LAB 1 Activity 1: Setting up Internet of Things [10 marks]

1.1 You can choose to connect the Raspberry Pi to either of the stated options below as a display:

Option 1: Screen Computer Monitor

Option 2: LCD Raspberry Pi Screen

1.2 Option 1: The Raspberry Pi can be connected to an external monitor via a VGA cable and adapter or a Micro-HDMI to HDMI cable. The connection using a VGA cable and adapter is shown in Figure 1. A computer monitor can be connected similarly using a Micro-HDMI to HDMI cable.

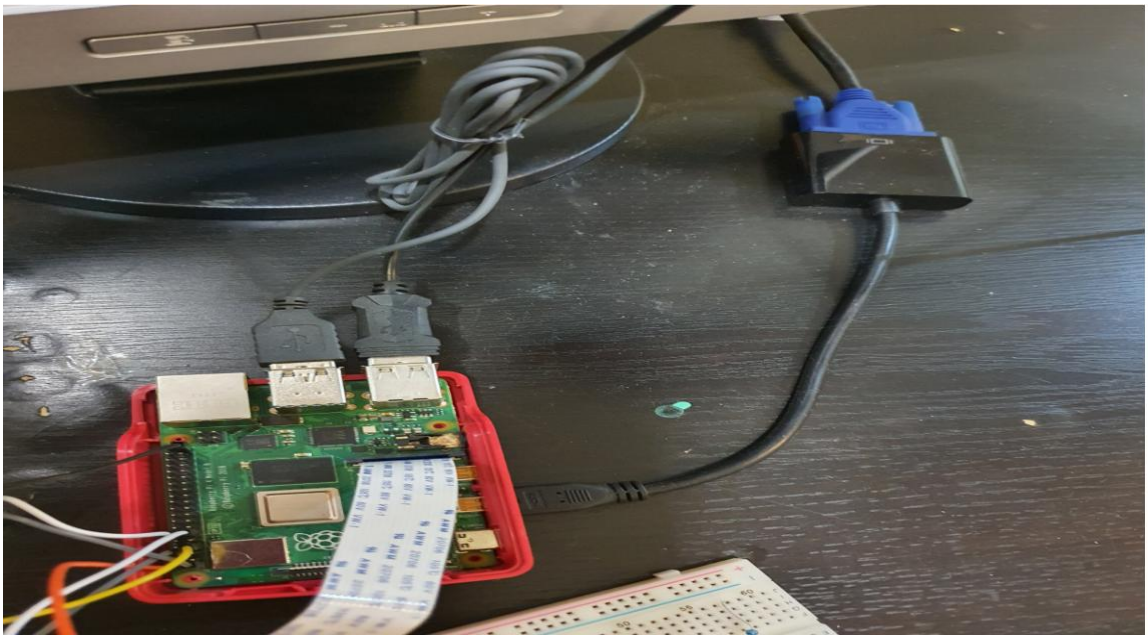


Figure 1: Connection of Raspberry Pi to external monitor.

1.3 Option 2: Follow the steps below to connect the Raspberry Pi Touch Display to the Raspberry Pi:

Step1: Introduction

The Raspberry Pi Touch Display is an LCD display which connects to the Raspberry Pi through the DSI connector. In some situations, it allows for the use of both the HDMI and LCD displays at the same time (this requires software support).

Step 2: Physical Installation (Important Notes)

Figure 2 shows how to attach the Raspberry Pi to the back of the Touch Display (if required), and how to connect both the data (ribbon cable) and power (red/black wires) from the Raspberry Pi to the display.

raspberrypi.org, pimylifeup.com, O'Reilly connecting a push switch with raspberry pi, Adel Kassah, Winfuture.de

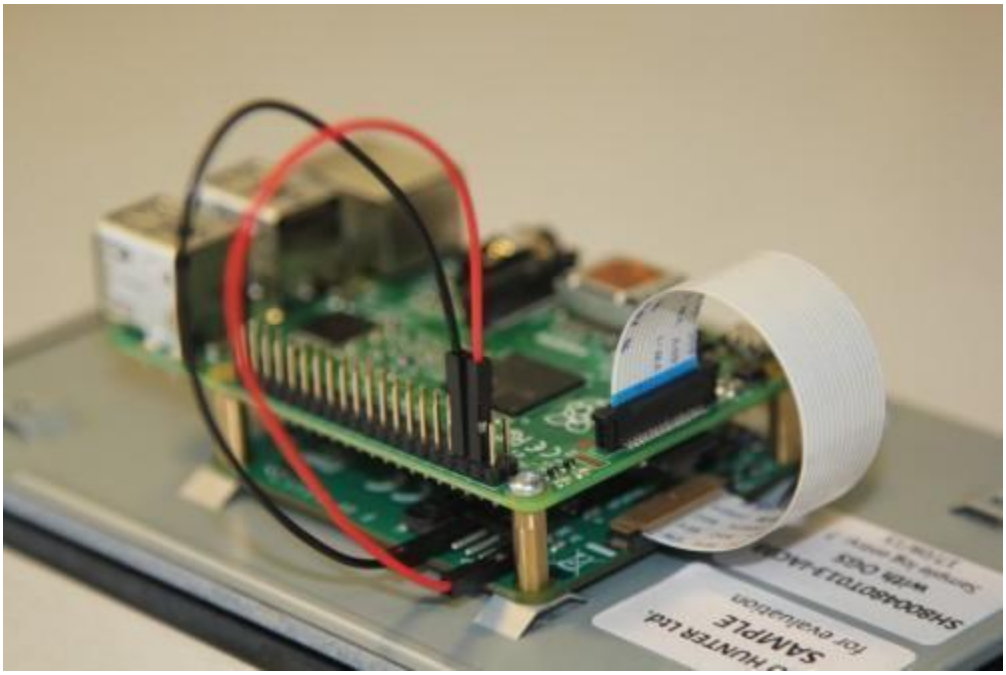
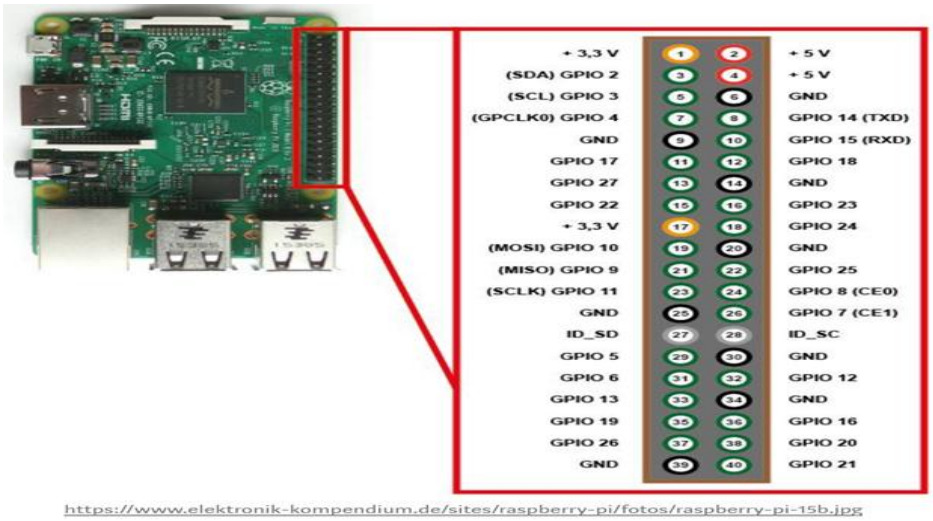


Figure 2: Connection of Raspberry Pi to LCD Raspberry Pi Screen.



https://www.elektronik-kompendium.de/sites/raspberry-pi/fotos/raspberry-pi-15b.jpg

Figure 3: Raspberry Pi pin diagram.

If you are not attaching the Raspberry Pi to the back of the display, take extra care when attaching the **ribbon cable to ensure it is the correct way round**. The black and red power wires should be attached to the GND and 5v pins respectively.

raspberrypi.org, pimylifeup.com, O'Reilly connecting a push switch with raspberry pi, Adel Kassah, Winfuture.de

1.4 Screen orientation

LCD displays have an optimum viewing angle and depending on how the screen is mounted it may be necessary to change the orientation of the display to give the best results. By default, the Raspberry Pi Touch Display and Raspberry Pi are set up to work best when viewed from slightly above, for example on a desktop. If viewing from below, you can physically rotate the display, and then tell the system software to compensate by running the screen upside down.

To set screen orientation when running the graphical desktop, select the Screen Configuration option from the Preferences menu. Right click on the DSI display rectangle in the layout editor, select Orientation then the required option.

1.5 Setup a thingspeak account : https://thingspeak.com/

The Python code for setting up Raspberry Pi to connect to ThingSpeak is shown below. Replace 'YOUR_API_KEY' in the 'RequestToThingSpeak' string with your ThingSpeak API key.

```
import requests
import time
import random

request = None

for count in range(3):
    RequestToThingspeak = 'https://api.thingspeak.com/update?api_key=YOUR_API_KEY&field1='
    RequestToThingspeak +=str((random.randint(1,100)))
    request = requests.get(RequestToThingspeak)
    print(request.text)
    time.sleep(15)
```

LAB 1 Activity 2: Distance Measurement

2.1 Connect one ultrasonic sensor to Raspberry Pi according to Lab 1 video uploaded on Moodle/Panopto.

2.2 Follow the instructions below to see how to wire your HC-SR04 distance sensor to your Raspberry Pi according to figure 4 below.

- VCC Connects to Pin 2 (5v)
- Trig Connects to Pin 7 (GPIO 4)
- Echo Connects to R1 (1k Ω)
- R2 (2k Ω) Connects from R1 to Ground
- Wire from R1 and R2 connects to Pin 11
- GND connects to Pin 6 (Ground)

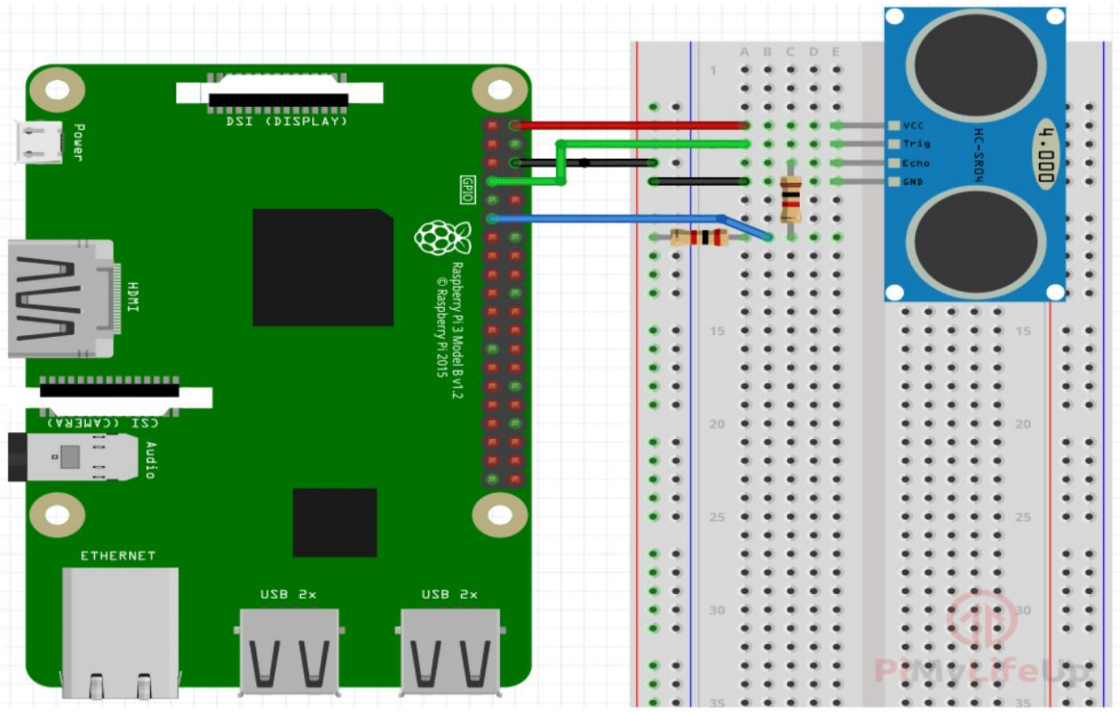
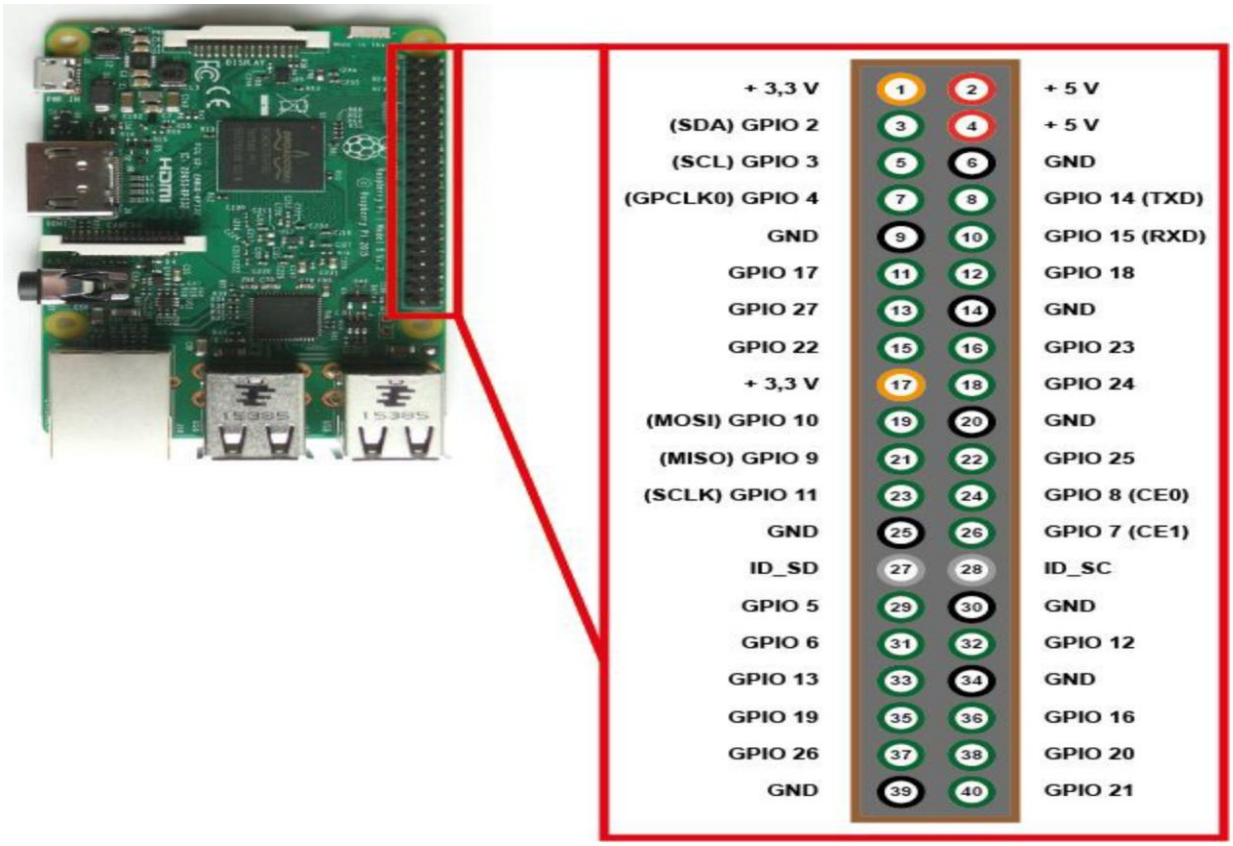


Figure 4: Connection of ultrasonic sensor to Raspberry Pi.

raspberrypi.org, pimylifeup.com, O'Reilly connecting a push switch with raspberry pi, Adel Kassah, Winfuture.de



https://www.elektronik-kompendium.de/sites/raspberry-pi/fotos/raspberry-pi-15b.jpg

Figure 5: Raspberry Pi pin diagram.

The Python code to configure and command the connected ultrasonic sensor is shown below. Remember to replace 'YOUR_API_KEY' in the highlighted line with your ThingSpeak WRITE API key before running the script.

```
import RPi.GPIO as GPIO
import time
import requests #Please install with PIP: pip install requests
request = None

try:
      GPIO.setmode(GPIO.BOARD)
      GPIO.setwarnings(False)

      PIN_TRIGGER = 7
      PIN_ECHO = 11

      GPIO.setup(PIN_TRIGGER, GPIO.OUT)
      GPIO.setup(PIN_ECHO, GPIO.IN)

      GPIO.output(PIN_TRIGGER, GPIO.LOW)

      for count in range(15):
```


raspberrypi.org, pimylifeup.com, O'Reilly connecting a push switch with raspberry pi, Adel Kassah, Winfuture.de

```
            print ("Waiting for sensor to settle")
            time.sleep(2)
            print ("Calculating distance")
            GPIO.output(PIN_TRIGGER, GPIO.HIGH)
            time.sleep(0.00001)
            GPIO.output(PIN_TRIGGER, GPIO.LOW)
            while GPIO.input(PIN_ECHO)==0:
                pulse_start_time = time.time()
            while GPIO.input(PIN_ECHO)==1:
              pulse_end_time = time.time()
            pulse_duration = pulse_end_time - pulse_start_time
            distance = round(pulse_duration * 17150, 2)
            print ("Distance:",distance,"cm")
            print ("Calculating distance")
            print('Sending to ThingSpeak')
            RequestToThingspeak = 'https://api.thingspeak.com/update?api_key=YOUR_API_KEY&field1='
            RequestToThingspeak +=str(distance)
            request = requests.get(RequestToThingspeak)
            print('Request Sent: you got this answer')
            print(request.text)
            time.sleep(15)
            print("completed")

finally:
      GPIO.cleanup()
```


raspberrypi.org, pimylifeup.com, O'Reilly connecting a push switch with raspberry pi, Adel Kassah, Winfuture.de

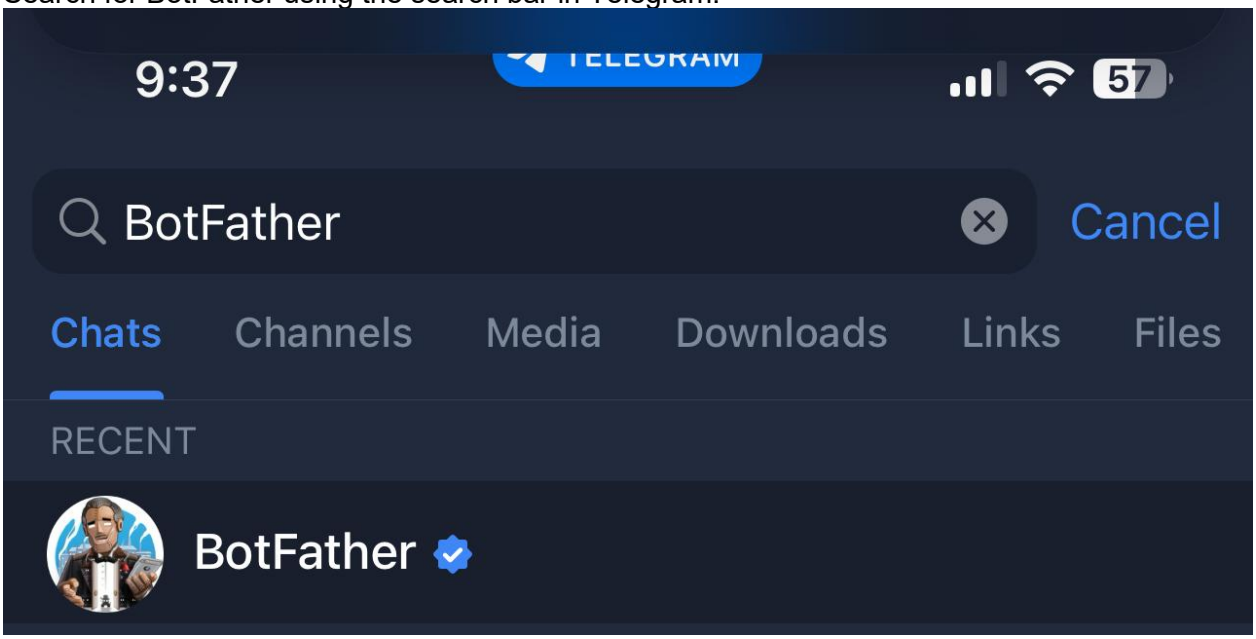
LAB 1 Activity 3: IoT Phone Notifications

Activity 1: Setting with IoT Phone Notifications

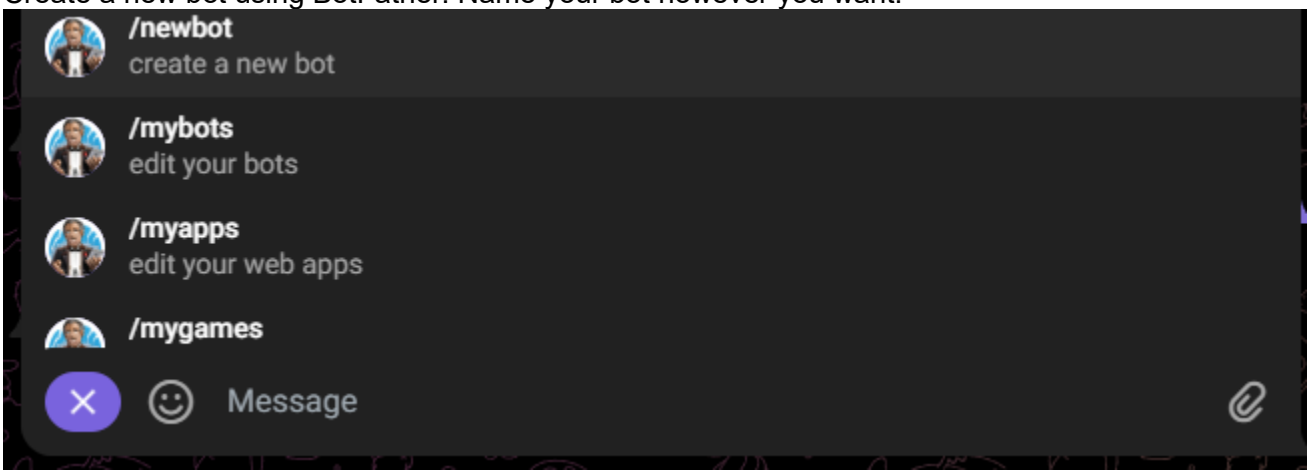
Telegram Bot Setup Guide

Step 1: Create a Telegram Bot

1. Search for BotFather using the search bar in Telegram.

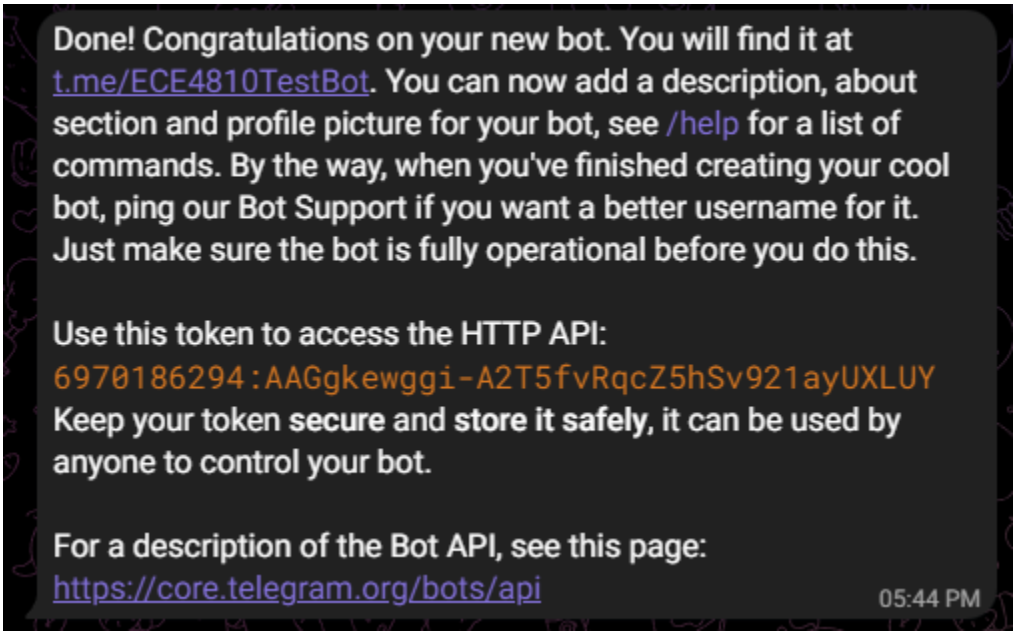


2. Create a new bot using BotFather. Name your bot however you want.



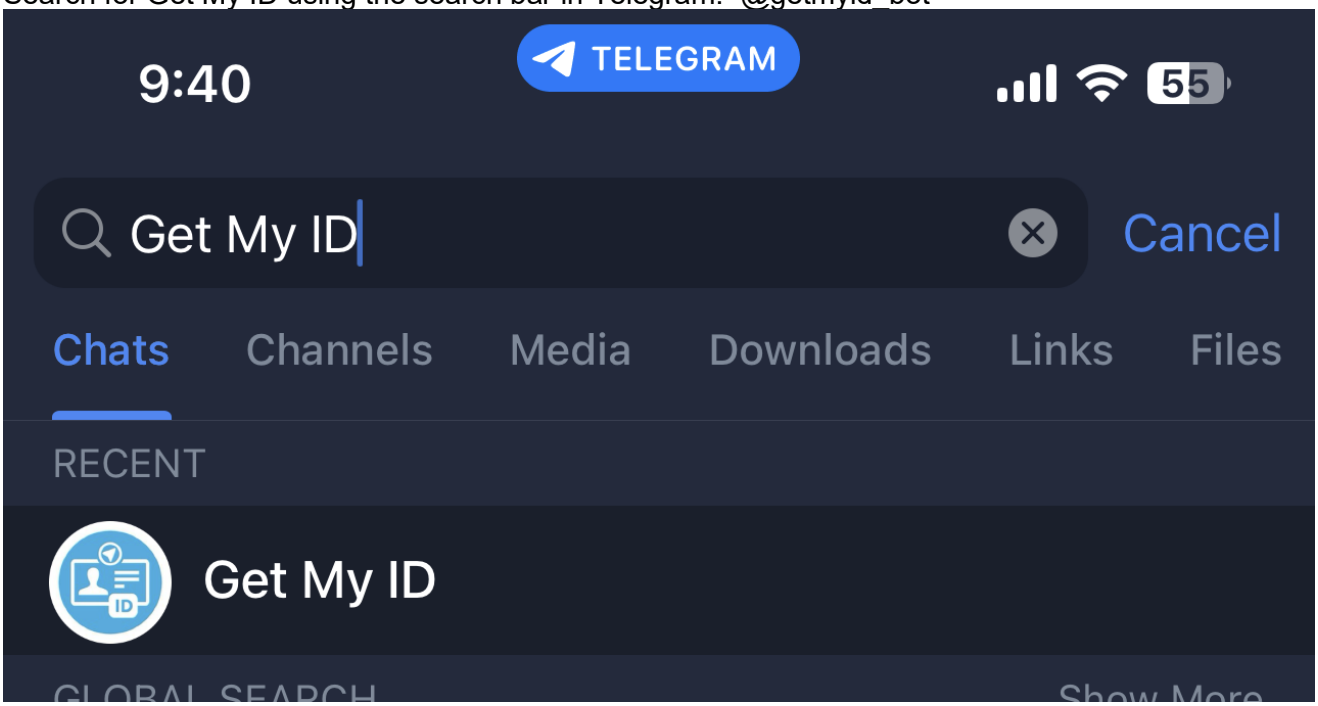
3. Take note of the token returned by BotFather. We will use it later.

raspberrypi.org, pimylifeup.com, O'Reilly connecting a push switch with raspberry pi, Adel Kassah, Winfuture.de



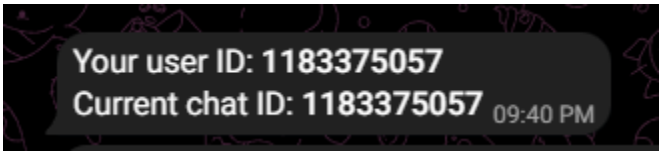
Step 2: Find your Telegram chat ID

1. Search for Get My ID using the search bar in Telegram. @getmyid_bot



2. Take note of your chat ID. We will use it later.

raspberrypi.org, pimylifeup.com, O'Reilly connecting a push switch with raspberry pi, Adel Kassah, Winfuture.de



Step 3: Create python script to send automated messages to Telegram bot

1. Download the code from here.

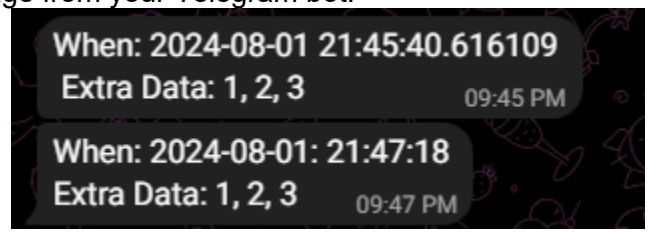
```
import requests
import datetime

TOKEN = "7413374898:AAEDeFQNyv0B7jgWaUU6TZuW_kA5gmd0-H0"
chat_id = "667359446"

def send_message_to_telegram(msg):
    url = "https://api.telegram.org/bot{}/sendMessage".format(TOKEN)
    response = requests.post(
        url = url,
        data={'chat_id': chat_id, 'text': msg}
    )
    print(response)

if __name__ == '__main__':
    value1 = 1
    value2 = 2
    value3 = 3
    now = datetime.datetime.now()
    msg = "When: {} \nExtra Data: {}, {}, {}".format(now.strftime("%Y-%m-%d %H:%M:%S"), value1, value2,
value3)
    send_message_to_telegram(msg)
```

2. Replace the TOKEN and chat_id in the python script with the ones you've gotten from previous steps.
3. Execute the code by running "python3 ece4810_telegram.py" and you should receive a message from your Telegram bot.



4. Code to send messages from the Raspberry pi to the Telegram bot. here.

```
import requests
from time import sleep
import RPi.GPIO as GPIO
import time
import requests #Please install with PIP: pip install requests
import datetime

request = None

TOKEN = "7413374898:AAEDeFQNyv0B7jgWaUU6TZuW_kA5gmd0-H0"
CHAT_ID = "667359446"

def send_message_to_telegram(msg):
    url = "https://api.telegram.org/bot{}/sendMessage".format(TOKEN)
    response = requests.post(
        url = url,
        data={'chat_id': CHAT_ID, 'text': msg}
    )

try:
    GPIO.setmode(GPIO.BOARD)
    GPIO.setwarnings(False)

    PIN_TRIGGER = 7
    PIN_ECHO = 11

    GPIO.setup(PIN_TRIGGER, GPIO.OUT)
    GPIO.setup(PIN_ECHO, GPIO.IN)

    GPIO.output(PIN_TRIGGER, GPIO.LOW)

    # Set GPIO sensor is connected to
    gpio=21

    for count in range(15):
        print ("Waiting for sensor to settle")
        time.sleep(2)
        print ("Calculating distance")
        GPIO.output(PIN_TRIGGER, GPIO.HIGH)
        time.sleep(0.00001)
        GPIO.output(PIN_TRIGGER, GPIO.LOW)
        while GPIO.input(PIN_ECHO)==0:
            pulse_start_time = time.time()
        while GPIO.input(PIN_ECHO)==1:
          pulse_end_time = time.time()
        pulse_duration = pulse_end_time - pulse_start_time
        distance = round(pulse_duration * 17150, 2)
        print ("Distance:",distance,"cm")
        print ("Calculating distance")
        print('Sending to ThingSpeak')
        RequestToThingspeak = 'https://api.thingspeak.com/update?api_key=WAEBMZYGGC2QKDXA&field1='
        RequestToThingspeak +=str(distance)
        request = requests.get(RequestToThingspeak)
        print(request.text)
        time.sleep(15)
        print("hello")
```


raspberrypi.org, pimylifeup.com, O'Reilly connecting a push switch with raspberry pi, Adel Kassah, Winfuture.de

```
        if distance < 10:
            if distance > 0:
              now = datetime.datetime.now()
              msg = "When: {} \nExtra Data: {}, {}, {}".format(now.strftime("%Y-%m-%d %H:%M:%S"), distance, distance,
distance)
              send_message_to_telegram(msg)
              print(distance)

finally:
    GPIO.cleanup()
```

Lab report:

Activity 1

a) Revise the code above to include ThingSpeak graphs for 4 ultrasonic and 4 Raspberry Pi placed at different locations. Discuss and summarize your findings thoroughly with sufficient analysis on how 4 different ultrasonics and 4 Raspberry Pis could be used to collect different data efficiently.
b) Using our previous code on ultrasonic sensor distance detection, design an efficient system to detect if someone is near the ultrasonic sensor. If yes, send a notification to telegram. Add extra features to enhance efficient usability.
c) Write your code in the space given below with comments to explain the code. Print screen the output/results you obtained from ThingSpeak and telegram. A comprehensive flow chart should be presented to explain the process of the code using the output/results. In-depth analysis and results should be presented.

[6 marks]

Lab 1 Demonstration

Performs acts with increasing efficiency, confidence and proficiency using IoT Tools

Develop practical skills in the IoT domain related to networking and data analytics using industry standard IoT tools.

[4 marks]

raspberrypi.org, pimylifeup.com, O'Reilly connecting a push switch with raspberry pi, Adel Kassah, Winfuture.de