



LAB 4: IoT Programming

DAS

LAB 4: IOT PROGRAMMING

Objective:

In this lab we are going to code ESP32 microcontroller using uPyCraft IDE. Throughout this lab, we will cover ESP32 programming using MQTT.

Steps:

ESP32 connect to WiFi

1. Connect the ESP32 to your PC/Laptop using USB cable.
2. Open upyCraft and write code as below.

```
connectNetwork.py X
1
2 import network
3 import time
4
5 SSID = "your WiFi SSID"           #WiFi name
6 PASSWORD = "your WiFi password"   #WiFi password
7
8 wlan = network.WLAN(network.STA_IF) #Create WLAN object
9 wlan.active(True)                  #Activate the interface
10 wlan.disconnect()
11 wlan.scan()                        #Scan access point
12 wlan.isconnected()                 #Check whether the site is connected to the AP
13 wlan.connect(SSID, PASSWORD)        #Connect to AP
14 wlan.config('mac')                  #Get the MAC address of the interface
15 wlan.ifconfig()                     #Get the IP/netmask/gw/DNS address of the interface
16
17 print('network config:', wlan.ifconfig())
18
19 #wlan.disconnect()
20
21
22
```

- edit the program at
 - ◆ SSID variable
 - ◆ password variable
- insert your WiFi SSID and password

3. Download and Run the program.
4. Make sure the ESP32 is connected by getting ESP32 IP address as below.

```
[0.32ml (198434) network: CONNECTED [0m
[0.32ml (199454) event: sta ip: 172.16.150.198, mask: 255.255.255.0, gw: 172.16.150.1 [0m
[0.32ml (199454) network: GOT_IP [0m
```

ESP32 MQTT

LAB 4: IOT PROGRAMMING

1. In this exercise, ESP32 will publish data to MQTT broker by Favoriot.
2. Sign up an account at platform.favoriot.com

The screenshot displays the Favoriot platform's user interface. On the left, a dark sidebar contains navigation links: 'Learn More', 'Pricing', 'Documentation', 'API', 'Github', 'Forum', and 'Tutorials'. The main content area is split into two sections. The top section, titled 'Log In', features input fields for 'Username' and 'Password', a 'Log In' button, and a 'Sign up for FREE/BEGINNER' button highlighted with a red rectangle. The bottom section, titled 'Create New Account', contains a 'Free (Free)' plan highlighted with a red rectangle and a 'Beginner (MYR 100) Per year' plan. The 'Free' plan is described as 'Great for simple projects or to be familiar with an IoT platform' and lists features: 'Support Three Devices', '500 APIs Per Day', 'Value Added Services: SMS', 'Free Email Alert', and '1-year Data Retention'. The 'Beginner' plan is described as 'Awesome for an individual developer or for student project' and lists features: 'Support Unlimited Devices', '5,000 APIs Per Day', 'Value Added Services: SMS & Telegram', 'Free Email Alert', and '1-year Data Retention'. Both plans have 'Register as Free' and 'Register as Beginner' buttons respectively. At the bottom, the 'Create New Account' form includes fields for 'First Name', 'Last Name', 'Company', 'Email', 'Password', 'Confirm Password', and a 'Submit' button. A link for 'Already registered? Log In' is also present.

Begin your IOT journey with
favoriot

Learn More
Pricing
Documentation
API
Github
Forum
Tutorials

© favoriot IOT Platform V2 2020
Privacy Policy V2.9.4
Check us out at [favoriot.com](https://platform.favoriot.com)

Log In
favoriot Platform

Username
Password
Log In
☐ Remember Me
[Forgot your password?](#)
Sign up for FREE/BEGINNER

Free (Free)
Great for simple projects or to be familiar with an IoT platform.
• Support Three Devices
• 500 APIs Per Day
• Value Added Services: SMS
• Free Email Alert
• 1-year Data Retention
Register as Free

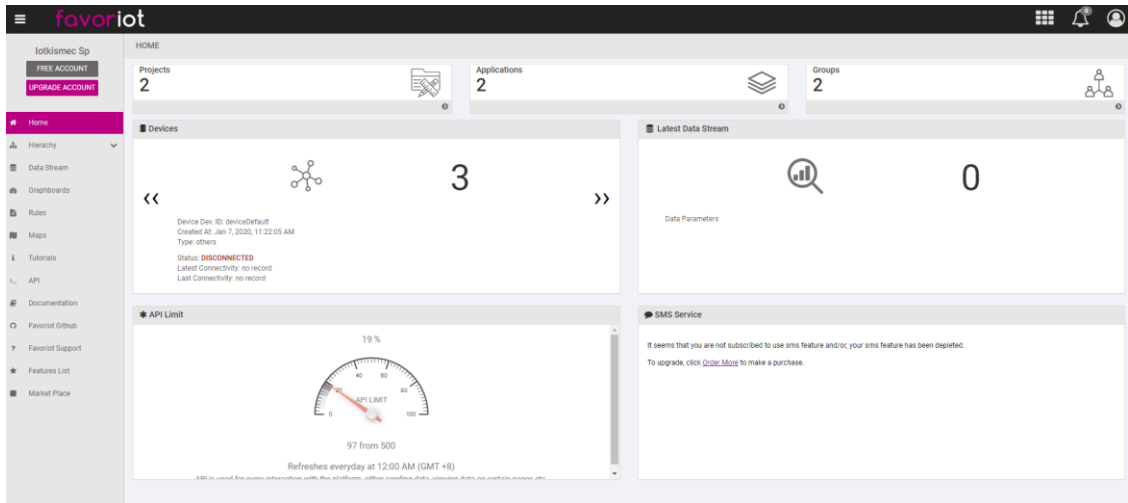
Beginner (MYR 100) Per year
Awesome for an individual developer or for student project.
• Support Unlimited Devices
• 5,000 APIs Per Day
• Value Added Services: SMS & Telegram
• Free Email Alert
• 1-year Data Retention
Register as Beginner

Create New Account

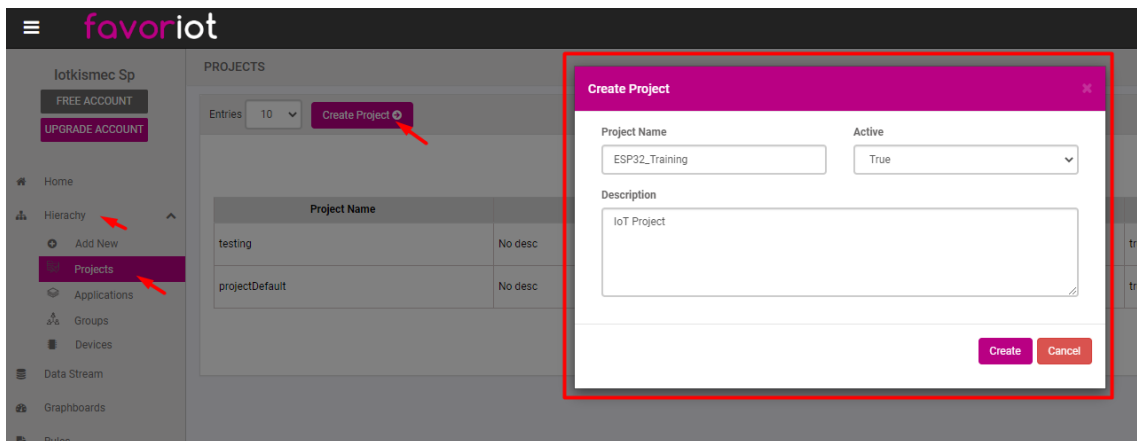
First Name
Last Name
Company
Select country
Email
Username
Password
Confirm Password
Free BEGINNER
Submit
Already registered? Log In

3. Login into your account in favoriot platform.

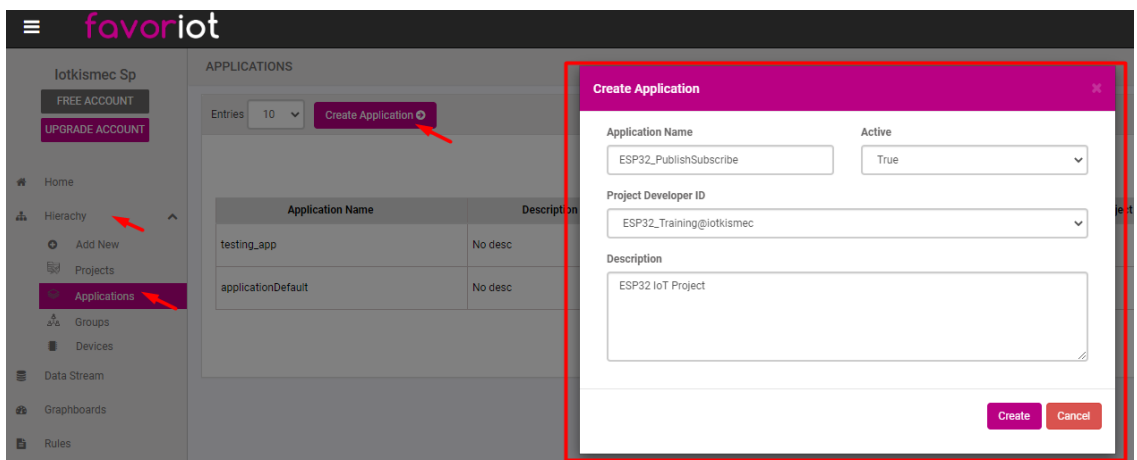
LAB 4: IOT PROGRAMMING



4. Add new project.
Hierarchy > Projects > Create Project

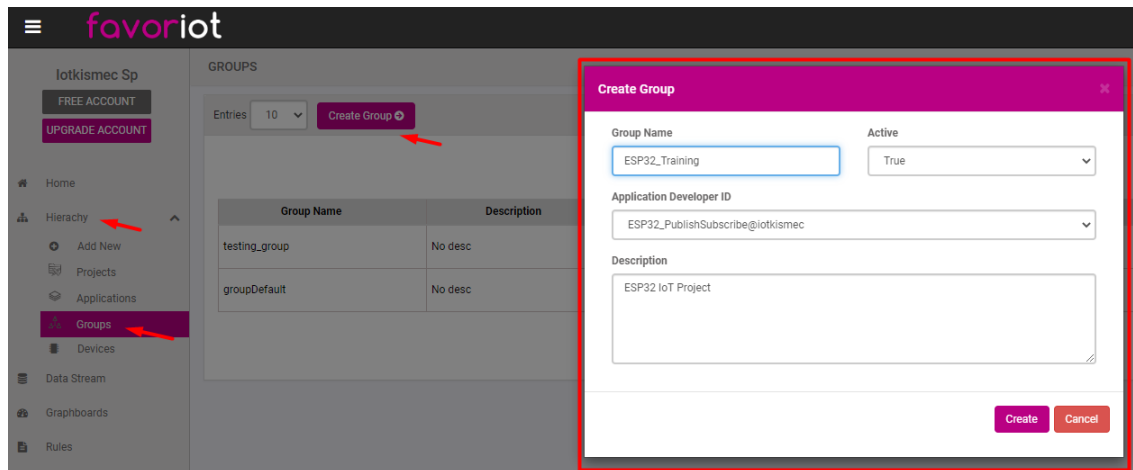


5. Create application.
Hierarchy > Applications > Create Application

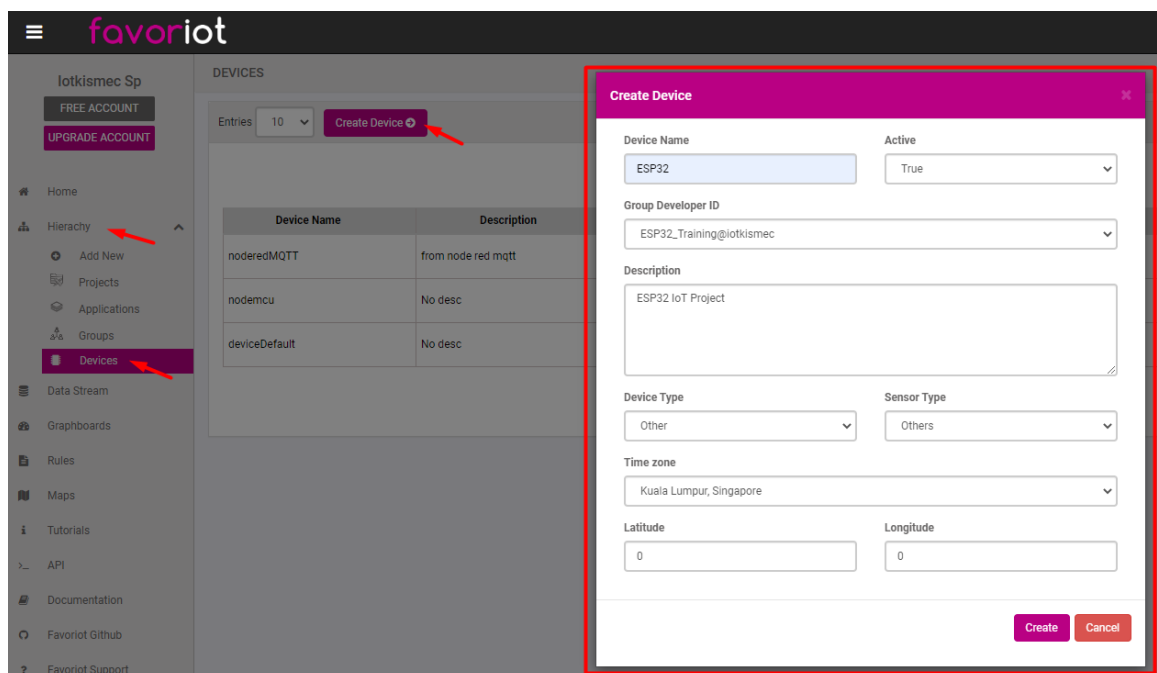


6. Create a new group.
Hierarchy > Groups > Create Group

LAB 4: IOT PROGRAMMING



7. Create a new device.
Hierarchy > Devices > Create Device



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

LAB 4: IOT PROGRAMMING

```
1 import network
2 import time
3 from umqtt.simple import MQTTClient
4
5 SERVER = "mqtt.favotiot.com"
6 client = MQTTClient("umqtt_client", SERVER, user="deviceAccessToken", password="deviceAccessToken")
7
8 wlan = network.WLAN(network.STA_IF)
9 wlan.active(True)
10 wlan.disconnect()
11
12 -if not wlan.isconnected():
13
14     print('connecting to network...')
15     wlan.connect('your SSID', 'your password') #Connect to AP
16         #'SSID': WiFi name
17         #'PASSWORD': WiFi password
18 - while not wlan.isconnected():
19     print('network config:', wlan.ifconfig())
20
21
22 -while wlan.isconnected():
23
24     topic = "deviceAccessToken/v2/streams"
25     client.connect()
26     print("ok")
27     client.publish(topic, '{"device_developer_id": "deviceDeveloperID", "data": {"test": "test"}}')
28     client.disconnect()
29     time.sleep(5)
30
```

Put your

- WiFi SSID
- WiFi Password
- deviceAccessToken
- deviceDeveloperID

How to get device Access Token and Developer ID :
Hierarchy > Devices > Edit > View (yellow eye icon)

LAB 4: IOT PROGRAMMING

The screenshot displays the Favoriot IoT platform interface. A modal window titled "Device Information" is open, showing fields for creating a new device. The fields are organized into two columns. The left column includes Device ID, Device Name, Connectivity, Latitude, Device Type, and Device Description. The right column includes Device Developer ID, Device Created, Access Token, Longitude, Sensor Type, Timezone, Device Updated, and an Active checkbox. The "Device Developer ID" and "Access Token" fields are highlighted with red boxes. The background shows a sidebar with navigation options like Home, Hierarchy, Add New, Projects, Applications, Devices, Dashboards, Rules, Maps, Tutorials, API, Documentation, Favoriot Github, Favoriot Support, Features List, and Market Place. A table of devices is also visible in the background.

Device Name	Description
ESP32	ESP32 IoT Project
nodemcuMQTT	from node red mqtt
deviceDefault	No desc

Device Information

Device ID: 9e391284-bb97-4949-8f18-fe2ddf17ac5b

Device Developer ID: ESP32@iotkismec

Device Name: ESP32

Device Created: 7/8/2020, 12:57:44 PM

Connectivity: false

Access Token: 2JB10thxu7dbpKDt5QuIVa2BgVzb

Latitude:

Longitude:

Device Type: others

Sensor Type: others

Timezone: Kuala Lumpur, Singapore

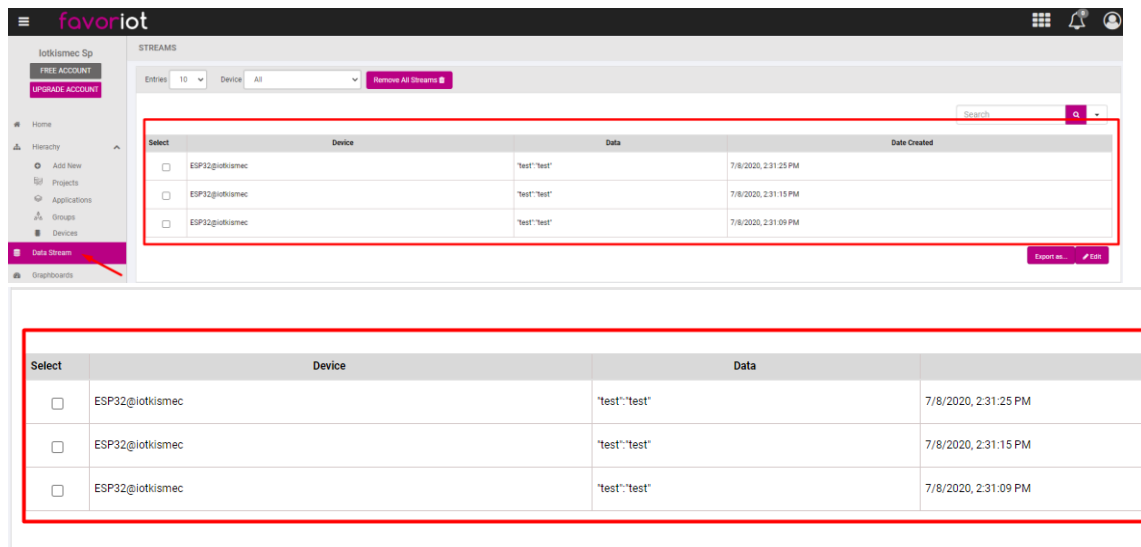
Device Updated: 7/8/2020, 12:57:44 PM

Active: Yes

Device Description: ESP32 IoT Project

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

LAB 4: IOT PROGRAMMING



The screenshot shows the Favoriot IoT platform interface. On the left is a sidebar with navigation options: Home, Hierarchy, Add New, Projects, Applications, Groups, Devices, Data Streams (highlighted), and Graphboards. The main area is titled 'STREAMS' and contains a table of data streams. The table has columns for 'Select', 'Device', 'Data', and 'Date Created'. Three entries are listed, all from the device 'ESP32@iotismec' with the data 'test:test'.

Select	Device	Data	Date Created
<input type="checkbox"/>	ESP32@iotismec	'test':'test'	7/8/2020, 2:31:25 PM
<input type="checkbox"/>	ESP32@iotismec	'test':'test'	7/8/2020, 2:31:15 PM
<input type="checkbox"/>	ESP32@iotismec	'test':'test'	7/8/2020, 2:31:09 PM

10. The program below is for susbcribing data from MQTT Broker Favoriot.

```
mqttFavoriotSubscribeTraining.py
1 import network
2 import time
3 from umqtt.simple import MQTTClient
4 SERVER = "mqtt.favoriot.com"
5 client = MQTTClient("umqtt_client", SERVER, user="deviceAccessToken", password="deviceAccessToken")
6 wlan = network.WLAN(network.STA_IF)
7 wlan.active(True)
8 wlan.disconnect()
9 def sub_callback(topic, msg):
10     print((topic, msg))
11 if not wlan.isconnected():
12     print('connecting to network...')
13     wlan.connect('your SSID', 'your password') #connect to AP
14     #SSID: wifi name
15     #PASSWORD: wifi password
16 while not wlan.isconnected():
17     print('network config:', wlan.ifconfig())
18 while wlan.isconnected():
19     topic1 = "deviceAccessToken"
20     client.set_callback(sub_callback)
21     client.connect()
22     print("ok")
23     client.subscribe(topic1)
24 while True:
25     if True:
26         # Blocking wait for message
27         client.wait_msg()
28     else:
29         # Non-blocking wait for message
30         client.check_msg()
31         # Then need to sleep to avoid 100% CPU usage (in a real app other useful actions would be performed instead)
32         time.sleep(1)
33     client.disconnect()
```

References:

1. <https://randomnerdtutorials.com/projects-esp32/>