Display LED in Pattern

Pin Diagram:

Name	Pin number
GPIO	40
GPIO	37
GPIO	35
GPIO	33
GND	6

Step 1:connect the led’s using the pin connection table

Step 2:open the terminal

Step 3:open the file

```
Sudo nano led.py
```

Step 4:write the code

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(37,GPIO.OUT)
GPIO.setup(35,GPIO.OUT)
GPIO.setup(33,GPIO.OUT)
GPIO.setup(31,GPIO.OUT)
while True:
        GPIO.output(37,True)
        GPIO.output(35,True)
        GPIO.output(33,True)
        GPIO.output(31,False)
        time.sleep(1)
        GPIO.output(37,False)
     GPIO.output(35,True)
     GPIO.output(33,True)
     GPIO.output(31,True)
     time.sleep(1)
        GPIO.output(37,True)
     GPIO.output(35,False)
     GPIO.output(33,True)
     GPIO.output(31,True)
     time.sleep(1)
        GPIO.output(37,True)
     GPIO.output(35,True)
     GPIO.output(33,False)
     GPIO.output(31,True)
     time.sleep(1)
```

step 5:Run the code

```
sudo  python led.py
```

Displaying Time over 4-Digit 7-Segment Display using Raspberry Pi

Pin diagram

Name	Pin number
VCC	2
GND	6
DIN	38
CLK	40

step 1 : download the zip file form the link
https://github.com/timwaizenegger/raspberrypi-examples/tree/master/actor-led-7segment-4numbers

step 2 : extract the zip file to the desktop

step 3 : open terminal

step 4 : **cd Desktop/**

step 5 : **cd actor-led-7segment-4numbers/**

step 6 : **sudo python clock.py**

code for clock.py

```
#!/usr/bin/env python

# -*- coding: utf-8 -*-


from time import sleep

import tm1637


try:

    import thread

except ImportError:
    import _thread as thread


# Initialize the clock (GND, VCC=3.3V, Example Pins are DIO-20 and CLK21)
Display = tm1637.TM1637(CLK=21, DIO=20, brightness=1.0)


try:
    print "Starting clock in the background (press CTRL + C to stop):"
    Display.StartClock(military_time=False)
    print 'Continue Python script and tweak Display!'
    sleep(5)
    Display.ShowDoublepoint(False)
    sleep(5)
    loops = 3
    while loops > 0:
      for i in range(0, 10):
        Display.SetBrightness(i / 10.0)
        sleep(0.5)
      loops -= 1
    Display.StopClock()
    thread.interrupt_main()
except KeyboardInterrupt:
    print "Properly closing the clock and open GPIO pins"
    Display.cleanup()
```

Raspberry Pi Based Oscilloscope

https://circuitdigest.com/microcontroller-projects/raspberry-pi-based-oscilloscope

https://github.com/adafruit/Adafruit_Python_ADS1x15

Name	Pin number
VDD	2
GND	6
SDA	3
SCL	5

Step 1: Enable Raspberry Pi I2C interface

sudo raspi-config

Step 2: Install the Adafruit ADS1115 library for ADC

sudo apt-get install build-essential python-dev python-smbus git

step 3 : download the Adafruit ADS fzip from git hub

https://github.com/adafruit/Adafruit_Python_ADS1x15.git

step 4 : Extact the file to desktop

step 5 : change the directory

cd Desktop

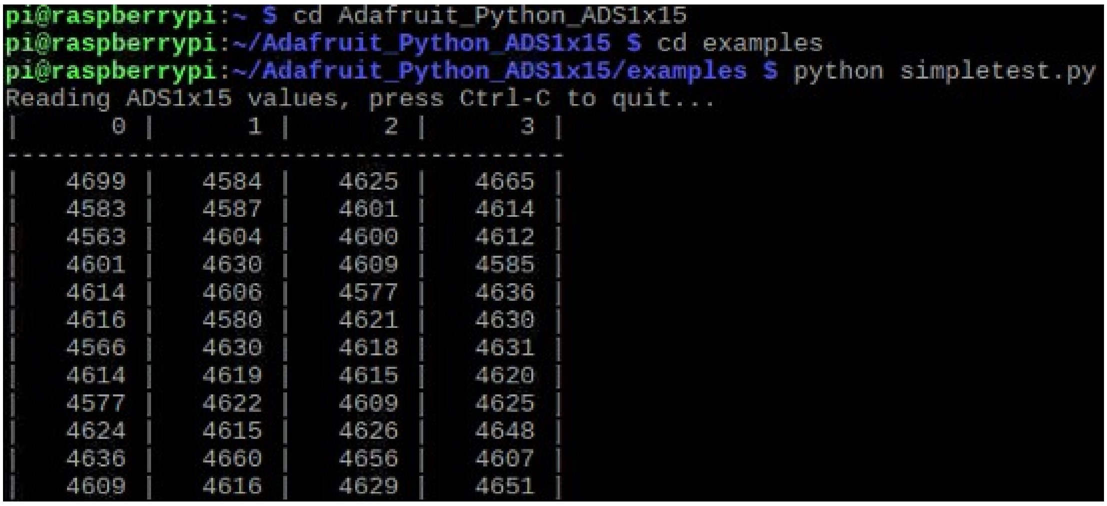
cd Adafruit_Python_ADS1x15-master

sudo python setup.py install

step 6 : Test the library and 12C communication.

cd examples

python simpletest.py



```
pi@raspberrypi:~ $ cd Adafruit_Python_ADS1x15
pi@raspberrypi:~/Adafruit_Python_ADS1x15 $ cd examples
pi@raspberrypi:~/Adafruit_Python_ADS1x15/examples $ python simpletest.py
Reading ADS1x15 values, press Ctrl-C to quit...
|      0 |      1 |      2 |      3 |
-------------------------------------
|   4699 |   4584 |   4625 |   4665 |
|   4583 |   4587 |   4601 |   4614 |
|   4563 |   4604 |   4600 |   4612 |
|   4601 |   4630 |   4609 |   4585 |
|   4614 |   4606 |   4577 |   4636 |
|   4616 |   4580 |   4621 |   4630 |
|   4566 |   4630 |   4618 |   4631 |
|   4614 |   4619 |   4615 |   4620 |
|   4577 |   4622 |   4609 |   4625 |
|   4624 |   4615 |   4626 |   4648 |
|   4636 |   4660 |   4656 |   4607 |
|   4609 |   4616 |   4629 |   4651 |
```

Step 6: Install Matplotlib

```
sudo apt-get install python-matplotlib

sudo apt-get install python-pip

sudo pip install drawnow

sudo nano scope.py
```

code for scope.py

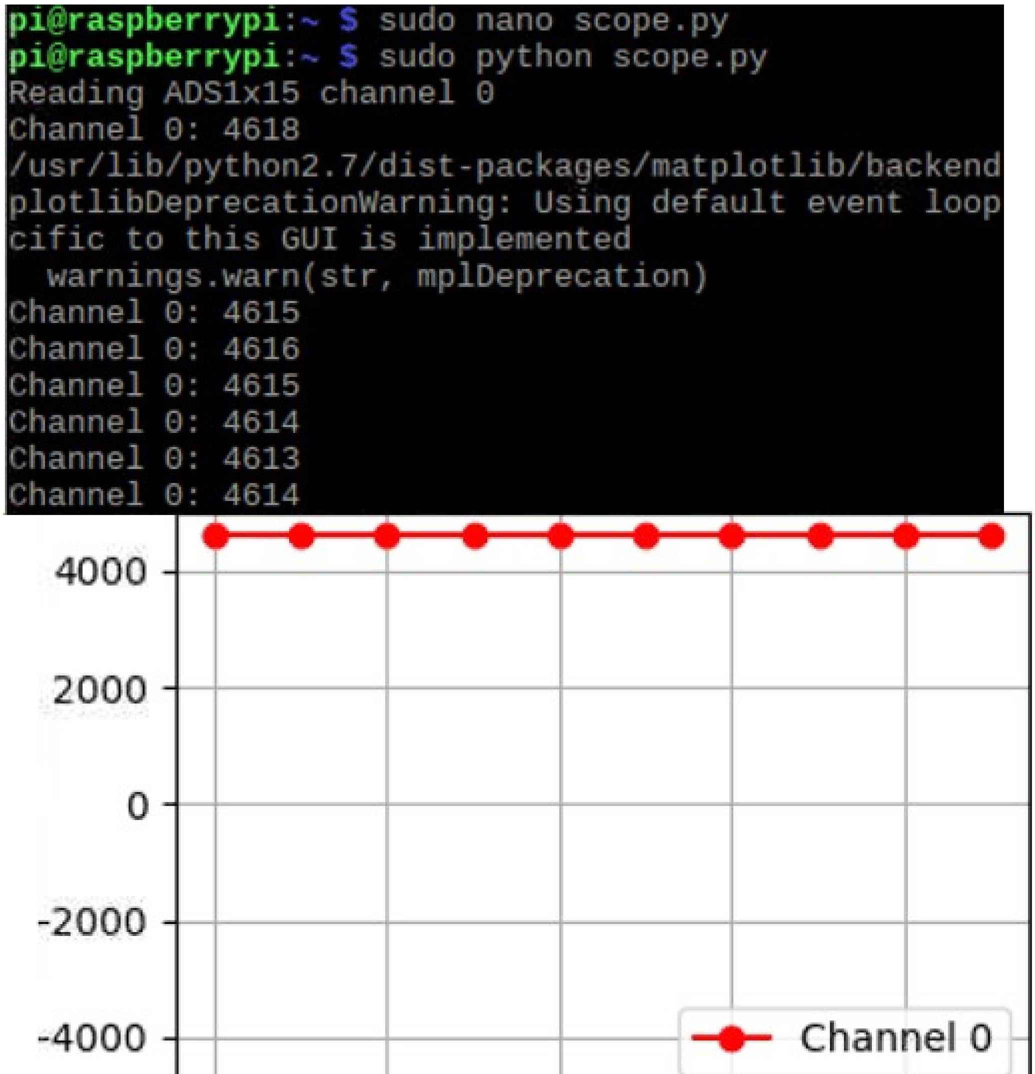
```
import time

import matplotlib.pyplot as plt

#import numpy

from drawnow import *

# Import the ADS1x15 module.

import Adafruit_ADS1x15

# Create an ADS1115 ADC (16-bit) instance.

adc = Adafruit_ADS1x15.ADS1115()


GAIN = 1

val = [ ]

cnt = 0

plt.ion()

# Start continuous ADC conversions on channel 0 using the previous gain value.

adc.start_adc(0, gain=GAIN)

print('Reading ADS1x15 channel 0')

#create the figure function

def makeFig():

    plt.ylim(-5000,5000)

    plt.title('Osciloscope')

    plt.grid(True)

    plt.ylabel('ADC outputs')

    plt.plot(val, 'ro-', label='Channel 0')

    plt.legend(loc='lower right')
```

```
while (True):
    # Read the last ADC conversion value and print it out.
    value = adc.get_last_result()
    print('Channel 0: {0}'.format(value))
    # Sleep for half a second.
    time.sleep(0.5)
    val.append(int(value))
    drawnow(makeFig)
    plt.pause(.000001)
    cnt = cnt+1
    if(cnt>50):
        val.pop(0)
```

step 7 : save and exit the file

step 8 : run the scope.py

sudo python scope.py



```
pi@raspberrypi:~ $ sudo nano scope.py
pi@raspberrypi:~ $ sudo python scope.py
Reading ADS1x15 channel 0
Channel 0: 4618
/usr/lib/python2.7/dist-packages/matplotlib/backend
plotlibDeprecationWarning: Using default event loop
cific to this GUI is implemented
  warnings.warn(str, mplDeprecation)
Channel 0: 4615
Channel 0: 4616
Channel 0: 4615
Channel 0: 4614
Channel 0: 4613
Channel 0: 4614
```

Controlling Raspberry Pi with Telegram

https://www.hackster.io/Salmanfarisvp/telegram-bot-with-raspberry-pi-f373da

Step 1 : Open Telegram app in your system or mobile

Step 2 : Start "BotFather"

Step 3 : Open "BotFather"

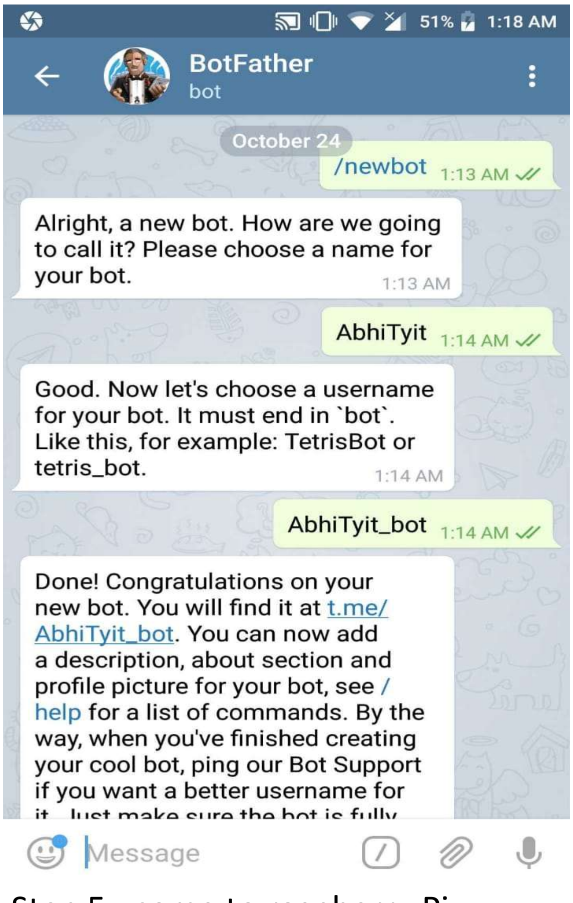
Step 4 : Create a new Bot

/new bot

Username : AbhiTyit

Bot name : AbhiTyit_bot

You will get the token number hightlighted there in given below picture



/newbot 1:13 AM

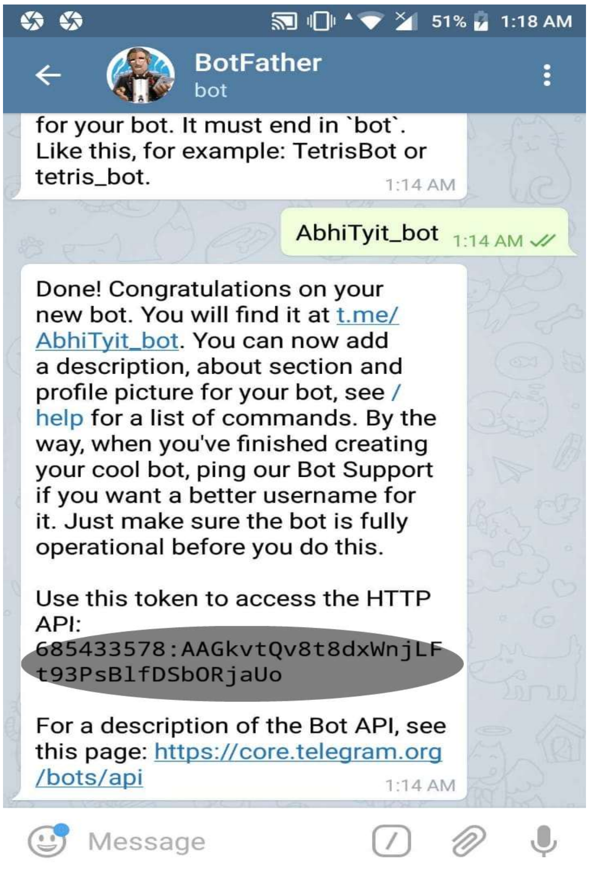
Alright, a new bot. How are we going to call it? Please choose a name for your bot. 1:13 AM

AbhiTyit 1:14 AM

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 1:14 AM

AbhiTyit_bot 1:14 AM

Done! Congratulations on your new bot. You will find it at t.me/AbhiTyit_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully



for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 1:14 AM

AbhiTyit_bot 1:14 AM

Done! Congratulations on your new bot. You will find it at t.me/AbhiTyit_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
685433578:AAGkvtQv8t8dxWnjLFt93PsBlfDSbORjaUo

For a description of the Bot API, see this page: https://core.telegram.org/bots/api 1:14 AM

Step 5 : come to raspberry Pi

Step 6 : open terminal

Step 7 : install telepot

```
sudo pip install telepot
```

step 8 : copy the code from the link

```
git clone https://github.com/salmanfarisvp/TelegramBot.git
```

step 9 : open the python file

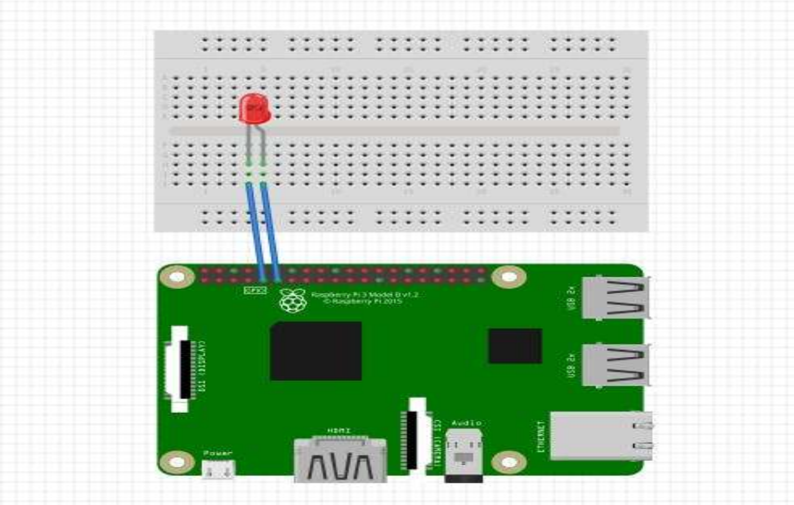
```
sudo nano telegrambot.py
```

step 10 : copy ur bot token in the line

```
bot = telepot.Bot('Bot Token')
```

step 11 : save the file and exit

step 12 :connect the led to the pi

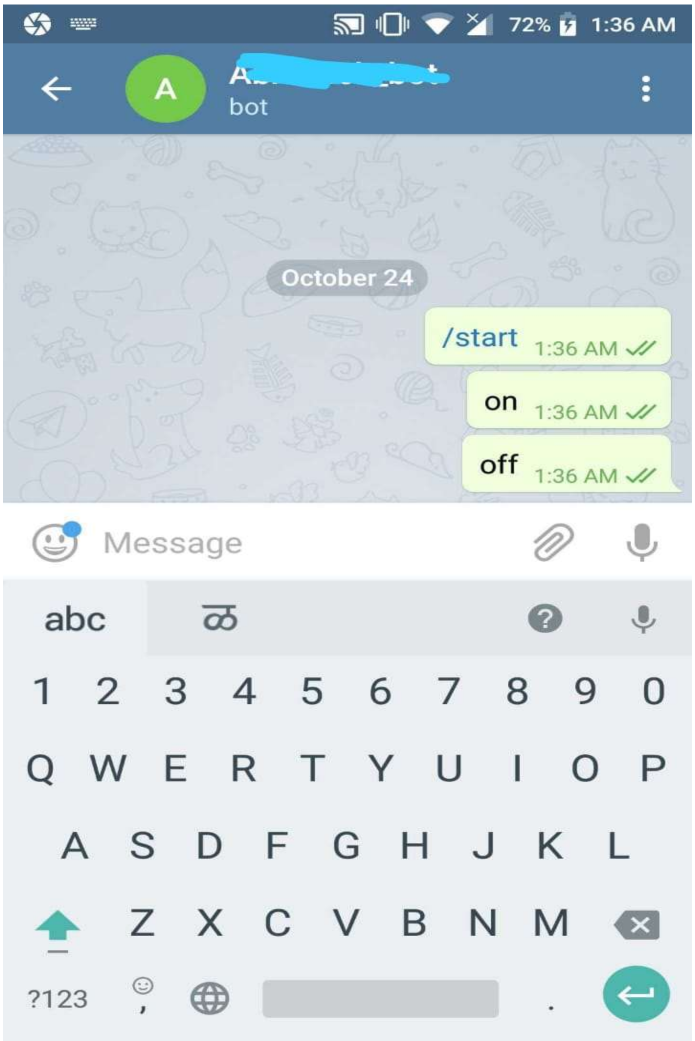


Name	Pin Number
3.3V	11
GND	6

step 13: run the python file

```
sudo python telegrambot.py
```

step 14 : send you command through your bot



[redacted]
bot

October 24

/start 1:36 AM

on 1:36 AM

off 1:36 AM

GPS Module

Pin Diagram:

Gps module pin	Usb port
VCC	VCC
GND	GND
RX	TX
TX	RX

Step 1: **sudo apt-get update**

Step 2: **sudo apt-get install gpsd gpsd-clients python-gps**

Step 3: **sudo systemctl start gpsd.socket**

Step 4: **cgps -s**

Home Automation

Pin Connection:-

Name	Pin number
GPIO	17
GND	6

Step 1: update the raspberry Pi

```
sudo apt-get update
sudo apt-get upgrade
sudo reboot
```

Step 2: Make sure you are in home directory using;

```
cd~
```

Step 3: Use wget to get the file from their source for page

```
wget http://sourceforge.net/projects/webiopi/files/WebIOPi-0.7.1.tar.gz
```

Step 4: When download is done, extract the file and go into the directory

```
tar xvzf WebIOPi-0.7.1.tar.gz
cd WebIOPi-0.7.1/
```

Step 5: install a patch as this version of the WebIOPi

```
wget https://raw.githubusercontent.com/doublebind/raspi/master/webiopi-pi2bplus.patch
patch -p1 -i webiopi-pi2bplus.patch
```

Step 6: we can run the setup installation for the WebIOPi

```
sudo ./setup.sh
```

Step 7: reboot your pi

```
sudo reboot
```

Step 8: test our WebIOPi installation

```
sudo webiopi -d -c /etc/webiopi/config
```

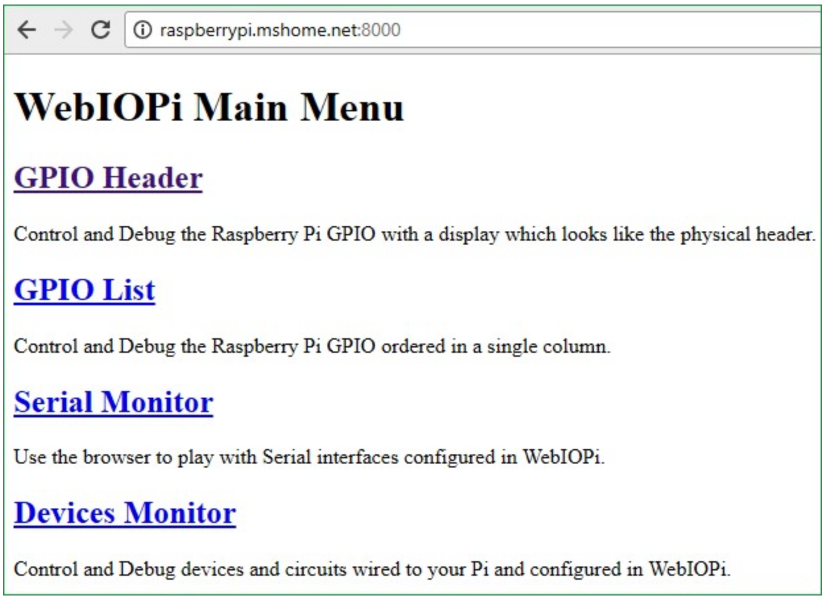
Step 9:web browser connected to the raspberry pi using

http://raspberrypi.mshome.net:8000 or http;//thepi'sIPaddress:8000. The system will prompt you for username and password.

Username is *webiopi*

Password is *raspberry*

Step 10: click on the GPIO header link.



raspberrypi.mshome.net:8000

WebIOPi Main Menu

GPIO Header

Control and Debug the Raspberry Pi GPIO with a display which looks like the physical header.

GPIO List

Control and Debug the Raspberry Pi GPIO ordered in a single column.

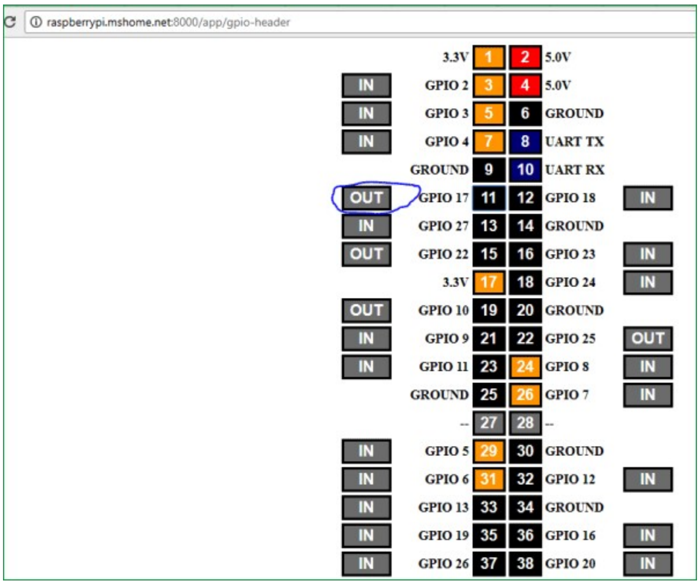
Serial Monitor

Use the browser to play with Serial interfaces configured in WebIOPi.

Devices Monitor

Control and Debug devices and circuits wired to your Pi and configured in WebIOPi.

Step 11: For this test, we will be connecting an LED to GPIO 17, so go on and set GPIO 17 as an output.



raspberrypi.mshome.net:8000/app/gpio-header

	3.3V	1	2	5.0V	
IN	GPIO 2	3	4	5.0V	
IN	GPIO 3	5	6	GROUND	
IN	GPIO 4	7	8	UART TX	
	GROUND	9	10	UART RX	
OUT	GPIO 17	11	12	GPIO 18	IN
IN	GPIO 27	13	14	GROUND	
OUT	GPIO 22	15	16	GPIO 23	IN
	3.3V	17	18	GPIO 24	IN
OUT	GPIO 10	19	20	GROUND	
IN	GPIO 9	21	22	GPIO 25	OUT
IN	GPIO 11	23	24	GPIO 8	IN
	GROUND	25	26	GPIO 7	IN
	--	27	28	--	
IN	GPIO 5	29	30	GROUND	
IN	GPIO 6	31	32	GPIO 12	IN
IN	GPIO 13	33	34	GROUND	
IN	GPIO 19	35	36	GPIO 16	IN
IN	GPIO 26	37	38	GPIO 20	IN

Pi Camera

Components:

1* Raspberry Pi 3

1* Camera

Jumper wires

Step 1:enable the camera

```
sudo raspi-config
```

Step 2: install camera module

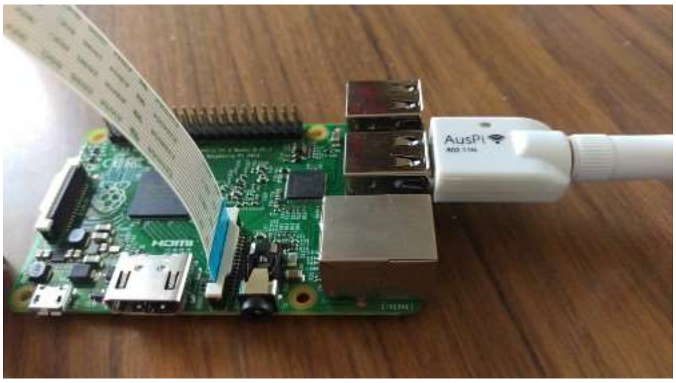
```
sudo apt-get update
```

```
sudo apt-get install python-picamera
```

Step 3: code for capture img

```
raspistill -o filename.jpg
```

Connection:



RFID Card Reading

Connection:

RFID pin	Pi pin
VCC	VCC
RX	TX
GND	GND

Step 1: make new file

```
Sudo nano card.py
```

Step 2: write a code

```
import RPi.GPIO as GPIO
import time
import  Serial
GPIO.setmode(GPIO.BOARD)
greenLED=37
redLED=35
buzzer=33
GPIO.setup(greenLED,GPIO.OUT)
GPIO.setup(redLED,GPIO.OUT)
GPIO.setup(buzzer,GPIO.OUT)
GPIO.output(geenLED,False)
GPIO.output(redLED,False)
GPIO.setup(buzzer,True)
time.sleep(0.1)
GPIO.setup(buzzer,False)
time.sleep(0.1)
GPIO.setup(buzzer,True)
time.sleep(0.1)
GPIO.setup(buzzer,False)
time.sleep(0.1)
```

```
def read_rfid():
        ser=serial.serial("/dev/ttyUSB0")
        value=data.decode("UTF-8")
        ser.baudrate=9600
        data=ser.read(12)
        ser.close()
        return data
Try:
        While True:
                id=read_rfid()
                print(id)
        if id=="400034E165F0":
                print("Access Granted")
                GPIO.output(greenLED,True)
                GPIO.output(redLED,False)
                GPIO.output(buzzer,False)
        else:
                print("Access denied")
                GPIO.output(greenLED,False)
                GPIO.output(redLED,True)
                GPIO.output(buzzer,True)
                time.sleep(2)
        GPIO.output(greenLED,False)
        GPIO.output(redLED,False)
        GPIO.output(buzzer,False)
finally:
        GPIO.cleanup()
```

Step 3: run the program

```
Sudo python card.py
```

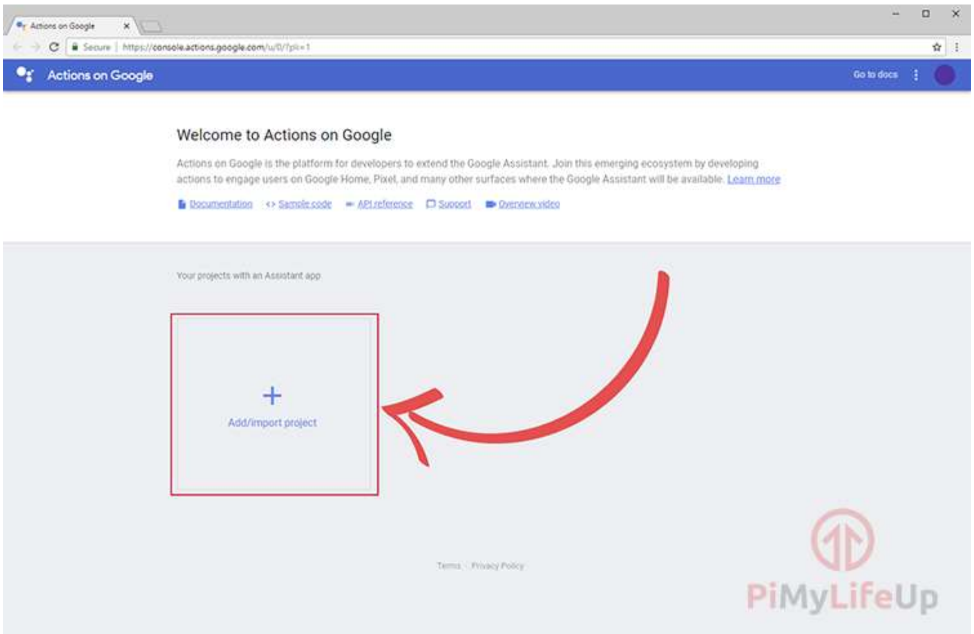
Google Assistant

Step 1: Before we get started with setting up the Google Assistant code on the Raspberry Pi itself, we must first register and set up a project on the Google Actions Console.

https://console.actions.google.com

Step 2:Once you have logged into your account, you will be greeted with the following screen.

On here you will want to click the “**Add/Import project**” button as shown in our screenshot below.



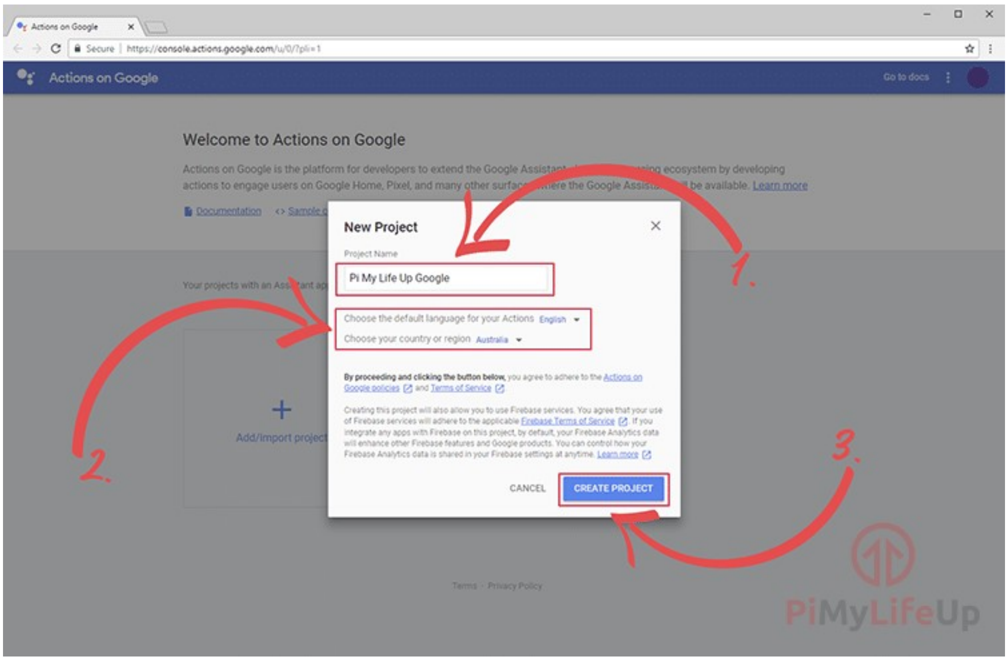
Welcome to Actions on Google

Actions on Google is the platform for developers to extend the Google Assistant. Join this emerging ecosystem by developing actions to engage users on Google Home, Pixel, and many other surfaces where the Google Assistant will be available. Learn more

Step 3: On this next screen, you will be asked to enter a “**Project Name**”

In addition to a project name you need to set both your country and your language as shown in the screenshot

Once you have set the Project Name and chosen your language and country, click the “**Create Project**” button



Actions on Google
Secure | https://console.actions.google.com/u/0/?pli=1
Actions on Google
Go to docs

Welcome to Actions on Google

Actions on Google is the platform for developers to extend the Google Assistant ... ecosystem by developing actions to engage users on Google Home, Pixel, and many other surfaces where the Google Assistant will be available. Learn more

Documentation
Sample c
Your projects with an Assistant ap
Add/import project

New Project

Project Name: Pi My Life Up Google

Choose the default language for your Actions: English

Choose your country or region: Australia

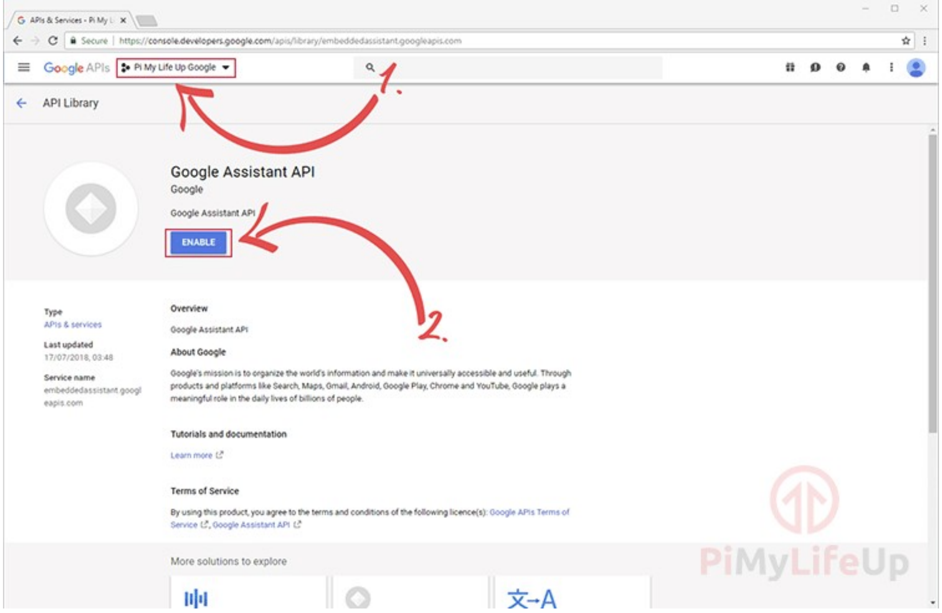
By proceeding and clicking the button below, you agree to adhere to the Actions on Google policies and Terms of Service

Creating this project will also allow you to use Firebase services. You agree that your use of Firebase services will adhere to the applicable Firebase Terms of Service. If you integrate any apps with Firebase on this project, by default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your Firebase settings at anytime. Learn more

Step 4: In a **new** tab, go to the Google developers console and enable the Google Embedded Assistant API.

Now before you go ahead and press the “**Enable**” button make sure that you have your project selected.

Once you are sure you have your current project selected, click the “**Enable**” button



APIs & Services - Pi My L
Secure | https://console.developers.google.com/apis/library/embeddedassistant.googleapis.com
Google APIs
Pi My Life Up Google
API Library
1.
2.
ENABLE

Google Assistant API

Google

Google Assistant API

Type: APIs & services

Last updated: 17/07/2018, 03:48

Service name: embeddedassistant.googleapis.com

Overview

Google Assistant API

About Google

Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.

Tutorials and documentation

Learn more

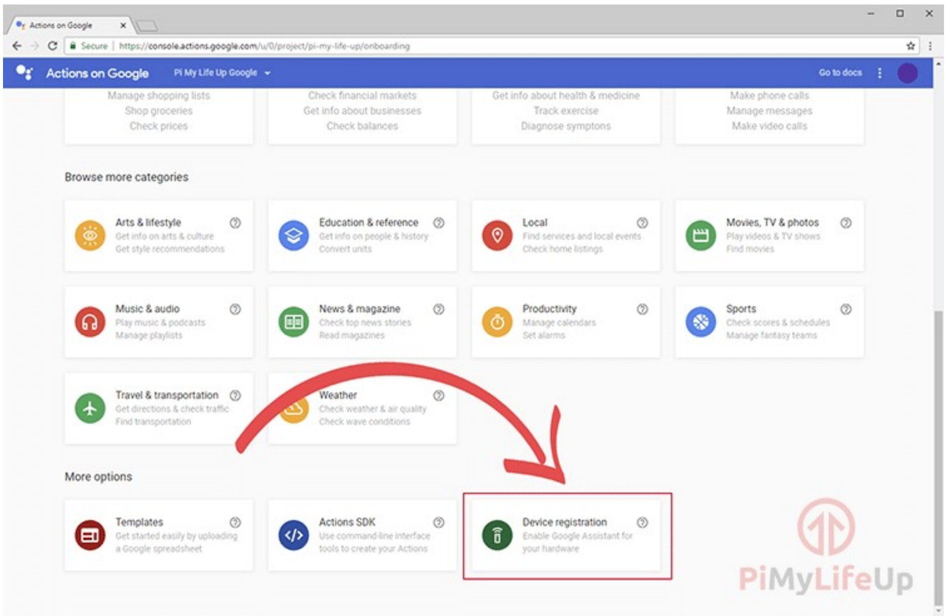
Terms of Service

By using this product, you agree to the terms and conditions of the following licence(s): Google APIs Terms of Service, Google Assistant API

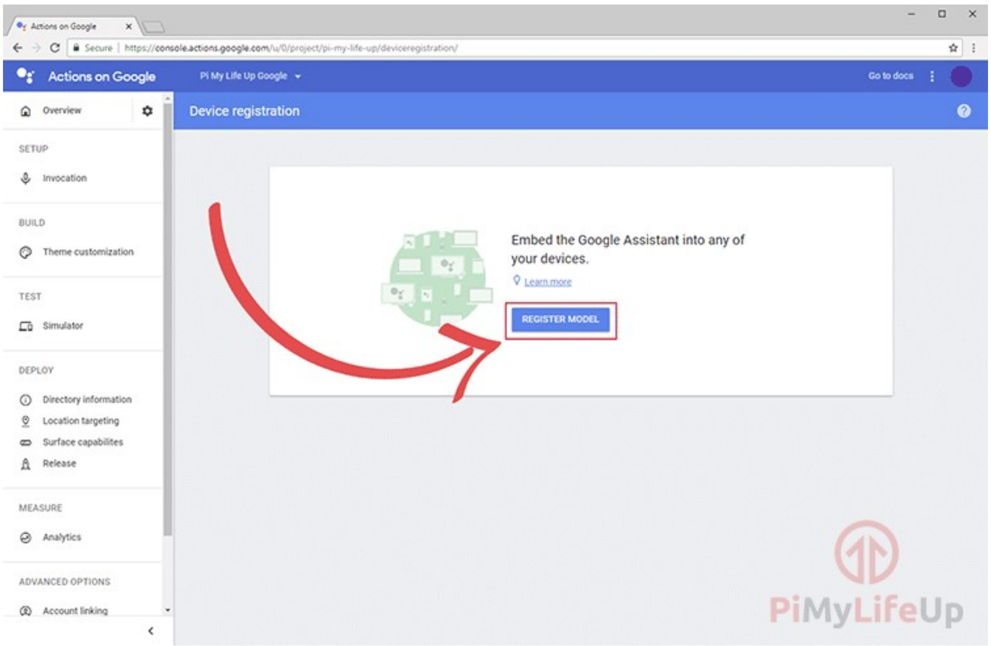
More solutions to explore

Step 5: Now back in the other tab where you created the project, scroll down to the bottom of the screen.

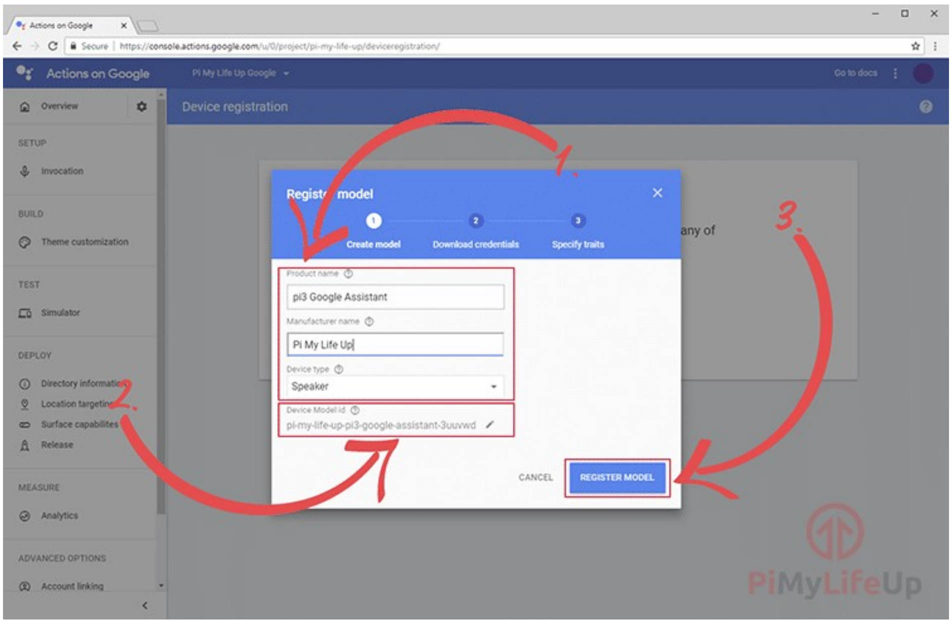
You should see a box with the text “**Device Registration**” on it as we have shown in the screenshot below. Click it to continue.



Step 6: You will now be taken to the following screen, click the “Register Model” button to continue.



Step 7: On this screen, you need to set a “Product Name“, “Manufacturer name” and set a “Device Type”



Product name: pi3 Google Assistant

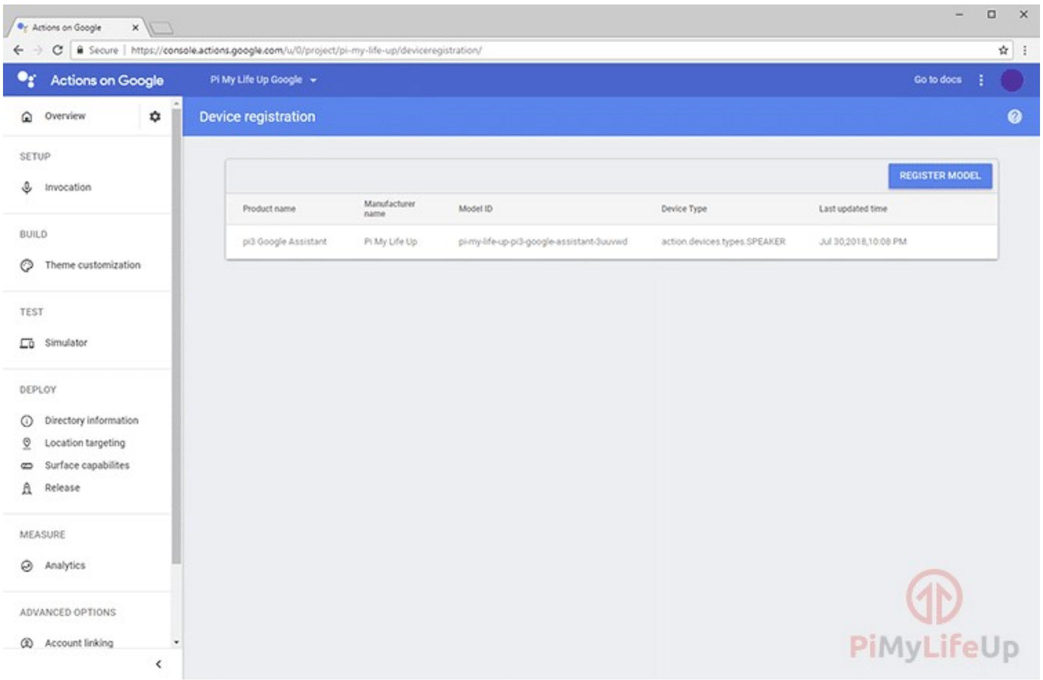
Manufacturer name: Pi My Life Up

Device type: Speaker

Device Model id: pi-my-life-up-pi3-google-assistant-3uuvwd

And Download the Crediantial.

Step 8:Once everything is done, you should be shown on this screen. We now only have one last thing we need to do before we can set up the Google Assistant on the Raspberry Pi itself.



Product name	Manufacturer name	Model ID	Device Type	Last updated time
pi3 Google Assistant	Pi My Life Up	pi-my-life-up-pi3-google-assistant-3uuvwd	action.devices.types.SPEAKER	Jul 30,2018,10:08 PM

Step 9: Finally, we need to go to the URL displayed below, on here you will need to activate the following activity controls to ensure that the Google Assistant API works correctly.

Web & App Activity

Location History

Device Information

Voice & Audio Activity

https://myaccount.google.com/activitycontrols

Step 10: Locate your USB microphone

```
arecord -l
```

Step 11: locate your speaker

```
aplay –l
```

Step 12:create a file with name .asoundrc

```
Sudo  nano .asoundrc
```

Step 13: Within this file enter the following lines.

Make sure that you replace **<card number>** and with **<device number>** their respective values that you retrieved during **Step 1**.

```
pcm.!default {
  type asym
  capture.pcm "mic"
  playback.pcm "speaker"
}
pcm.mic {
  type plug
  slave {
    pcm "hw:<card number>,<device number>"
  }
}
pcm.speaker {
  type plug
  slave {
    pcm "hw:<card number>,<device number>"
  }
}
```

Step 13:Test the Speaker

```
speaker-test -t wav
```

Step 14:Test micro phone

```
arecord --format=S16_LE --duration=5 --rate=16000 --file-type=raw out.raw
```

Step 15: Doing this is a crucial task as you don't want your Raspberry Pi picking up every little noise but you also don't want it being able to barely hear you when you say "Ok Google".

```
aplay --format=S16_LE --rate=16000 out.raw
```

Step 16:Update the Raspberry pi

```
sudo apt-get update
```

Step 17:make Directory and make a credential file.

```
mkdir ~/googleassistant
```

```
sudo nano ~/googleassistant/credentials.json
```

step 18: install Python3 and the Python 3 Virtual Environment

```
sudo apt-get install python3-dev python3-venv
```

Step 19: enable python3 as our virtual environment variable

```
python3 -m venv env
```

Step 20:install the new version of pip and setuptool.

```
env/bin/python -m pip install --upgrade pip setuptools --upgrade
```

Step 21:Activete the python Environment

```
source env/bin/activate
```

Step 22:utilize the pip and install the new version of python package

```
python -m pip install --upgrade google-assistant-library
```

```
python -m pip install --upgrade google-assistant-sdk[samples]
```

Step 23:Install the python authorization tool

```
python -m pip install --upgrade google-auth-oauthlib[tool]
```

step 24:Run the Google authorization tool

```
google-oauthlib-tool --client-secrets ~/googleassistant/credentials.json \
--scope https://www.googleapis.com/auth/assistant-sdk-prototype \
--scope https://www.googleapis.com/auth/gcm \
--save --headless
```

Step 25: Please visit this URL to authorize this application

Please copy this code, switch to your application and paste it there" followed by a long authentication code.

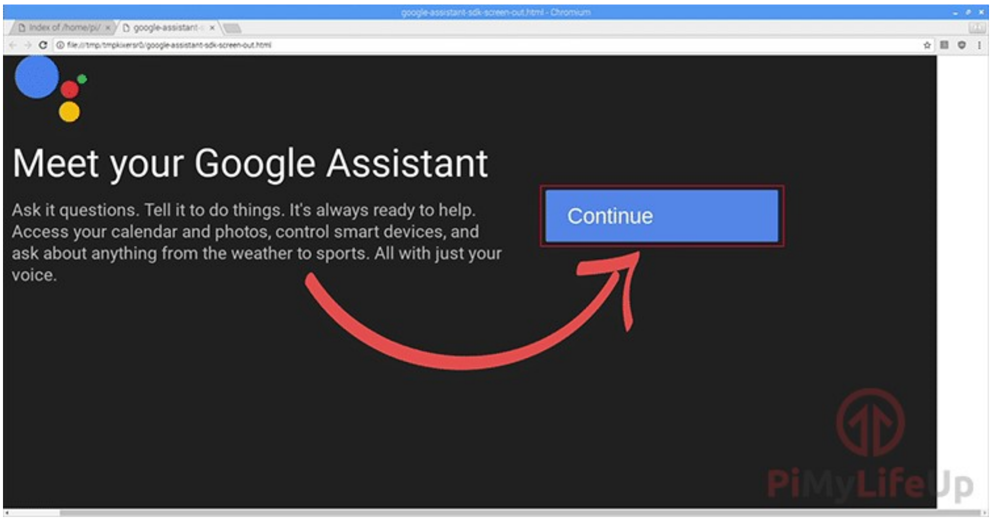
Copy the authentication code and paste it back into your terminal session and press enter.

Step 26:run the google sample code

```
googlesamples-assistant-pushtotalk --project-id <projectid> --device-model-id <deviceid> --display
```

Step 27: After running the push to talk sample, press "**Enter**" to trigger it and ask it any question.

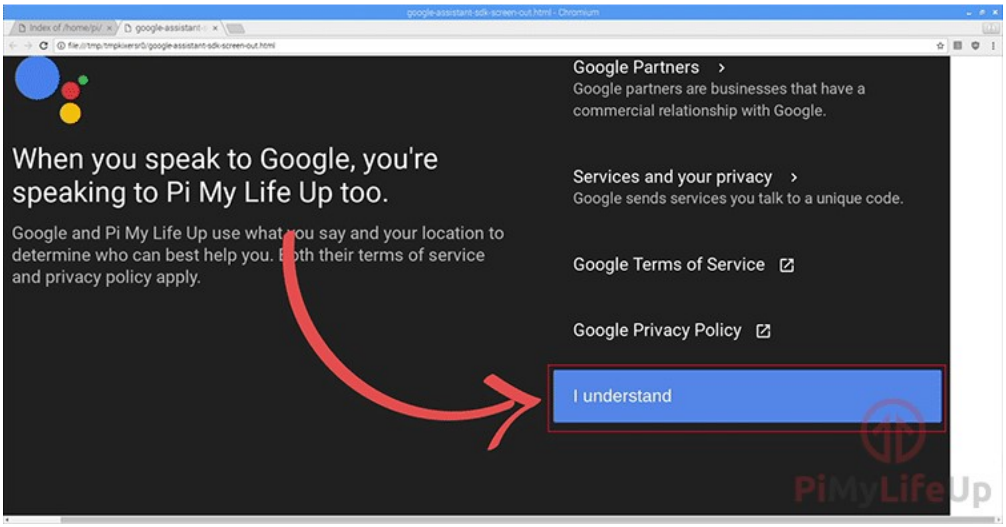
Upon asking your first question, you will be shown the screen below, begin by clicking the "**Continue**".


google-assistant-sdk-screen-out.html - Chromium
Index of /home/pi/
google-assistant-
file:///tmp/tmpkixersr0/google-assistant-sdk-screen-out.html
Continue
PiMyLifeUp

Meet your Google Assistant

Ask it questions. Tell it to do things. It's always ready to help. Access your calendar and photos, control smart devices, and ask about anything from the weather to sports. All with just your voice.

Step 28: After selecting "Continue" you will now be asked if you agree to a variety of different Google policies, to continue you must click the "I understand" button.


google-assistant-sdk-screen-out.html - Chromium
Index of /home/pi/
google-assistant-
file:///tmp/tmpkixersr0/google-assistant-sdk-screen-out.html
PiMyLifeUp

When you speak to Google, you're speaking to Pi My Life Up too.

Google and Pi My Life Up use what you say and your location to determine who can best help you. Both their terms of service and privacy policy apply.

Google Partners >
Google partners are businesses that have a commercial relationship with Google.

Services and your privacy >
Google sends services you talk to a unique code.

Google Terms of Service

Google Privacy Policy

I understand

Step 29: Finally, you will be asked to allow Google and your project the right to be able to share information with each other. Without this, the Google Assistant project will not function correctly.

To continue on you must click the “Allow” button as showcased below.

Step 30: With that now done we can now use the push to talk Google Assistant sample and hear a response.

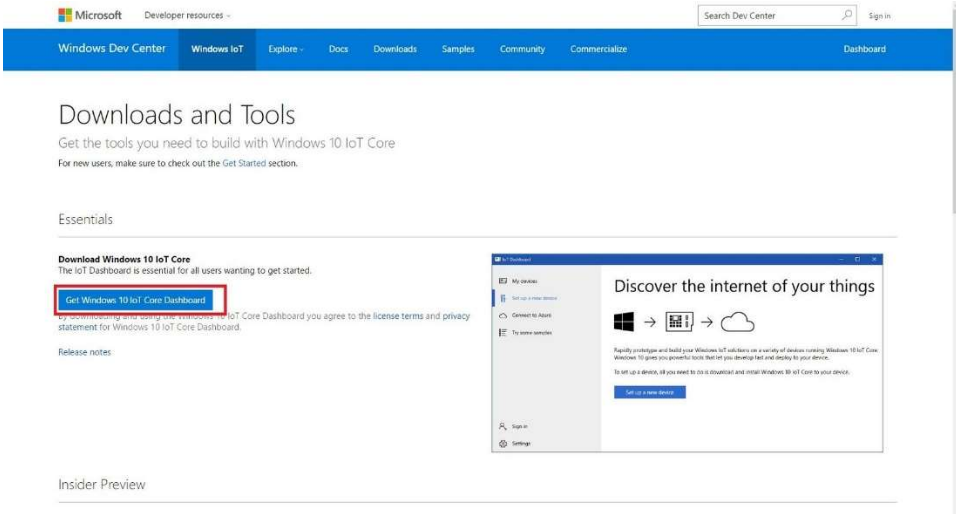
This time when you press the “Enter” in the terminal and speak an action such as “What is the time” you should hear a verbal response and another tab will be automatically opened displaying the action you just called.

Don’t worry you can disable the tab behavior by removing the –display argument on the command. We only needed this to get up the authorization screen.

IOT Core

Step 1: Go to the Windows 10 developer center.

Step 2: Click **Get Windows 10 IoT Core Dashboard** to download the necessary application.



Microsoft Developer resources

Search Dev Center Sign in

Windows Dev Center | Windows IoT | Explore | Docs | Downloads | Samples | Community | Commercialize | Dashboard

Downloads and Tools

Get the tools you need to build with Windows 10 IoT Core

For new users, make sure to check out the Get Started section.

Essentials

Download Windows 10 IoT Core
The IoT Dashboard is essential for all users wanting to get started.

Get Windows 10 IoT Core Dashboard

By downloading and using the Windows 10 IoT Core Dashboard you agree to the license terms and privacy statement for Windows 10 IoT Core Dashboard.

Release notes

IoT Dashboard

My devices
Set up a new device
Connect to Azure
Try some samples
Sign in
Settings

Discover the internet of your things

Rapidly prototype and build your Windows IoT solutions on a variety of devices running Windows 10 IoT Core. Windows 10 gives you powerful tools that let you develop fast and deploy to your device.

To set up a device, all you need to do is download and install Windows 10 IoT Core to your device.

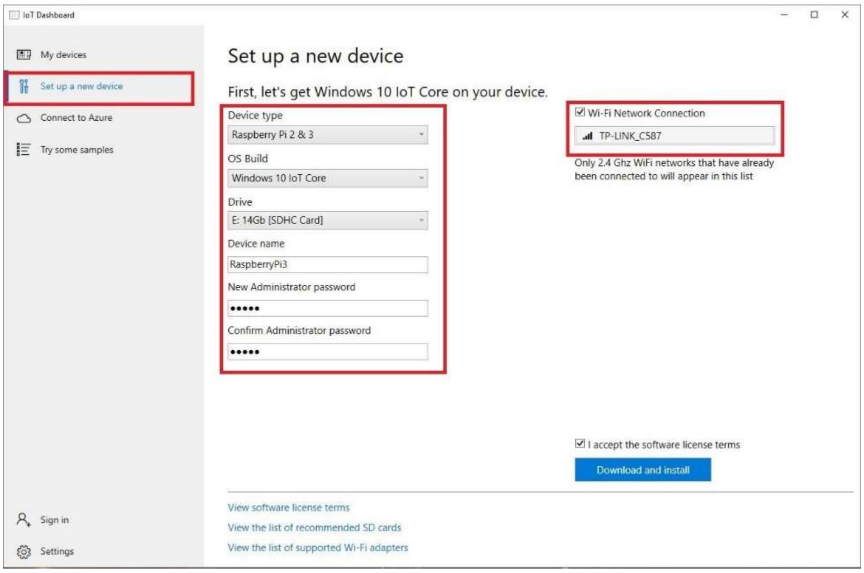
Set up a new device

Insider Preview

Step 3: Install the application and open it.

Step 4: Select **set up a new device** from the sidebar.

Step 5: Select the options as shown in the image below. Make sure you select the correct drive for your microSD card and give your device a name and admin password.



IoT Dashboard

My devices
Set up a new device
Connect to Azure
Try some samples
Sign in
Settings

Set up a new device

First, let's get Windows 10 IoT Core on your device.

Device type: Raspberry Pi 2 & 3

OS Build: Windows 10 IoT Core

Drive: E: 14Gb [SDHC Card]

Device name: RaspberryPi3

New Administrator password: •••••

Confirm Administrator password: •••••

☑ Wi-Fi Network Connection

TP-LINK_C587

Only 2.4 Ghz WiFi networks that have already been connected to will appear in this list

☑ I accept the software license terms

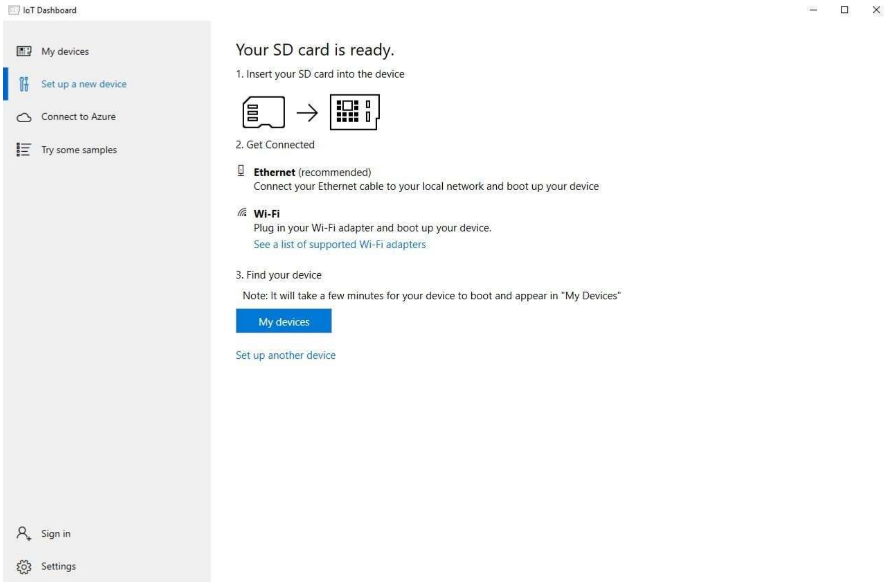
Download and install

View software license terms
View the list of recommended SD cards
View the list of supported Wi-Fi adapters

Step 6: Select the WiFi network connection you want your Raspberry Pi to connect to, if required. Only networks your PC connects to will be shown.

1. Step 7: Click **download and install**.

The application will now download the necessary files from Microsoft and flash them to your microSD card. It'll take a little while, but the dashboard will show you the progress.



Your SD card is ready.

1. Insert your SD card into the device

2. Get Connected

Ethernet (recommended)
Connect your Ethernet cable to your local network and boot up your device

Wi-Fi
Plug in your Wi-Fi adapter and boot up your device.
See a list of supported Wi-Fi adapters

3. Find your device

Note: It will take a few minutes for your device to boot and appear in "My Devices"

Once the image has been installed on the microSD card, it's time to eject it from your PC and go over to the Raspberry Pi. First connect up the micro USB cable and power supply, HDMI cable and USB WiFi adapter or Ethernet cable. Connect the HDMI cable to your chosen display, insert the microSD card into the Raspberry Pi and power it up.

LED Matrix

Components :

1-RaspberryPi3

2-8x8matrix

3 –Jumper wires

Pin Connection:

Name	Pin number	Rpi Pin
VCC	1	2
GND	2	6
DIN	3	19
CS	4	24
CLK	5	23

Step1: **git clone https://github.com/rm-hull/max7219.git**

Step2: **sudo python max7219/setup.py install**

Step3: enable SPI

```
sudo raspi-config
```

Step4: install module of 8x8 matrix

```
Sudo apt-get install python-dev python-pip
```

```
Sudo pip install max7219
```

Step 5:open the terminal

Step 6:open new file

```
Sudo nano matrix.py
```

Step 7:write the program

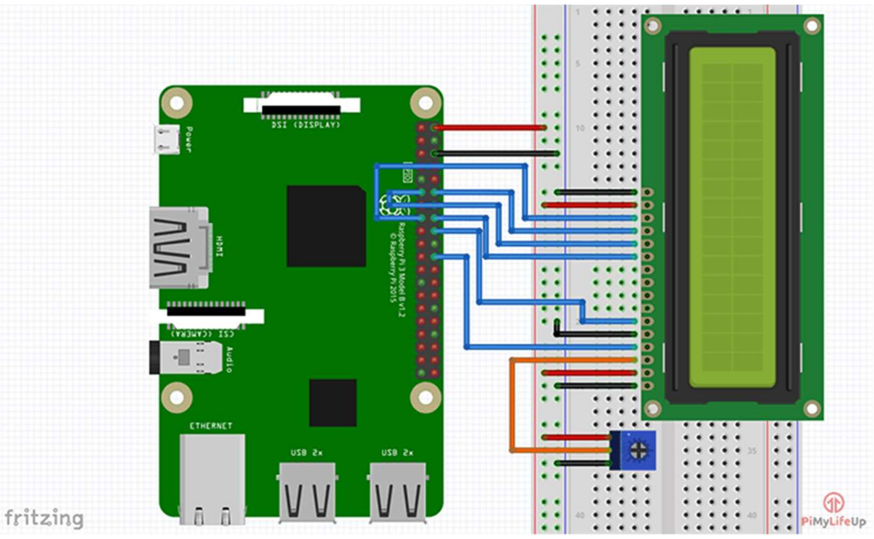
```
Import max7219.led as led

device=led.matrix(cascaded=3)

device.show_message("Akshay")
```

step 8: Run the program

```
sudo  python matrix.py
```

LCD Display

Circuit Diagram:



Step 1: clone the required git directory to the Raspberry Pi

```
git clone https://github.com/adafruit/Adafruit_Python_CharLCD.git
```

step 2: change into the directory we just cloned and run the setup file

```
cd ./Adafruit_Python_CharLCD
sudo python setup.py install
```

step 3: update the pin variable in the file char_lcd.py

if you follow my circuit then the value is

```
lcd_rs      = 25
lcd_en       = 24
lcd_d4       = 23
lcd_d5       = 17
lcd_d6       = 18
lcd_d7       = 22
lcd_backlight = 4
lcd_columns = 16
lcd_rows = 2
```

Step 3: go to directory

```
cd ~/Adafruit_Python_CharLCD/examples/
```

Step 4:open the char_lcd.py and update the value which we listed above

```
Sudo nano char_lcd.py
```

Step 5:run the program

```
Sudo python char_lcd.py
```