**LAB Activity 2: IoT Security Camera [5 marks]**

Group Number:

Group Members (Name and ID):

1.

2.

3.

4.

Activity 1: Setting with IoT Security Camera

Setting up Raspberry Pi with Camera



Figure 1: Connection of Raspberry Pi Camera to Raspberry Pi.

1. Ensure your Raspberry Pi is turned off.
2. Locate the Camera Module port
3. Gently pull up on the edges of the port’s plastic clip
4. Insert the Camera Module ribbon cable; make sure the cable is the right way round
5. Push the plastic clip back into place
6. Start up your Raspberry Pi.
7. Go to the main menu and open the Raspberry Pi Configuration tool.

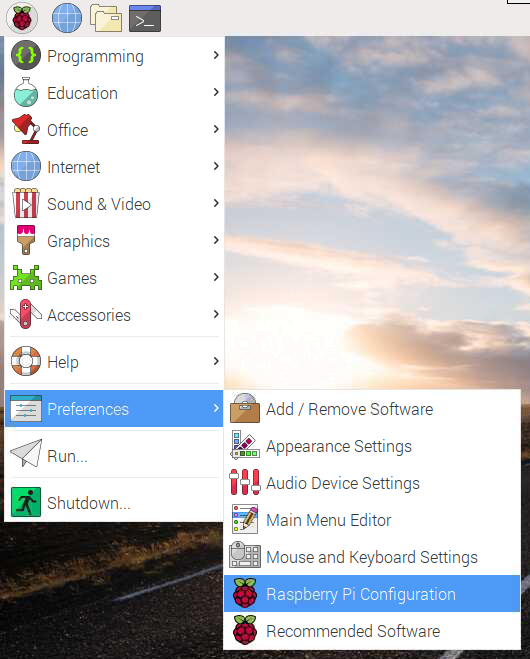


Figure 2: Open the Raspberry Pi Configuration via the main menu.

1. Select the Interfaces tab and ensure that the camera is enabled:

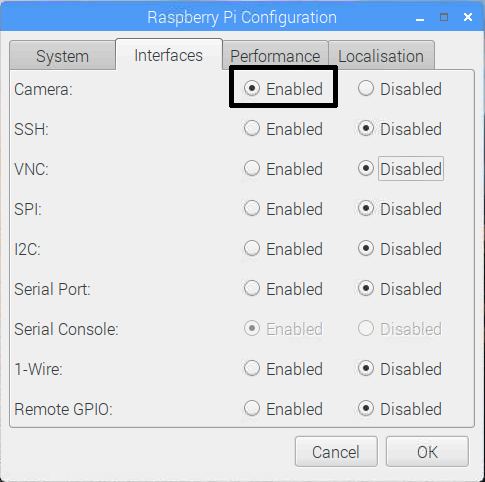


Figure 3: Ensure camera interface is enabled.

1. Reboot your Raspberry Pi.
2. Now your Camera Module is connected and the software is enabled, try out the command line tools raspistill and raspivid.
3. Open a terminal window by clicking the black monitor icon in the taskbar:

Open terminal

Figure 4: Open the terminal window (black monitor icon).

1. Type in the following command to take a still picture and save it to the Desktop:

raspistill -o Desktop/image.jpg

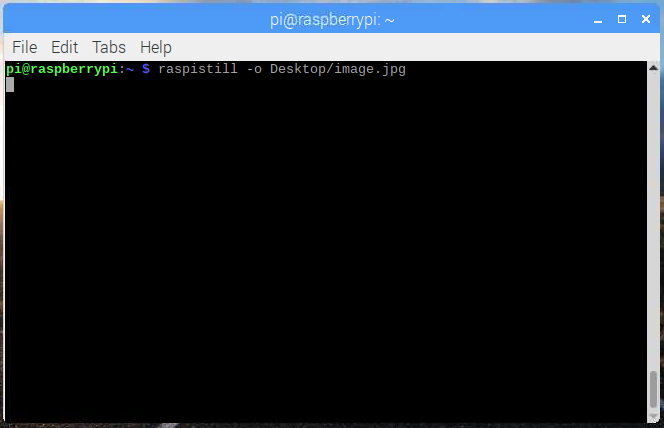


Figure 5: Run command in the terminal window to capture image.

1. Press Enter to run the command. When the command runs, you can see the camera preview open for five seconds before a still picture is taken.
2. Look for the picture file icon on the Desktop and double-click the file icon to open the picture.

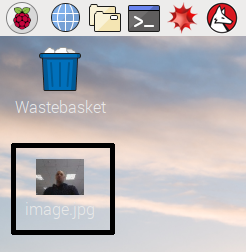


Figure 6: Double-click the image saved on the desktop.

1. By adding different options, you can set the size and look of the image the raspistill command takes.
2. For example, add -h and -w to change the height and width of the image:

raspistill -o Desktop/image-small.jpg -w 640 -h 480

1. Now record a video with the Camera Module by using the following raspivid command:

raspivid -o Desktop/video.h264

1. In order to play the video file, double-click the video.h264 file icon on the Desktop to open it in VLC Media Player.

Signing Up with Telegram and IFTTT

1. Sign up for a Telegram account using your phone or other devices, which you must do via the service's Android or iOS app. Then download and install the Telegram desktop app on your PC or Mac and keep it running. You will need it to make the connection work with IFTTT.

2. Visit https://ifttt.com and sign up for a free account. IFTTT stands for IF This, Then That, and it is used to trigger actions based on inputs. In this project we will create a webhook, a real-time link from our code to the web that will trigger a message on Telegram that alerts us about an intruder. Leave the browser window open.

3. Connect your Telegram account to your IFTTT account. In Telegram desktop app, search for IFTTT and select it.

Software Setup

1. Sign up for a Telegram account, which you must do via the service's Android or iOS app. Then download and install the Telegram desktop app on your PC or Mac and keep it running. You will need it to make the connection work with IFTTT.

2. Visit https://ifttt.com and sign up for a free account. IFTTT stands for IF This, Then That, and it is used to trigger actions based on inputs. In this project we will create a webhook, a real-time link from our code to the web that will trigger a message on Telegram that alerts us about an intruder. Leave the browser window open.

3. Connect your Telegram account to your IFTTT account. In Telegram desktop app, search for IFTTT and select it.

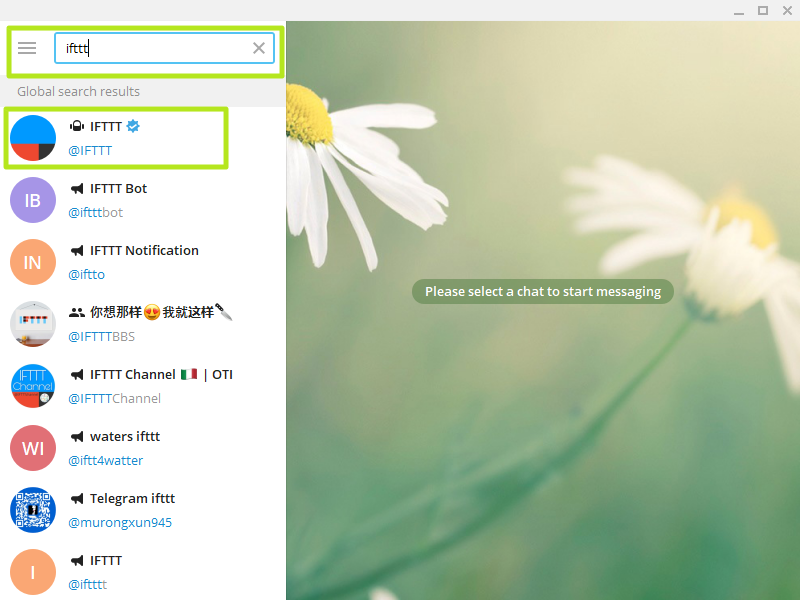


Figure 7: Select the IFTTT chat in Telegram.

4. Click the Start button.

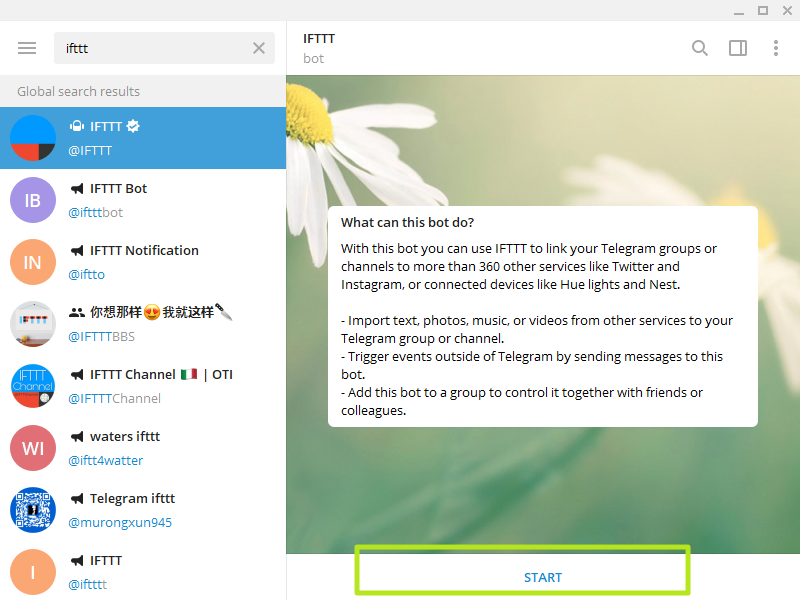


Figure 8: Click ‘Start’ in the IFTTT chat.

5. Then click the https://iftttt.com/telegram link in Telegram which will open your browser to the Telegram page on IFTTT.

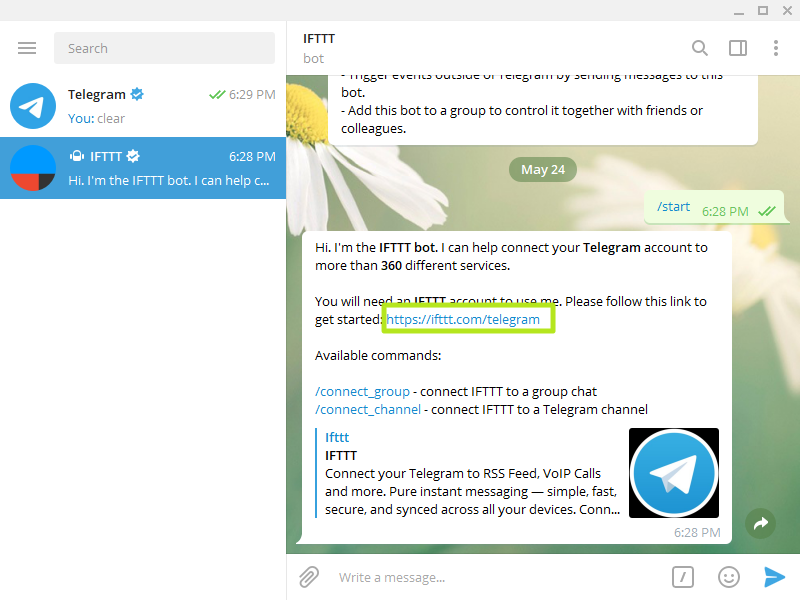


Figure 9: Click on the link to Telegram’s webpage.

6. Click Connect.

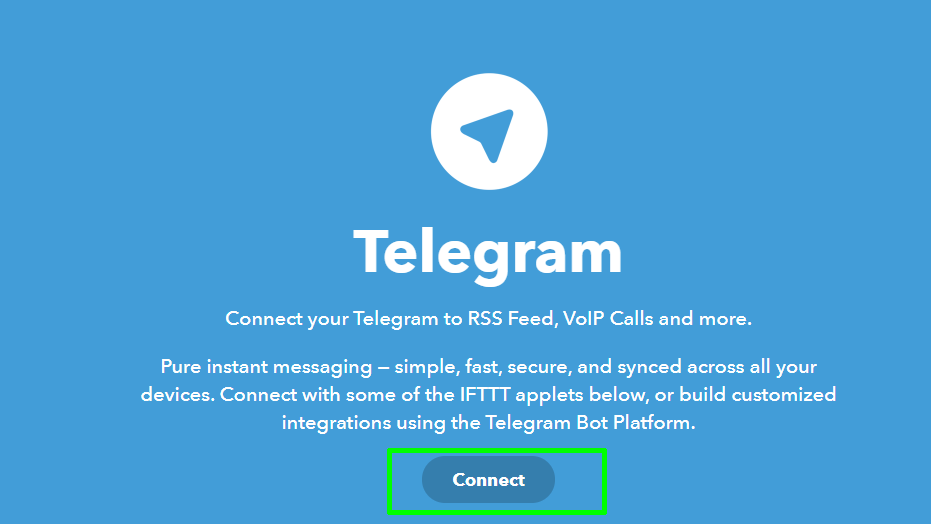


Figure 10: Click on ‘Connect’ on the Telegram webpage.

7. Then click Send Message and allow your browser to Open Telegram Desktop.

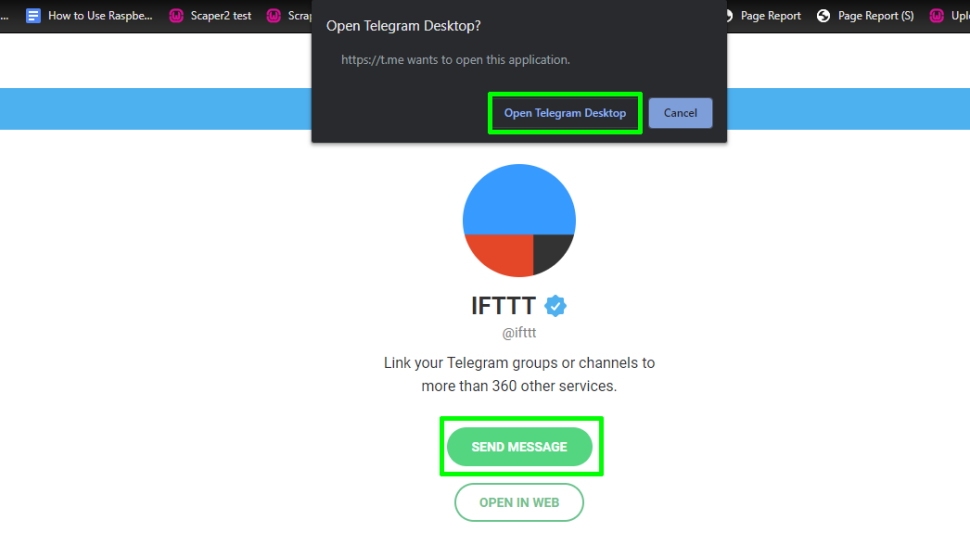


Figure 11: Click on ‘Send Message’ to open Telegram Desktop.

8. In Telegram desktop, click "Authorize IFTTT."

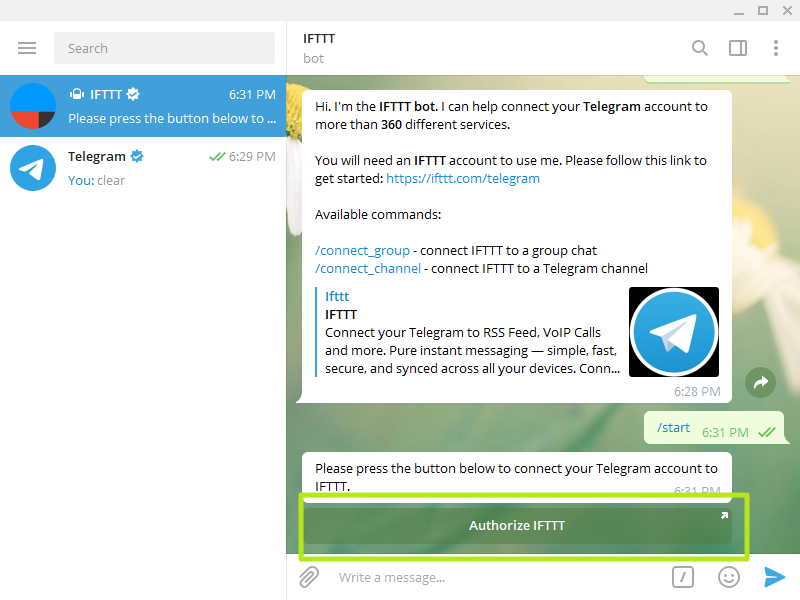


Figure 12: Click ‘Authorize IFTTT’ in the IFTTT chat.

9. Create the trigger in IFTTT by clicking on your account avatar icon in the top right of the screen, then select Create.

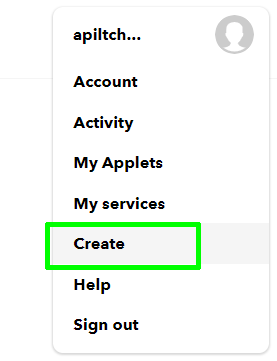


Figure 13: Click ‘Create’ in the drop-down menu after clicking on your profile icon in IFTTT.

10. Now click on If +.

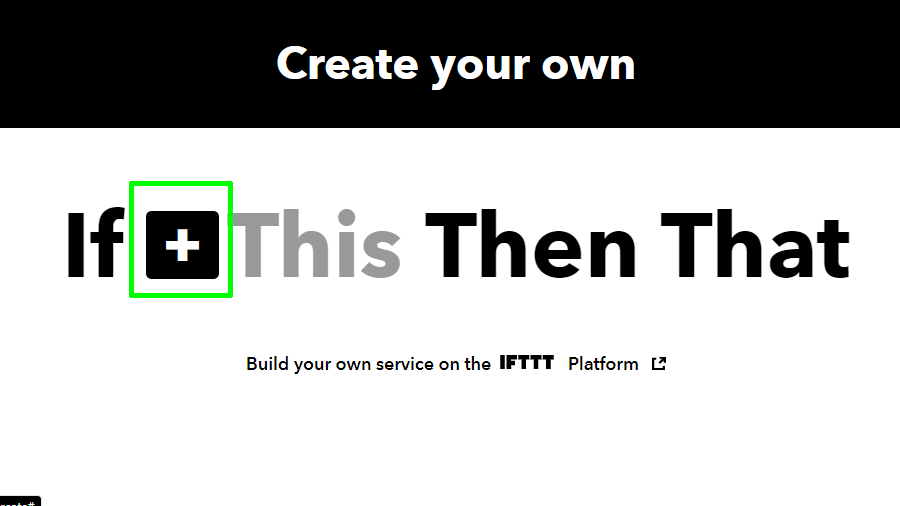


Figure 14: Click on the first ‘+’ symbol.

11. Type in webhook and click on the blue icon.

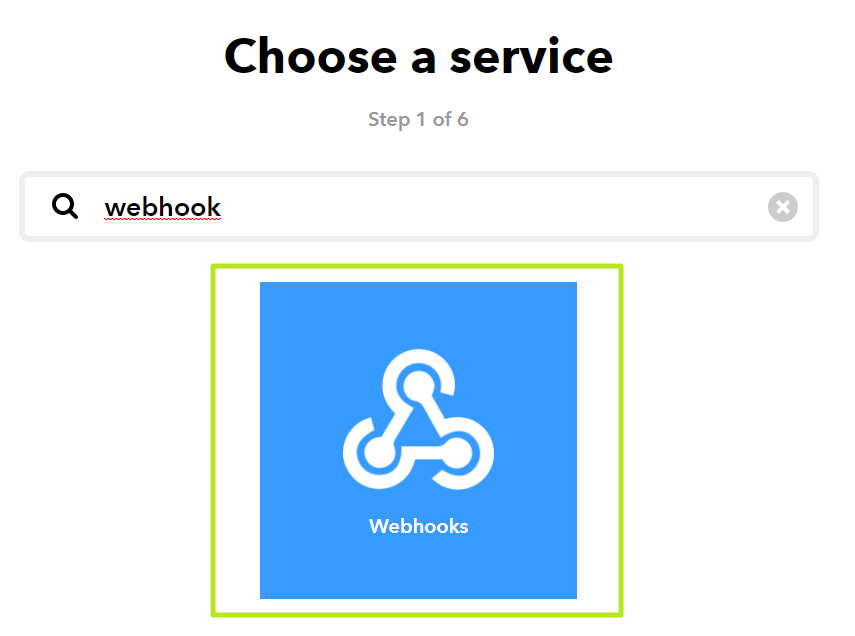


Figure 15: Select ‘Webhooks’.

12. On the next screen, click on Receive A Web Request.

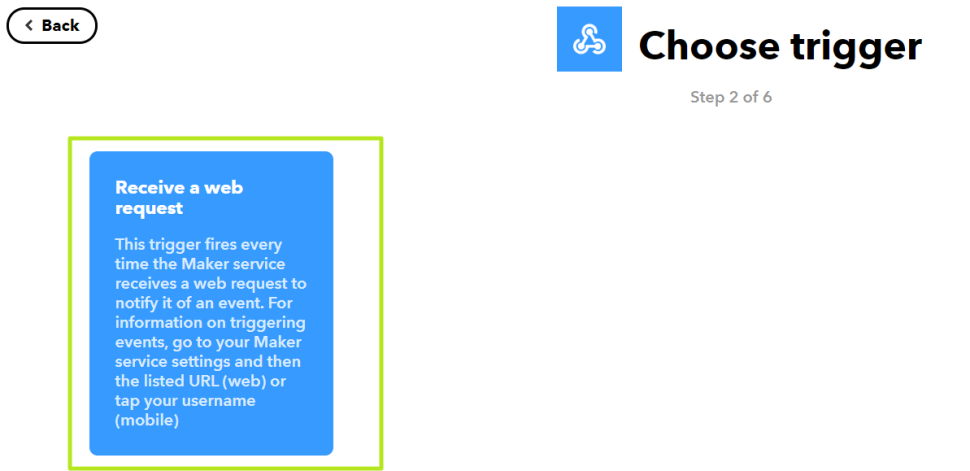


Figure 16: Select ‘Received a web request’.

13. Name the event “trigger”, without the quotation marks and click Create trigger.



Figure 17: Create a trigger named ‘trigger’.

14. Click on Then +.



Figure 18: Click on the second ‘+’ symbol.

15. Search for Telegram then select it.

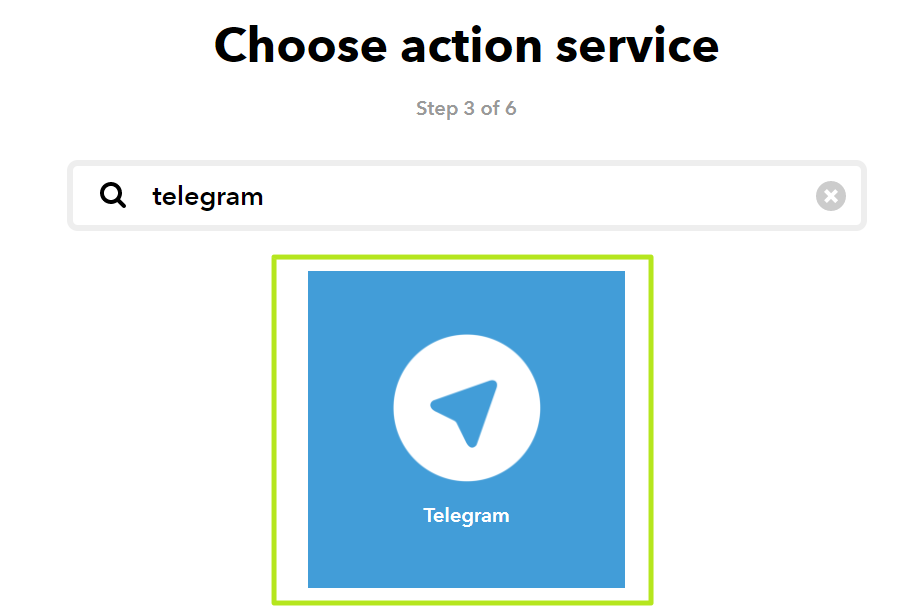


Figure 19: Select Telegram.

16. Click on Send Photo as the action.

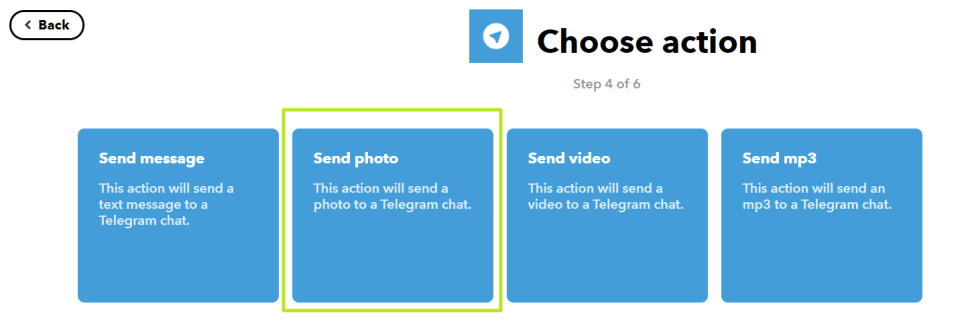


Figure 20: Select ‘Send photo’.

17. On the next screen, set target chat to "Private chat with @IFTTT" if it's not set to that already. In the Photo URL section click on Add Ingredient and select Value1. Ensure that Value1 has a capital V. In the Caption section, change the caption to “Movement detected” and then click Create Action.



Figure 21: Fill in the appropriate information as shown.

18. Click Finish to create and start the service.

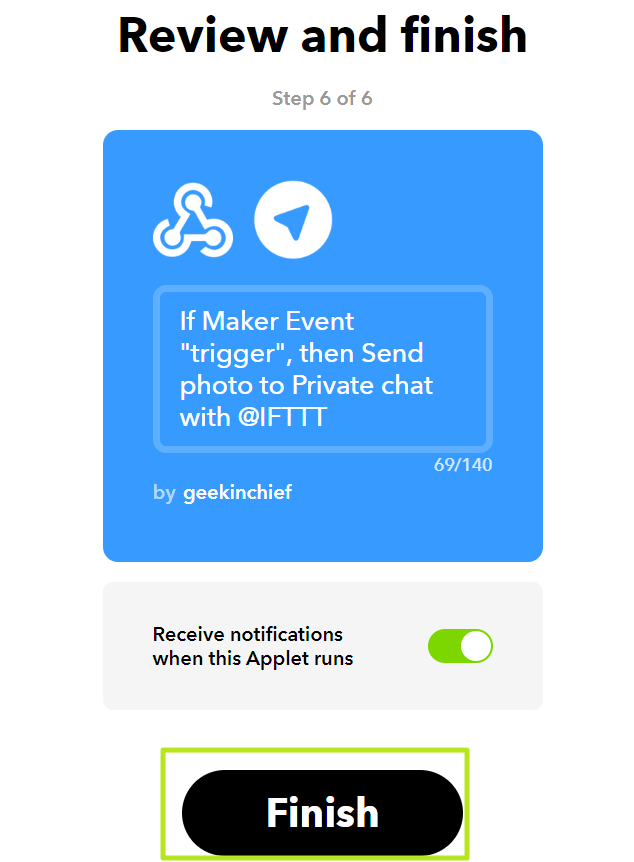


Figure 22: Click ‘Finish’ to complete the service.

19. Now click on Home, and then click Webhooks.

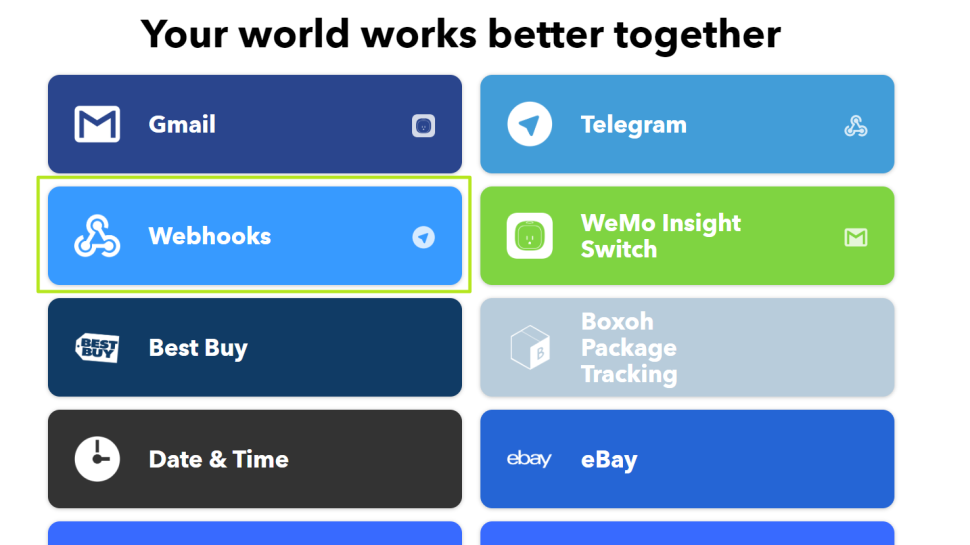


Figure 23: Go to Home and select ‘Webhooks’.

20. On the next screen, click Documentation and you will see how the webhook is constructed.

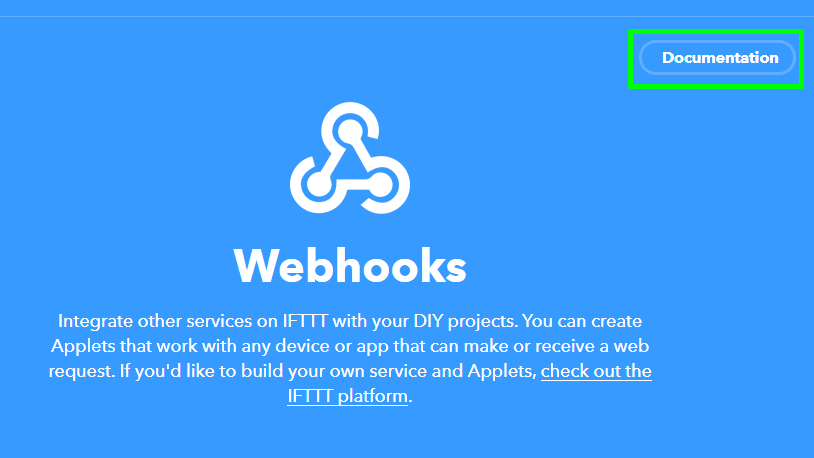


Figure 24: Click on ‘Documentation’ to obtain trigger link.

21. In the Make a POST or GET section, click on {event} and change it to trigger. Now, on the bottom line, which starts curl -X POST, select from https to the end of the line, then copy the link and save it in a notepad for later use. When you’re done, click < Back to service.



Figure 25: Copy the trigger link as shown.

22. Create an account with www.filestack.com, which will be used to store images uploaded from the Pi. These images are then attached to the Telegram message. Sign up for a free account, and when logged in to the Dashboard your API key will be in the top right corner. We will need this later.

Graphical user interface, application, website

Description automatically generated

Figure 26: Copy API key shown in the top right corner of Filestack dashboard.

23. Install the filestack library. To use filestack with Python we need to install the library. In a terminal type the following and then press Enter to run:

sudo pip3 install filestack-python

Coding the Motion Detection Project

1. To write the Python code for this project, choose your favorite Python 3 editor – we used Thonny – and create a new file called trigger.py. Remember to save very regularly.

The first block of code consists of the imports for the libraries that will make up the project. The ‘requests’ library is used to send a webhook to IFTTT. The ‘RPi’ library is used to define the pin numbers of the Raspberry Pi. The ‘picamera’ library is used to control the camera. The ‘sleep()’ method from ‘time’ is used to pause the program. The ‘filestack’ library is used to upload camera images to Filestack.

import requests

import time

import RPi.GPIO as GPIO

from picamera import PiCamera

from time import sleep

from filestack import Client

2. Next, we will need to create an object called client, which will contain a link to Filestack, including our API Key.

client = Client("YOUR\_API\_KEY")

3. We now create a PiCamera object and set the resolution of the camera to 1920x1080.

camera = PiCamera()

camera.resolution = (1920, 1080)

4. The next step is to create a function called send\_alert() which contains all of the steps necessary to send an image via webhooks to IFTTT and then to Telegram. When the function is called, an image will be captures and saved as ‘image.jpg’. The image file is then uploaded to Filestack and saved in the object ‘new\_filelink’. The URL for the uploaded image is printed to the Python shell for debugging purposes.

5. To send the image to IFTTT, the ‘requests’ library is utilized to create an HTTP POST request with the Webhook address that we have previously noted down. This link contains our trigger word, ‘trigger’ and our API key for IFTTT. A JSON body containing ‘value 1’ is expected by the POST request. This should contain the URL of the image that was previously uploaded to Filestack. The URL of the image can be accessed as the ‘url’ property of the ‘new\_filelink’ object (new\_filelink.url).

6. To handle any errors when sending the image, the status code returned by the request is returned. The request is successful of the status code returned is 200. Otherwise, an error has occurred, and an error will be triggered.

def send\_alert():

camera.capture("image.jpg")

new\_filelink = client.upload(filepath="image.jpg")

print(new\_filelink.url)

r = requests.post("https://maker.ifttt.com/trigger/trigger/with/key/YOUR\_API\_KEY",

json = {"value1": new\_filelink.url})

if r.status\_code == 200:

print("Alert Sent")

else:

print("Error")

7. After the function is completed, setup the ultrasonic sensor similarly to Lab 1. Define the threshold distance that will trigger the system. Define a constant that will be used for the infinite loop of the main code.

GPIO.setmode(GPIO.BOARD)

PIN\_TRIGGER = YOUR\_PIN\_TRIGGER

PIN\_ECHO = YOUR\_PIN\_ECHO

GPIO.setup(PIN\_TRIGGER, GPIO.OUT)

GPIO.setup(PIN\_ECHO, GPIO.IN)

threshold = 10

systemStart = True

8. An infinite loop is created so that the ultrasonic sensor constantly returns the distance of the object in front of it. If the detected distance falls below a threshold, the ‘send\_alert()’ function will be called. Then, the code will be paused until a prompt is received.

try:

while systemStart:

GPIO.output(PIN\_TRIGGER, GPIO.LOW)

print("Waiting for sensor to settle.")

time.sleep(2)

print("Calculating distace")

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")

if distance < threshold:

if distance > 0:

send\_alert()

sleep(1)

else:

continue

finally:

camera.close()

GPIO.cleanup()

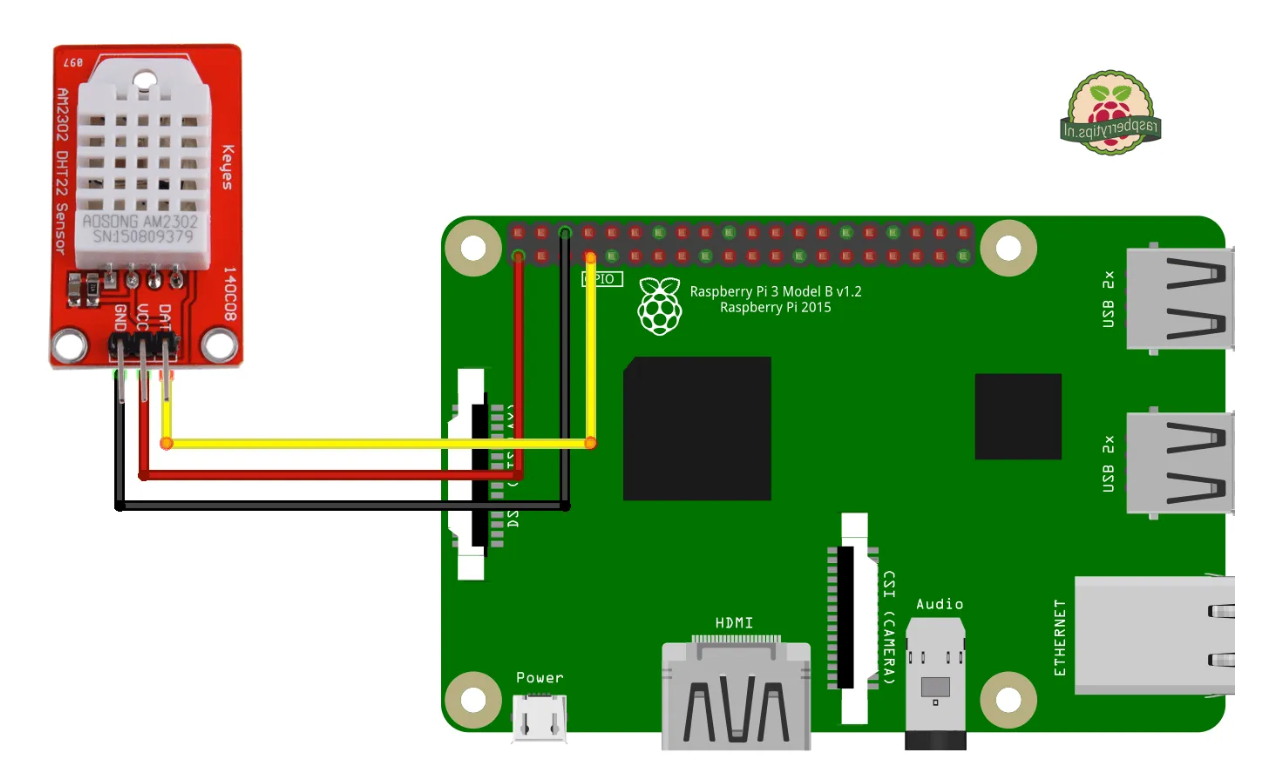
8. Save and run the completed code.

Activity 2 [To be Included in report – 2 marks]

Using our previous code on ultrasonic sensor distance detection, design an efficient system to detect if someone is near the ultrasonic sensor within 15 minutes (use loop). If yes, capture an image and send to telegram. Add extra features to enhance efficient usability. Write your code in the space given below with comments to explain the code. Print screen the output/results you obtained from thingspeak, telegram and filestack. 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.

Activity 3:

1. Connect the AM2302 (temperature and humidity sensor) according to the diagram below:
2. Sensor Pin VCC -> Raspberry Pi pin (1) 3.3v
3. Sensor Pin DAT -> Raspberry Pi pin (7) GPIO 4 **(if you have used pin (7) GPIO 4 previously for ultrasonic sensor, you can connect to pin (12) GPIO 18)**
4. Sensor Pin GND -> Raspberry Pi pin (6) Ground



1. In the codes below, we will use pin (12) GPIO 18.
2. Software Setup

To start with update your package lists and install a few Python libraries:

sudo apt-get update

sudo apt-get install build-essential python-dev

Then clone the Adafruit library from their repository:

sudo git clone https://github.com/adafruit/Adafruit\_Python\_DHT.git

cd Adafruit\_Python\_DHT

Then install the library for Python 2 and Python 3:

sudo python setup.py install

sudo python3 setup.py install

Hopefully at this point the library is installed and ready to be used within a Python script.

1. Adafruit Example Python Script

Adafruit provide an example script that you can use to check your sensor is operating correctly.

cd ~

cd Adafruit\_Python\_DHT

cd examples

Then, run the following Python script:

python AdafruitDHT.py 2302 18

1. The example script takes two parameters. The first is the sensor type so is set to “2302” to represent the AM 2302. The second is the GPIO number so for my example I am using “18” for GPIO18 pin. You can change this if you are using a different GPIO pin for your data/out wire.
2. You should see an output similar to this:



Figure 27: Readings printed out in the console.

1. There may also be the possibility that the console does not show you any readings after running the script. After some time, the following error will be printed out.

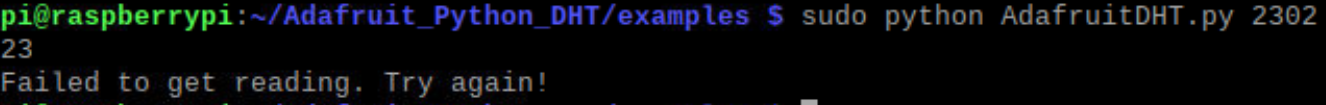


Figure 28: Error message printed out after running the script.

If this occurs, open the File Manager of your Raspberry Pi and open the ‘Adafruit\_Python\_DHT’ folder that we have just downloaded.

Graphical user interface, application

Description automatically generated

Figure 29: Open the recently downloaded ‘Adafruit\_Python\_DHT’ folder.

Then, open the ‘Adafruit\_DHT’ folder and open the ‘platform\_detect.py’ Python script.

Graphical user interface, application, PowerPoint

Description automatically generated

Figure 30: Open the ‘Adafruit\_DHT’ folder.

Graphical user interface, application

Description automatically generated

Figure 31: Open the ‘platform\_detect.py’ Python script.

Scroll down the script and look for the ‘pi\_version()’ function.

Graphical user interface, text, application, email

Description automatically generated

Figure 32: Look for the function called ‘pi\_version()’.

Scroll down the function and you should see an ‘if…else…’ statement as shown in Figure 33.

Graphical user interface, text, application, email

Description automatically generated

Figure 33: ‘If…else…’ statement in ‘pi\_version()’ function.

Add an additional ‘elif’ statement that checks for the condition where ‘match.group(1) == ‘BCM2711’ and returns 3 if the condition is fulfilled as shown in Figure 33.

Graphical user interface, text, application, email

Description automatically generated

Figure 34: Add an ‘elif’ statement that checks for ‘BCM2711’ and returns 3 if it is true.

Save the file after the modifications are implemented. There may be instances where the following error is shown when you attempt to save the changes.

Graphical user interface, application

Description automatically generated

Figure 35: Error printed when changes are to be saved.

To solve this, open the console terminal. And run the following commands.

cd ~

cd Adafruit\_Python\_DHT/Adafruit\_DHT

sudo chmod 777 platform\_detect.py

After this is done, you should be able to save the changes you have made. After the changes are saved, run the following command in the console terminal.

sudo chmod 644 platform\_detect.py

cd ~

cd Adafruit\_Python\_DHT

sudo python setup.py install

sudo python3 setup.py install

Run the script again as described in step 4. You should be able to see the appropriate results being printed out.

1. Create a python file and run the following:

import Adafruit\_DHT

# Set sensor type : Options are DHT11,DHT22 or AM2302

sensor=Adafruit\_DHT.AM2302

# Set GPIO pin that temperature & humidity sensor is connected to

GPIO=18

# Use read\_retry method. This will retry up to 15 times to

# get a sensor reading (waiting 2 seconds between each retry).

humidity, temperature = Adafruit\_DHT.read\_retry(sensor, GPIO)

# Reading the DHT11 is very sensitive to timings and occasionally

# the Pi might fail to get a valid reading. So, check if readings are valid.

if humidity is not None and temperature is not None:

print('Temp={0:0.1f}\*C Humidity={1:0.1f}%'.format(temperature, humidity))

else:

print('Failed to get reading. Try again!')

Activity 4: [To be included in report – 2 marks]

Design a loop and collect around 30 data on temperature and humidity within 15 minutes using thingspeak. Based on the data obtained from thingspeak, send a text to telegram on the average temperature and humidity after 15 minutes. Write your code and comment on the code to explain each line/block of code. Show the results from thingspeak and analyse the results comprehensively. Print screen the output/results you obtained from thingspeak and telegram. Present the whole code using comprehensive flow chart. Observe the trend in the data obtained from ThingSpeak. If you need more data, collect more. Explain the trend of the temperature and humidity results collected. If there are any noise in the data, explain why and how to further improve. Explain whether using this data you can predict the next temperature and humidity in the next time frame. Explain your analysis. Add extra features to enhance the efficient usability.

Activity 5: Lab 1 Demonstration [1 mark]

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.