Date 5/5/2023

CSM3313

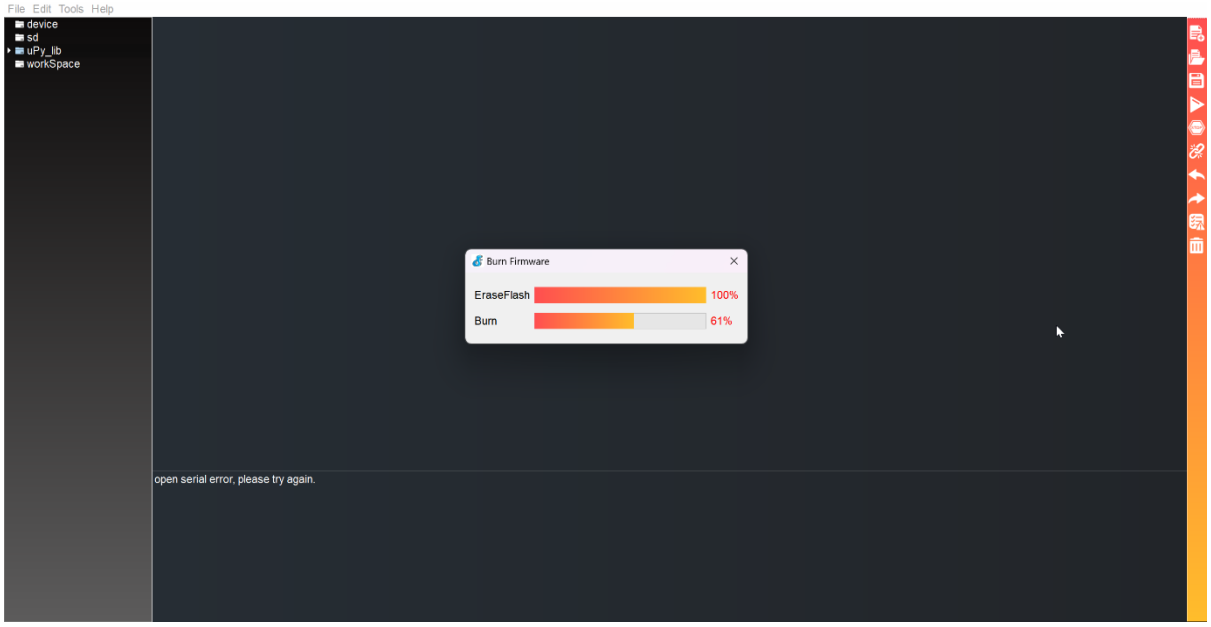
IOT COMPUTING

DR AHMAD SHUKRI BIN MOHD NOOR

OMAR ISMAIL ABDJALEEL ALOMORY

LAB 3(MP2)

LAB 1 GETTING STARTED WITH MICROPYTHON:



Installation of micropython, driver, and bin file are completed.

LAB 2 BASIC MICROPYTHON PROGRAMMING:

Arithmetic operation:

```
>>> 3+5
8
>>> 6-5
1
>>> 8*9
72
>>> 20/10
2.0
```

Boolean expression(logical operation):

```
>>> 2==5
False
>>> 4==4
True
>>> 69874 !=65
True
>>> 3>2
True
>>>
```

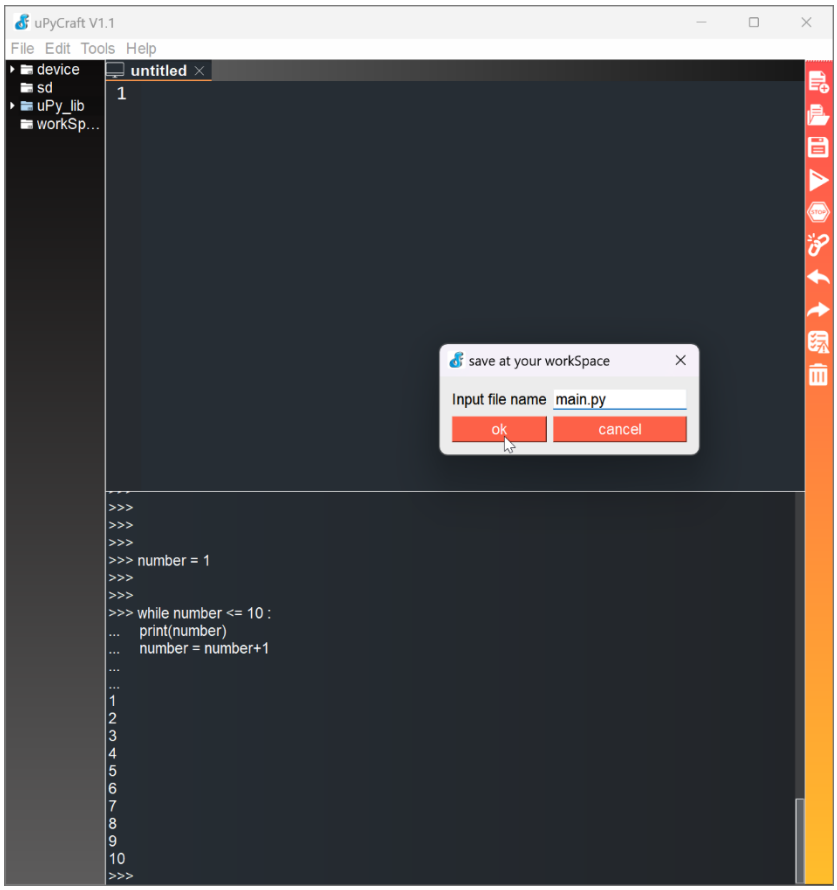
Assigning values and exploring micropython datatypes:

```
>>>
>>> a = 10
>>> b = 12
>>> c = 20.6
>>> text = 'abcdef'
>>> d = True
>>>
>>>
>>> type(a)
<class 'int'>
>>> type(b)
<class 'int'>
>>> type(b)
<class 'int'>
>>> type(c)
<class 'float'>
>>> type(text)
<class 'str'>
>>> type(d)
<class 'bool'>
>>>
```

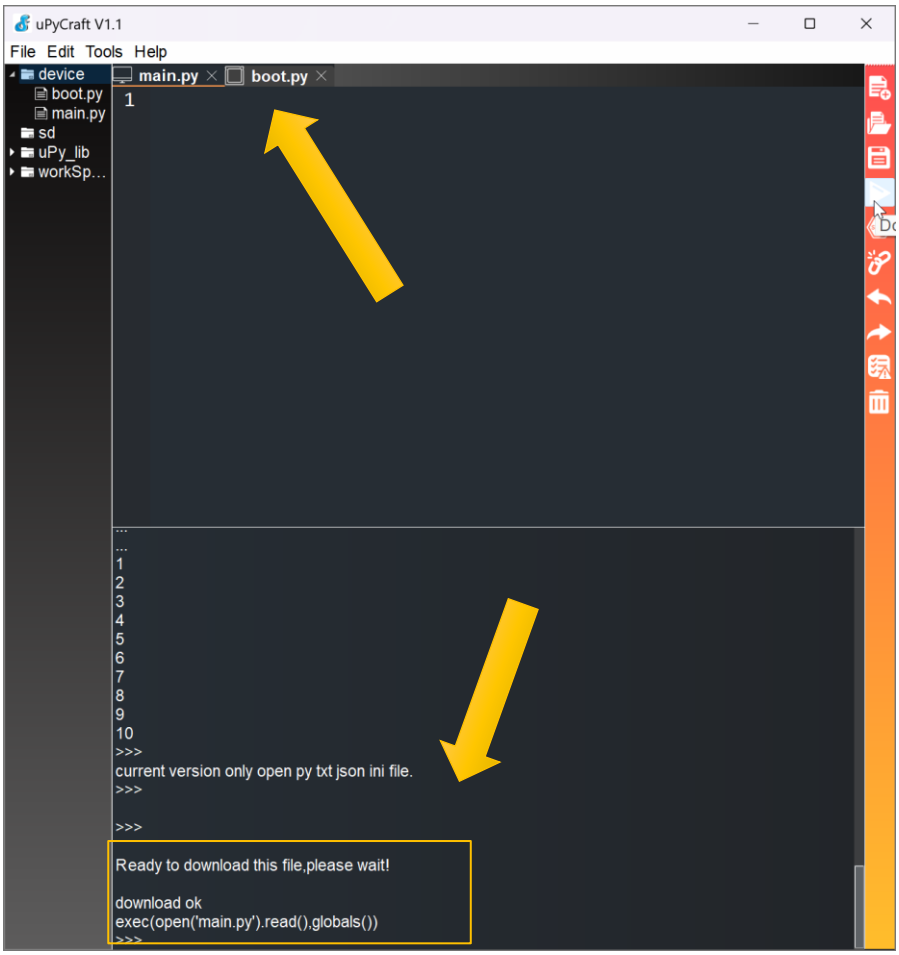
“**`while`**” loops in Micropython:

```
>>> number = 1
>>>
>>>
>>> while number <= 10 :
...     print(number)
...     number = number+1
...
...
1
2
3
4
5
6
7
8
9
10
>>>
```

Creating main.py in the board:



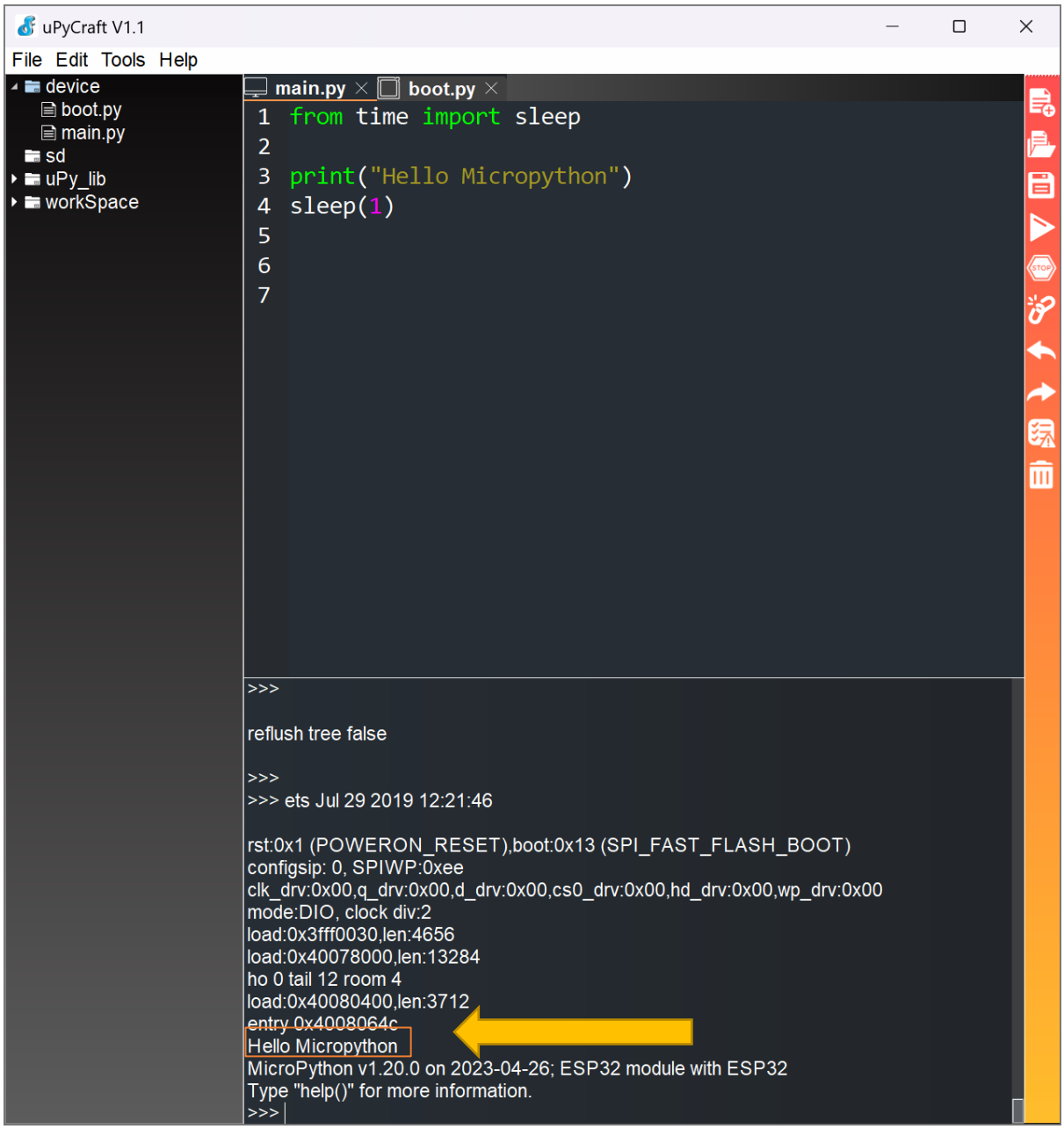
- After that, you should see the following in your uPyCraft IDE (the boot.py file in your device and a new tab with the main.py file)
- After clicking Download and run, the device directory should now load the main.py file. Your ESP has the file main.py stored.



After clicking the reset button (EN) in the ESP32 the output looks like this.

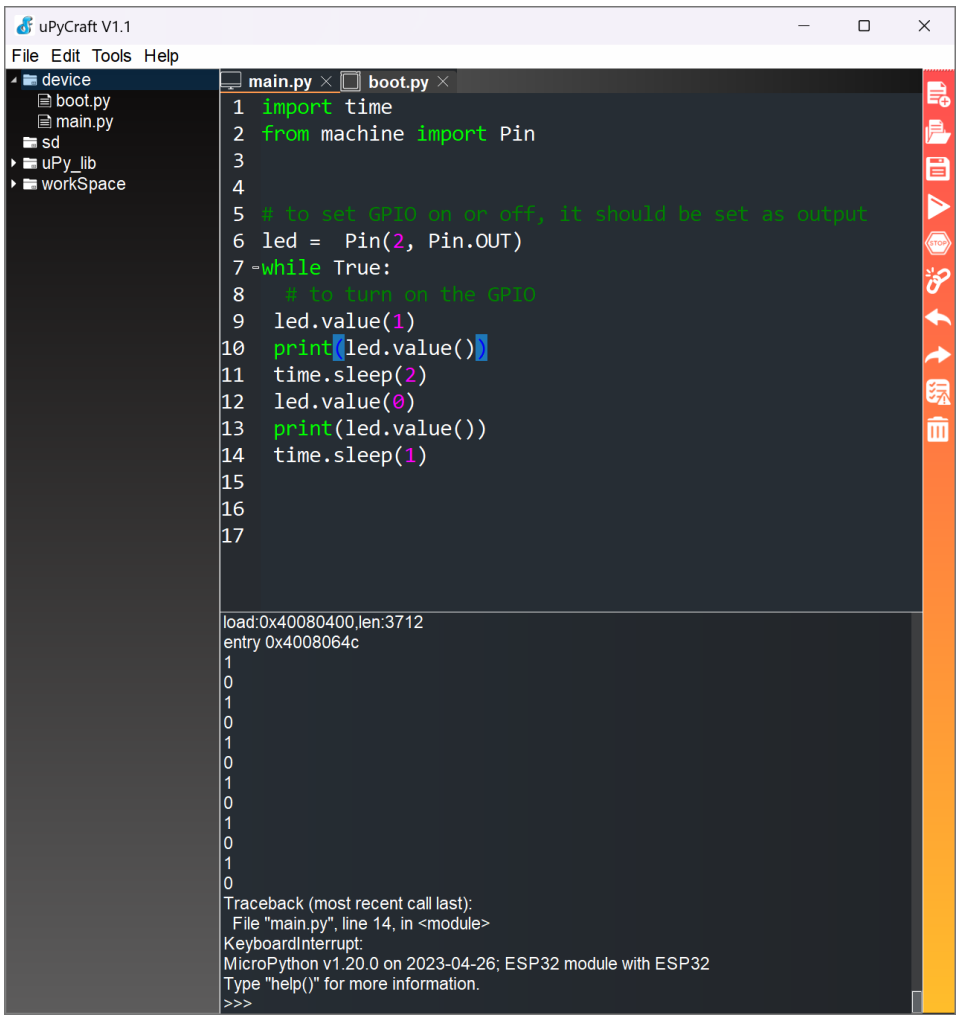
```
>>> ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:4540
ho 0 tail 12 room 4
load:0x40078000,len:12344
ho 0 tail 12 room 4
load:0x40080400,len:4124
entry 0x40080680
MicroPython v1.19.1 on 2022-06-18; ESP32 module with ESP32
Type "help()" for more information.
>>>
>>>
```

Print and sleep methods:

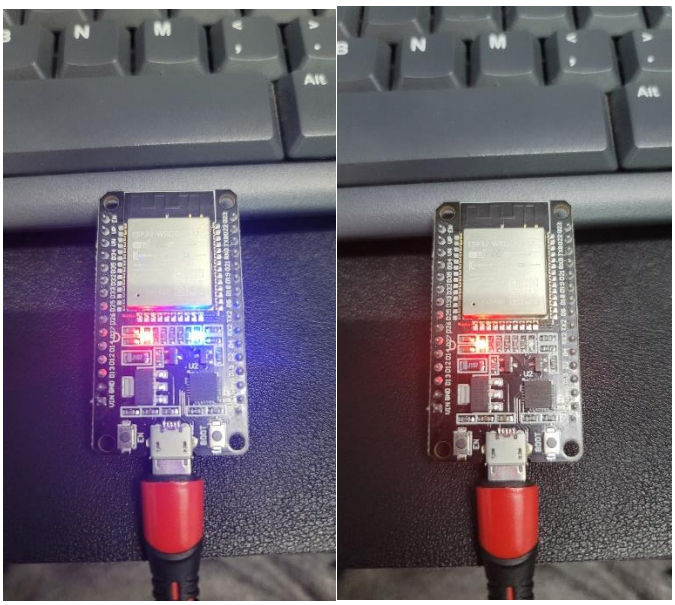


LAB 3 ESP32 PROGRAMMING:

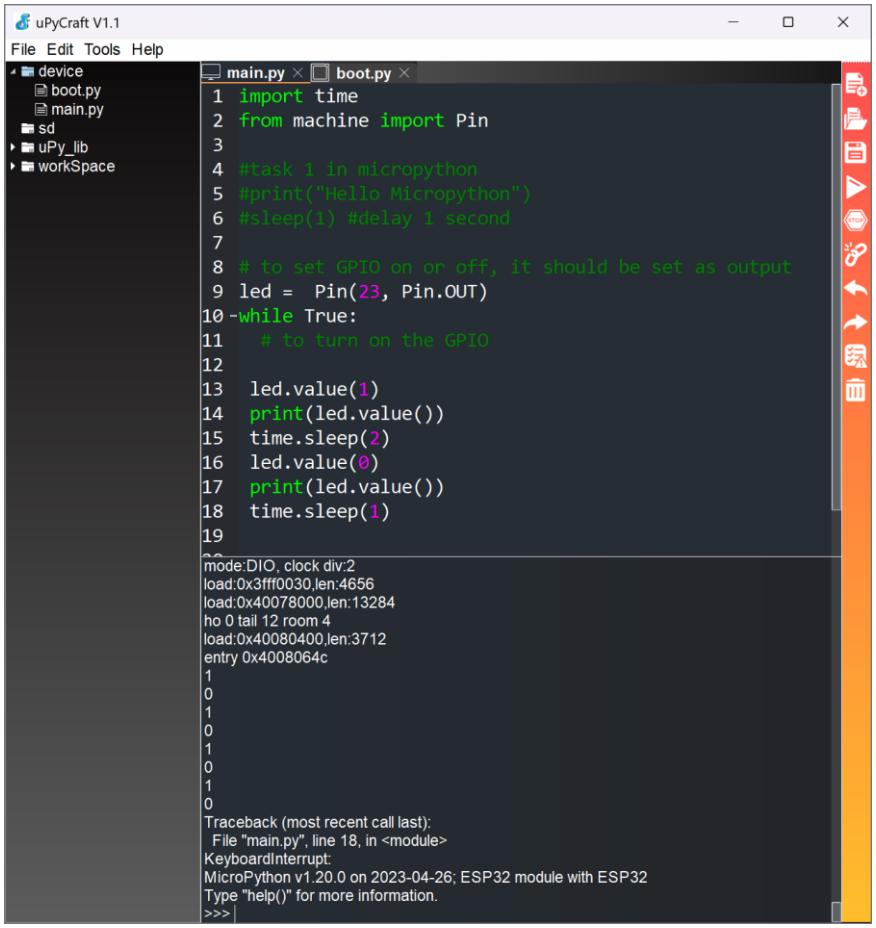
Turning on the LED in the ESP32 board(internal)



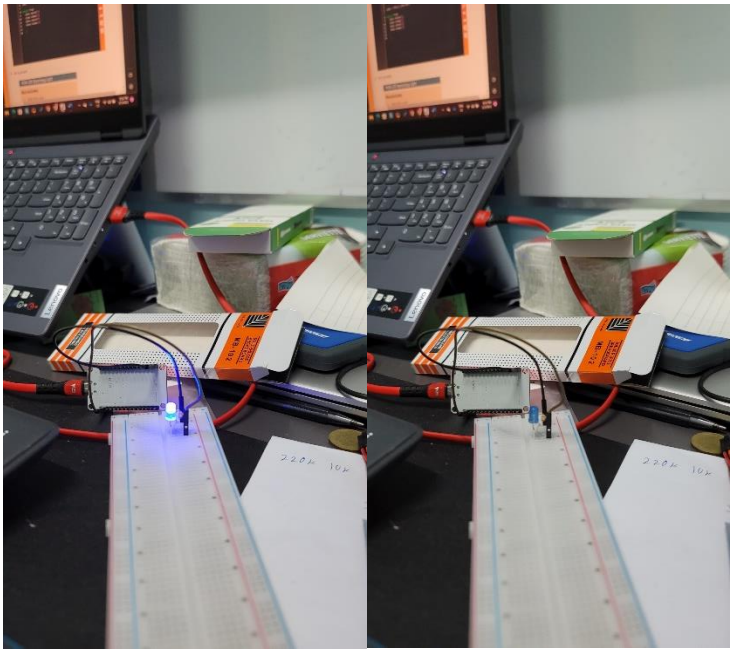
Output



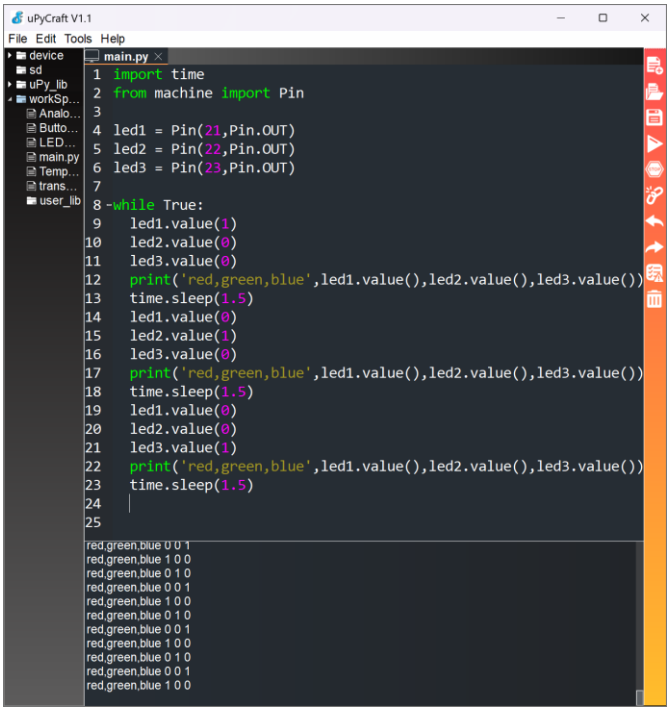
Blinking LED (external):



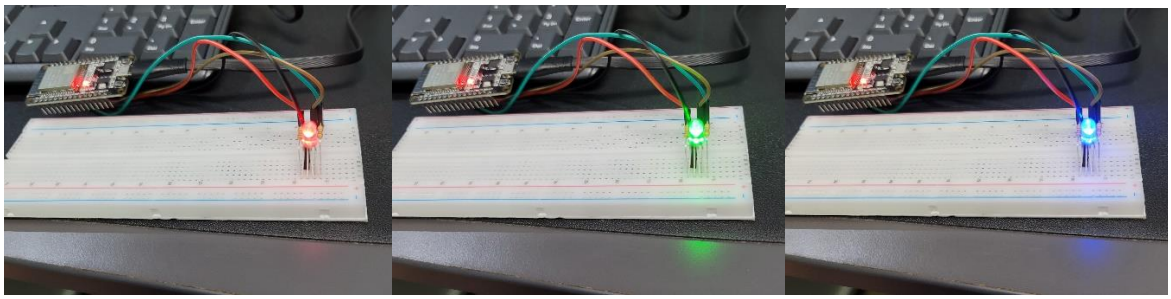
Output:



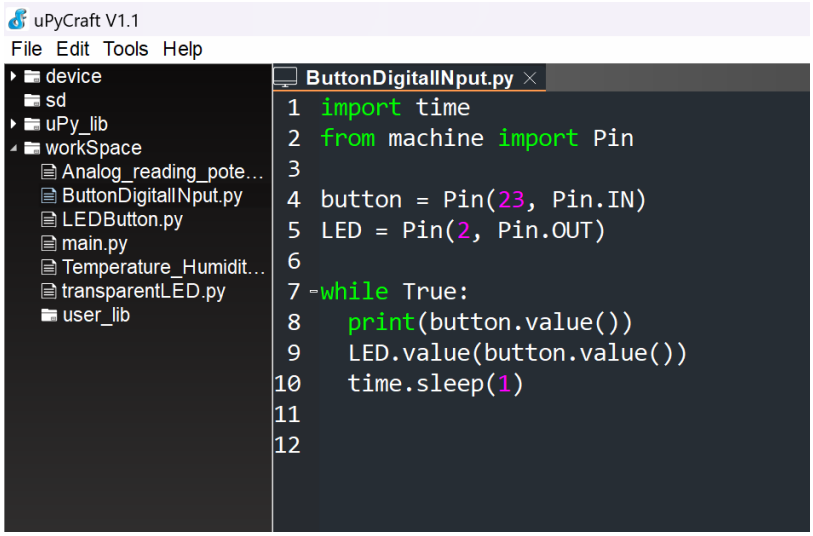
RGB LED:



Output

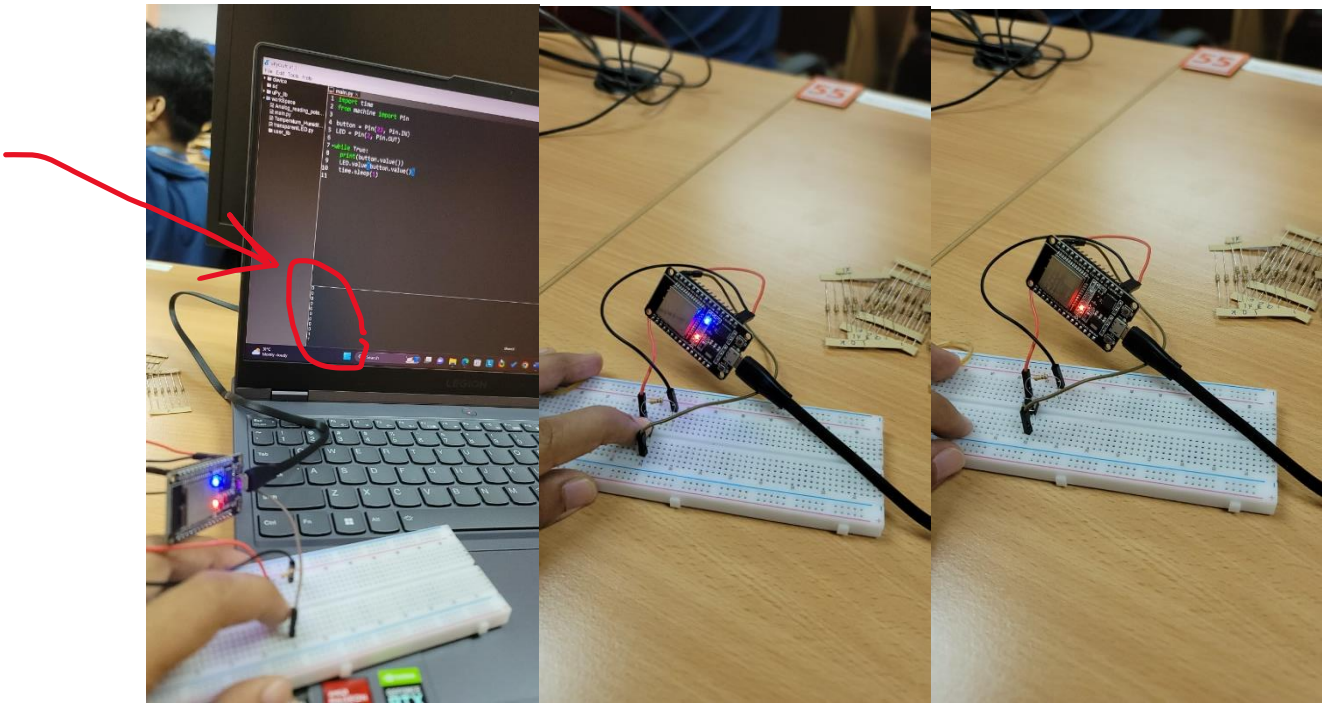


Button interaction with ESP32:

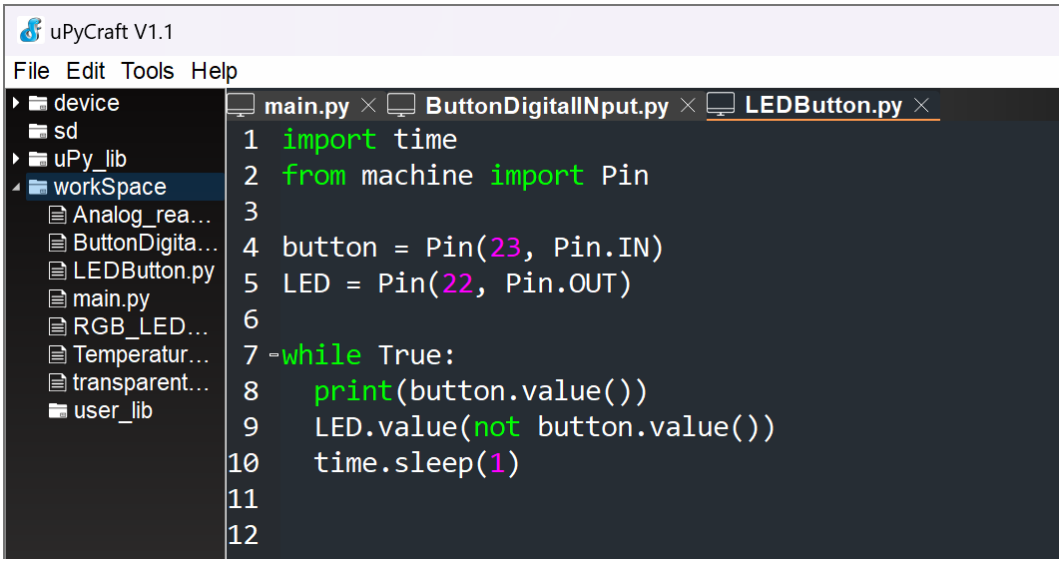


Output

When the button is clicked the value = 1, otherwise 0, plus if it 1 the internal LED will blink as it shown in the picture below.

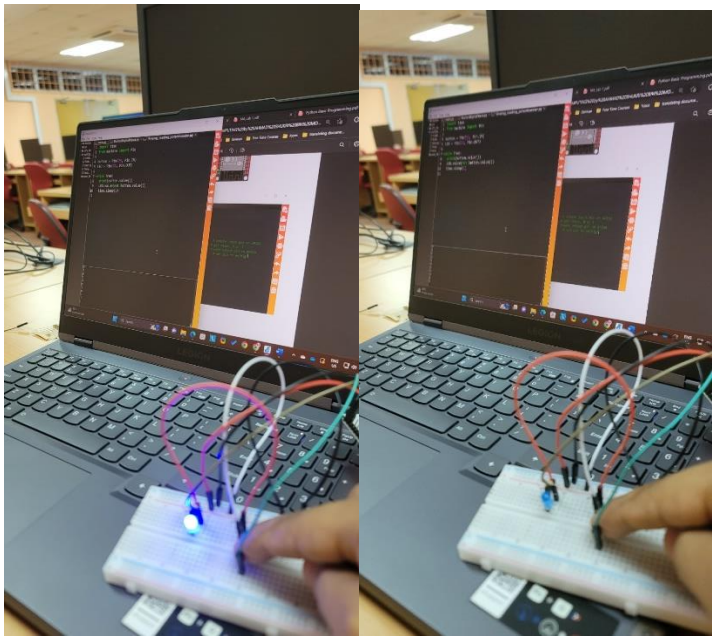


Button interaction to light LED in ESP32:



Output

When the button is clicked the LED any color will light up, otherwise it will be down.



Temperature and Humidity measurement:

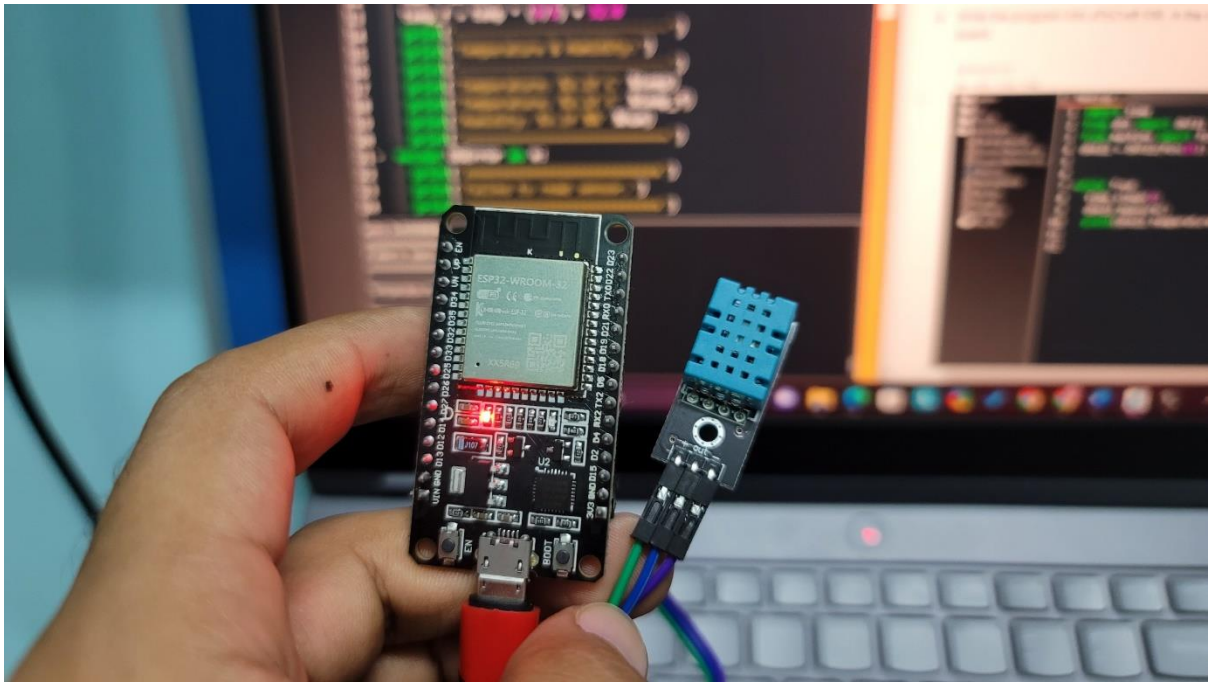
File Edit Tools Help

device
sd
uPy_lib
workSp...
main.py
user_lib

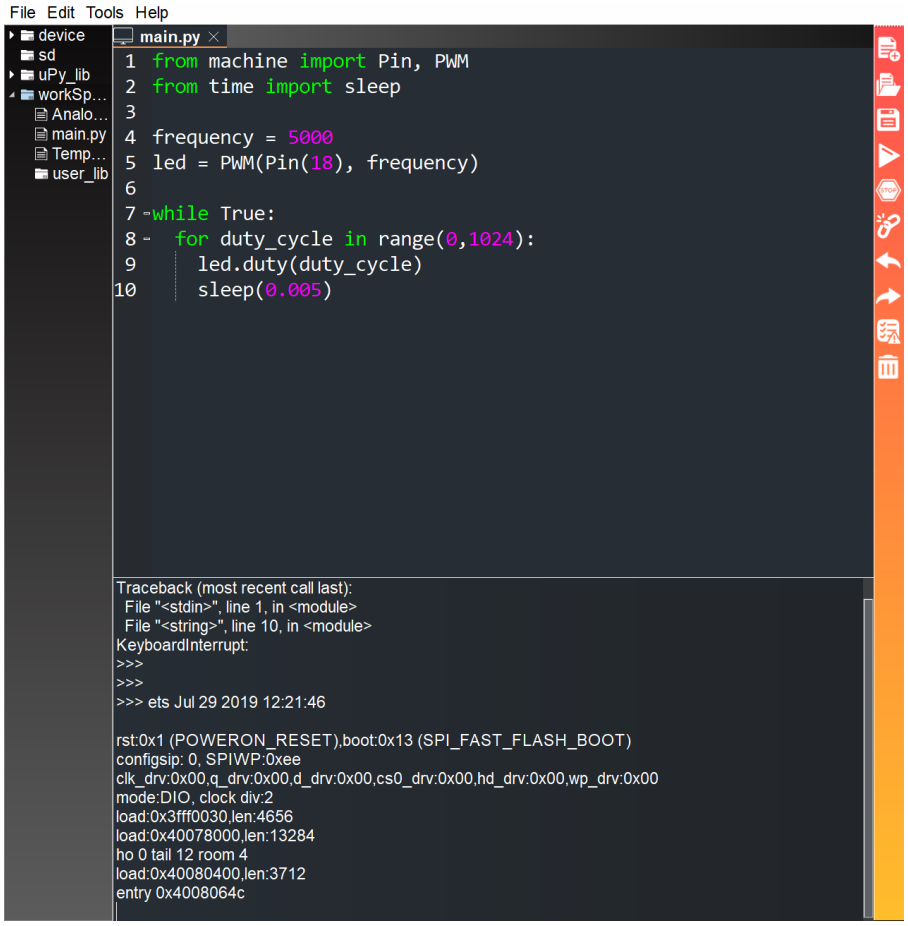
main.py

```python
from machine import Pin
from time import sleep
import dht

sensor = dht.DHT11(Pin(14))

while True:
  try:
    sleep(3)
    sensor.measure()
    temp = sensor.temperature()
    hum = sensor.humidity()
    temp_f = temp * (9/5) + 32.0
    print('---------------------------')
    print('Temperature & Humidity:')
    print('---------------------------')
    print('Temperature: %3.1f C' %temp)
    print('Temperature: %3.1f F' %temp_f)
    print('Humidity: %3.1f %%' %hum)
    print('===========================')
  except OSError as e:
    print('xxxxxxxxxxxxxxxxxxxxxxxxxxx')
    print('Failed to read sensor.')
    print('xxxxxxxxxxxxxxxxxxxxxxxxxxx')
```

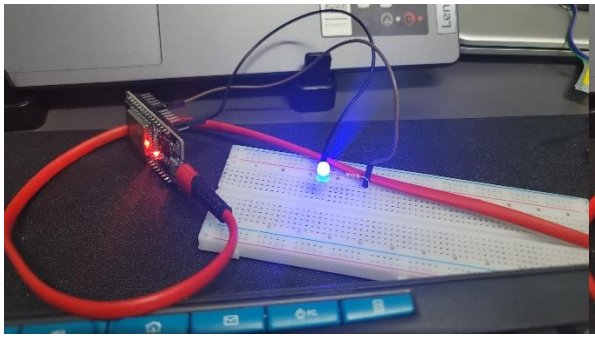
```
entry 0x4008064c
xxxxxxxxxxxxxxxxxxxxxxxxxxx
Failed to read sensor.
xxxxxxxxxxxxxxxxxxxxxxxxxxx
---------------------------
Temperature & Humidity:
---------------------------
Temperature: 31.0 C
Temperature: 87.8 F
Humidity: 75.0 %
===========================
---------------------------
Temperature & Humidity:
```

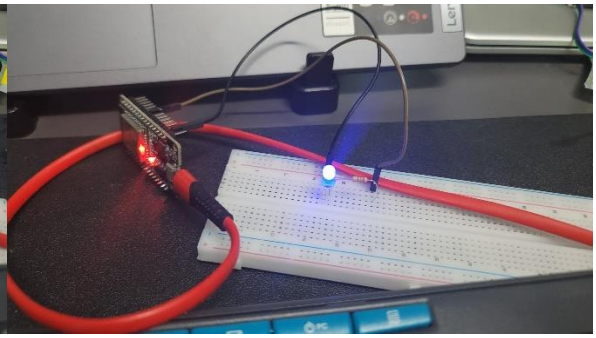


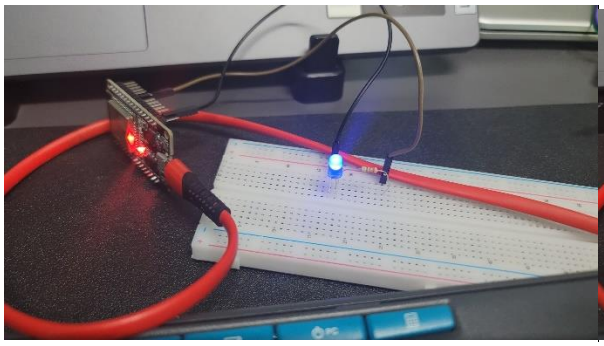
LED fade:

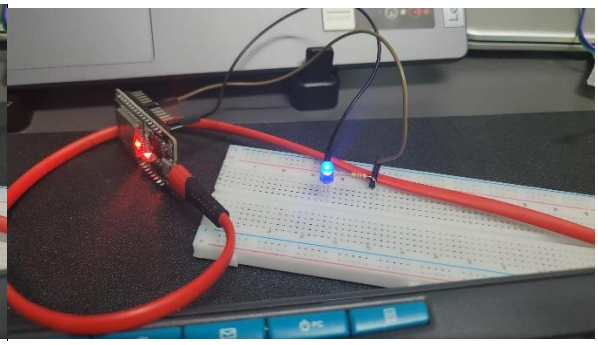


Output(max, medium-high. medium-low, low)



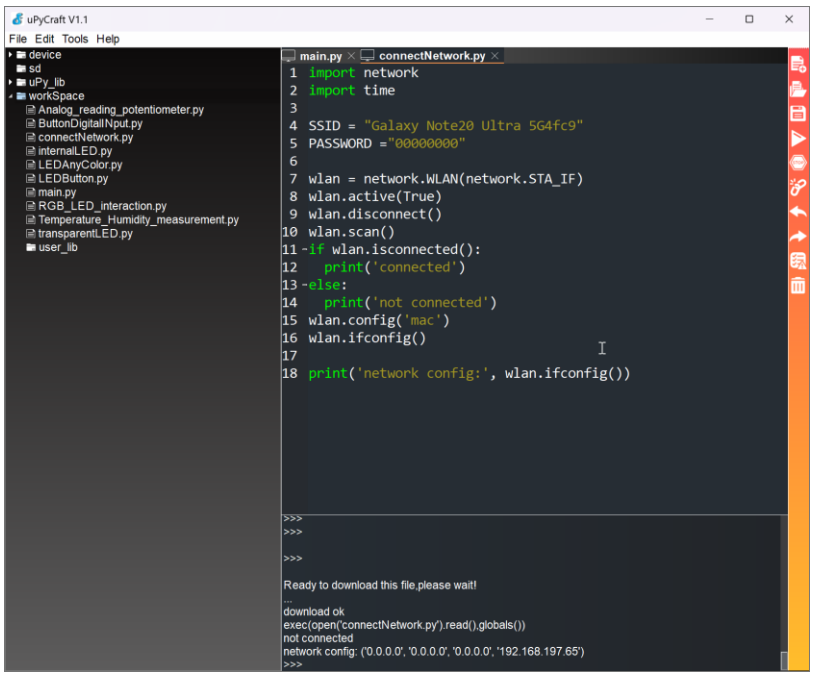




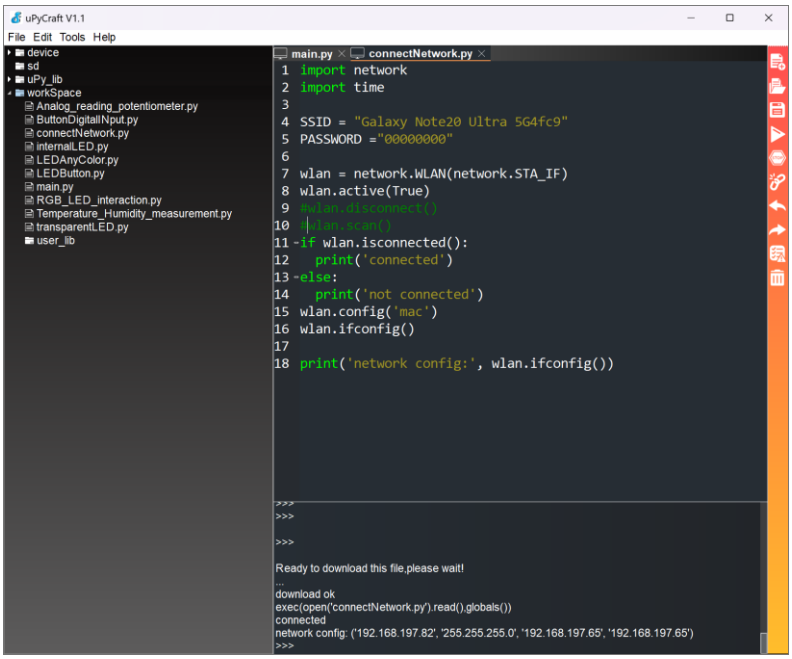


LAB 2 IOT PROGRAMMING:

ESP32 connect to Wi-Fi.

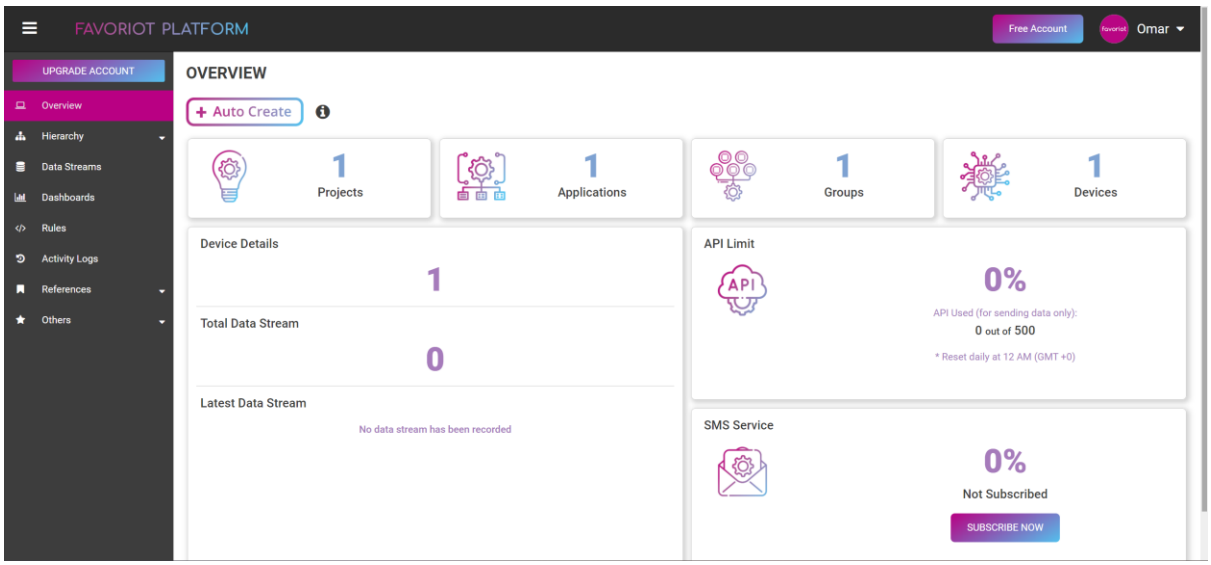


If we comment the wlan.disconnect() and wlan.scan(), we can get more parameters value such as subnet mask and other.

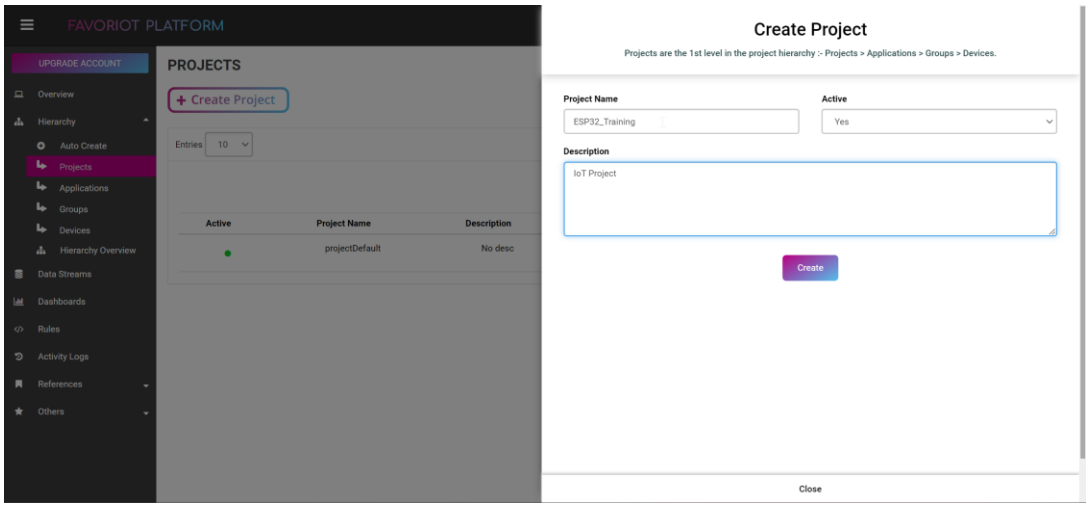


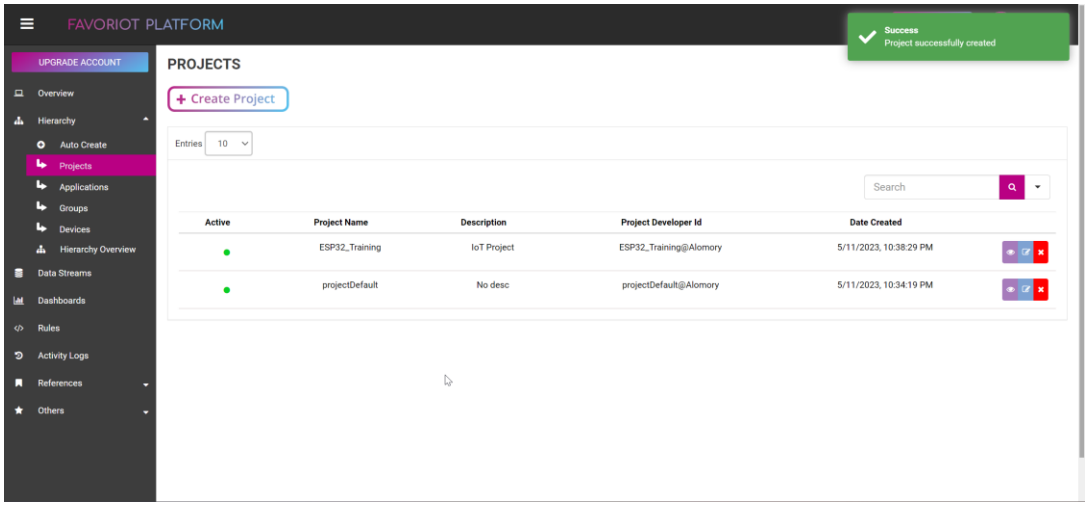
ESP32 MQTT

Registeration Completed Successfully.

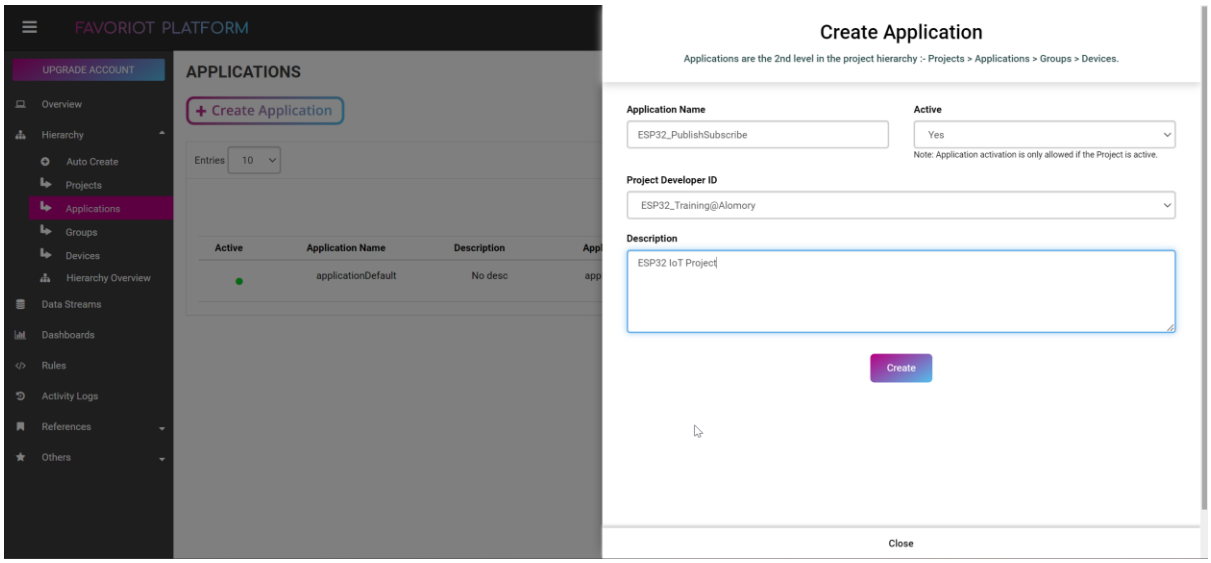


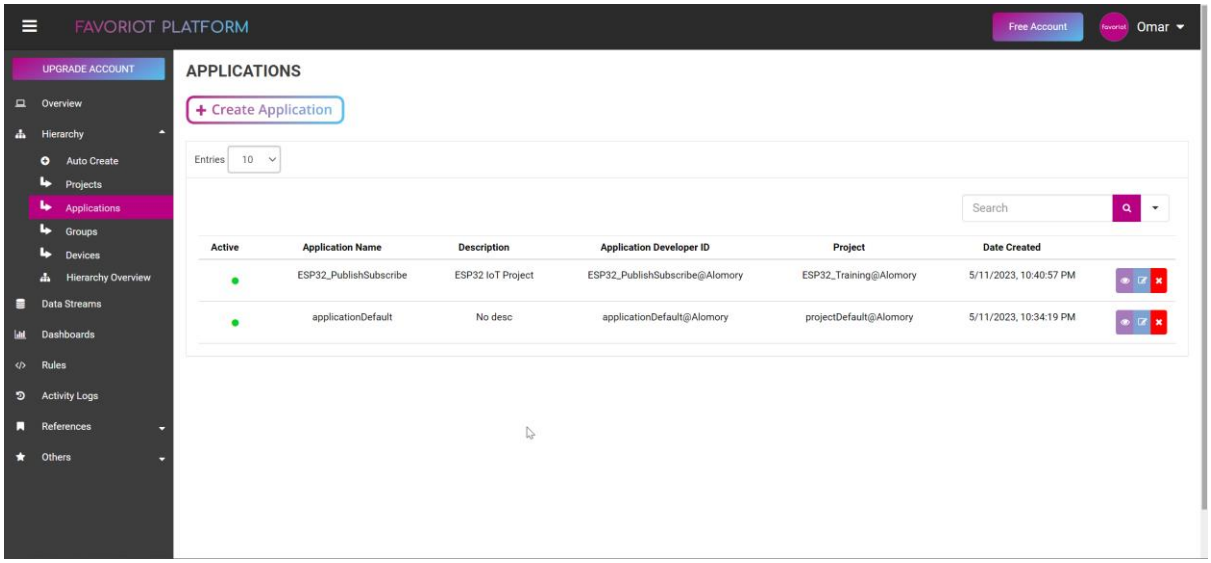
adding new project.



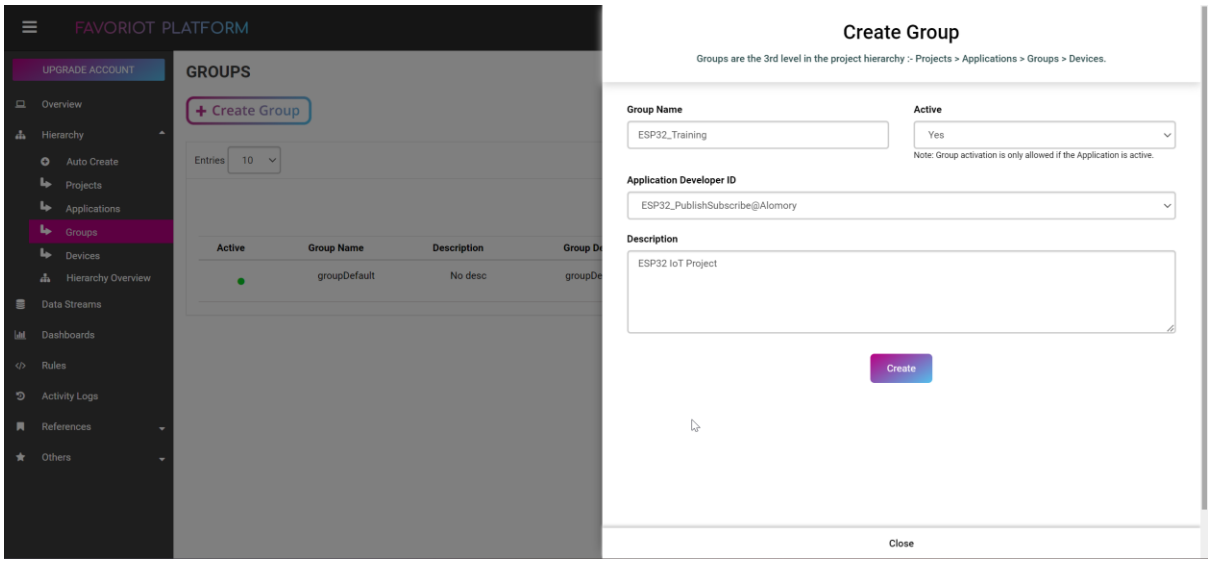


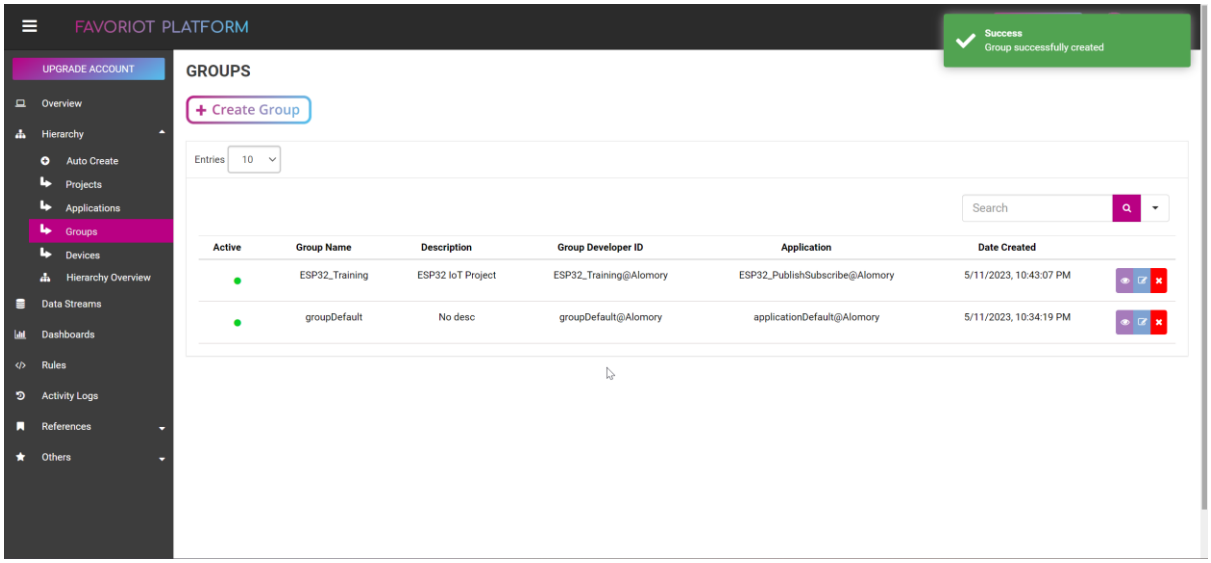
Create application.



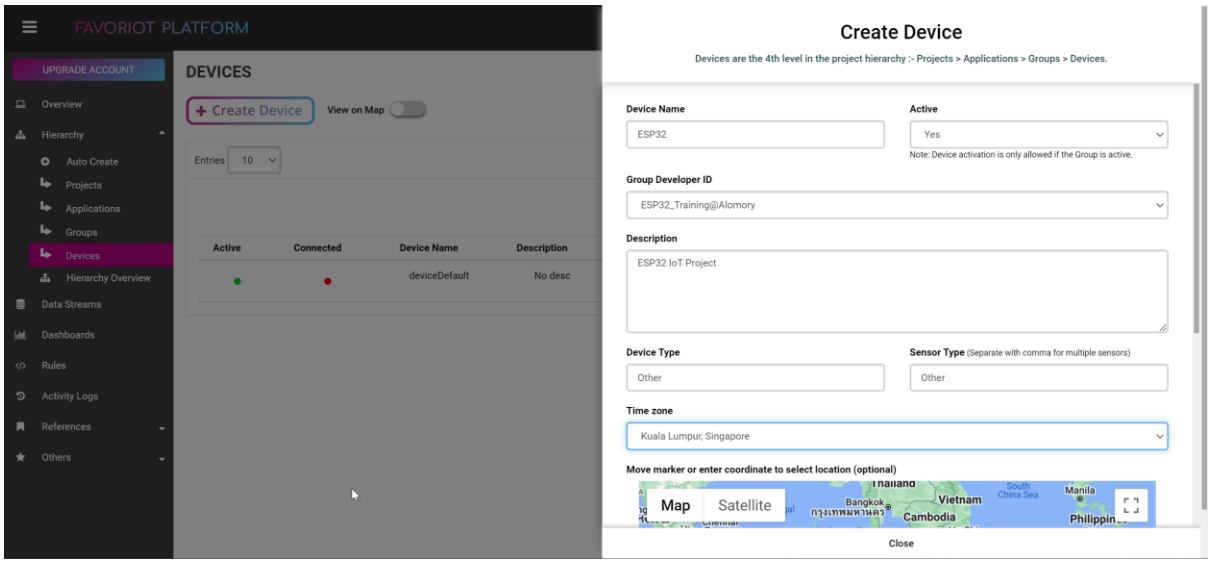


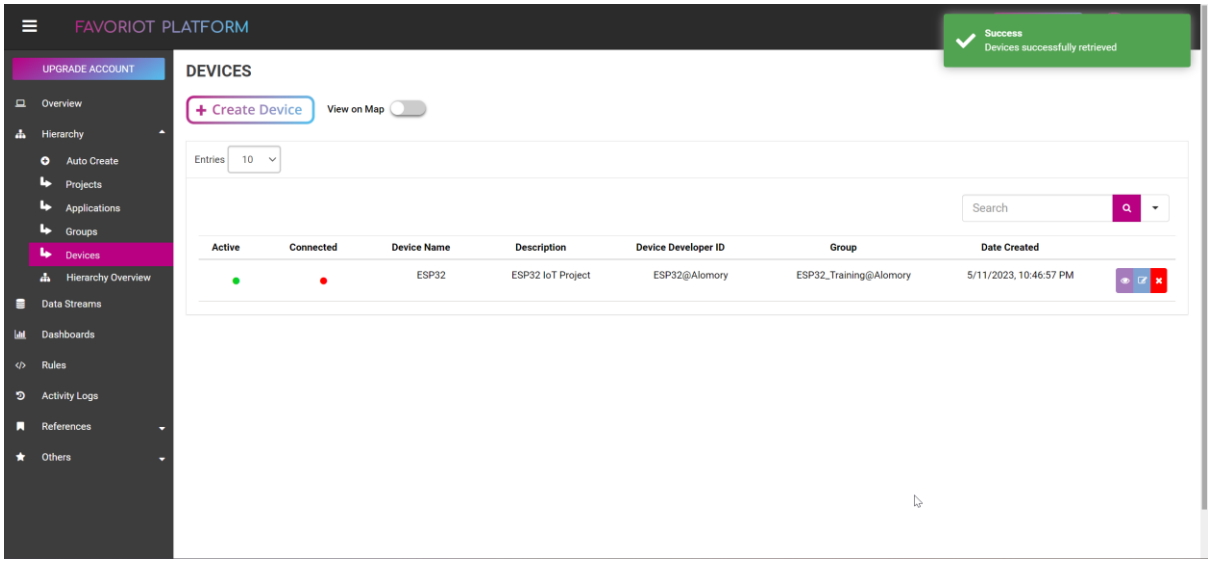
Create Group.



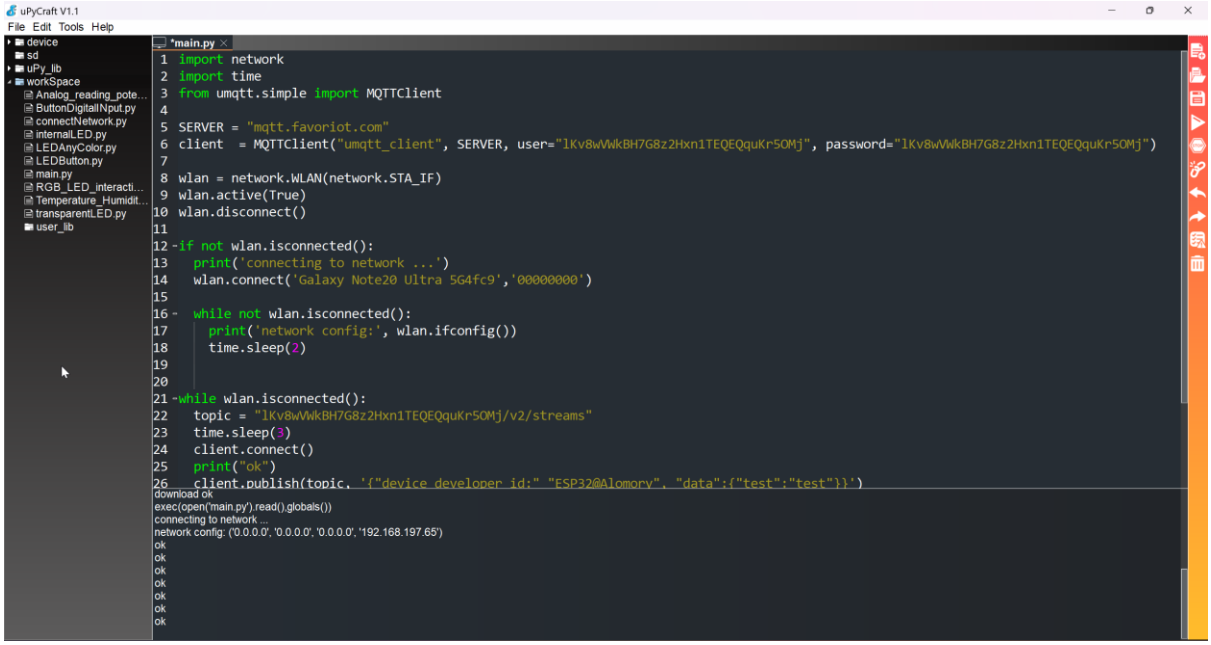


Create Device.

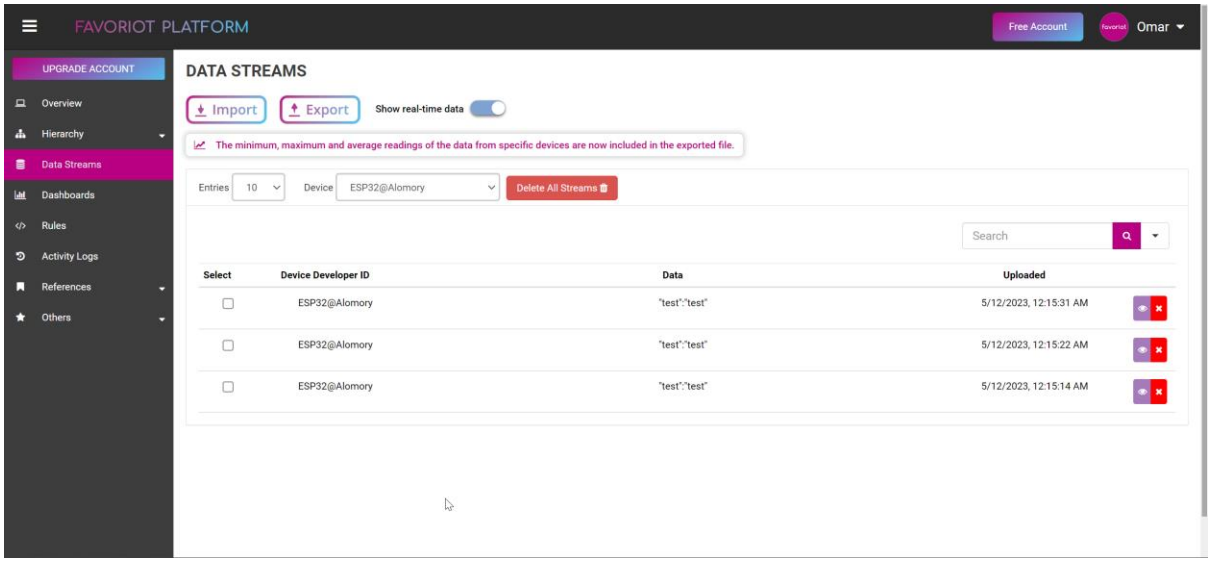




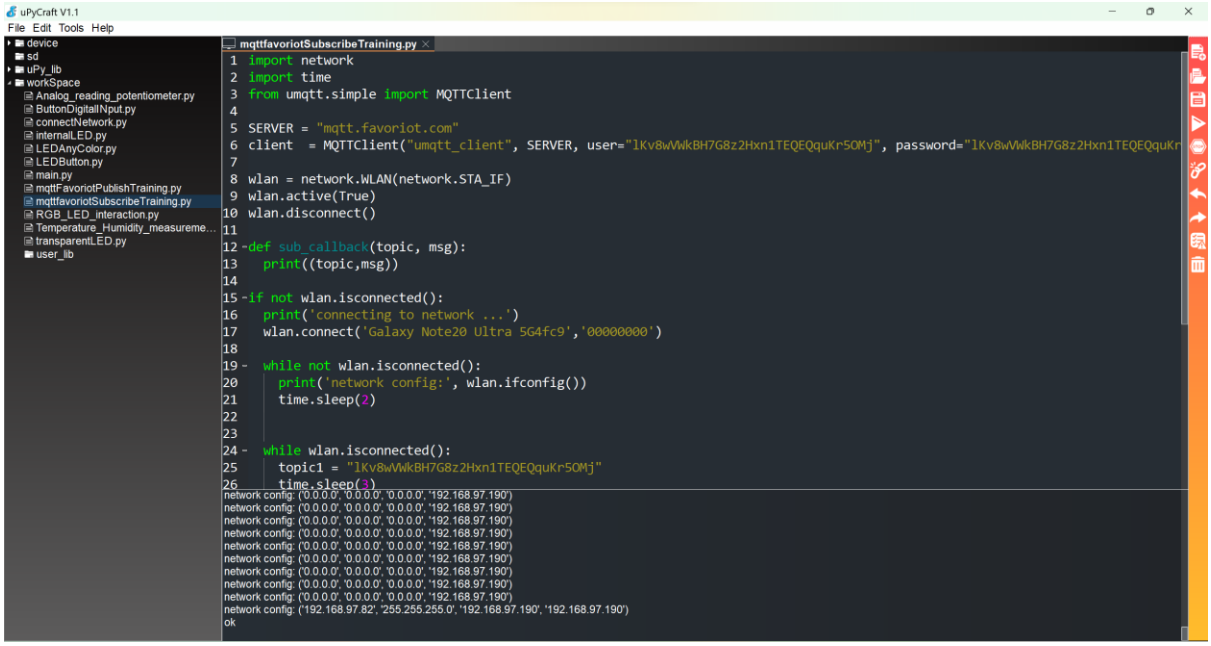
Program your ESP32 as below and edit the program. After that download the program into ESP32.



The dummy “test” data will be published to MQTT Broker Favoriot every 5 seconds. You can see your data at Data Stream tab



Subscriber file



```python
import network
import time
from umqtt.simple import MQTTClient

SERVER = "mqtt.favoriot.com"
client  = MQTTClient("umqtt_client", SERVER, user="lKv8wVWkBH7G8z2Hxn1TEQEQquKr5OMj", password="lKv8wVWkBH7G8z2Hxn1TEQEQquKr

wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.disconnect()

def sub_callback(topic, msg):
  print((topic,msg))

if not wlan.isconnected():
  print('connecting to network ...')
  wlan.connect('Galaxy Note20 Ultra 5G4fc9','00000000')

  while not wlan.isconnected():
    print('network config:', wlan.ifconfig())
    time.sleep(2)


  while wlan.isconnected():
    topic1 = "lKv8wVWkBH7G8z2Hxn1TEQEQquKr5OMj"
    time.sleep(3)
```

```
network config: ('0.0.0.0', '0.0.0.0', '0.0.0.0', '192.168.97.190')
network config: ('0.0.0.0', '0.0.0.0', '0.0.0.0', '192.168.97.190')
network config: ('0.0.0.0', '0.0.0.0', '0.0.0.0', '192.168.97.190')
network config: ('0.0.0.0', '0.0.0.0', '0.0.0.0', '192.168.97.190')
network config: ('0.0.0.0', '0.0.0.0', '0.0.0.0', '192.168.97.190')
network config: ('0.0.0.0', '0.0.0.0', '0.0.0.0', '192.168.97.190')
network config: ('0.0.0.0', '0.0.0.0', '0.0.0.0', '192.168.97.190')
network config: ('0.0.0.0', '0.0.0.0', '0.0.0.0', '192.168.97.190')
network config: ('0.0.0.0', '0.0.0.0', '0.0.0.0', '192.168.97.190')
network config: ('192.168.97.82', '255.255.255.0', '192.168.97.190', '192.168.97.190')
ok
```

As we can see the connection is successful (“ok”) and its waiting whenever data is published to the server it will be retrieved from here.