

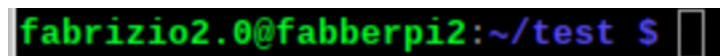
## STEG 2

1. Här kommer vi att använda oss av "python". Vi kommer att utföra följande steg för att kontrollera så att vår "buzzer" fungerar korrekt med python. Gör följande:

- Börja med att skapa en mapp i terminalen: ``mkdir test/``

- Öppna mappen: ``cd test/``

- Din terminal bör nu visa: ``~/test $``



- För att skapa en fil och namnge den som (buzzer\_flash.py), använd följande kommando: ``nano buzzer_flash.py``

- Kopiera in följande kod (finns även i mappen "CODE" på GitHub):

```
#!/usr/bin/python

import RPi.GPIO as GPIO
import time

led = 17

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(led,GPIO.OUT)

# Turn on the BUZZER
GPIO.output(led,1)

#Wait 0.5s
time.sleep(0.5)

#Turn off the BUZZER
GPIO.output(led,0)
```

- För att spara din fil och komma tillbaka till terminalen, använd följande kommandon: Tryck ``Ctrl + X`` --> Tryck ``Y`` för att bekräfta att du vill spara ändringarna --> Tryck ``Enter``. Du bör nu återvända till terminalen. Om du vill kontrollera att din fil har sparats korrekt, kan du använda kommandot ``ls`` för att visa alla filer i "test"-mappen.

- Nu kan du prova din fil för att se om din "buzzer" låter i 0,5 sekunder. Kör följande kommando: ``python buzzer_flash.py``

- Din buzzer bör låta i 0,5 sekunder. 😊

----->

I detta steg kommer vi att styra vår "buzzer" via IoT-Open. Vi kommer att använda protokollet "MQTT" för att uppnå detta. Nu kör vi!

För att lyckas med detta måste du vara inloggad i IoT-Open. Gå till din installation och under "Översikt" bör du se en ruta som heter "Edge Client". Den bör vara grön (se bifogad bild).

The screenshot shows the IoT-Open web interface. At the top, there is a navigation bar with the IoT-Open logo and links for Home, Installations, Users, and Organizations. On the left, there is a sidebar menu with options: Labbar Fabbe, Overview, Functions, Devices, Map, Schedules, Notifications, Edge app instances, History, MQTT, and Settings. The main content area is titled "Overview" and contains two sections. The first section, "Installation Information", shows the installation ID "TUC IOTD23 / FAMU05", the organization "Labbar Fabbe", and the purpose "Träning för min andra Pi". The second section, "Status", contains three buttons: "Devices", "Functions", and "Edge client".

I följande steg ska vi använda MQTT för att styra vår "buzzer":

- Kör följande kommando för att installera "paho python lib" på din RPi: ``sudo apt-get install python3-paho-mqtt``
- Skapa en ny fil och namnge den "buzzer\_integration\_v1.py" genom att skriva följande kommando: ``nano buzzer_integration_v1.py``
- Klistra in följande kod:

```
#!/usr/bin/python

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

topic_write = "set/test/buzzer1"
buzzer = 17

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(buzzer, GPIO.OUT)

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

def on_message(client, userdata, message):
    payload = json.loads(message.payload)
    if message.topic == topic_write:
        if payload["value"] == 0:
            print("Turning buzzer off")
            GPIO.output(buzzer, 0)
        else:
            print("Turning buzzer on")
            GPIO.output(buzzer, 1)
    else:
        print("Ignoring " + message.topic)

print("Subscribing to topic " + topic_write)
client.subscribe(topic_write)
client.on_message = on_message

client.loop_forever()
```

- Spara på filen på samma sätt som för: Tryck `Ctrl + X` --> Tryck `Y` för att bekräfta att du vill spara ändringarna --> Tryck `Enter`.
- Nu när du är tillbaka i terminalen, kör kommandot: `python buzzer\_integration\_v1.py`. När du kör detta kommando kommer det att ta över din terminal. Du kan stoppa programmet med "Ctrl-C", men gör inte det nu. Öppna istället

en annan terminal och kör följande kommando: ``mosquitto_pub -t set/test/buzzer1 -m '{"value": 0}'`` och ``mosquitto_pub -t set/test/buzzer1 -m '{"value": 1}'``. Återigen kan vi se 0 & 1. Din buzzer bör låta när värdet är 1 och sluta låta när värdet är 0.

----->

Nu ska vi styra "buzzer" via IoT-Open Plattformen istället för terminalen.

- Gå till din installation och välj "funktioner" och skapa en ny+. Du kan kalla den för "MyBuzzer" och under typ kan du skriva "switch".

I MetaData-sektionen kan du välja "Lägg till metadata". Under "Nyckel" skriver du "topic\_write" och under "Värde" skriver du "set/test/buzzer1". Tryck på "Spara" när du är klar. Det bör se såhär när du är klar :

Installations

Users

Organizations

Device

Raspberry Pi 3

Go to device

Name

MyBuzzer

Type

switch

Created

May 10, 2024 1:30 PM

Updated

May 10, 2024 1:31 PM

Metadata

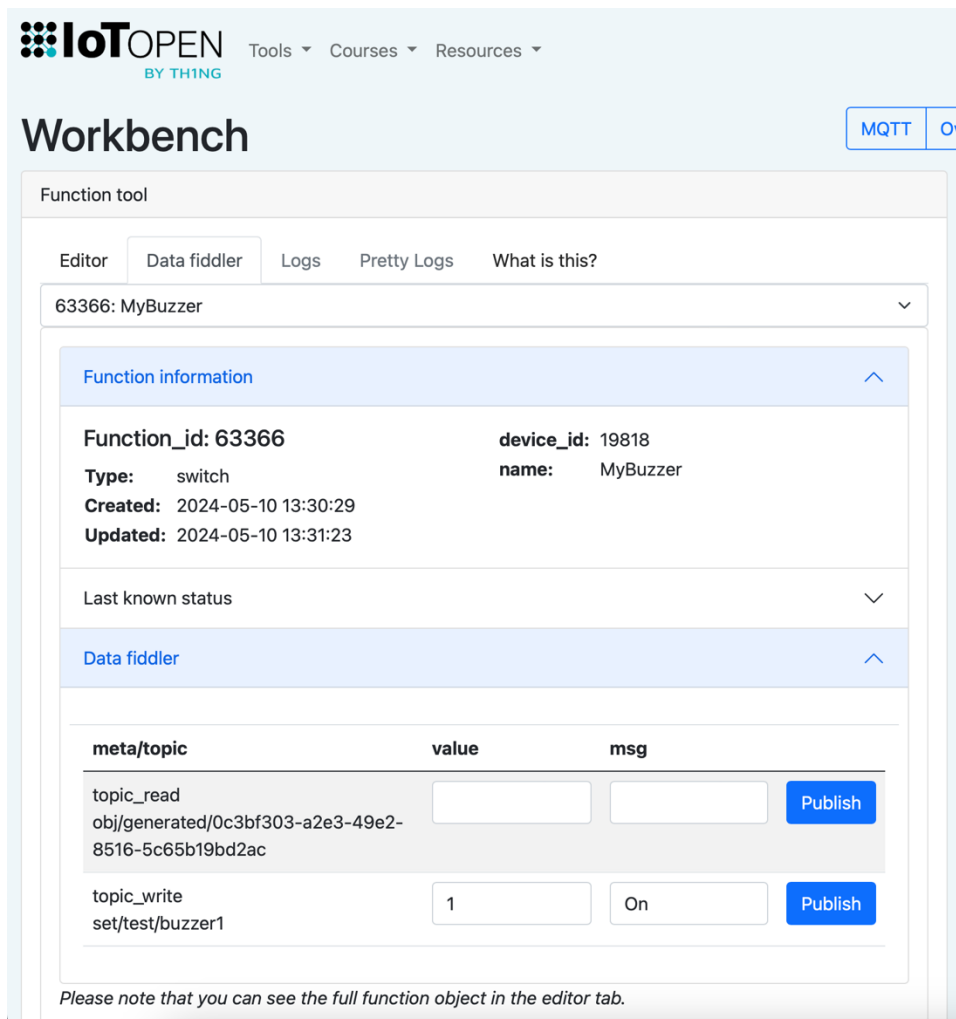
	Key	Value		
	device_id	19818		<input type="checkbox"/>
	name	MyBuzzer		<input type="checkbox"/>
	topic_read	obj/generated/0c3bf303-a2e3-		<input type="checkbox"/>
	topic_write	set/test/buzzer1		<input type="checkbox"/>

Add metadata

Propagate

Save

- Nu kan du öppna en ny flik i din webbläsare och öppna IoT-open Workbench. När du har gjort detta, gå längst upp till höger och välj din installation. När du har kommit in i din installation, välj "Funktioner" ovanför MQTT-fönstret. När du är inne där kommer du att se en flik som heter "Data fiddler". Välj din nyss skapade funktion "MyBuzzer" och välj sedan "Data fiddler". Det bör se ut såhär för dig:



- Nu ska du kunna styra din "buzzer" via IoT-open. Under "set/test/buzzer1", välj återigen 1 eller 0 för att aktivera eller inaktivera den. Du ska även kunna följa detta live på MQTT-rutan bredvid dig.

Så nu kan vi helt enkelt styra vår "buzzer" via IoT-Open.