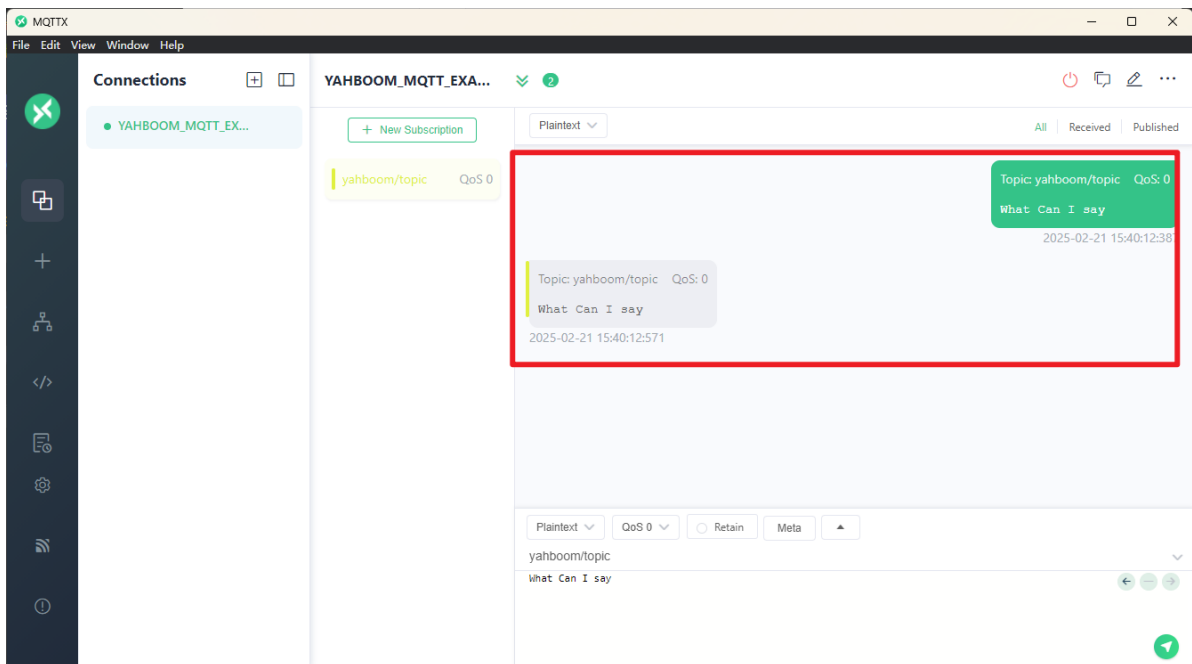
We follow the steps in the previous section to connect to the public broker and create a subscription for yahboom/topic

If you are coming from the previous section and MQTTX is not turned off, then no additional operations are required because the steps here are exactly the same as in the previous section.





We can see that the content we sent appears on MQTTX side

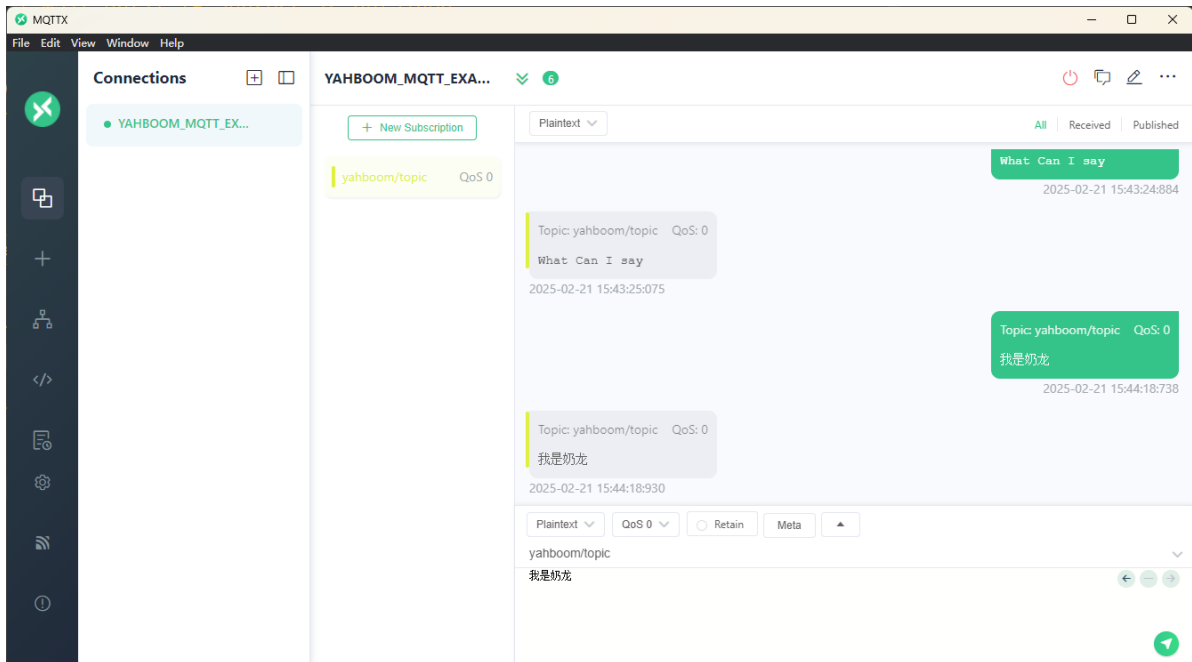


Similarly, the serial terminal of K230 also outputs this content

```
Topic: b'yahboom/topic', Message: b'What Can I say'
```

You can see that the message output here starts with b. In this case, Chinese characters will appear garbled.

So we can modify the msg in the routine code to msg.decode(), so that Chinese can be received normally.



```
连接WIFI: Yahboom.. Connecting..  
WIFI连接成功 WIFI Connected!  
subscribe success!  
Topic: b'yahboom/topic', Message: 我是奶龙
```

## Code Explanation

For complete comments and code, please refer to the file [Source Code/ 11.Network/ 06.mqtt/ mqtt\_subscribe.py]

1. Module import part:

```
from mqtt import MQTTClient # 导入MQTT客户端模块 / Import MQTT client module  
import network # 导入网络模块, 用于WiFi连接 / Import network module for WiFi  
connection  
import os # 导入操作系统模块 / Import operating system module  
import machine # 导入硬件控制模块 / Import hardware control module  
import time # 导入时间模块, 用于延时操作 / Import time module for delays
```

This part imports the core modules required to implement the MQTT client functions, including MQTT communication, network connection, system operation, hardware control and time control functions.

1. Configuration parameters: The code defines two important sets of configuration parameters:

- WiFi connection parameters: WIFI\_SSID and WIFI\_PASSWORD
- MQTT server parameters: Use the public MQTT server broker.emqx.io, port 1883, and the custom topic "yahboom/topic"

1. WiFi connection function:

```
def connect_wifi(ssid, password):  
    """  
    连接WiFi并返回IP地址  
    Connect to WiFi and return IP address  
  
    参数 / Parameters:  
    ssid: WiFi名称 / WiFi name  
    password: WiFi密码 / WiFi password  
  
    返回 / Returns:  
    str: IP地址 / IP address  
    """
```

This function implements the WiFi connection function:

- Create a WiFi site object
- Connect to WiFi using the provided credentials
- Wait until a valid IP address is obtained by polling
- Returns the obtained IP address

1. MQTT message callback function:

```
def on_message(topic, msg):  
    print("Topic: {}, Message: {}".format(topic, msg.decode()))
```

This is a simple message processing callback function that will be called when an MQTT message is received. It will print out the subject and content of the message.

1. Main program flow:

- First connect to the WiFi network
- Create an MQTT client instance and use "YAHBOOM-K230" as the client ID
- Connecting to an MQTT server
- Set the message callback function
- Subscribe to the specified MQTT topic
- Enter an infinite loop, continuously checking for new messages
- Disconnect the MQTT connection when the program ends (in fact, this line of code will never be executed because it is an infinite loop)

