JNAN VIKAS MANDAL'S

PADMASHREE DR. R.T.DOSHI DEGREE COLLEGE OF COMPUTER SCIENCE
MOHANLAL RAICHAND MEHTA COLLEGE OF COMMERCE
DIWALIMAA DEGREE COLLEGE OF SCIENCE
AMRATLAL RAICHAND MEHTA COLLEGE OF ARTS
JVM'S DEGREE COLLEGE OF INFORMATION TECHNOLOGY
AIROLI, NAVI MUMBAI – 400708

CERTIFICATE

This is to certify that the Mr./Miss. ____ of S.Y.B.Sc.(C.S) Semester IV having roll number ___________ has completed the practical work in the subject of **IoT Technology** during the Academic year 2023-24 under the guidance of **Mrs. Janhavi Kshirsagar** being the partial requirement for the fulfilment of the curriculum of Degree of Bachelor of Science in Computer Science, University of Mumbai.

Place: **Date:**

______________________ ____________________________

Sign of Subject In-Charge Sign of External Examiner

____

Sign of CS-IT Coordinator

INDEX

Practical No : 1
Raspberry Pi Hardware Preparation and Installation

Hardware Guide:

For getting started with raspberry pi for the first time you will require the following hardware

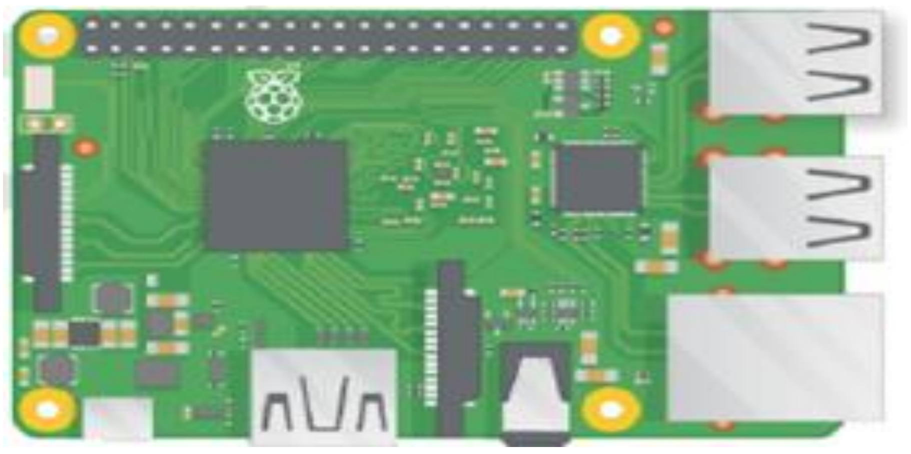
1. Raspberry Pi (latest Model)
2. Monitor or TV
3. HDMI cable
4. Ethernet cable
5. USB keyboard
6. USB mouse
7. Micro USB power supply
8. 8GB or larger microSD card
9. SD Card Reader

Raspberry Pi 3 Model B:

The Raspberry Pi 3 is the third generation Raspberry Pi. It replaced the Raspberry Pi 2 ModelB in February 2016. Compared to the Raspberry Pi 2 it has:

A 1.2GHz 64-bit quad-core ARMv8 CPU802.11n Wireless LAN
Bluetooth 4.1
Bluetooth Low Energy (BLE)Like the Pi 2, it also has:

4 USB ports
40 GPIO pins Full HDMI portEthernet port
Combined 3.5mm audio jack and composite videoCamera interface (CSI)
Display interface (DSI)
Micro SD card slot (now push-pull rather than push-push)Video Core IV 3D graphics core
The Raspberry Pi 3 has an identical form factor to the previous Pi 2 (and Pi 1 Model B+) andhas complete compatibility with Raspberry Pi 1 and 2.



Monitor or TV:

A monitor or TV with HDMI in can be used as a display with a Raspberry Pi. Most modern television sets and monitors have an HDMI port, and are the easiest to get working with the Raspberry Pi. You can use an HDMI cable to connect the Raspberry Pi directly to the t elevision or monitor.

Some older monitors have a DVI port. These work well with the Raspberry Pi, although you'll need an HDMI-to-DVI adapter to attach to an HDMI cable, or a one-piece HDMI-to-DVI cable. Some old monitors have a VGA port. These can be trickier to use as you'll need an HDMI-to-VGA converter, which can change digital video to analogue video. A simple port adapter won't work.

HDMI to HDMI Cable:

Connect Raspberry Pi to a Monitor or TV with a HDMI to HDMI cable.

Ethernet cable:

Ethernet cable will allow your Pi to connect with the internet. It is also useful for headless setup of Raspberry Pi

USB Keyboard and Mouse:

Any standard USB keyboard and mouse can be used with the Raspberry Pi. This plug and play devices will work without any additional driver. Simply plug them into the Raspberry Pi and they should be recognised when it starts up.

Power Supply:

It is recommended that you use a 5V, 2A USB power supply for all models of Raspberry Pi.

SD Card:

The latest version of Raspbian, the default operating system recommended for the Raspberry Pi, requires an 8GB (or larger) micro SD card. SD card will store the operating systems as well as all the file and applications created by you.

Installation Guide:

Now since you have all the required hardware, we will now learn how to get the operating system onto your microSD card so that you can start using software on your Raspberry Pi

Get Raspbian OS on your microSD card:

Raspbian comes pre-installed with plenty of software for education, programmin g and general use. It has Python, Scratch, Sonic Pi, Java, Mathematica and more.

1. To download Raspbian log on to raspberrpi.org and click on the download, then click on Raspbian and lastly download the RASPBIAN BUSTER WITH DESKTOP file. You can chooseeither the Torrent file or ZIP file.
2. The downloaded file will be in zip format. To unzip the file, you will require an unzip tool. You can use any unzipping tool viz. WINRAR, 7ZIP etc. After unzipping the file, you will find adisc image file in the unzipped folder.
3. Now format the SD Card before writing the disc image file on the SD card. You can use SD Formatter tool or any other tool of your wish.
4. To write the image file of the operating system on the SD card you will require a Disk Imagertool. For this you can use Win32 Disk Imager tool.
5. Once the image is written on the SD Card, your untitled SD card will now have the name boot. Your SD Card will now hold the Raspbian Operating system required for the first-timesetup.

Plugging in your Raspberry Pi:

1. Begin by placing your SD card into the SD card slot on the Raspberry Pi. It will only fit oneway.
2. Next, plug your keyboard and mouse into the USB ports on the Raspberry Pi.
3. Make sure that your monitor or TV is turned on, and that you have selected the right in put(e.g. HDMI 1, DVI, etc).
4. Connect your HDMI cable from your Raspberry Pi to your monitor or TV.
5. If you intend to connect your Raspberry Pi to the internet, plug an Ethernet cable into the Ethernet port, or connect a WiFi dongle to one of the USB ports (unless you have a Raspberry Pi 3).
6. When you're happy that you have plugged all the cables and SD card in correctly, connectthe micro USB power supply. This action will turn on and boot your Raspberry Pi.

NOTE: for Raspberry Pi 4 please follow the instruction given below.

1. To download Raspberry pi OS (buster)log on to **raspberrypi.com** and click on the download and thenclick of rasberry pi os and click on **Raspberry pi OS buster.**

2. Follow the above mention step from 2 to 5 of **INSTALLATION GUIDE** section.
3. After writing the image on SD Card plug in the SD Card in to the Raspberry pi 4. If boot normally andyou see the raspberry pi screen than ok. But if not than make the following changes in the SD card ‘config' filewith notepad++.
    1. Uncomment the line 'hdmi_force_hotplug=1’
    2. Uncomment the line 'hdmi_group=1’
    3. Uncomment the line 'hdmi_mode=1'
    4. Save and exit the 'config' file
    5. Now insert again the memory card into the raspberry pi it will work.

Practical No : 2
GPIO: Light the LED with Python

Hardware Guide:

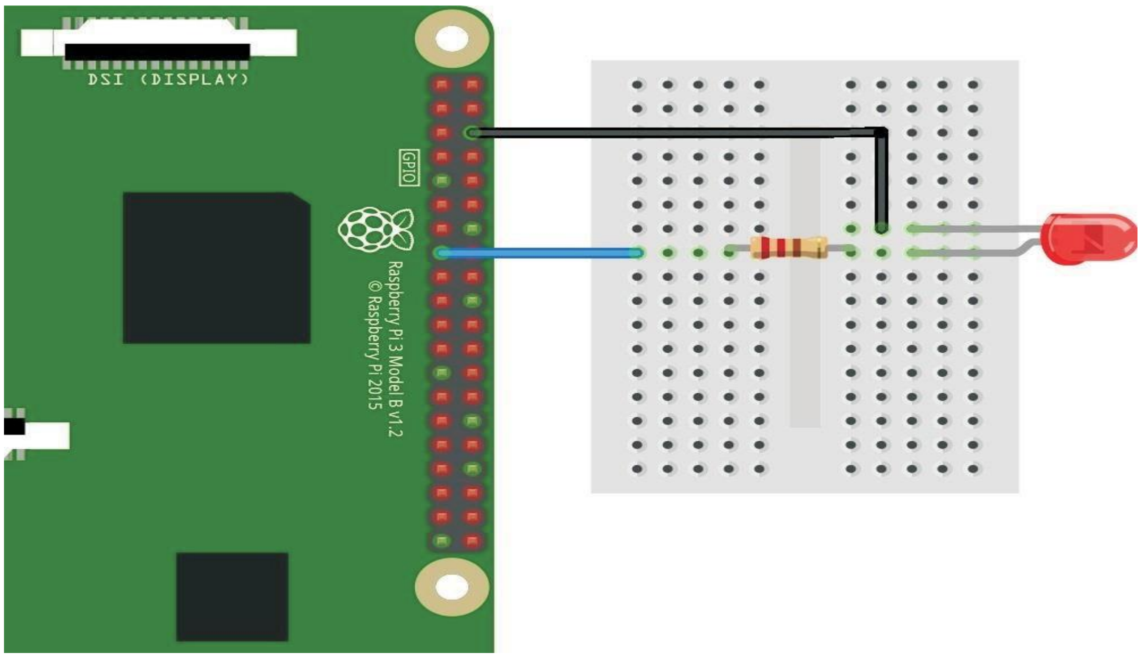
Along with the basic setup you will require the following components to get started with the GPIO pins as follows:
1. LED
2. Resistor
3. Connecting wires
4. Breadboard

Before learning this lesson, you must understand the pin numbering system of the GPIO pins.

Wiring up your Circuit:

1. Connect the GPIO22 (i.e. Physical Pin 15) Pin od raspberry pi to one end of the resistor.
2. Connect another end of resistor to the positive end (anode) of LED
3. Connect the negative end (cathode) of LED to Ground of raspberry pi.
4. Then Power on your raspberry pi

Circuit Diagram:



Code:

```
#Blink LED Program
#Connect the LED to GPIO 22 Pin
#LED Blink Progarm
#Connect the LED to GPIO22 (i.e. Physical Pin15)


#import GPIO and time libraryimport
RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)                  #set the Pin mode you will be working with

ledPin = 22                             #this is GPIO22 pin i.e

Physical Pin15#setup the ledPin(i.e. GPIO22) as output
GPIO.setup(ledPin, GPIO.OUT)
GPIO.output(ledPin, False)

try:
while True:
GPIO.output(ledPin, True) #Set the LED Pin to HIGH
print("LED ON")
sleep(1)                                #Wait for 1
sec GPIO.output(ledPin, False) #Set the LED Pin to LOW
print("LED OFF")
    sleep(1)
                                        #wai
t for 1 secfinally:
#reset the GPIO Pins
GPIO.output(ledPin, False)
GPIO.cleanup()

#end of code
```

After writing your code save is on a desired location and run it to enjoy the fun.
Great! You have completed your first GPIO program. Now you are confident enough to go forward to explore and interface new things with raspberry Pi.
If you were unable to blink the LED, don't worry, recheck your connection and debug the error so that you don't repeat it again.
Now in the next lessons we will explore and interface new things

Practical No : 3
Displaying different LED Patterns with Raspberry Pi

For displaying Differen LED pattern connecte 8 LEDs in the same format to the pin number given in the below python Code :

CODE FOR LED PATTERN:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
led1 = 29
led2 = 31
led3 = 33
led4 = 35
led5 = 36
led6 = 37
led7 = 38
led8 = 40
#setup the ledPin(i.e. GPIO22) as output
GPIO.setup(led1, GPIO.OUT)
GPIO.setup(led2, GPIO.OUT)
GPIO.setup(led3, GPIO.OUT)
GPIO.setup(led4, GPIO.OUT)
GPIO.setup(led5, GPIO.OUT)
GPIO.setup(led6, GPIO.OUT)
GPIO.setup(led7, GPIO.OUT)
GPIO.setup(led8, GPIO.OUT)
GPIO.output(led1, False)
GPIO.output(led2, False)
GPIO.output(led3, False)
GPIO.output(led4, False)
GPIO.output(led5, False)
GPIO.output(led6, False)
GPIO.output(led7, False)
GPIO.output(led8, False)
def ledpattern(ledVal1, ledVal2, ledVal3, ledVal4, ledVal5, ledVal6, ledVal7,ledVal8):
GPIO.output(led1, ledVal1)
GPIO.output(led2, ledVal2)
GPIO.output(led3, ledVal3)
GPIO.output(led4, ledVal4)
GPIO.output(led5, ledVal5)
GPIO.output(led6, ledVal6)
GPIO.output(led7, ledVal7)
GPIO.output(led8, ledVal8)
10 EDKITS ELECTRONICS
11 EDKITS ELECTRONICS
def patterOne():
for i in range (0, 3):
ledpattern(1, 0, 1, 0, 1, 0, 1, 0)
time.sleep(1)
ledpattern(0, 1, 0, 1, 0, 1, 0, 1)
```

```
time.sleep(1)
def patternTwo():
for i in range (0, 5):
ledpattern(1, 0, 0, 0, 0, 0, 0, 0)
time.sleep(0.1)
ledpattern(0, 1, 0, 0, 0, 0, 0, 0)
time.sleep(0.1)
ledpattern(0, 0, 1, 0, 0, 0, 0, 0)
time.sleep(0.1)
ledpattern(0, 0, 0, 1, 0, 0, 0, 0)
time.sleep(0.1)
ledpattern(0, 0, 0, 0, 1, 0, 0, 0)
time.sleep(0.1)
ledpattern(0, 0, 0, 0, 0, 1, 0, 0)
time.sleep(0.1)
ledpattern(0, 0, 0, 0, 0, 0, 1, 0)
time.sleep(0.1)
ledpattern(0, 0, 0, 0, 0, 0, 0, 1)
time.sleep(0.1)
def patternThree():
for i in range (0, 5):
ledpattern(0, 0, 0, 0, 0, 0, 0, 1)
time.sleep(0.1)
ledpattern(0, 0, 0, 0, 0, 0, 1, 0)
time.sleep(0.1)
ledpattern(0, 0, 0, 0, 0, 1, 0, 0)
time.sleep(0.1)
ledpattern(0, 0, 0, 0, 1, 0, 0, 0)
time.sleep(0.1)
ledpattern(0, 0, 0, 1, 0, 0, 0, 0)
time.sleep(0.1)
ledpattern(0, 0, 1, 0, 0, 0, 0, 0)
time.sleep(0.1)
ledpattern(0, 1, 0, 0, 0, 0, 0, 0)
time.sleep(0.1)
ledpattern(1, 0, 0, 0, 0, 0, 0, 0)
time.sleep(0.1)
```

```
def patternFour():
for i in range (0, 5):
ledpattern(0, 1, 1, 1, 1, 1, 1, 1)
time.sleep(0.1)
ledpattern(1, 0, 1, 1, 1, 1, 1, 1)
time.sleep(0.1)
ledpattern(1, 1, 0, 1, 1, 1, 1, 1)
time.sleep(0.1)
ledpattern(1, 1, 1, 0, 1, 1, 1, 1)
time.sleep(0.1)
ledpattern(1, 1, 1, 1, 0, 1, 1, 1)
time.sleep(0.1)
ledpattern(1, 1, 1, 1, 1, 0, 1, 1)
time.sleep(0.1)
ledpattern(1, 1, 1, 1, 1, 1, 0, 1)
```

```
time.sleep(0.1)
ledpattern(1, 1, 1, 1, 1, 1, 1, 0)
time.sleep(0.1)
def patternFive():
for i in range (0, 5):
ledpattern(1, 1, 1, 1, 1, 1, 1, 0)
time.sleep(0.1)
ledpattern(1, 1, 1, 1, 1, 1, 0, 1)
time.sleep(0.1)
ledpattern(1, 1, 1, 1, 1, 0, 1, 1)
time.sleep(0.1)
ledpattern(1, 1, 1, 1, 0, 1, 1, 1)
time.sleep(0.1)
ledpattern(1, 1, 1, 0, 1, 1, 1, 1)
time.sleep(0.1)
ledpattern(1, 1, 0, 1, 1, 1, 1, 1)
time.sleep(0.1)
ledpattern(1, 0, 1, 1, 1, 1, 1, 1)
time.sleep(0.1)
ledpattern(0, 1, 1, 1, 1, 1, 1, 1)
time.sleep(0.1)
try:
while True:
patterOne()
patternTwo()
patternThree()
patternFour()
patternFive()
finally:
#reset the GPIO Pins
GPIO.cleanup()
```

Practical 4:
GPIO: Program the 8x8 LED Grid Module

In this lesson, we will be interfacing 8x8 LED matrix Module with raspberry pi. Since you are now familiar with the GPIO pins, it will be a fun to know more about the python coding and installing libraries to do more advance things.

Hardware Guide:

For completing this lesson, you will require the following things along with your initial raspberry pi setup

1. 8x8 LED matrix module
2. 7219 driver board
3. Connecting wires

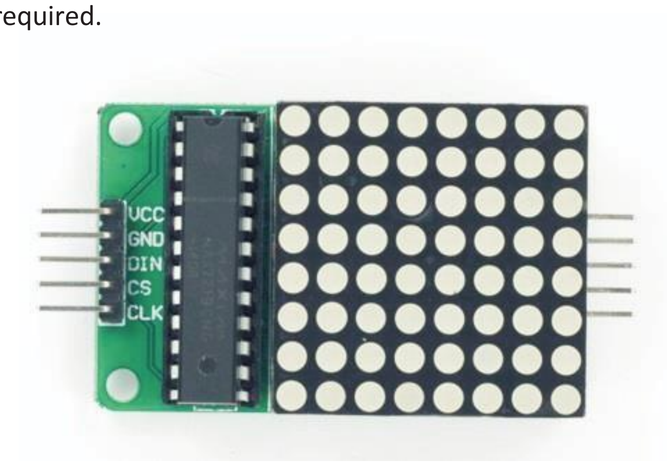
7219 Driver board:

Before interfacing LED matrix with raspberry pi, we need to connect the Max7219 IC which isan Led driver to the LED matrix display. The reason behind using this led driver is that it drives the 64Led's simultaneously which in turn reduces the number of wires so that the user will find it easy to connect the display to the raspberry pi.

The MAX7219 has four wire SPI interface (we need only these four wires to interface it to theraspberry pi):

1. Din - MOSI - Master Output Serial Input.
2. Chip select - Load (CS) - active low Chip select.
3. Clock – SCK
4. Ground.

And off course VCC (5V) is required.



Software Guide:

Here we will be writing our code I python. But before writing our code we need to enable SPI and we need to install the library for driving the LED Matrix Module using our raspberry pi.

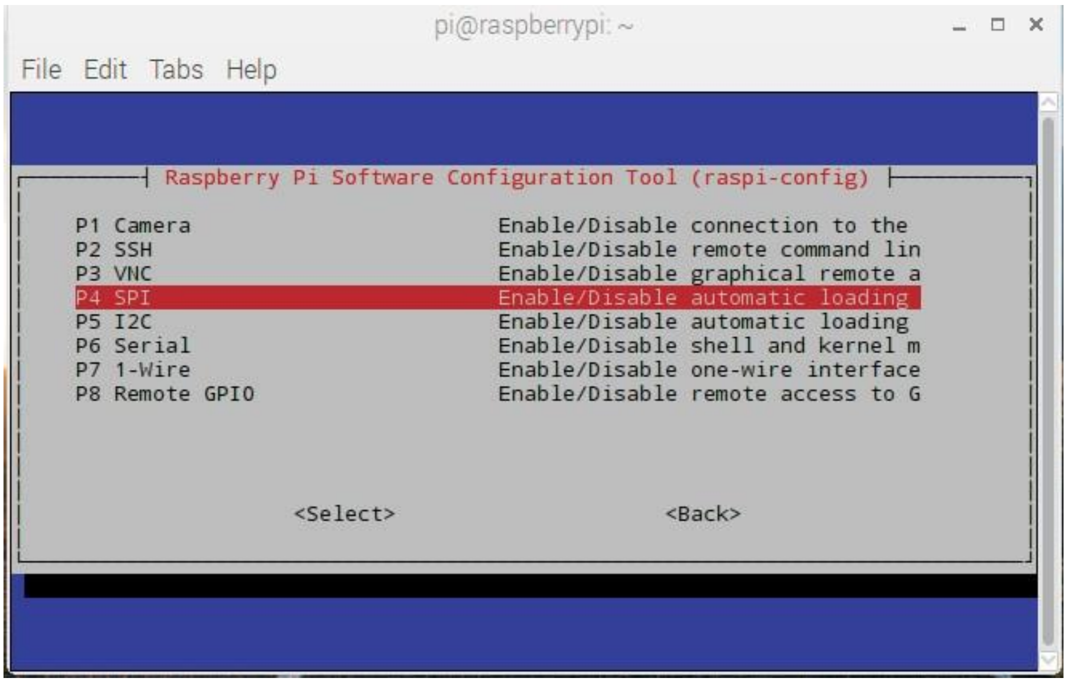
Pre-requisites:

By default, the SPI kernel driver is NOT enabled on the Raspberry Pi Raspian image . Enable the SPI as follows:

1. Open terminal and type sudo raspi-config and press Enter.
2. Use the down arrow to select 5 Interfacing options
3. Arrow down to P4 SPI.
4. Select yes when it asks you to enable SPI,
5. Also select yes when it asks about automatically loading the kernel module.
6. Use the right arrow to select the <Finish> button.
7. Select yes when it asks to reboot.

Alternatively using GUI, you can also follow the following steps to enable SPI :

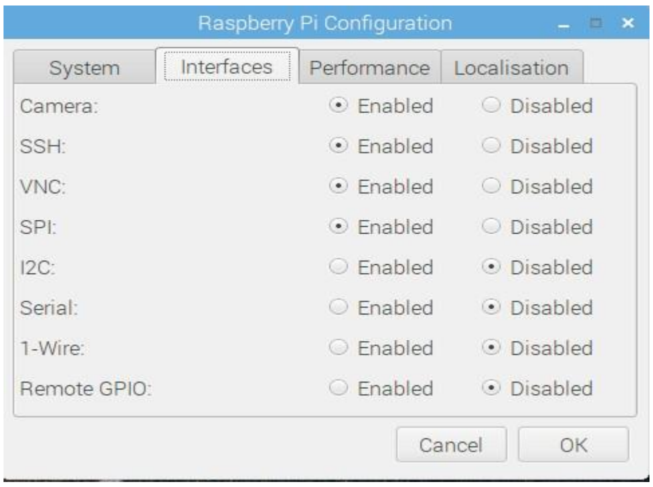
1. Select Preference from the raspberry pi application menu.
2. From Preference select Raspberry Pi Configuration.
3. Now form the Raspberry Pi Configuration window, navigate to Interfaces option.
4. Select the enabled radio button in front of SPI to enable it and click on OK.
5. Finally, do not forget to reboot your raspberry pi after changing this setting.



```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera               Enable/Disable connection to the
P2 SSH                  Enable/Disable remote command lin
P3 VNC                  Enable/Disable graphical remote a
P4 SPI                  Enable/Disable automatic loading
P5 I2C                  Enable/Disable automatic loading
P6 Serial               Enable/Disable shell and kernel m
P7 1-Wire               Enable/Disable one-wire interface
P8 Remote GPIO          Enable/Disable remote access to G

            <Select>                 <Back>
```



Camera:	[✓ Enabled]	Disabled
SSH:	[✓ Enabled]	Disabled
VNC:	[✓ Enabled]	Disabled
SPI:	[✓ Enabled]	Disabled
I2C:	Enabled	[✓ Disabled]
Serial:	Enabled	[✓ Disabled]
1-Wire:	Enabled	[✓ Disabled]
Remote GPIO:	Enabled	[✓ Disabled]

Installing the Library:

Before installing the library make sure Raspberry pi is connected to Internet.

Clone the code from github:

```
$ git clone https://github.com/freedomwebtech/max7219voicecontrol
```

Now change directory to max7219voicecontrol-main

```
$ pi@raspberrypi:~ $ cd max7219voicecontrol-main
```

Run ls command

```
$ pi@raspberrypi:~/max7219voicecontrol-main $ ls
buttons.py  install.sh  ledmatrix.py  __pycache___________
```

Run following command

```
pi@raspberrypi:~/max7219voicecontrol-main $ sudo chmod 775 install.sh
```

For installation run the following command

```
$ pi@raspberrypi:~/max7219voicecontrol-main $ sudo ./install.sh
```

Now it is the time to write our code. Open Python3, navigate to files and open a new file and writethe code given below

Code:

```
from luma.led_matrix.device import max7219
from luma.core.interface.serial import spi, noop
from luma.core.render import canvas
from luma.core.virtual import viewport
from luma.core.legacy import text, show_message
from luma.core.legacy.font import proportional, CP437_FONT, TINY_FONT, SINCLAIR_FONT,
LCD_FONT
from datetime import datetime
import time

serial = spi(port=0, device=0, gpio=noop())
device = max7219(serial, cascaded=1, block_orientation=-90,
blocks_arranged_in_reverse_order=True)device.contrast(16)

def test():
now = datetime.now()
#  dt1_string = now.strftime("%H:%M:%S")dt1_string = now.strftime("%I:%M:%S")

with canvas(device) as draw:
text(draw, (3, 1), dt1_string, fill="white", font=proportional(TINY_FONT))
#show_message(device, "Hello EDKITS", fill="red",font=(CP437_FONT),scroll_delay=0.08)

while True:
test()
```

IMPORTANT:

There is some change is main code please change the code in ledmatrix.py as shown below

Comment this line in ledmatrix.py file

```
#           text(draw, (3, 1), dt1_string, fill="white", font=proportional(TINY_FONT))
```

and uncomment this line in ledmatrix.py file

```
        show_message(device, "Hello EDKITS",
```

fill="red",font=(CP437_FONT),scroll_delay=0.08)save the file. then run the

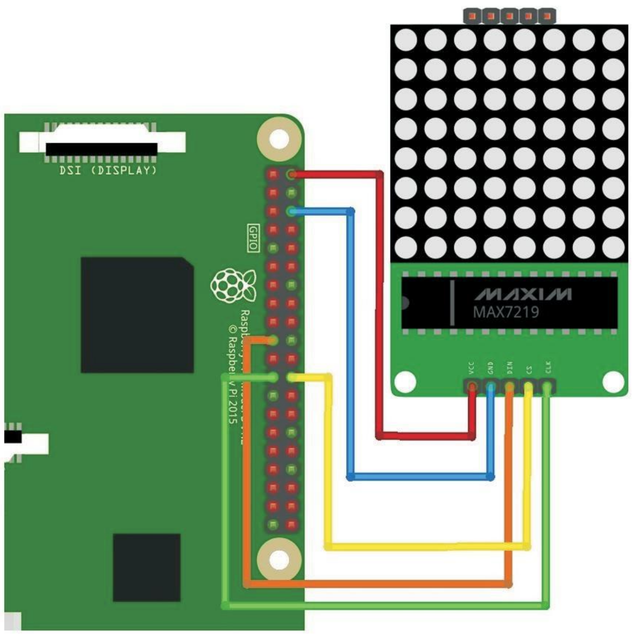
command after conneting LEDmatrix module to Raspberry pi.

Wiring up your Circuit:

We are using SPI protocol for wiring LED matrix module to raspberry pi, since it reduces the number pins required for wiring the circuit. You can follow the diagram given below while wiring your circuit.

1. Connect the VCC pin of 7219 driver board to Pin2 of raspberry pi.
2. Connect the Gnd pin of 7219 driver board to Pin6 of raspberry pi.
3. Connect the DIN pin of 7219 driver board to Pin19 of raspberry pi.
4. Connect the CS pin of 7219 driver board to Pin24 of raspberry pi.
5. Lastly, connect the CLK Pin of 7219 driver board to Pin23 of raspberry pi.

Circuit Diagram:



After ensuring that the connections are done properly, power on your raspberry pi. Now open Python3 and run the code that you have written for this lesson.
So now you have learned how to interface 8x8 LED matrix module with your raspberry pi, how to install libraries and how to enable SPI.
Next lesson will bring new things and new fun, so stay tuned!

Practical No : 4
Camera Connection and capturing Images

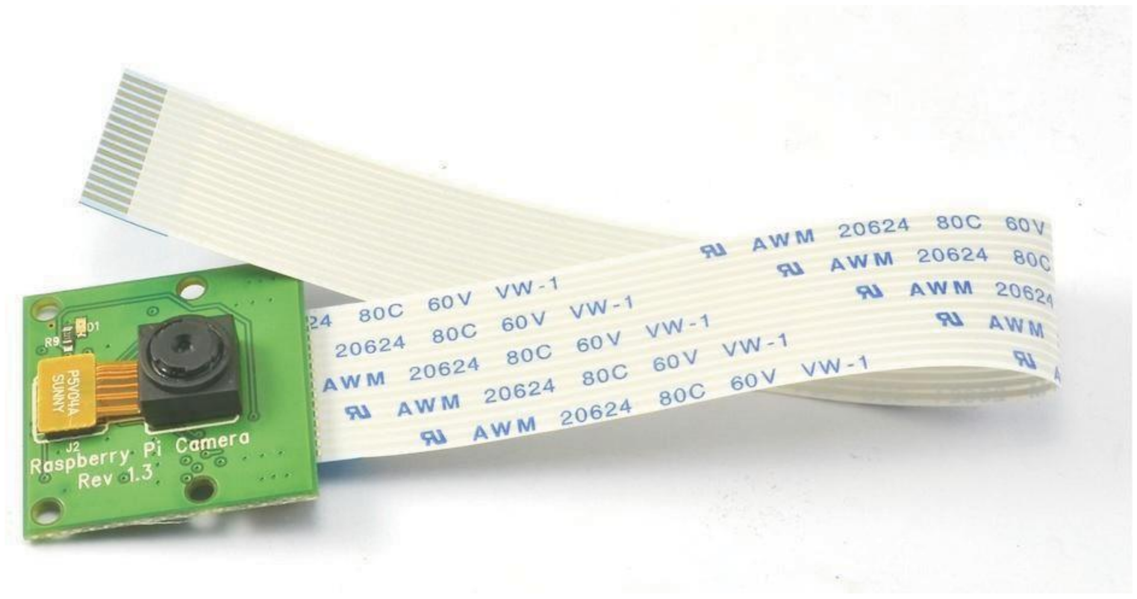
The Camera Module is a great accessory for the Raspberry Pi, allowing users to take still pictures and record video in full HD.

Hardware Guide:

For completing this lesson, you will require the Camera Module along with your initial raspberry pi setup.

Camera Module:

The Raspberry Pi Camera Board plugs directly into the CSI connector on the Raspberry Pi. The camera is supported in the latest version of Raspbian, the Raspberry Pi's preferred operating system.



The Raspberry Pi Camera Board Features:
1. Fully Compatible with Both the Model A and Model B Raspberry Pi
2. 5MP Omnivision 5647 Camera Module
3. Still Picture Resolution: 2592 x 1944
4. Video: Supports 1080p @ 30fps, 720p @ 60fps and 640x480p 60/90 Recording
5. 15-pin MIPI Camera Serial Interface – Plugs Directly into the Raspberry Pi Board
6. Size: 20 x 25 x 9mm
7. Weight 3g
8. Fully Compatible with many Raspberry Pi cases

Connect the Camera Module:

First of all, with the Pi switched off, you'll need to connect the Camera Module to the Raspberry Pi's camera port, then start up the Pi and ensure the software is enabled.

1. Locate the camera port and connect the camera:
2. Start up the Pi.
3. Open the Raspberry Pi Configuration Tool from the main menu.
4. Ensure the camera software is enabled. If it's not enabled, enable it and reboot your Pi tobegin.

Software Guide:

Now your camera is connected and the software is enabled, you can get started by capturing animage.

You can capture an image by just typing a single line command. Open terminal window and type thecommand as follows:

```
$ sudo raspistill -o /home/pi/Desktop/image.jpg
```

Code:

```
from time import sleep
from picamera import PiCamera

camera = PiCamera()
camera.resolution = (1280, 720)
camera.start_preview()
sleep(2)
camera.capture('/home/pi/Pictures/newImage.jpg')
camera.stop_preview()


#end of code
```

Practical No : 5
Controlling Stepper Motor with Raspberry Pi

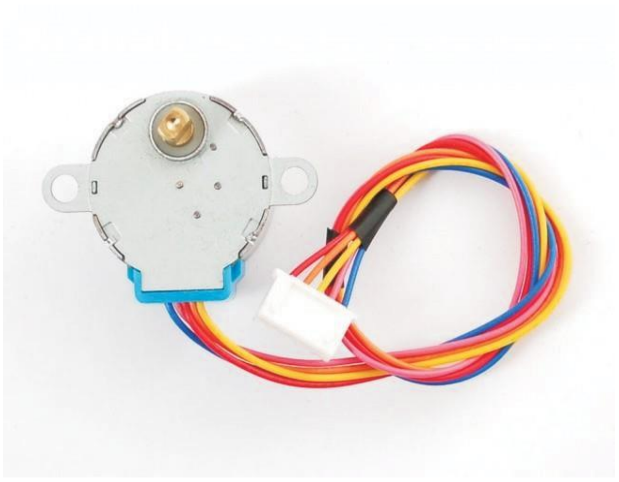
Hardware Guide:

For completing this lesson, you will require the following things along with your init ial raspberry pi setup

1. Stepper Motor
2. Stepper motor driver
3. Connecting wires

Stepper Motor:

This is a great first stepper motor, good for small projects and experimenting with steppers. This uni - polar motor has a built in mounting plate with two mounting holes. There are only 32 step (11.25 degree) per revolution, and inside is a 1/16 reduction gear set. (Actually its 1/16.032 but for most purposes 1/16 is a good enough approximation) What this means is that there are really 32*16.032 steps per revolution = 513 steps! The shaft is flattened so it's easy to attach stuff to it with a set-screw.

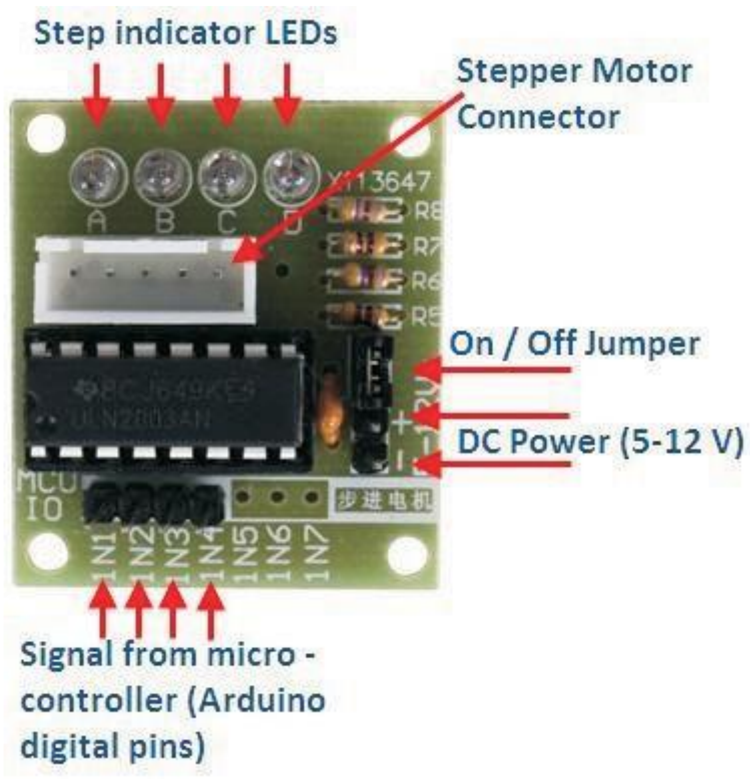


Technical details:

1. Unipolar stepper with 0.1" spaced 5-pin cable connector
2. 513 steps per revolution
3. 1/16.032 geared down reduction
4. 5V DC suggested operation
5. Weight: 37 g.
6. Dimensions: 28mm diameter, 20mm tall not including 9mm shaft with 5mm diameter
7. 9" / 23 cm long cable
8. Holding Torque: 150 gram -force*cm, 15 N*mm/ 2 oz-force*in
9. Shaft: 5mm diameter flattened
10. Approx. 42-ohm DC impedance per winding

Stepper motor driver board:

The ULN2003 stepper motor driver board allows you to easily control t he 28BYJ-48 stepper motor. One side of the board side has a 5-wire socket where the cable from the stepper motor hooks up and 4 LEDs to indicate which coil is currently powered. The motor cable only goes in one way, wh ich always helps. On the side, you have a motor on / off jumper (keep it on to enable power to the stepper). The two pins below the 4 resistors, is where you provide power to the stepper. Note that powering the stepper from the 5 V rail of the Raspberry Pi is not recommended. A separate 5-12 V 1 Amp power supply or battery pack should be used, as the motor may drain more current than the microcontroller can handle and could potentially damage it. In the middle of the bo ard we have the ULN2003 chip. At the bottom are the 4 control inputs that should be connected to four GPIO pins of raspberry pi.



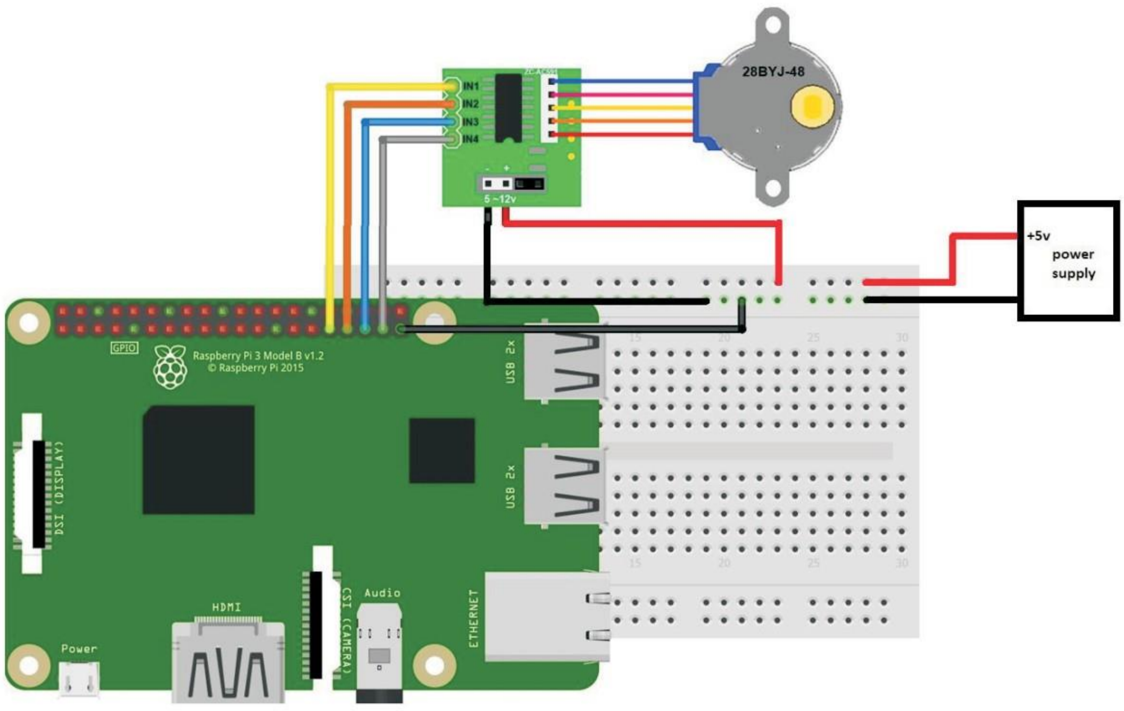
Wiring up Your circuit:

Wire your circuit as follows

1. Connect the IN1 pin of driver board to Physical Pin31 of raspberry pi
2. Connect the IN2 pin of driver board to Physical Pin33 of raspberry pi
3. Connect the IN3 pin of driver board to Physical Pin35 of raspberry pi
4. Connect the IN4 pin of driver board to Physical Pin37 of raspberry pi
5. Connect the + pin of driver board to and external 5v power supply
6. Connect the – (ground) pin of driver board, the ground pin of power supply and the groundpin of raspberry pi (Physical Pin6)to each other using a bread board.

Now connect the Stepper motor to the stepper motor connector on the driver board. It will fit only in one way. After ensuring that the connections are proper, power on your raspberry pi

Circuit diagram:



Software Guide:

Now open Python3, navigate to files and create a new file write the code as follows:

Code:

```
 #import GPIO and time library
 import RPi.GPIO as GPIO
 import time
 GPIO.setmode(GPIO.BOARD)                    #set the mode to be used (BCM/BOARD)
 stepPin1 = 31                               #define the pins used for
 stepper motorstepPin2 = 33
 stepPin3 = 35
 stepPin4 = 37
 GPIO.setup(stepPin1, GPIO.OUT)              #set the GPIO pins
 as outputGPIO.setup(stepPin2, GPIO.OUT)
 GPIO.setup(stepPin3, GPIO.OUT)
 GPIO.setup(stepPin4, GPIO.OUT)
 GPIO.output(stepPin1, False)
 GPIO.output(stepPin2, False)
 GPIO.output(stepPin3, False)
 GPIO.output(stepPin4, False)
def  singleStep(stepVal1,  stepVal2,  stepVal3,  stepVal4):  #defining  function
 GPIO.output(stepPin1, stepVal1)
 GPIO.output(stepPin2,     stepVal2)
 GPIO.output(stepPin3,     stepVal3)
```

```
    GPIO.output(stepPin4, stepVal4)
def clockWiseRotate(delay, steps 1):#defining functionfor i in range (0,
    steps1):
    singleStep(1, 0, 0, 0)
    time.sleep(delay)
    singleStep(1, 1, 0, 0)
    time.sleep(delay)
    singleStep(0, 1, 0, 0)
    time.sleep(delay)
    singleStep(0, 1, 1, 0)
    time.sleep(delay)
    singleStep(0, 0, 1, 0)
    time.sleep(delay)
    singleStep(0, 0, 1, 1)
    time.sleep(delay)
    singleStep(0, 0, 0, 1)
    time.sleep(delay)
    singleStep(1, 0, 0, 1)
    time.sleep(delay)
def anticlockWiseRotate(delay, steps2):
for i in range (0, steps2):
    singleStep(1, 0, 0, 1)
    time.sleep(delay)
    singleStep(0, 0, 0, 1)
    time.sleep(delay)
    singleStep(0, 0, 1, 1)
    time.sleep(delay)
    singleStep(0, 0, 1, 0)
    time.sleep(delay)
    singleStep(0, 1, 1, 0)
    time.sleep(delay)
    singleStep(0, 1, 0, 0)
    time.sleep(delay)
    singleStep(1, 1, 0, 0)
    time.sleep(delay)
    singleStep(1, 0, 0, 0)
    time.sleep(delay)
    try:
    while 1:                                                                    #infinite loop
    delay = input("Enter delay betwwen steps (in miliseconds): ")
    steps1 = input("How many steps clockwise?: ")
    steps2 = input("How many steps Anticlcokwise?: ")
    clockWiseRotate(int(delay)/1000.0, int(steps2))
    anticlockWiseRotate(int(delay)/1000.0, int(steps2))
    finally:
    GPIO.cleanup()
```

Practical 6
Node RED: Connect LED to Internet of Things

Node-RED is a drag-and-drop visual tool which comes pre-installed on Raspbian. In this lesson, we will use Node-RED to control LEDs via the Raspberry Pi's GPIO pins.

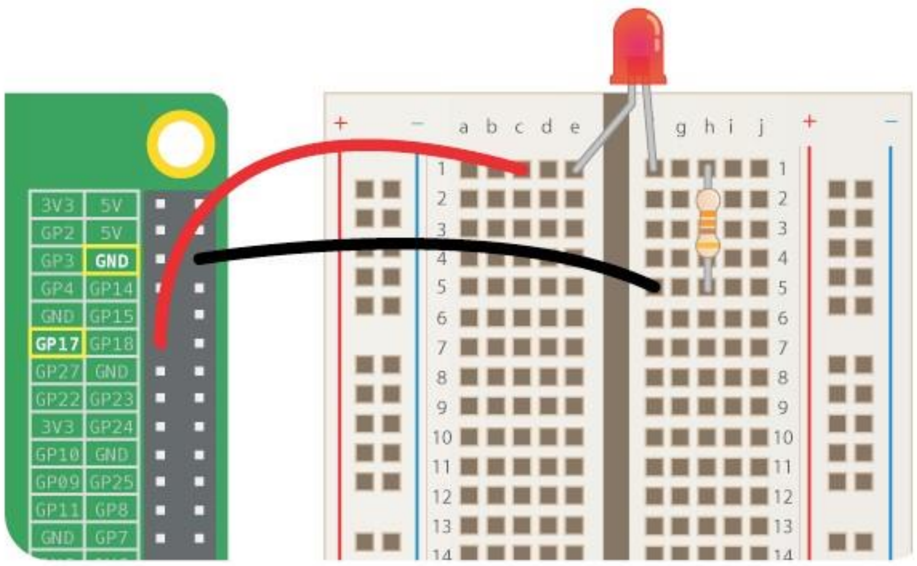
Hardware Guide:

Along with the basic setup you will require the following components to get started with the Node RED as follows:

1. LED
2. Resistor
3. Connecting wires
4. Breadboard

Wiring up the circuit:

1. Connect the raspberry pi to internet by connecting ethernet cable to the ethernet port or by connecting the on board WIFI module to the router.
2. Wire up an LED to GPIO17 (i.e. Physical pin11) on your Raspberry Pi



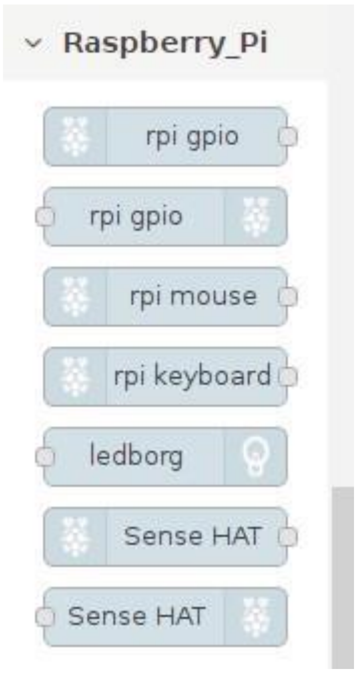
Software Guide:

1. Start up your Raspberry Pi. Click on the Raspberry icon, then the Programming menu to openNode-RED.
2. You should see a window displaying information about Node -RED starting up.
3. Now go to the Internet menu and open Chromium Web Browser.
4. In Chromium, locate the address bar at the top and type in localhost:1880, then press Enter.This will display the Node-RED interface. (Your Raspberry Pi does not need to be connected to the internet to use Node-RED: localhost is the address the Raspberry Pi uses to refer to itself and :1880 means that it is looking at port 1880.)

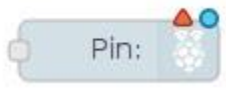
Programming in Node RED:

Programs in Node-RED are called flows. You can see that your blank page is labelled as Flow 1 in the tab at the top. You can create as many flows as you want and they can all run at the same time. For this guide, we will only need one flow.

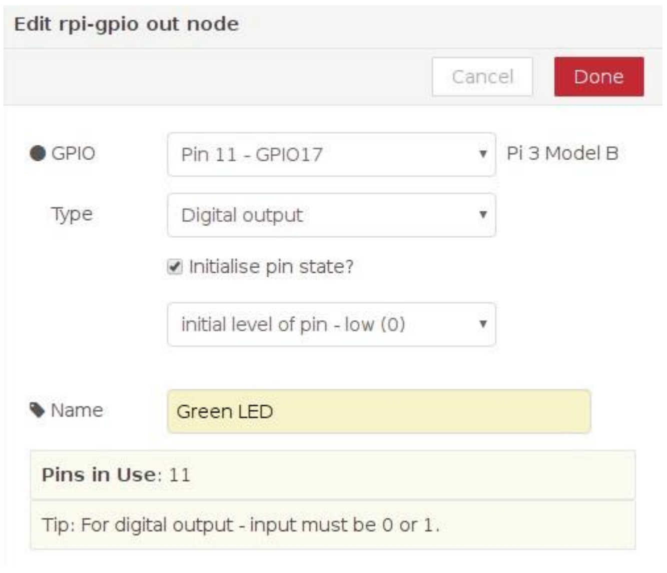
1. The coloured blocks on the left side of the interface are the nodes. Scroll right down to thebottom of the list and you will see some nodes labelled Raspberry Pi.



2. You will see two nodes with the label rpi gpio: these are the ones we will use to talk to the GPIO pins on the Raspberry Pi. The first one in the list, with the raspberry icon on the left, isfor inputs. Using a button push to control something would be an example of an input . The second node, with the raspberry icon on the right, is for outputs. Switching on an LED wouldbe an example of an output. Drag an output node onto the blank page in the middle.



3. Double-click on the node and a box will appear to let you configure the node. Change the GPIO pin to be GPIO17 and tick Initialise pin state? Leave the setting for Initial level of pin onlow. Give the node a name - we called it Green LED because the LED we used was green, butif yours is a different colour feel free to change the name. When you are finished, click Done.



Edit rpi-gpio out node

GPIO: Pin 11 - GPIO17 — Pi 3 Model B

Type: Digital output

☑ Initialise pin state?

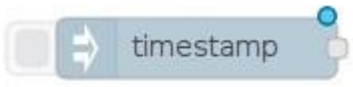
initial level of pin - low (0)

Name: Green LED

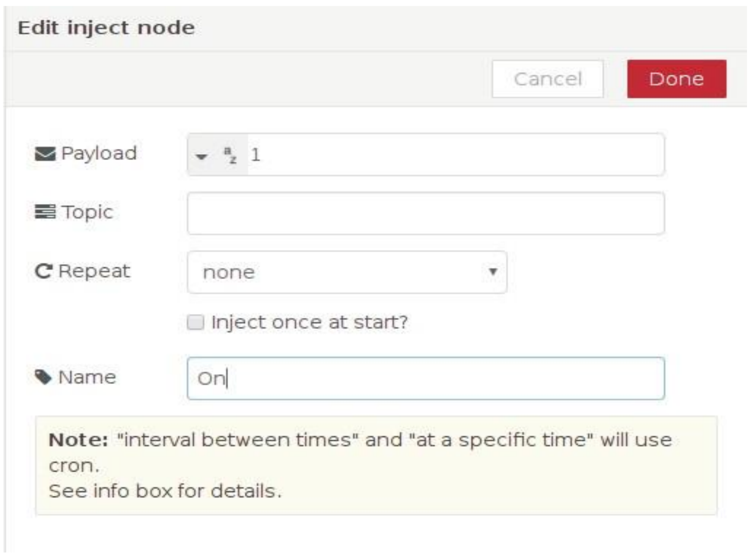
Pins in Use: 11

Tip: For digital output - input must be 0 or 1.

4. Now scroll back up to the list of nodes. To turn the LED on and off, we need an input. In Node-RED we can inject messages into the flow and cause things to happen as a result. Dragan inject node onto the flow.



5. Double-click on the inject node. Use the drop down next to Payload to change the data typeto string and type 1 in the Payload box - this will be the message. Type On in the Name box.Press Done.



Edit inject node

Payload: 1

Topic:

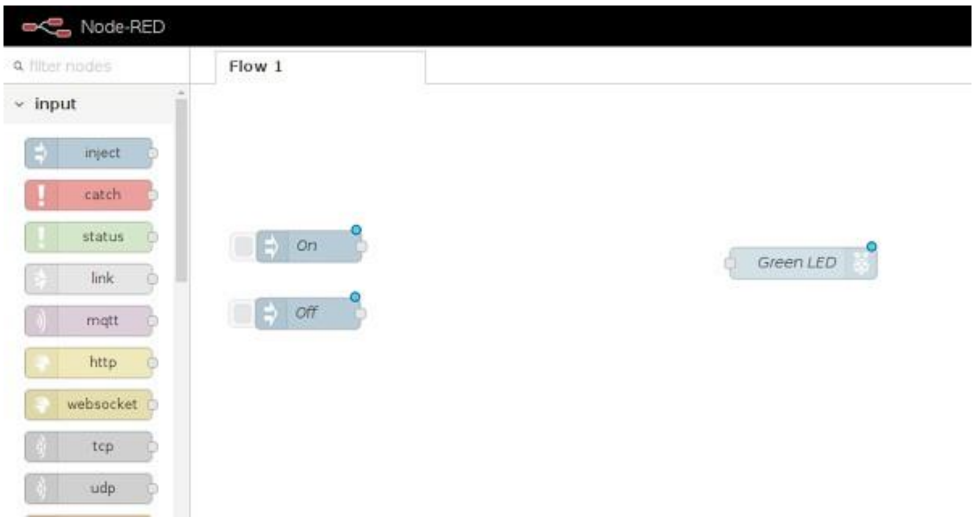
Repeat: none

☐ Inject once at start?

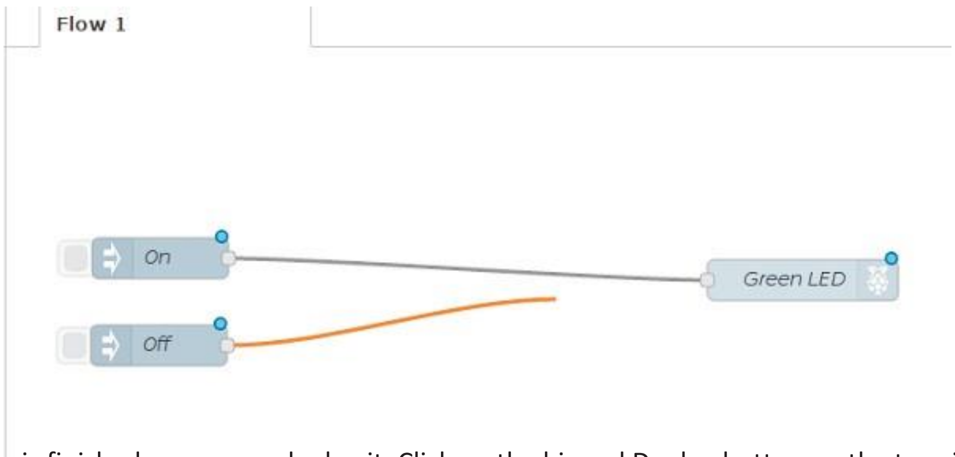
Name: On

Note: "interval between times" and "at a specific time" will use cron.
See info box for details.

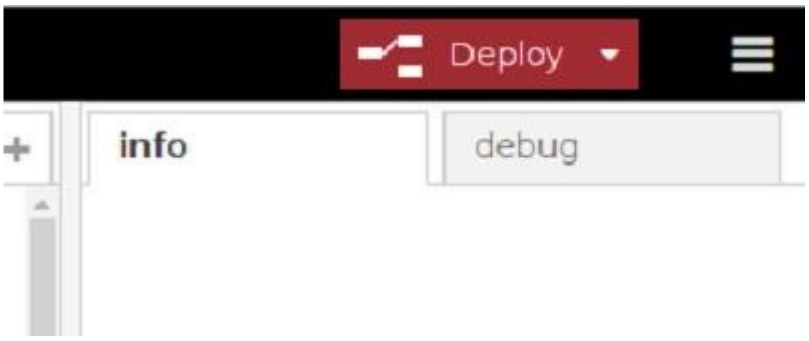
6. Repeat the previous steps to create another inject node, except this time add 0 as thepayload message, and call this node Off.



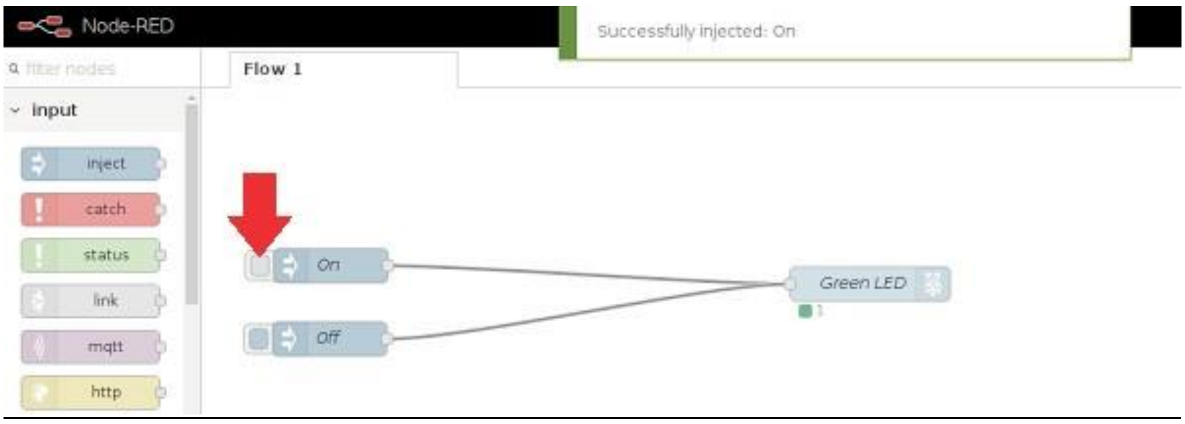
7. Now look for the grey dot on the right side of the inject nodes. Click and drag from the greydot on the On node to the grey dot on your LED node to join them up. Repeat for the Off node, also joining it to the LED node.



8. Our flow is finished, so we can deploy it. Click on the big red Deploy button on the top right of the screen. A message should pop up at the top saying "Successfully deployed". This is likepressing the green flag in Scratch or F5 to run your code in Python.



9. Now click on the blue square on the left of the On node to inject the message 1. The Green LED node receives the message and your LED should light up. You should be able to turn theLED off by clicking the blue square on the Off node, which injects the message 0.


Node-RED
Successfully injected: On
filter nodes
Flow 1
input
inject
catch
status
link
mqtt
http
On
Off
Green LED
1