

Steg 2 (Buttons)

I det här steget ska vi egentligen göra samma sak, fast denna gång är vi ute efter att kunna se MQTT-trafik i vår IoT-Open-plattform. För att uppnå detta behöver vi göra följande steg:

- Kontrollera noggrant så att din "Edge client" är online och lyser grönt.
- Öppna din mapp i terminalen genom att skriva: `cd test/`
- Skapa en ny fil genom att skriva: `nano följt av buzzer_buttons_integration.py`
- Inne i filen skriver du följande kod:

```
#!/usr/bin/python

import paho.mqtt.client as mqtt
import RPi.GPIO as GPIO
import time
import json

topic_read="obj/tuc/button"
button_on = 14
button_off = 15

client = mqtt.Client()
client.connect("localhost")

GPIO.setmode(GPIO.BCM)

last_pressed_button = 0

def button_callback(gpio):
    global last_pressed_button
    now=time.time()

    # Ignore repeated press on the same button
    if last_pressed_button == gpio:
        return

    if gpio == button_on:
        client.publish(topic_read, '{"value":1, "timestamp":'
```

```

+ str(now) + '}')
    print("Button ON pushed! ")
    elif gpio == button_off:
        client.publish(topic_read, '{"value":0, "timestamp":'
+ str(now) + '}')
        print("Button OFF pushed! ")

    last_pressed_button = gpio

GPIO.setup(14, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.add_event_detect(14,GPIO.RISING,callback=button_callback)

GPIO.setup(15, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.add_event_detect(15,GPIO.RISING,callback=button_callback)

client.loop_forever()

```

- Spara filen genom att trycka: `Ctrl + X` --> Tryck `Y` för att bekräfta sparningen --> Tryck `Enter`.
- Kör följande kommando för att installera python3-paho-mqtt: `sudo apt-get install python3-paho-mqtt`
- Kör din nyligen skapade fil genom att använda kommandot: `python buzzer_buttons_integration.py`
- Nu ska du kunna gå till ditt MQTT-fönster i IoT-Open och se alla meddelanden när du trycker på ON och OFF på dina knappar. OBS!! Du kommer fortfarande inte höra något när du trycker på knapparna.