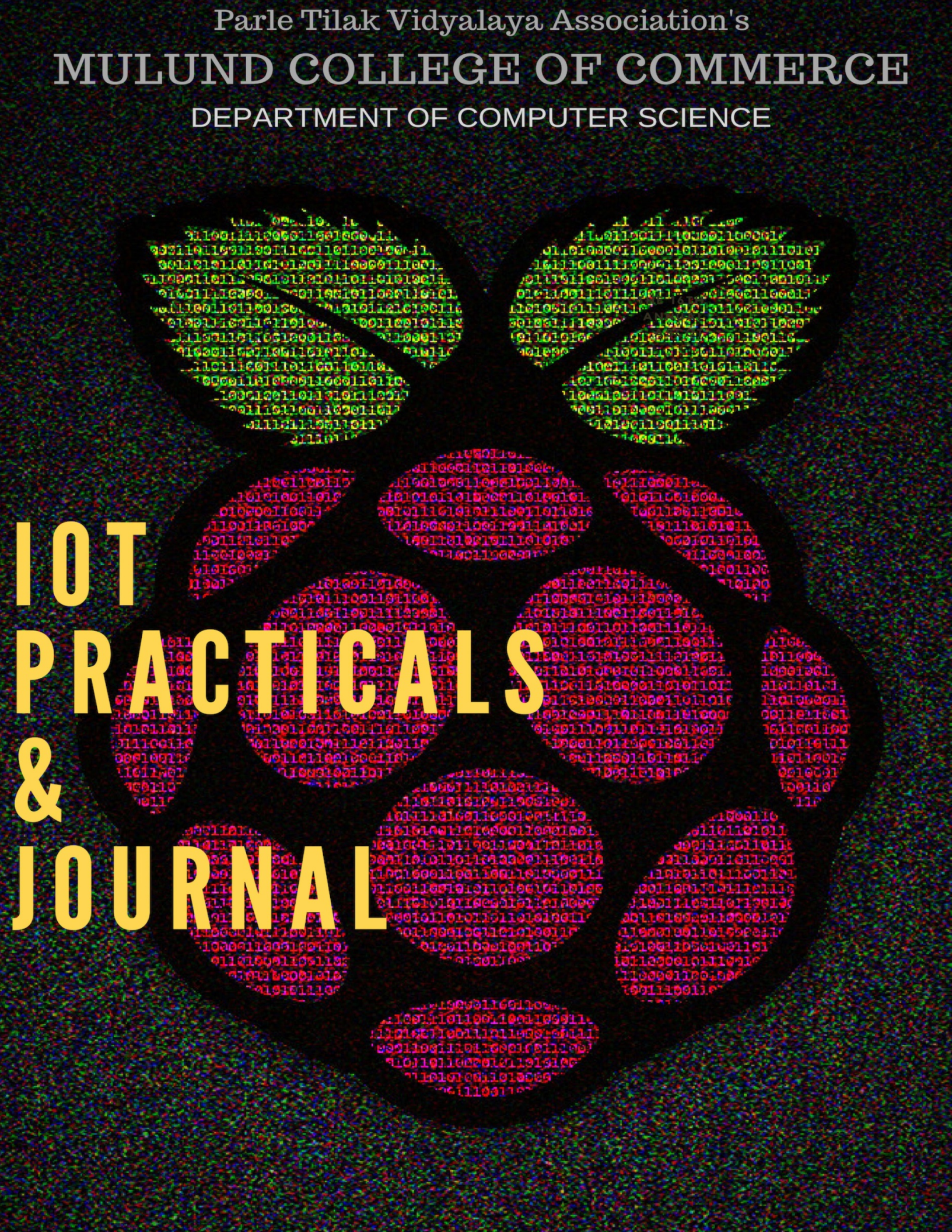
Parle Tilak Vidyalaya Association's

MULUND COLLEGE OF COMMERCE

DEPARTMENT OF COMPUTER SCIENCE



IOT PRACTICALS & JOURNAL

Table of Contents

PRACTICAL NO . 1

INTRODUCTION TO RASPBERRY PI

This is an introduction of Raspberry Pi 3 Model B. We will explore what Raspberry Pi 3 has to offer in terms of its features and performance. Raspberry Pi as a world's most inexpensive and powerful Single Board Computer. Ever since the launch of Raspberry Pi from 2012, we have seen several version of it. This is world's cheapest microprocessor unit specially built for learner and makers. We can easily learn how software and hardware work together without been worrying about damage/cost. We can buy Raspberry Pi board with just somewhere around 35$. The cost of Pi allows newbies to celebrate mistakes and learn most out of it. Also Raspberry Pi has a huge community and plenty of online resources which make learning smooth.

- **Raspberry Pi 3 Model B Features** : CPU: Raspberry Pi 3 uses Broadcom BCM2837 SOC 64-bit quad-core ARM Cortex A53 (ARMv8 CPU) with 512KB shared L2 cache.

- **Memory** : Provided with 1 GB of RAM Wi-Fi Support: 802.11n .

- **Wireless LAN Bluetooth**: Supports Bluetooth 4.1 Bluetooth Low Energy (BLE).

- **USB Ports** : 4-USB ports which allow attaching four different USB devices like keyboard, mouse, etc.

- **Ethernet Port** : Standard Ethernet port to quickly setup and access internet. This can be very useful when we want to setup raspberry pi for the first time without a monitor.

- **GPIO Pins**: Raspberry Pi 3 supports 40 GPIO Pins General Purpose Input Output. These digital input/output pins can be used to drive LED, Switches, and Sensors etc.

- **Full HDMI Port**: Support HDMI port (High-Definition Multimedia Interface) which can be used to quickly connect raspberry pi to HDMI Monitor. With HDMI Cable and Monitor we can add Screen to Raspberry Pi.

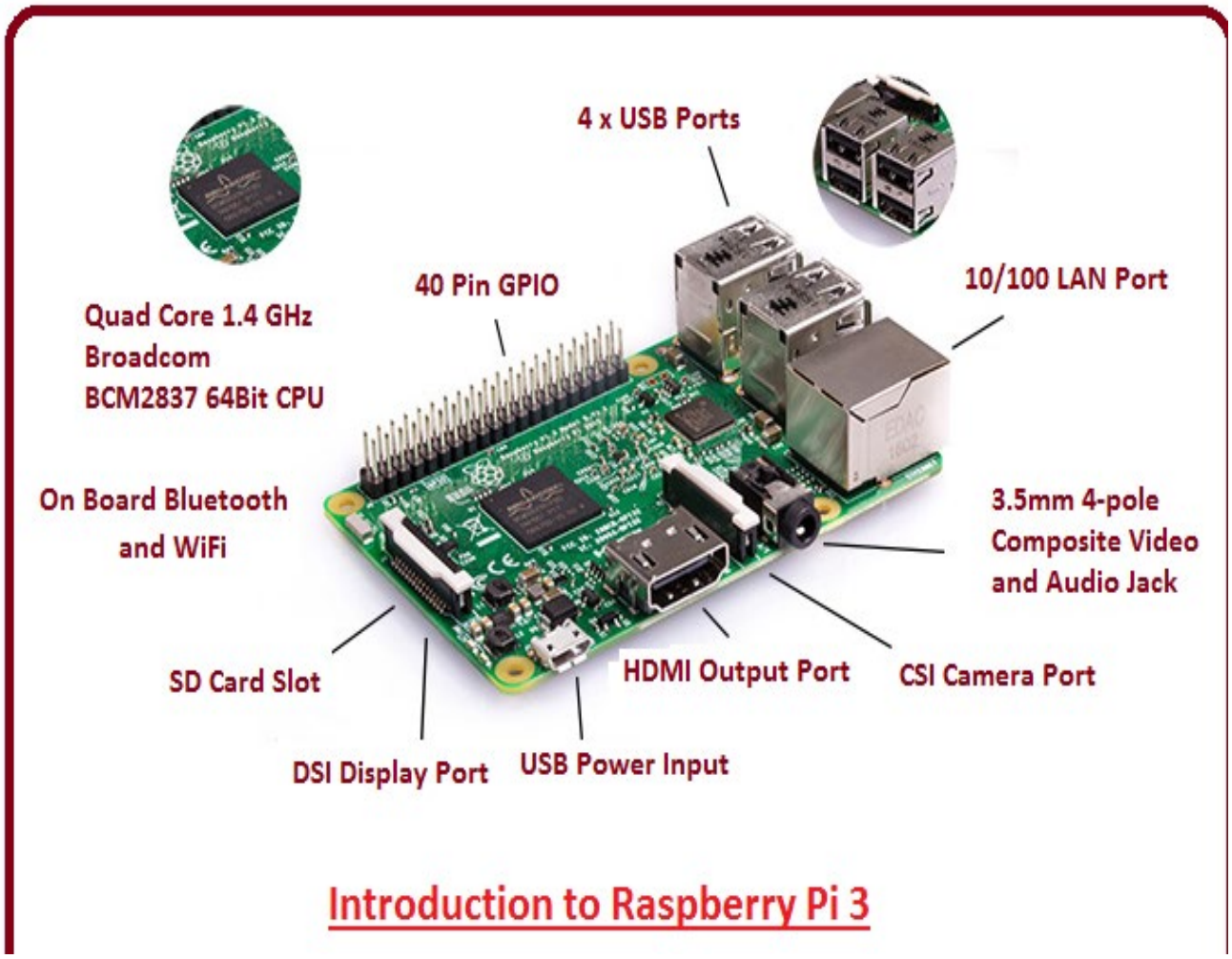
- **Micro SD card slot**: The Micro SD Card will hold the operating system which will boot while we power on Raspberry Pi 3. In next tutorial, we will learn how to setup and prepare SD card with Raspbian OS.

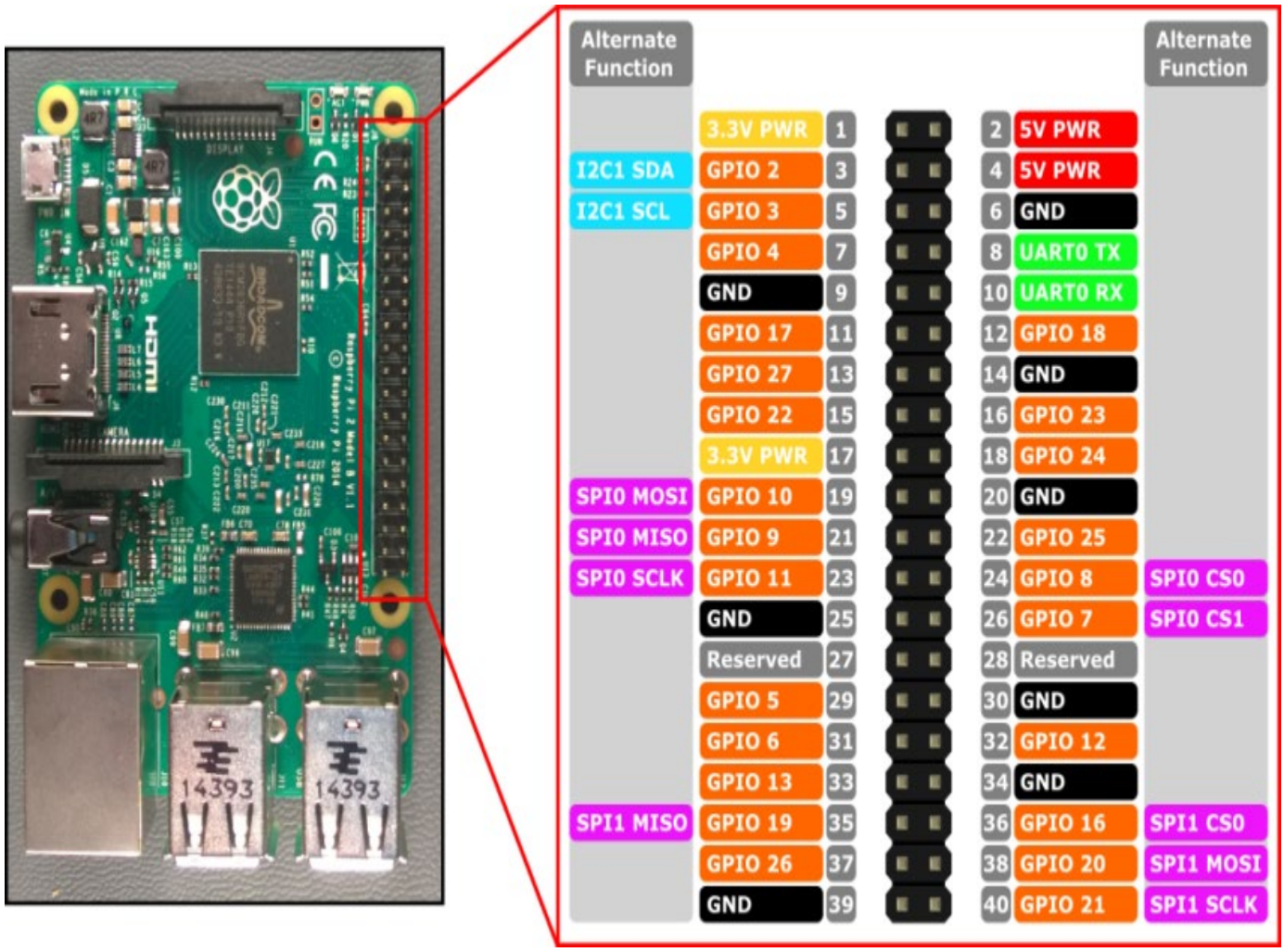
- **Audio/Video**: Combined 3.5mm audio jack and composite video .

- **Display interface (DSI):** enable us to interface Display Module .

- **Camera interface (CSI):** enable us to interface Camera Module

- **Graphics Support**: VideoCore IV 3D graphics core for advance graphics capabilities.



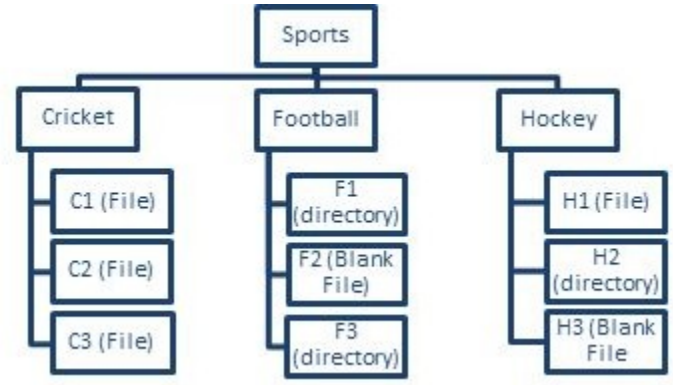

Alternate Function
Alternate Function
3.3V PWR 1
2 5V PWR
I2C1 SDA GPIO 2 3
4 5V PWR
I2C1 SCL GPIO 3 5
6 GND
GPIO 4 7
8 UART0 TX
GND 9
10 UART0 RX
GPIO 17 11
12 GPIO 18
GPIO 27 13
14 GND
GPIO 22 15
16 GPIO 23
3.3V PWR 17
18 GPIO 24
SPI0 MOSI GPIO 10 19
20 GND
SPI0 MISO GPIO 9 21
22 GPIO 25
SPI0 SCLK GPIO 11 23
24 GPIO 8 SPI0 CS0
GND 25
26 GPIO 7 SPI0 CS1
Reserved 27
28 Reserved
GPIO 5 29
30 GND
GPIO 6 31
32 GPIO 12
GPIO 13 33
34 GND
SPI1 MISO GPIO 19 35
36 GPIO 16 SPI1 CS0
GPIO 26 37
38 GPIO 20 SPI1 MOSI
GND 39
40 GPIO 21 SPI1 SCLK

PRACTICAL NO.2

Linux Commands: Exploring the Raspbian

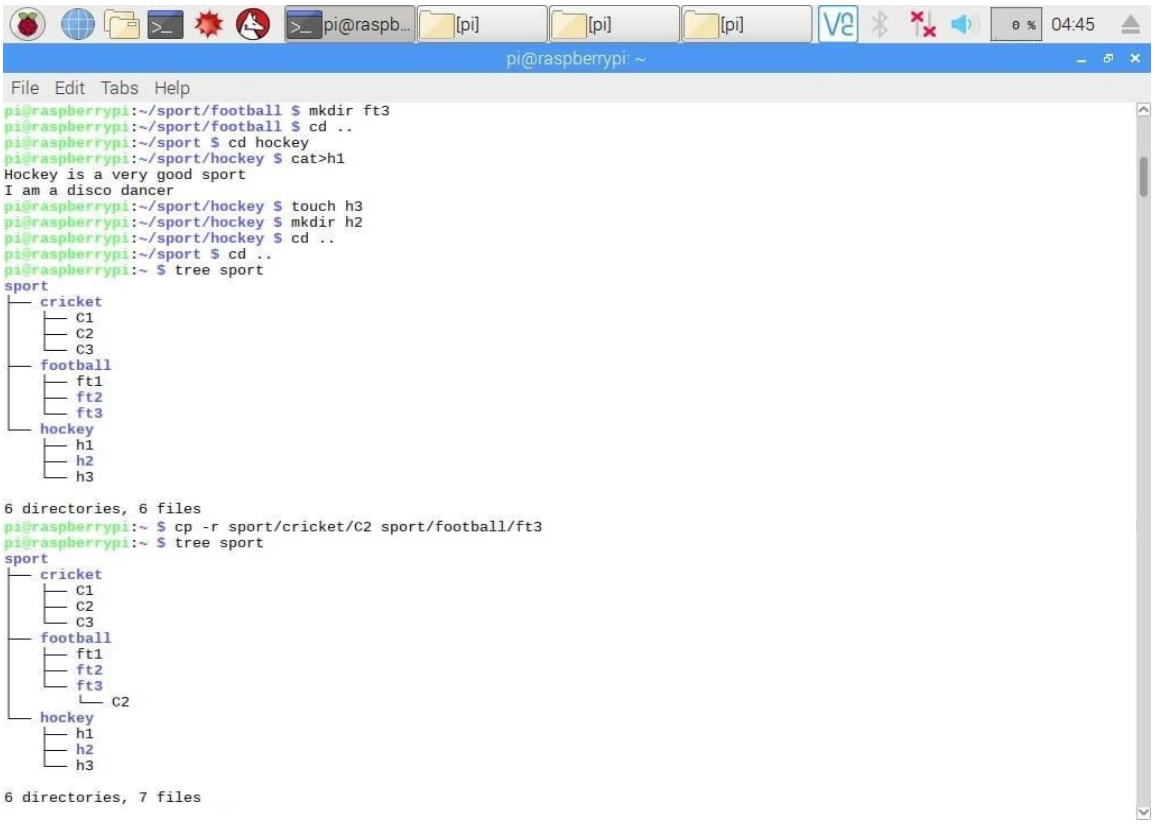
Q1].

a. Create a structure as given bellow by using Linux.



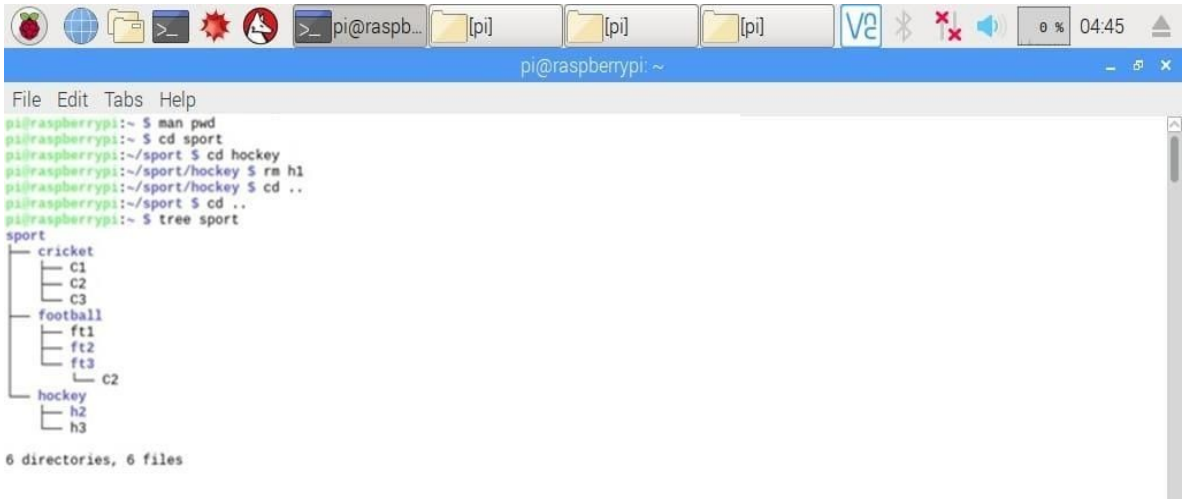

```
pi@raspberrypi: ~
File  Edit  Tabs  Help
pi@raspberrypi:~ $ mkdir sport
pi@raspberrypi:~ $ cd sport
pi@raspberrypi:~/sport $ mkdir cricket
pi@raspberrypi:~/sport $ mkdir football
pi@raspberrypi:~/sport $ mkdir hockey\
> ;
pi@raspberrypi:~/sport $ mkdir hockey
mkdir: cannot create directory ‘hockey’: File exists
pi@raspberrypi:~/sport $ cd cricket
pi@raspberrypi:~/sport/cricket $ cat<C1>
bash: syntax error near unexpected token `newline'
pi@raspberrypi:~/sport/cricket $ cat <C1>
bash: syntax error near unexpected token `newline'
pi@raspberrypi:~/sport/cricket $ cat<C1
bash: C1: No such file or directory
pi@raspberrypi:~/sport/cricket $ cat>C1
abcd
heman pawar
topper
vikram patil
pi@raspberrypi:~/sport/cricket $ cat>C2
prasad
pawar playboy
pi@raspberrypi:~/sport/cricket $ cat>C3
Heman is a very good boy.
He is the best
pi@raspberrypi:~/sport/cricket $ cd football
bash: cd: football: No such file or directory
pi@raspberrypi:~/sport/cricket $ cd..
bash: cd..: command not found
pi@raspberrypi:~/sport/cricket $ cd ..
pi@raspberrypi:~/sport $ cd football
pi@raspberrypi:~/sport/football $ cat>ft1
pi@raspberrypi:~/sport/football $ touch ft1
pi@raspberrypi:~/sport/football $ mkdir ft2
pi@raspberrypi:~/sport/football $ mkdir ft3
pi@raspberrypi:~/sport/football $ cd ..
pi@raspberrypi:~/sport $ cd hockey
pi@raspberrypi:~/sport/hockey $ cat>h1
Hockey is a very good sport
I am a disco dancer
pi@raspberrypi:~/sport/hockey $ touch h3
pi@raspberrypi:~/sport/hockey $ mkdir h2
pi@raspberrypi:~/sport/hockey $ cd ..
```

b. Display the complete structure of Sports directory.
c. Copy the file C2 from cricket to F3.



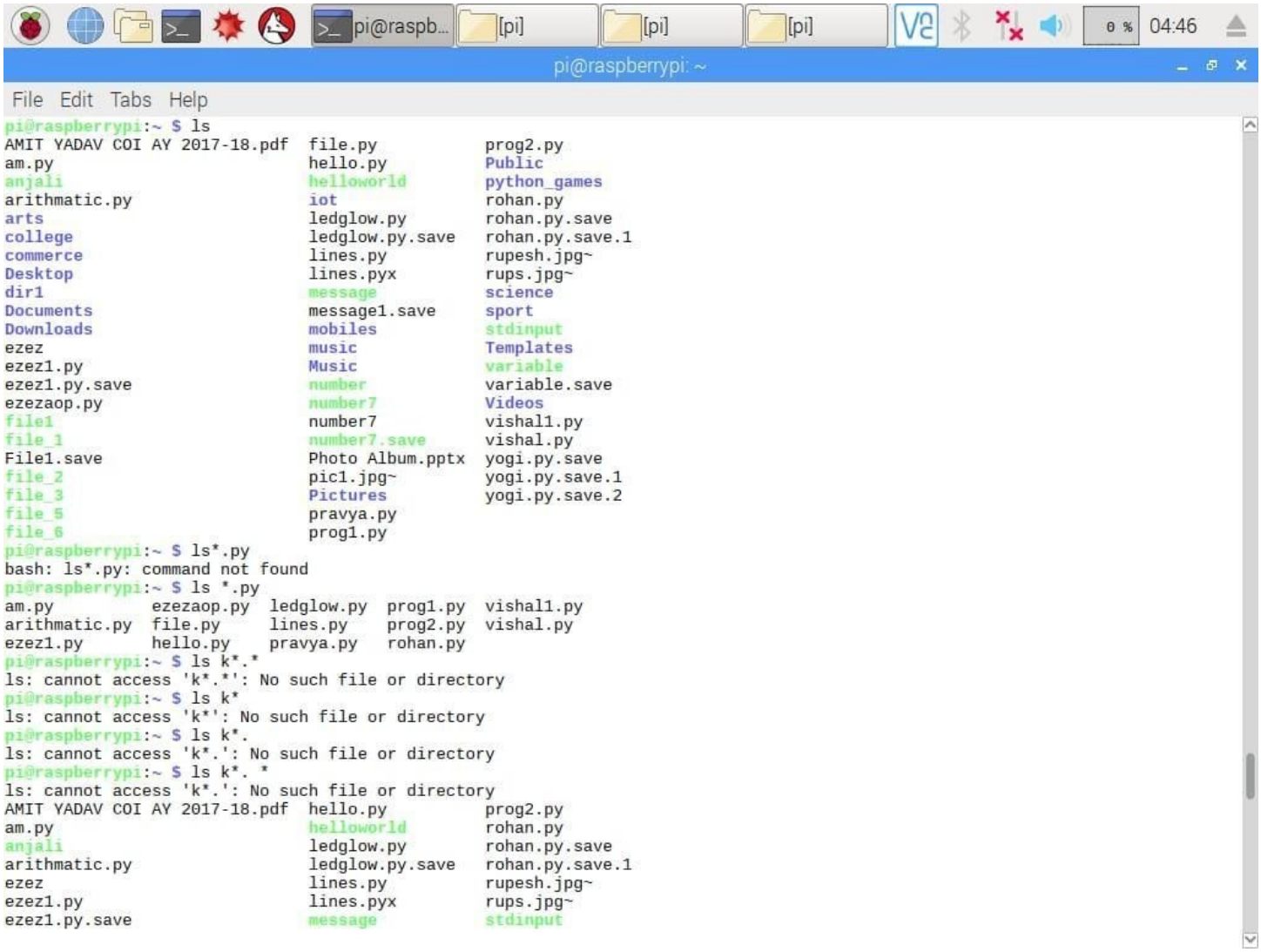
Q2].

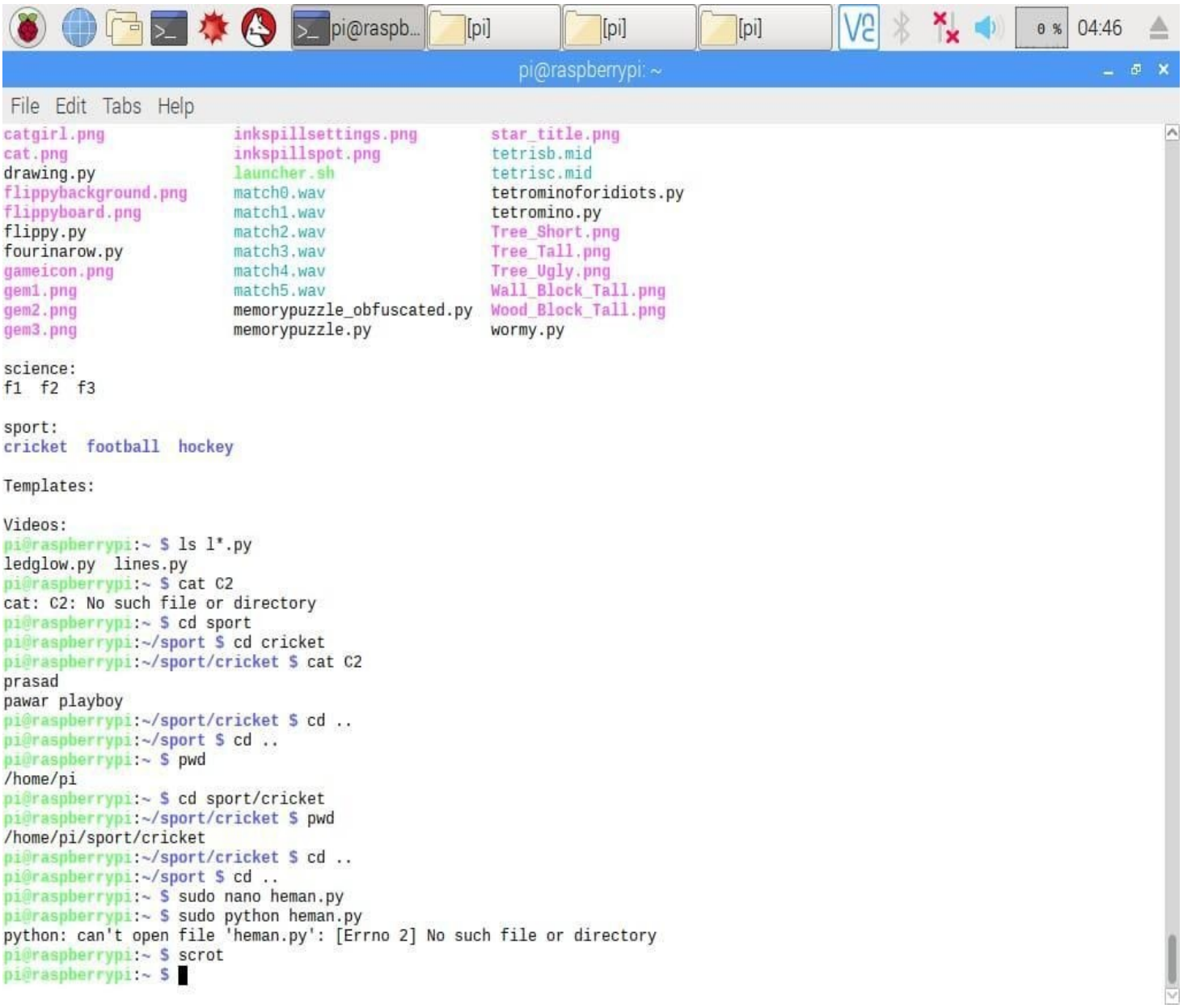
a. Display the help file for the command pwd.
b. Remove the file H1.



c. List all the files present under root directory .($ ls)
d. Display all python files.($ ls +.py)

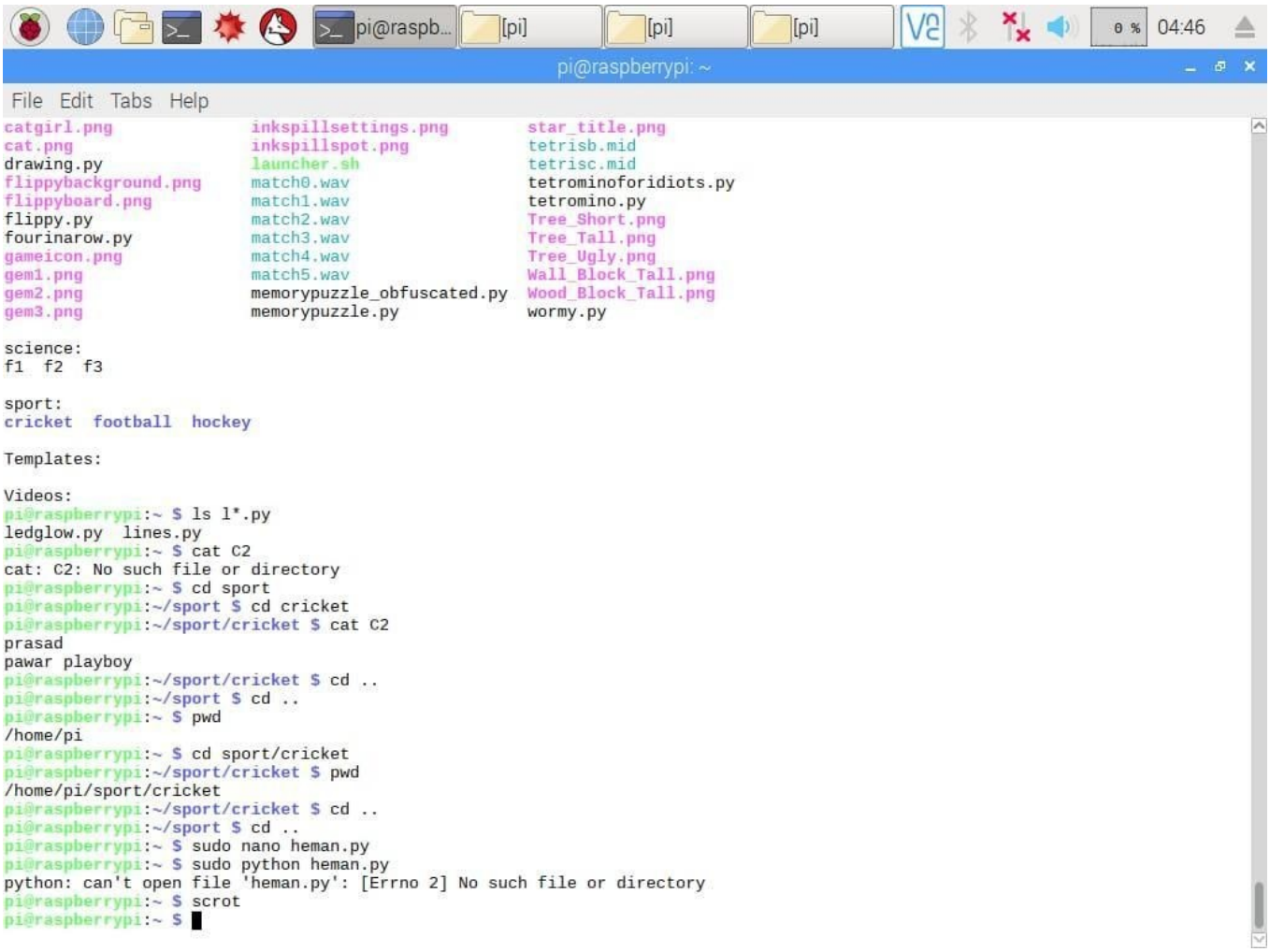
e. List all the files which starts with letter ‘k’.($ ls k**)
f. Display content of file C2.




pi@raspb...
[pi]
[pi]
[pi]
0 %
04:46
pi@raspberrypi: ~
File Edit Tabs Help
catgirl.png inkspillsettings.png star_title.png
cat.png inkspillspot.png tetrisb.mid
drawing.py launcher.sh tetrisc.mid
flippybackground.png match0.wav tetrominoforidiots.py
flippyboard.png match1.wav tetromino.py
flippy.py match2.wav Tree_Short.png
fourinarow.py match3.wav Tree_Tall.png
gameicon.png match4.wav Tree_Ugly.png
gem1.png match5.wav Wall_Block_Tall.png
gem2.png memorypuzzle_obfuscated.py Wood_Block_Tall.png
gem3.png memorypuzzle.py wormy.py
science:
f1 f2 f3
sport:
cricket football hockey
Templates:
Videos:
pi@raspberrypi:~ $ ls l*.py
ledglow.py lines.py
pi@raspberrypi:~ $ cat C2
cat: C2: No such file or directory
pi@raspberrypi:~ $ cd sport
pi@raspberrypi:~/sport $ cd cricket
pi@raspberrypi:~/sport/cricket $ cat C2
prasad
pawar playboy
pi@raspberrypi:~/sport/cricket $ cd ..
pi@raspberrypi:~/sport $ cd ..
pi@raspberrypi:~ $ pwd
/home/pi
pi@raspberrypi:~ $ cd sport/cricket
pi@raspberrypi:~/sport/cricket $ pwd
/home/pi/sport/cricket
pi@raspberrypi:~/sport/cricket $ cd ..
pi@raspberrypi:~/sport $ cd ..
pi@raspberrypi:~ $ sudo nano heman.py
pi@raspberrypi:~ $ sudo python heman.py
python: can't open file 'heman.py': [Errno 2] No such file or directory
pi@raspberrypi:~ $ scrot
pi@raspberrypi:~ $

Q3]. Write the commands :

a. Display pwd.
b. Write the steps to create a python file.
c. Write the shortcut for saving a python file, exit the window, come back to the terminal.
d. How to execute a python file ?



FILESYSTEM commands:

- LS : The ls command lists the content of the current directory (or one that is specified). It can be used with the -l flag to display additional information (permissions, owner, group, size, date and timestamp of last edit) about each file and directory in a list format. The -a flag allows you to view files beginning with . (i.e. dotfiles).

- CD : Using cd changes the current directory to the one specified. You can use relative (i.e. cd directoryA) or absolute (i.e. cd /home/pi/directoryA) paths.

- PWD : The pwd command displays the name of the present working directory: on a Raspberry Pi, entering pwd will output something like /home/pi.

- MKDIR : You can use mkdir to create a new directory, e.g. mkdir newDir would create the directory newDir in the present working directory.

- RMDIR : To remove empty directories, use rmdir. So, for example, rmdir oldDir will remove the directory oldDir only if it is empty.

- RM : The command rmremoves the specified file (or recursively from a directory when used with -r). Be careful with this command: files deleted in this way are mostly gone for good!

- CP : Using cp makes a copy of a file and places it at the specified location (this is similar to copying and pasting). For example, cp ~/fileA /home/otherUser/would copy the file fileA from your home directory to that of the user otherUser (assuming you have permission to copy it there). This command can either take FILE FILE (cp fileA fileB), FILE DIR(cp fileA /directoryB/) or -r DIR DIR (which recursively copies the contents of directories) as arguments.

- MV : The mv command moves a file and places it at the specified location (so where cp performs a 'copy-paste', mv performs a 'cut-paste'). The usage is similar to cp. So mv ~/fileA /home/otherUser/ would move the file fileA from your home directory to that of the user otherUser. This command can either take FILE FILE (mv fileA fileB), FILE DIR (mv

fileA /directoryB/) or DIR DIR (mv /directoryB /directoryC) as arguments. This command is also useful as a method to rename files and directories after they've been created.

- TOUCH : The command touch sets the last modified time-stamp of the specified file(s) or creates it if it does not already exist.

- CAT : You can use cat to list the contents of file(s), e.g. cat thisFile will display the contents of thisFile. Can be used to list the contents of multiple files, i.e. cat *.txt will list the contents of all .txt files in the current directory.

- HEAD : The head command displays the beginning of a file. Can be used with -n to specify the number of lines to show (by default ten), or with -c to specify the number of bytes.

- TAIL : The opposite of head, tail displays the end of a file. The starting point in the file can be specified either through -b for 512 byte blocks, -c for bytes, or -n for number of lines.

- CHMOD : You would normally use chmod to change the permissions for a file. The chmodcommand can use symbols u (user that owns the file), g (the files group) , and o (other users) and the permissions r (read), w (write), and x (execute). Using chmod u+x *filename* will add execute permission for the owner of the file.

- CHOWN : The chown command changes the user and/or group that owns a file. It normally needs to be run as root using sudo e.g. sudo chown pi:root *filename* will change the owner to pi and the group to root.

- SSH : ssh denotes the secure shell. Connect to another computer using an encrypted network connection. For more details see SSH (secure shell) .

- SCP : The scp command copies a file from one computer to another using ssh. For more details see SCP (secure copy) .

- SUDO : The sudo command enables you to run a command as a superuser, or another user. Use sudo -s for a superuser shell. For more details see Root user / sudo .

- DD : The dd command copies a file converting the file as specified. It is often used to copy an entire disk to a single file or back again. So, for example, dd if=/dev/sdd of=backup.img will create a backup image from an SD card or USB disk drive at /dev/sdd.

Make sure to use the correct drive when copying an image to the SD card as it can overwrite the entire disk.

- DF : Use df to display the disk space available and used on the mounted filesystems. Use df -h to see the output in a human-readable format using M for MBs rather than showing number of bytes.

- UNZIP : The unzip command extracts the files from a compressed zip file.

- TREE : Use the tree command to show a directory and all subdirectories and files indented as a tree structure. & Run a command in the background with &, freeing up the shell for future commands.

- WGET : Download a file from the web directly to the computer with wget.

- CURL :Use curl to download or upload a file to/from a server. By default, it will output the file contents of the file to the screen.

- MAN : Show the manual page for a file with man. To find out more, run man man to view the manual page of the man command.

SEARCH COMMANDS :

- GREP : Use grep to search inside files for certain search patterns. For example, grep "search" *.txt will look in all the files in the current directory ending with .txt for the string search. The find command searches a directory and subdirectories for files matching certain patterns.

- WHEREIS : Use whereis to find the location of a command. It looks through standard program locations until it finds the requested command.

NETWORKING COMMANDS :

- PING : The ping utility is usually used to check if communication can be made with another host. It can be used with default settings by just specifying a hostname (e.g.

ping raspberrypi.org) or an IP address (e.g. ping 8.8.8.8). It can specify the number of packets to send with the -c flag.

- NMAP : nmap is a network exploration and scanning tool. It can return port and OS information about a host or a range of hosts. Running just nmap will display the options available as well as example usage.

- HOSTNAME : The hostname command displays the current hostname of the system. A privileged (super) user can set the hostname to a new one by supplying it as an argument (e.g. hostname new-host).

- IFCONFIG : Use ifconfig to display the network configuration details for the interfaces on the current system when run without any arguments (i.e. ifconfig). By supplying the command with the name of an interface (e.g. eth0 or lo) you can then alter the configuration.

PRACTICAL NO. 3

GPIO: Light the LED with Python

Once you've setup your Raspberry Pi according to the getting started tutorial, you are ready for your first real project. Let's light up an led using the Python programming language and the GPIO pins on your Raspberry Pi, hereafter
called **RPi**.

What You Will Learn:

- You will construct a basic electrical circuit and attach it to your RPi GPIO pins
- You will write a simple Python program to control the circuit using IDLE IDE

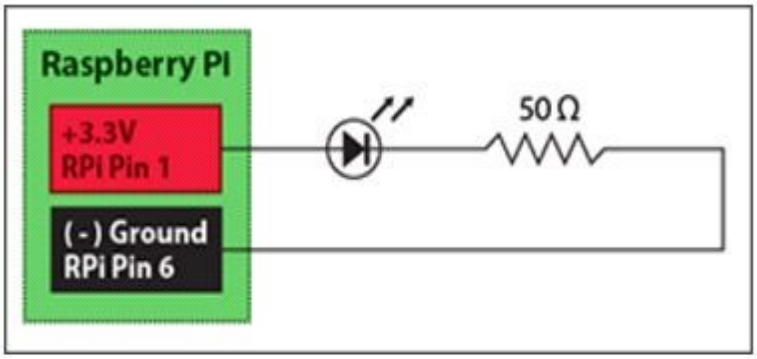
What You Will Need:

- Raspberry Pi configured with the GPIO library
- 1 - small led, any color
- 1 - 50 ohm resistor
- Some small-gauge solid core wire
- Breadboard and/or alligator clips to hold connections

Let There Be Light

Before we get around to writing any code, let's first get acquainted with the pin numbering of our RPi and create a simple circuit. We'll start by simply lighting our led using the 3.3v pin and the ground pin on our RPi. We will use

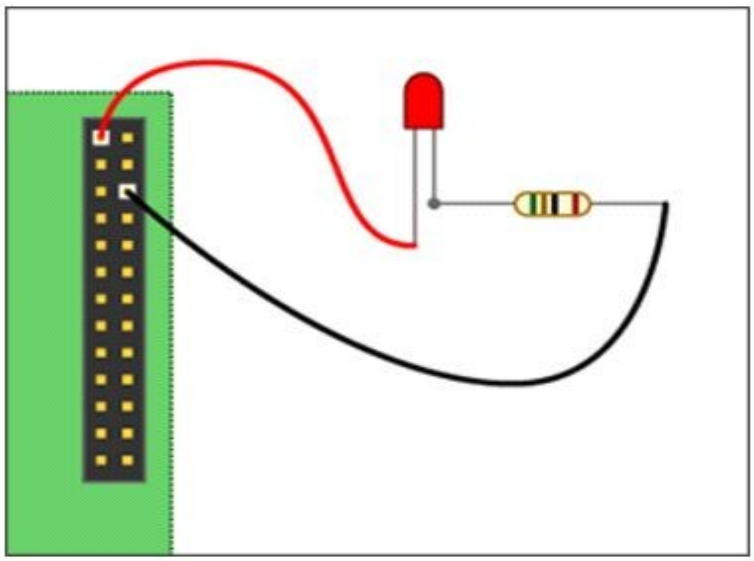
the following schematic for our circuit:



Before starting, unplug your RPi. You wouldn't want to risk shorting it out while working with it 'powered on', especially since this is our first project.

- Using your assortment of materials, create the circuit on either your breadboard, or using your alligator clips.
- Pin 1 (+3.3 volts) the longer leg of your led. This pin provides a steady supply of 3.3v. Unlike the GPIO pins on your RPi, this pin is not programmable, and cannot be controlled by software.
- Attach the shorter leg of the led to the resistor. Finally, attach the other should go to
- end of the resistor to Pin 6 (- ground) on your RPi.

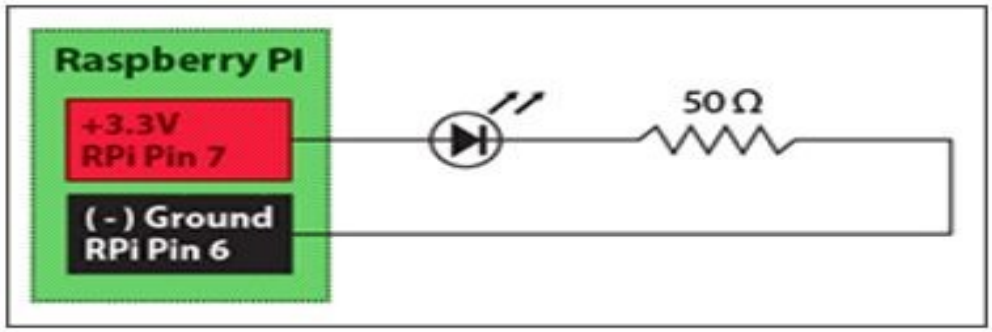
Double-check your connections. When you are done, your circuit should look like this:



- Power on your RPi - the led should immediately turn on.

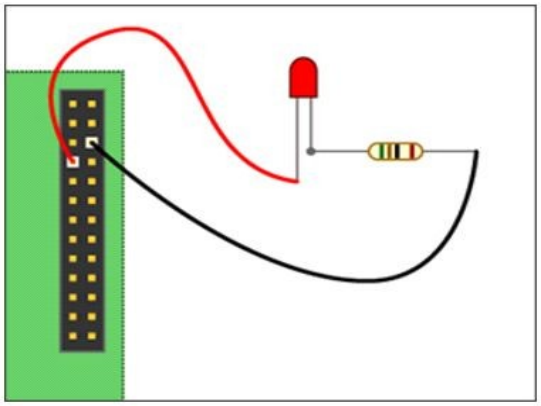
Controlling The Led With Python :

Now that we've tested our basic circuit, it's time to move the positive lead from the 'always on' 3.3v pin to one of the programmable GPIO pins. Here is what our circuit will look like:



- Power-off your RPi again before making any changes to your wiring.
- Move the positive lead from pin 1 to pin 7.

When you are done, your circuit should look like this:



Writing The Code

To write any python scripts using the IDLE IDE because it comes packaged with the Raspbian distribution, it's free, and it makes writing and debugging code a bit simpler than when using Python command line or a text editor. It's important to note that when writing python scripts that utilize the GPIO pins, you must run them as a **superuser** or your scripts will not run properly.

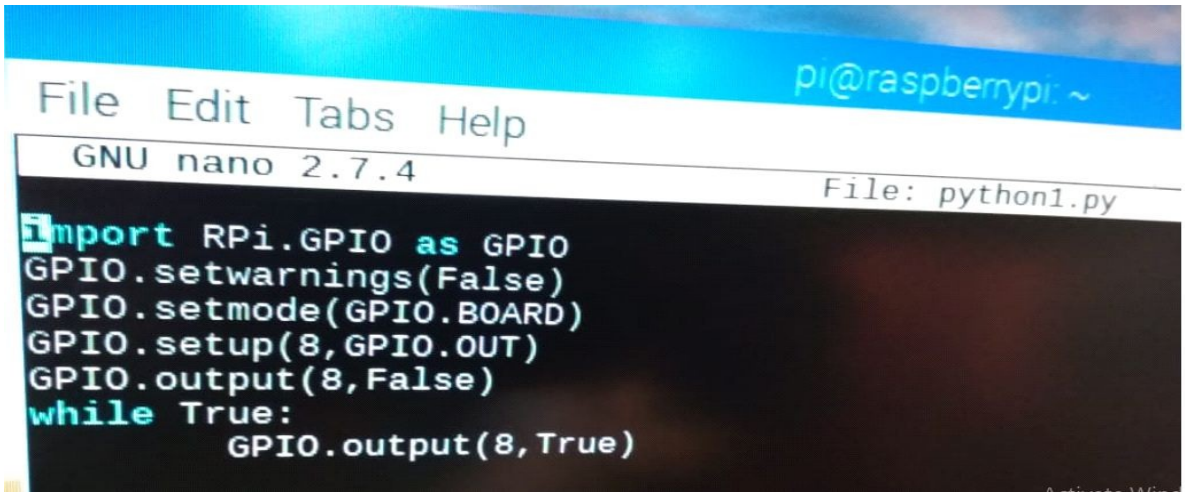
- Power on your RPi and boot all the way into the operating system GUI
- Open the terminal and launch the IDLE IDE as a superuser: `sudo idle`
- Wait for IDLE to open, then click **File > New** to open a new window (Ctrl + N)

- Type the code into the window
- Click **File > Save** when you are done (Ctrl + S).
- To run your program, click **Run > Run** (Ctrl + F5). You should see your led light up! We just told the RPi to supply voltage (+3.3v) to our circuit using GPIO pin 7.

If you are having trouble getting the led to light up, double-check your wiring, and make sure you have installed the GPIO Python library according to my instructions.

1) Write a Program to glow the LED at pin number 8 in BOARD mode.

- Type the command " sudo nano filename.py" to open the editor where you can write your python code.
- The below given image is the code for glowing the LED on.



After writing the code press "ctrl + 0" to save the code and "ctrl + x" to exit from the editor.

- Once, you are back on the terminal enter the command "sudo python filename.py" to run your program . If your connections are proper and the code is written correctly the LED will be glowing .
- The Output looks like this:

2) Write a Program to glow the LED at pin number 8 in BCM mode.

- **Program :**

```
GNU nano 2.7.4

import RPi.GPIO as GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(14,GPIO.OUT)
GPIO.output(14,False)
while True:
        GPIO.output(14,True)
```

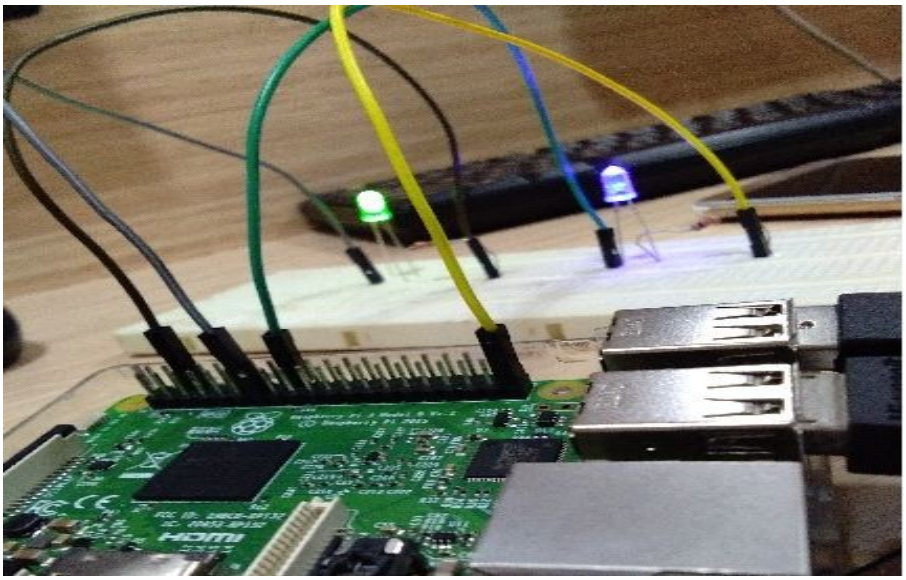
Output :

3) **Write a Program to glow any two LED's in BOARD mode**.

- **Program :**

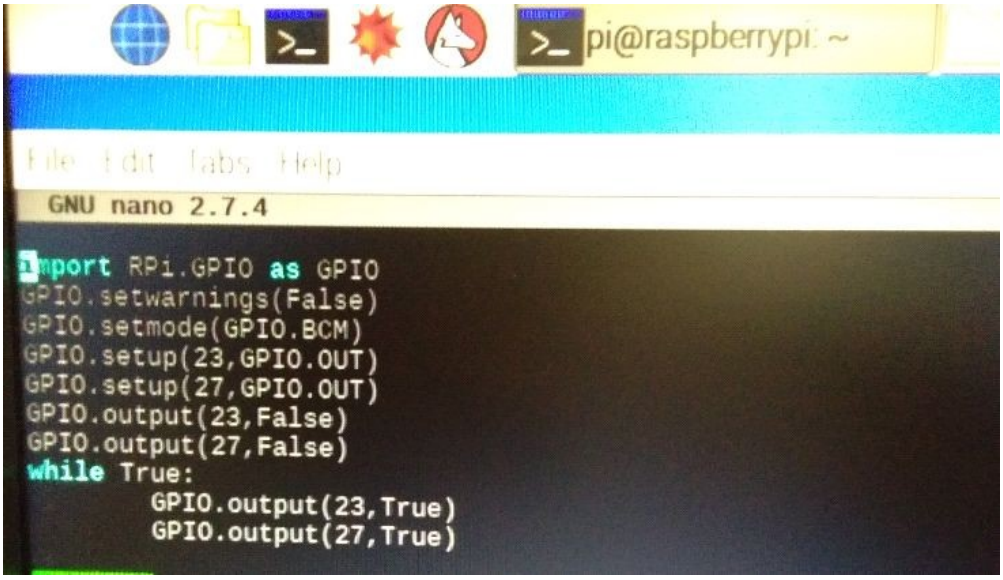
```
File  Edit  Tabs  Help
GNU nano 2.7.4

import RPi.GPIO as GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(15,GPIO.OUT)
GPIO.setup(18,GPIO.OUT)
GPIO.output(15,False)
GPIO.output(18,False)
while True:
        GPIO.output(15,True)
        GPIO.output(18,True)
```

Output :

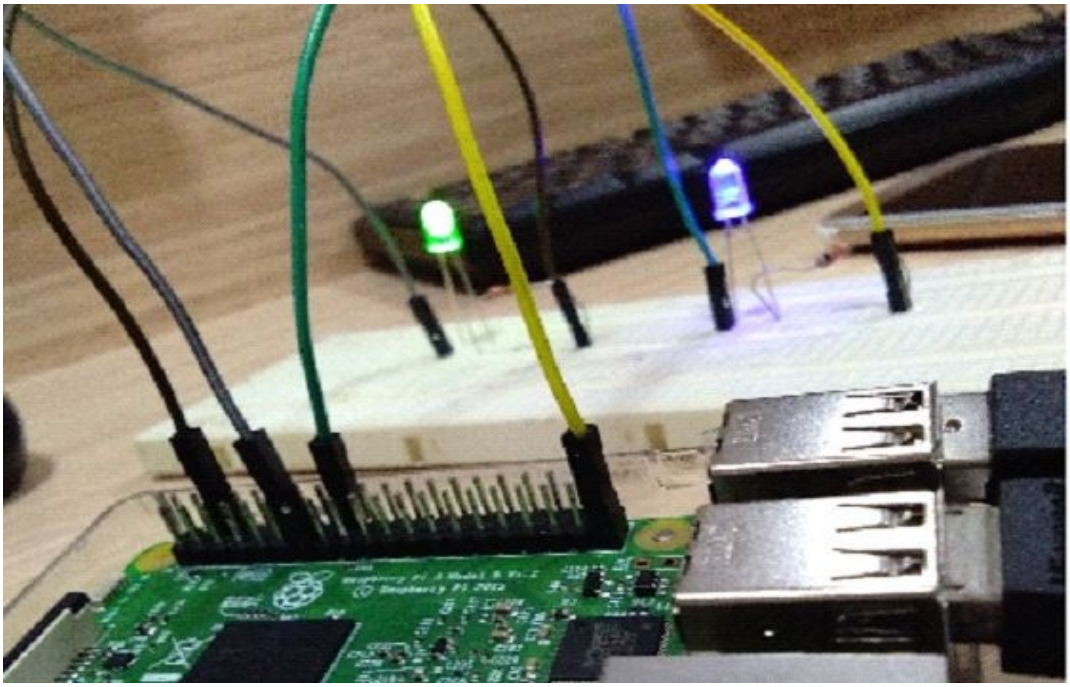


4) **Write a Program to glow any two LED's in BCM mode.**

- **Program :**



Output :

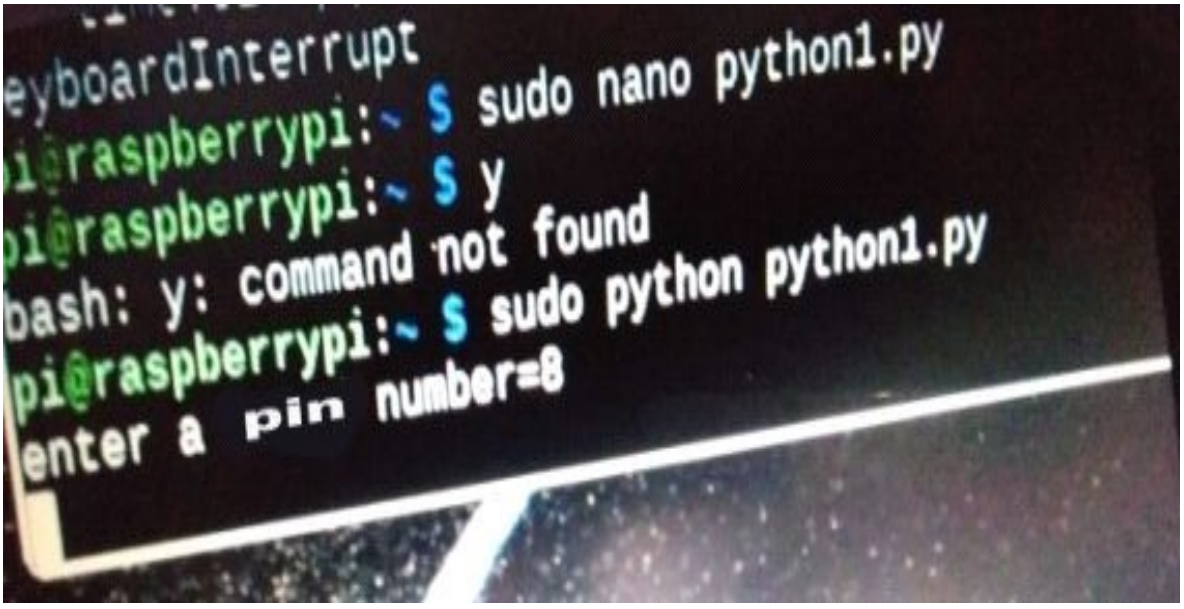


5) **Write a Program to glow the LED by accepting pin number from user in BOARD mode.**

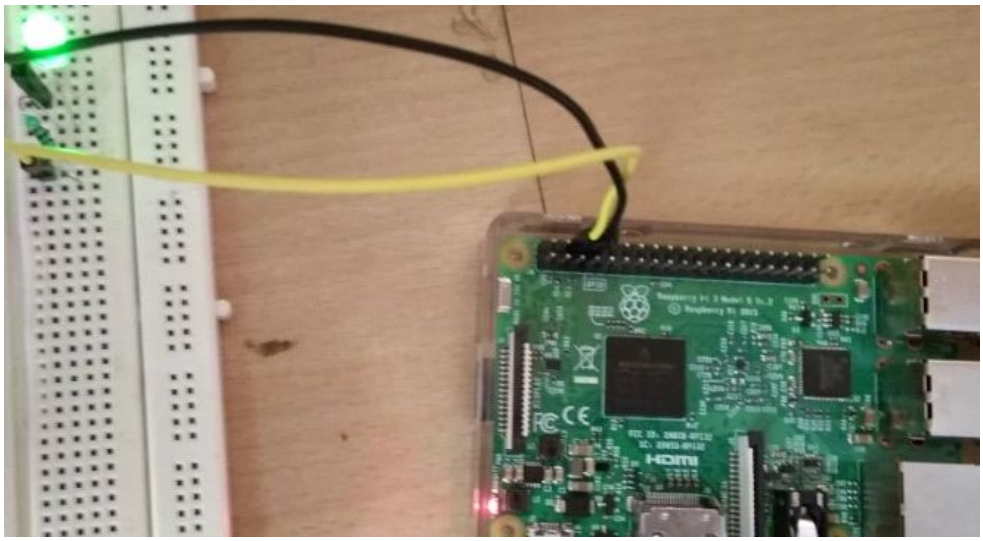
- **Program :**

```
File  Edit  Tabs  Help
  GNU nano 2.7.4

import RPi.GPIO as GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
x=int(input('Enter a pin number:'))
GPIO.setup(x,GPIO.OUT)
GPIO.output(x,False)
while True:
        GPIO.output(x,True)
```

Output :

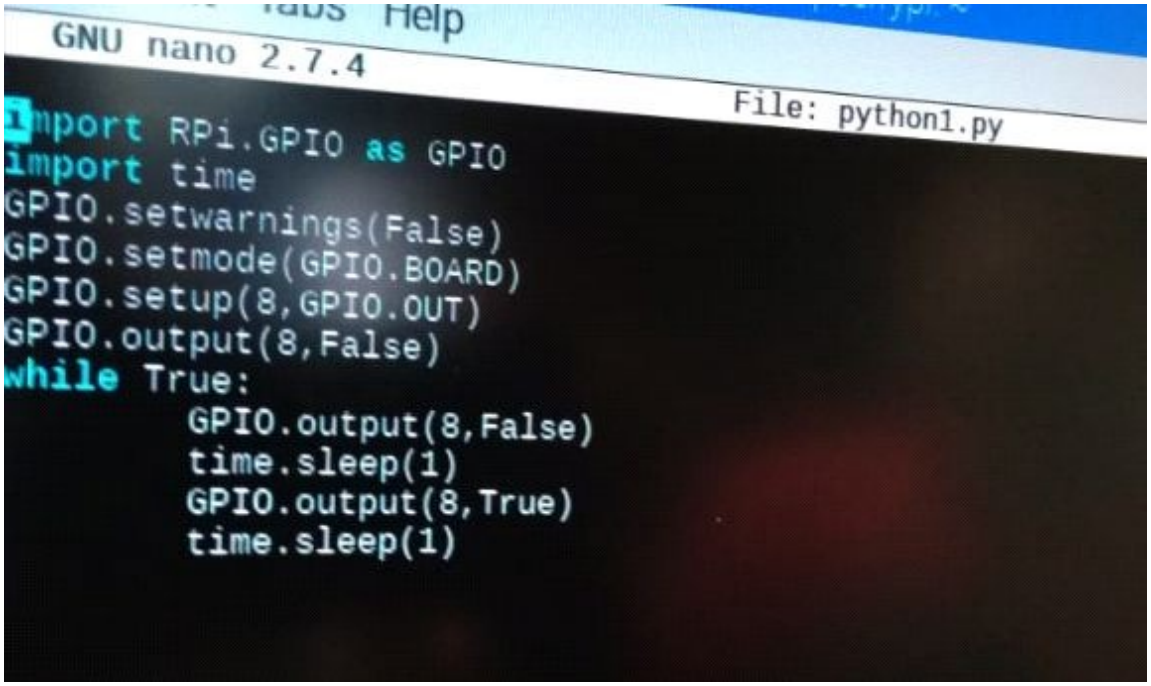


Mulund College Of Commerce

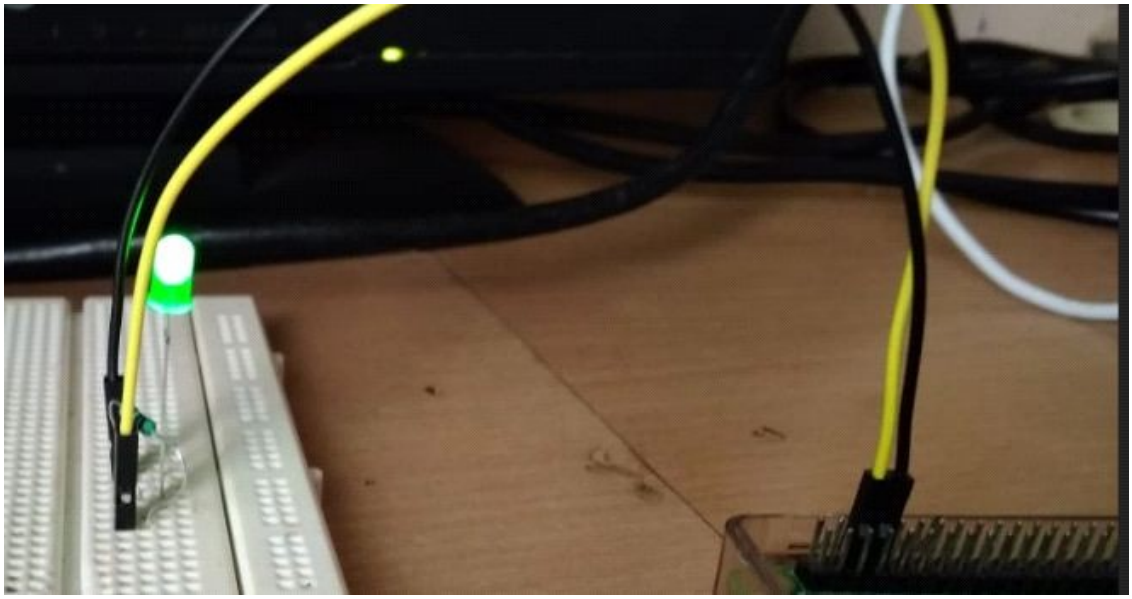


6) Write a Program to blink the LED at pin number 8 in BOARD mode.

- **Program :**

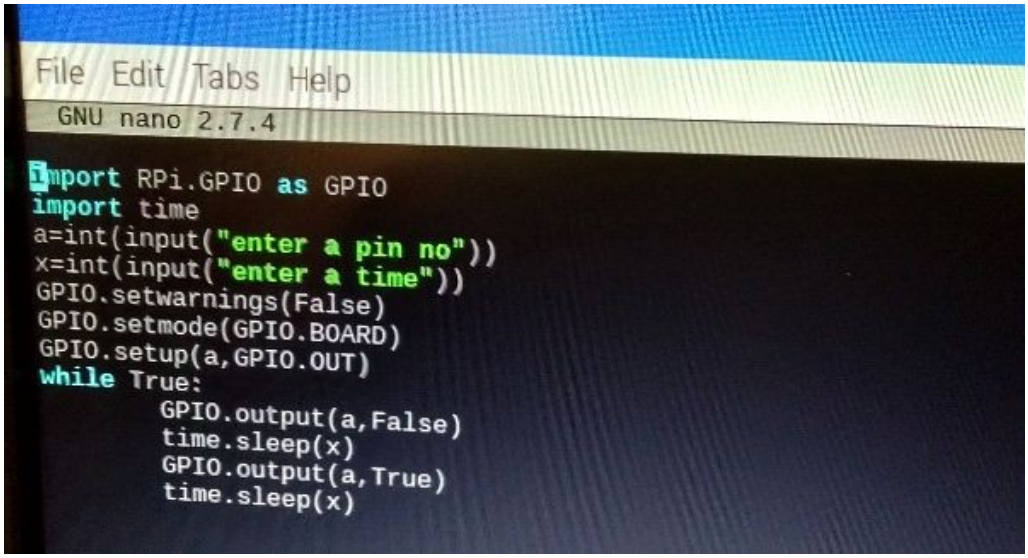


Output :

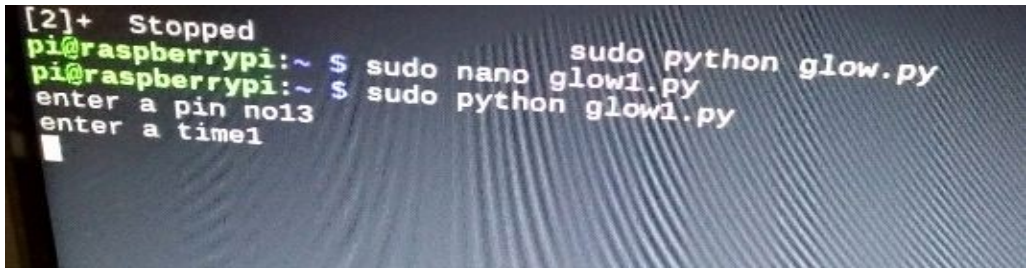


7) **Write a Program to blink the LED accepted pin number and time from user.**

- **Program :**



Output :



8) Write a Program to glow the LED when you enter a even number .

- **Program :**

```
GNU nano 2.7.4

import RPi.GPIO as GPIO
a=int(input("enter the pin no."))
if(a%2==0):
        print("led will  glow")
        GPIO.setwarnings(False)
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(8,GPIO.OUT)
        GPIO.output(8,False)
while True:
        GPIO.output(8,True)
else:
        print("sorry  led can't glow")
```

Output :

```
[1]+  Stopped                 sudo python glow.py
pi@raspberrypi:~ $ sudo nano glow.py
pi@raspberrypi:~ $ sudo python glow.py
enter the pin no.4
led will  glow
```

PRACTICAL NO. 4

GPIO: LED Grid Module: Program the 8X8 Grid with different formulas .

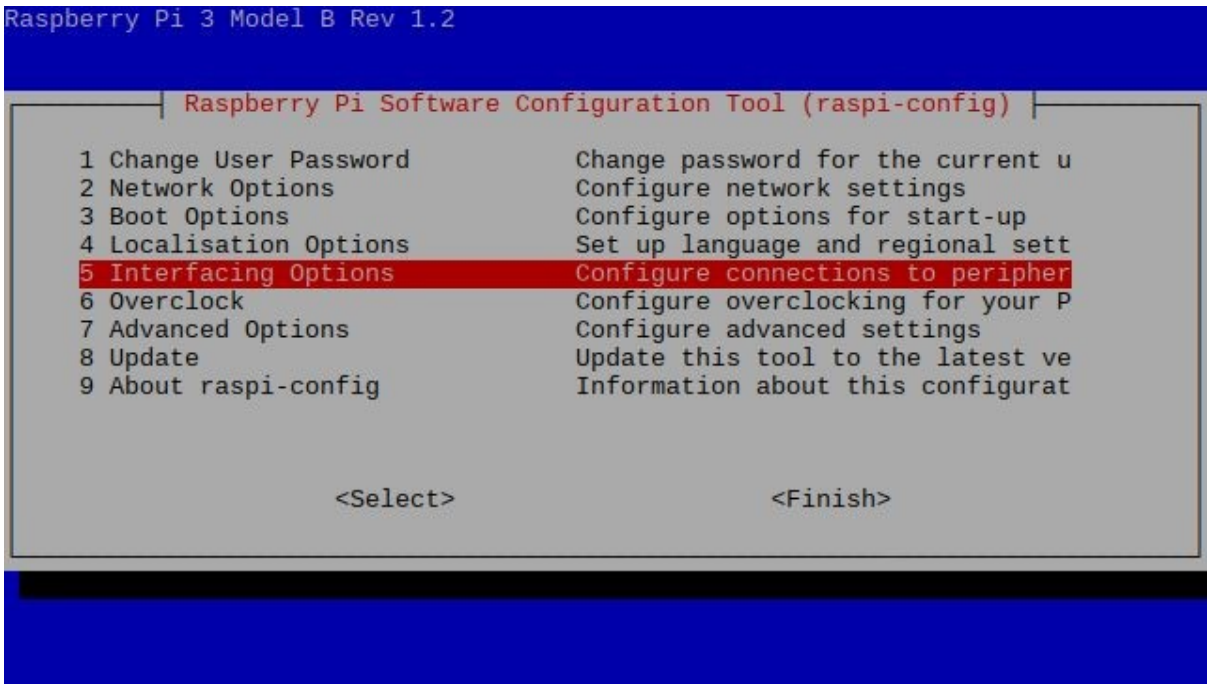
The 8x8 LED Matrix and the max7219 chip allow us to control 64 LEDs per module. Because of the max7219 chip on the module we don't need to connect every single led to the Raspberry Pi. We only need to connect the chip to the Pi and the chip will control all of the LEDs on our behalf. The SPI communication protocol will be utilized to provide communication between the chip and the Pi, following are the steps to activate SPI

Step 1:

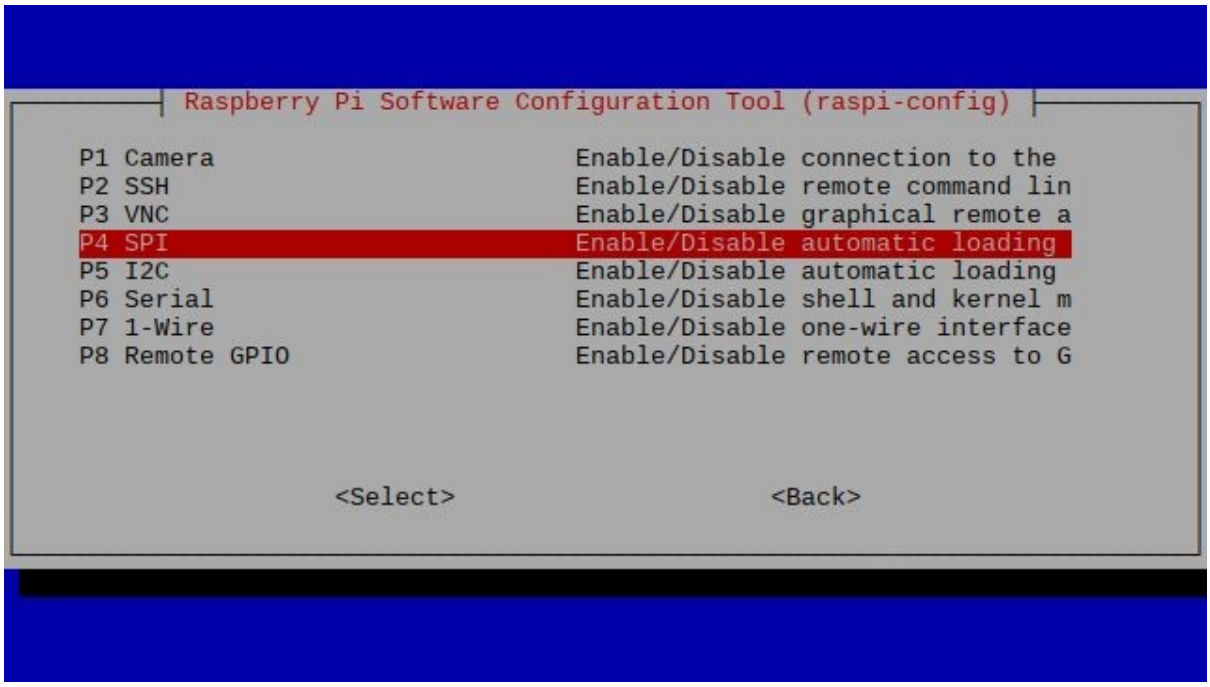
From the command line or Terminal window start by running the following command :

Sudoraspi-config

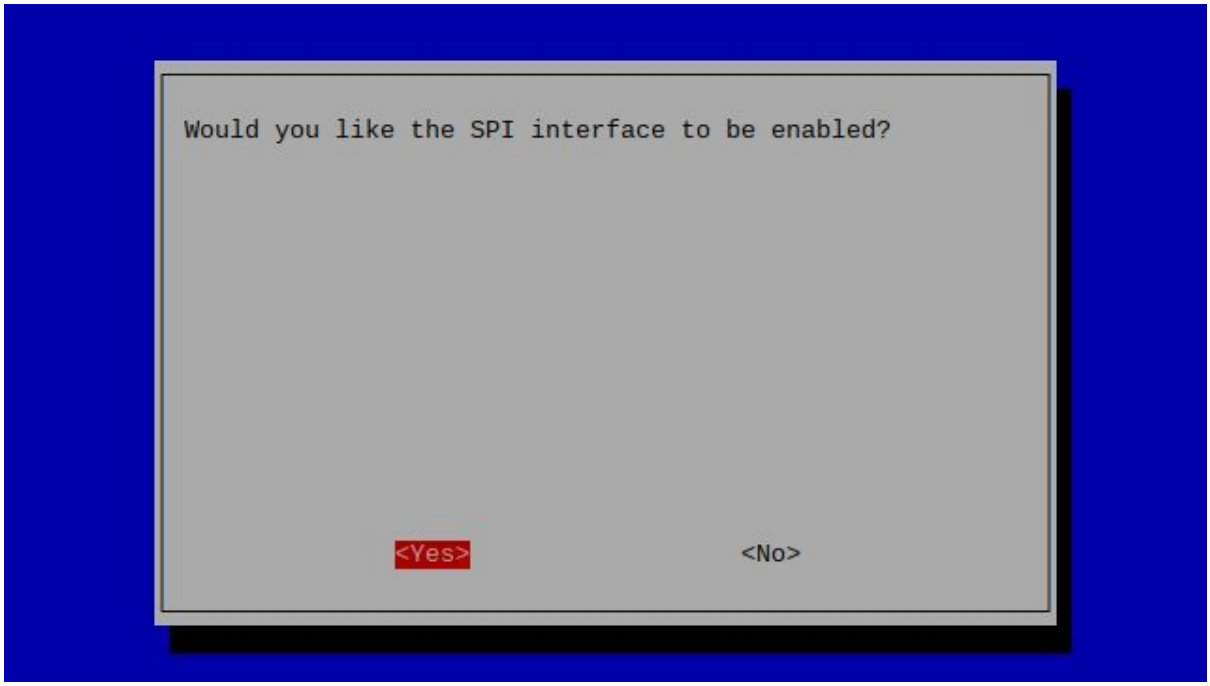
This will launch the raspi-config utility. Select "Interfacing Options" :



Highlight the "SPI" option and activate "<Select>".



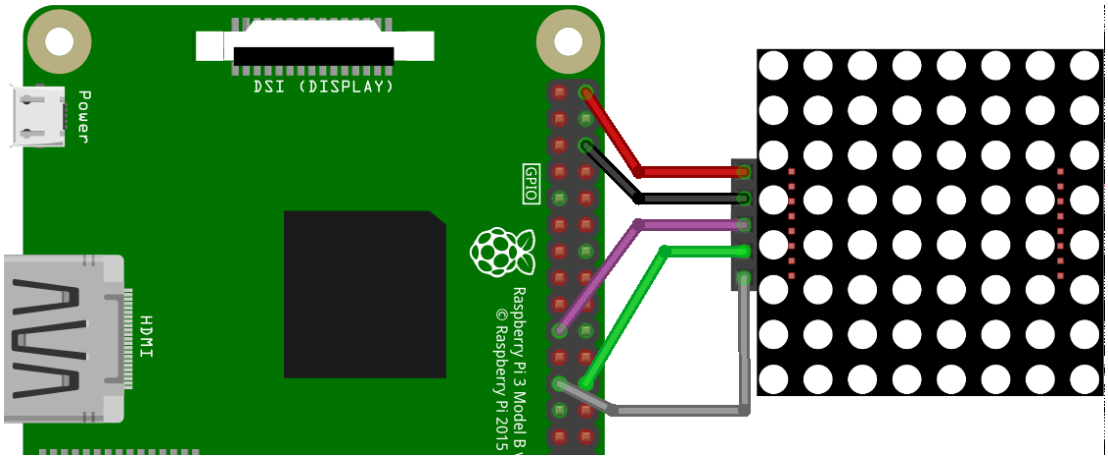
Select and activate "<Yes>" :



Now click on finish and your spi is enabled .

Step 2 : Connections

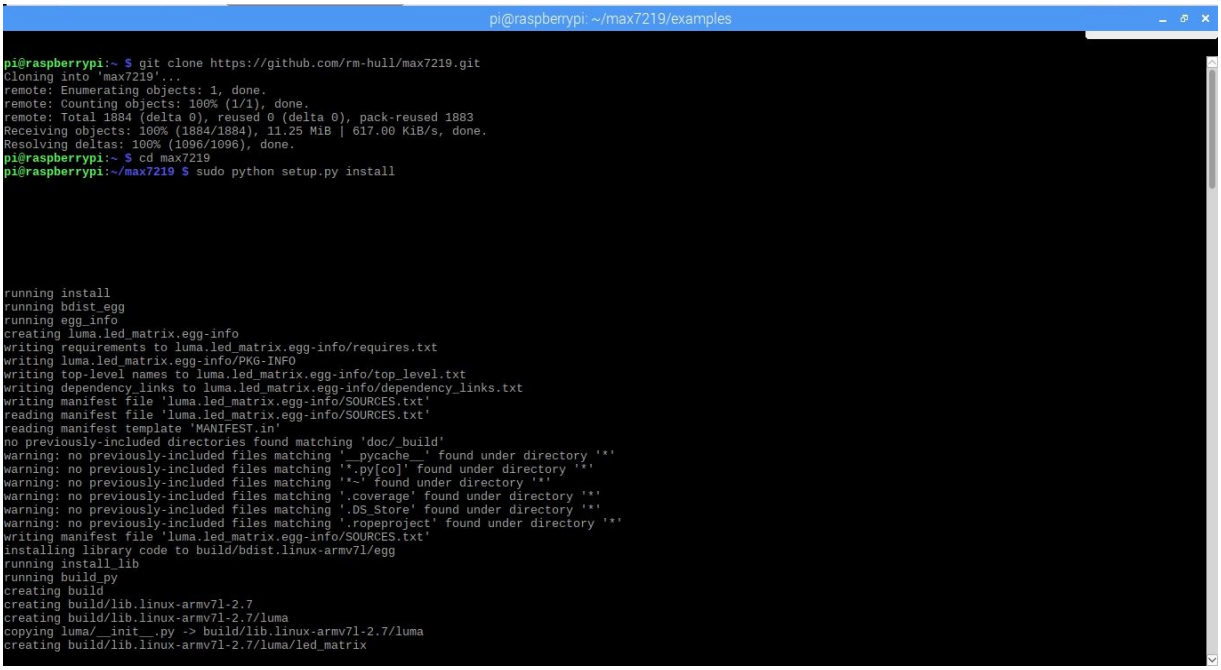
Now that we have enabled the spi, we next need to connect the 8x8 LED matrix to the Raspberry Pi.



Connections Raspberry Pi	Connections LED Matrix
5V	1 - VCC
GND	2 - GND
GPIO 10 (MOSI)	3 - DIN
GPIO 8 (SPI CE0)	4 - CS
GPIO 11 (SPI CLK)	5 - CLK

Step 3: After connections, we have to install drivers, below given are the following commands .

git clone https://github.com/rm-hull/max7219.git sudo python max7219/setup.py install

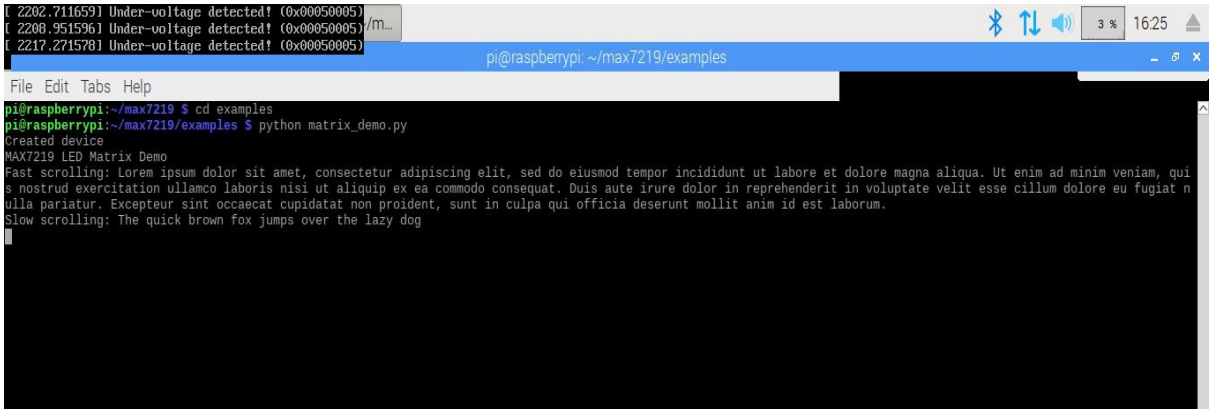


Step 4: Executing the program

As the drivers are installed,run the demo program in the examples subdirectory which is in max7129 directory , following is the cmd to enter into the examples directory.

cd max7219/examples

After entering into examples folder, execute matrix.py Sudo python matrix.py



PRACTICAL NO.5

SPI: Camera Connection and capturing Images using SPI

Step 1 :

Camera slot port is available in between audio port and HDMI port insert the camera in the slot.

Step 2 :

Just pull the camera and check the connection.

Step 3 :

Update & upgrade the raspberry pi by using following commands.

”sudo apt-get update”

“sudo apt-get upgrade”

Step 4 :

Run:”sudo raspi-config”

Go to the camera option & enable it.

Step 5 :

To take a still picture follow the below given commands. “rapistill -o image.jpg”

Step 6 :

To capture the video follow the below given commands. “raspivid -o video.h264-t 1000”

To open the video.

“omxplayer video.h264”.

PRACTICAL NO.6

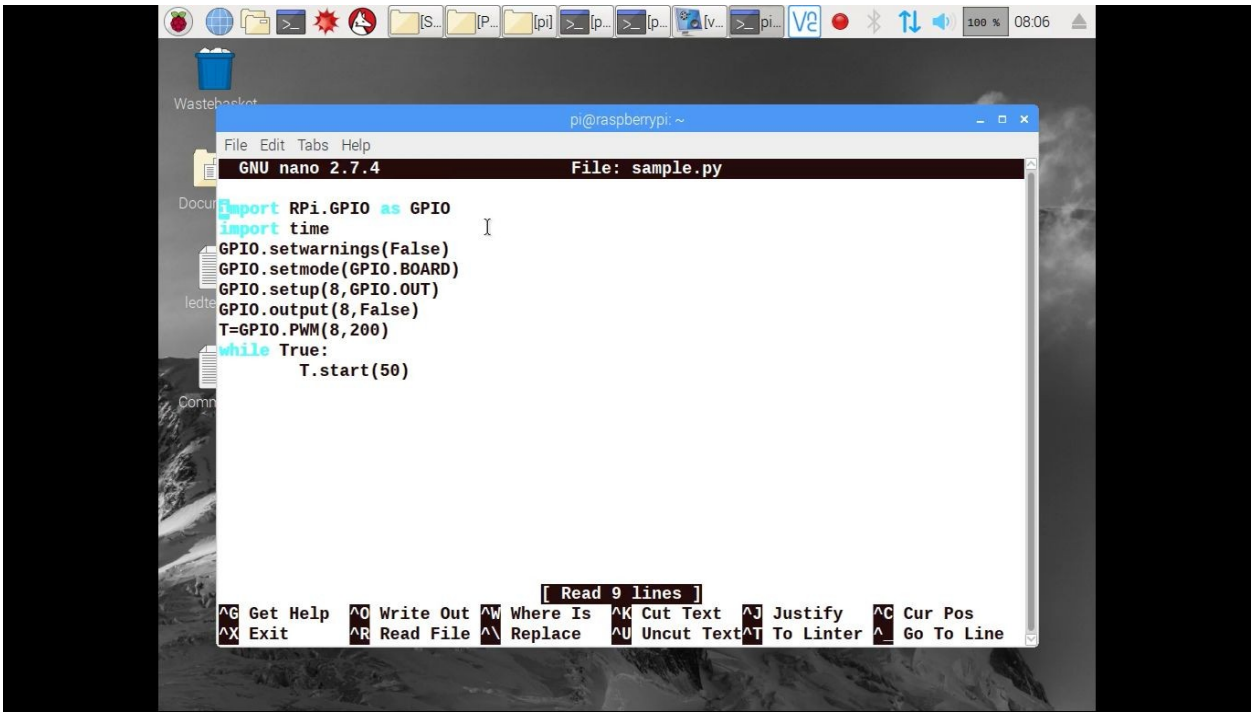
DEMONSTRATE THE USE OF PWM

1) WRITE A PROGRAM TO INCREASE THE INTENSITY OF AN LED BY USING PWM .

→

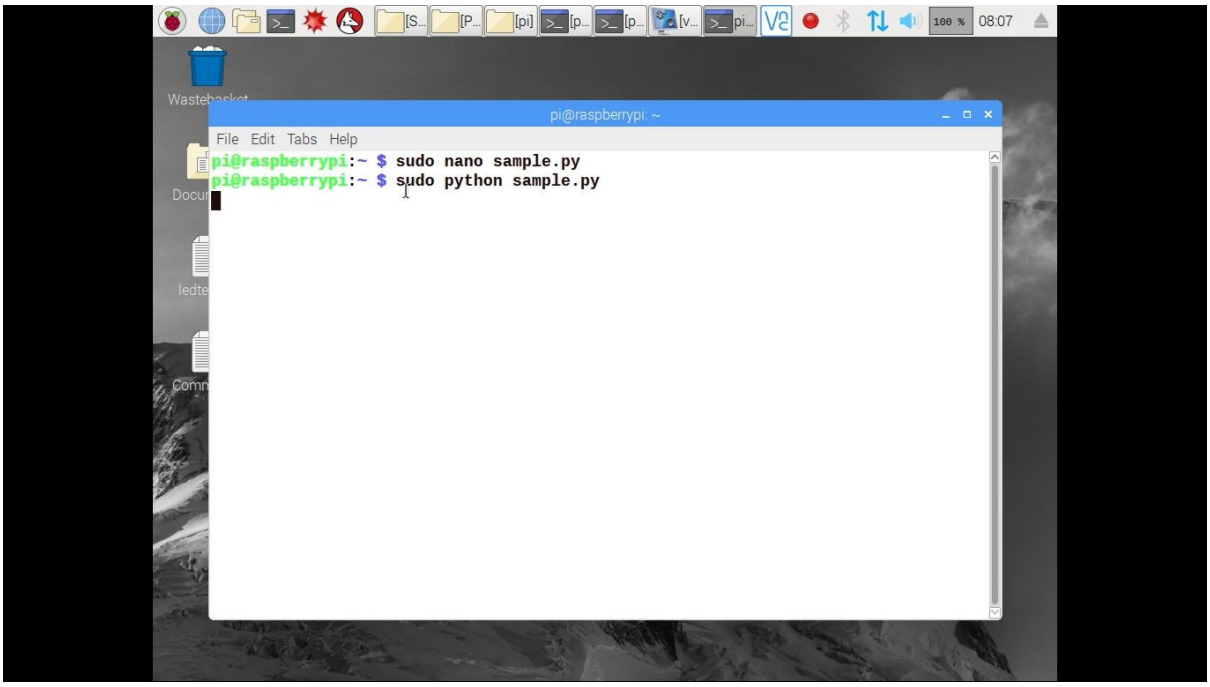
- Type the command "sudo nano filename.py" to open the editor where you can write your python code

- The below given image is the code for glowing the LED on.

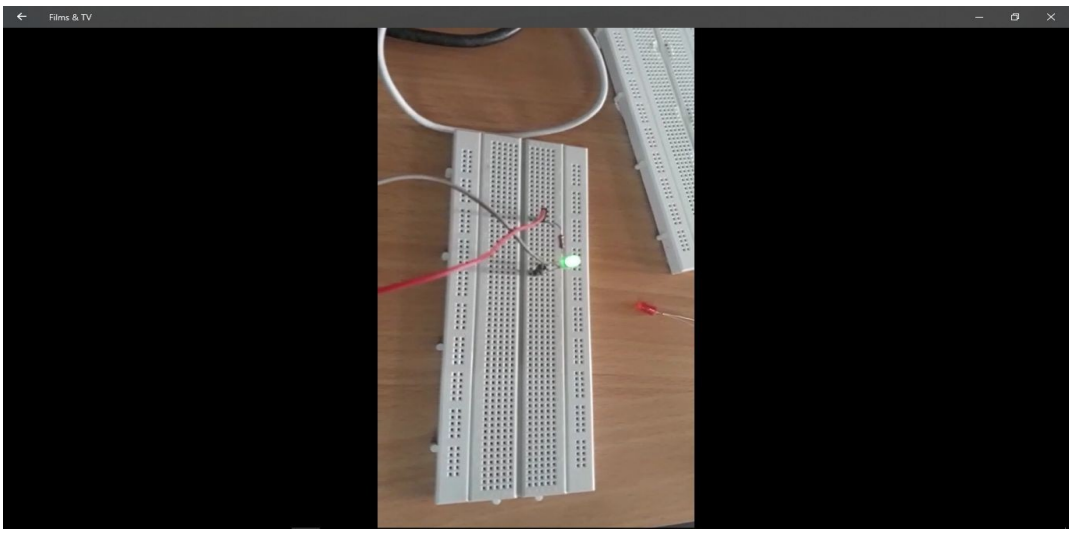


- After writing the code press "ctrl+0" to save the code and "ctrl+X" to exit from the editor.

- Once you are back on the terminal enter the command "sudo python filename.py"



- to run your program.if your connections are proper and the code is written correctly the LED will be glowing.
- The output looks like this :

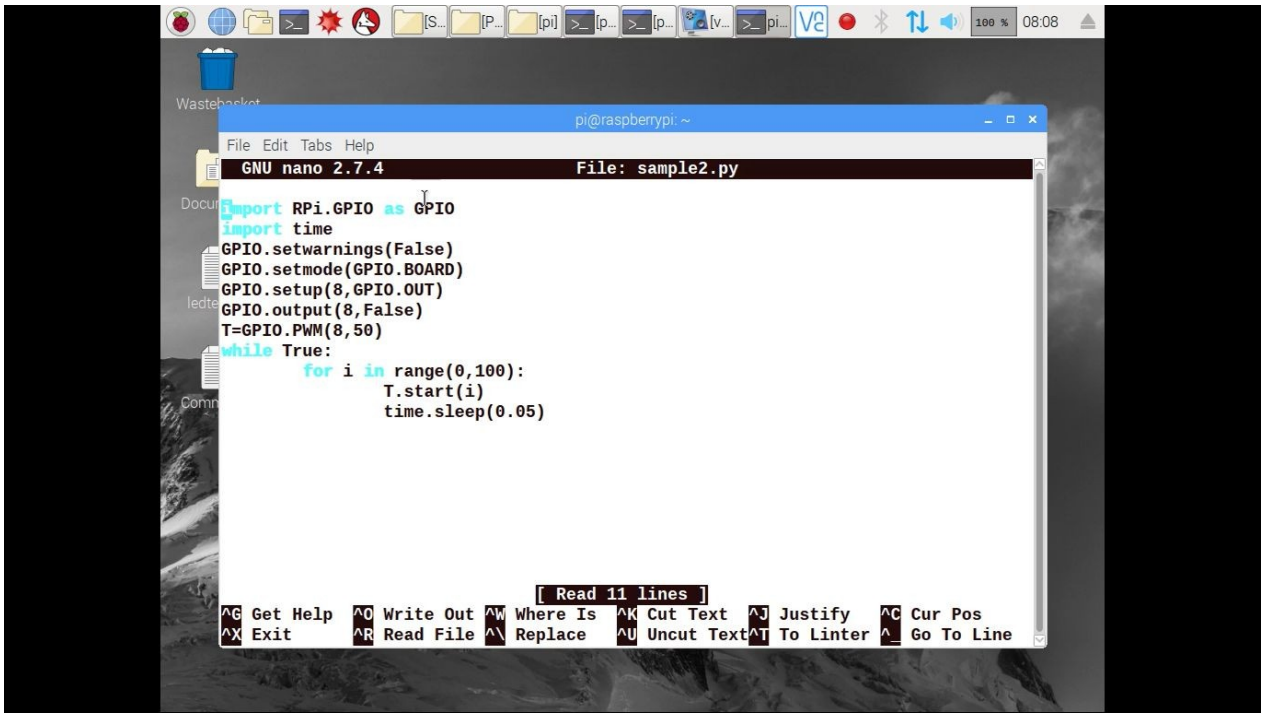


2) WRITE A PROGRAM TO INCREASE THE INTENCITY OF AN LED BY USING FOR LOOP IN PWM .

→

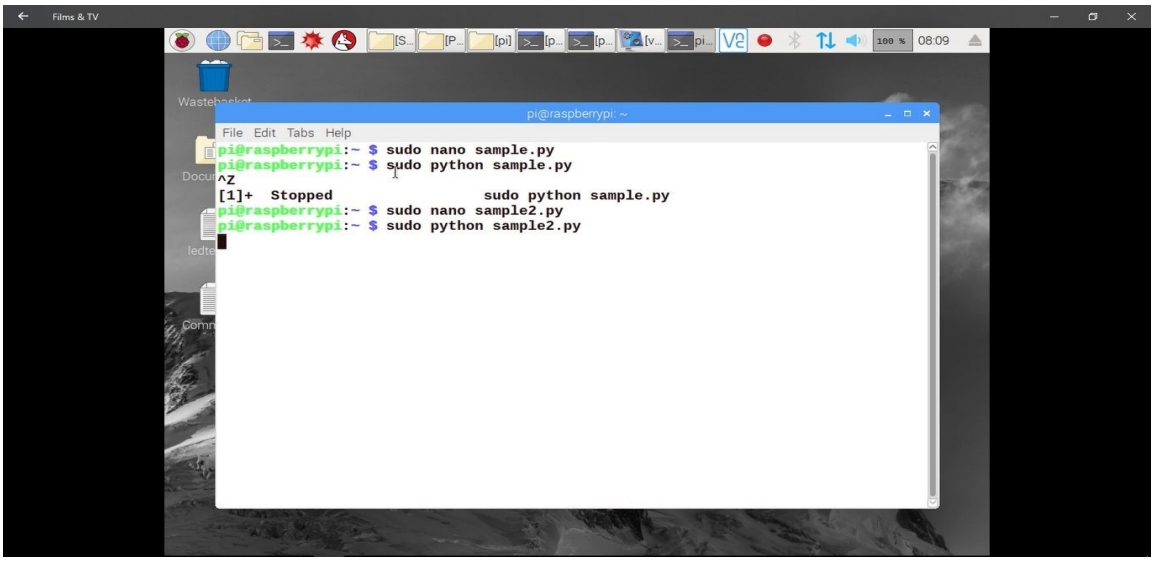
- Type the command "sudo nano filename.py" to open the editor where you can write your python code

- The below given image is the code for glowing the LED on :



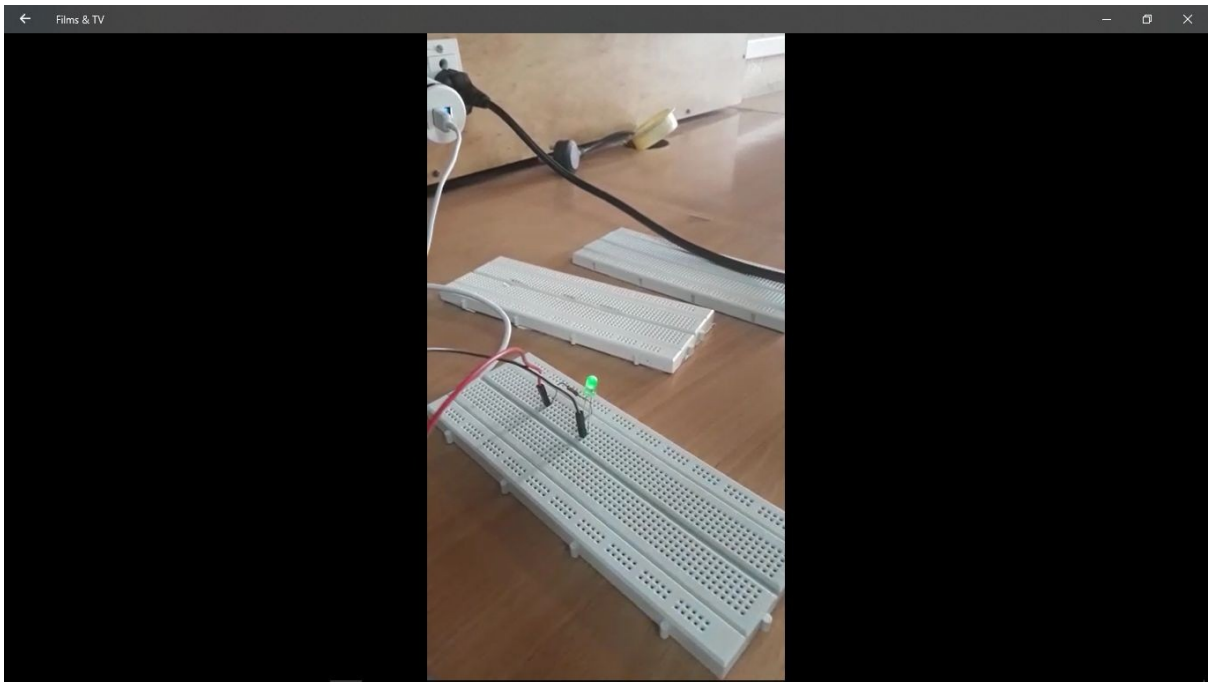
- After writing the code press "ctrl+0" to save the code and "ctrl+X" to exit from the editor.

- Once you are back on the terminal enter the command "sudo python filename.py"



- to run your program.if your connections are proper and the code is written correctly the LED will be glowing.

- The output looks like this:

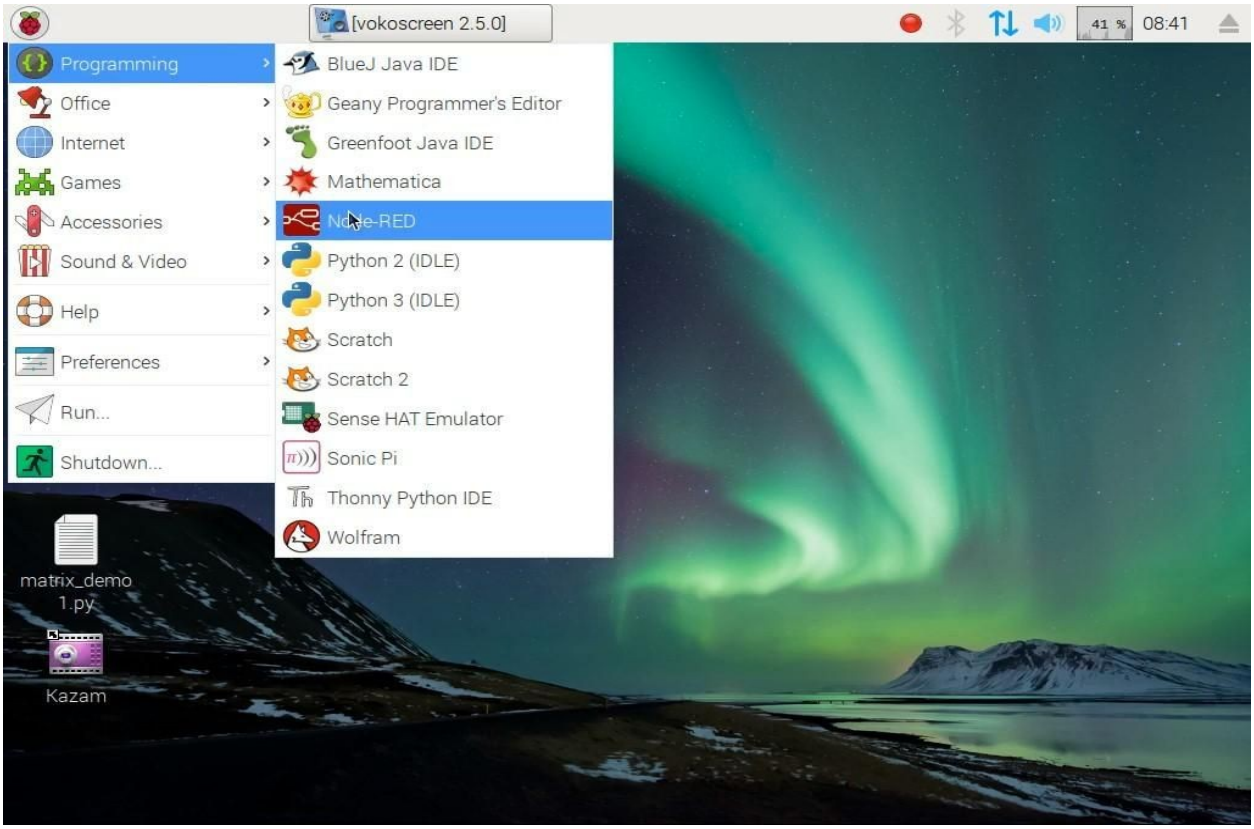


PRACTICAL NO.7

Node RED: Connect LED to Internet of Things

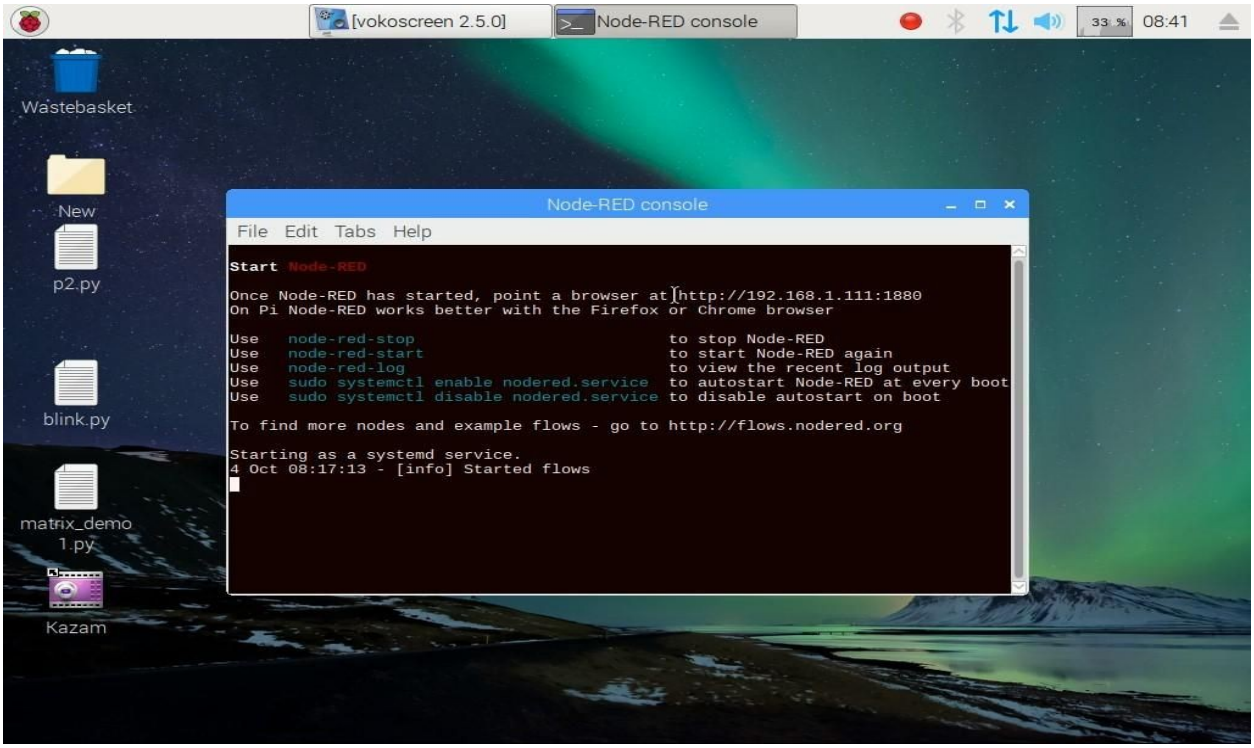
Step 1 →

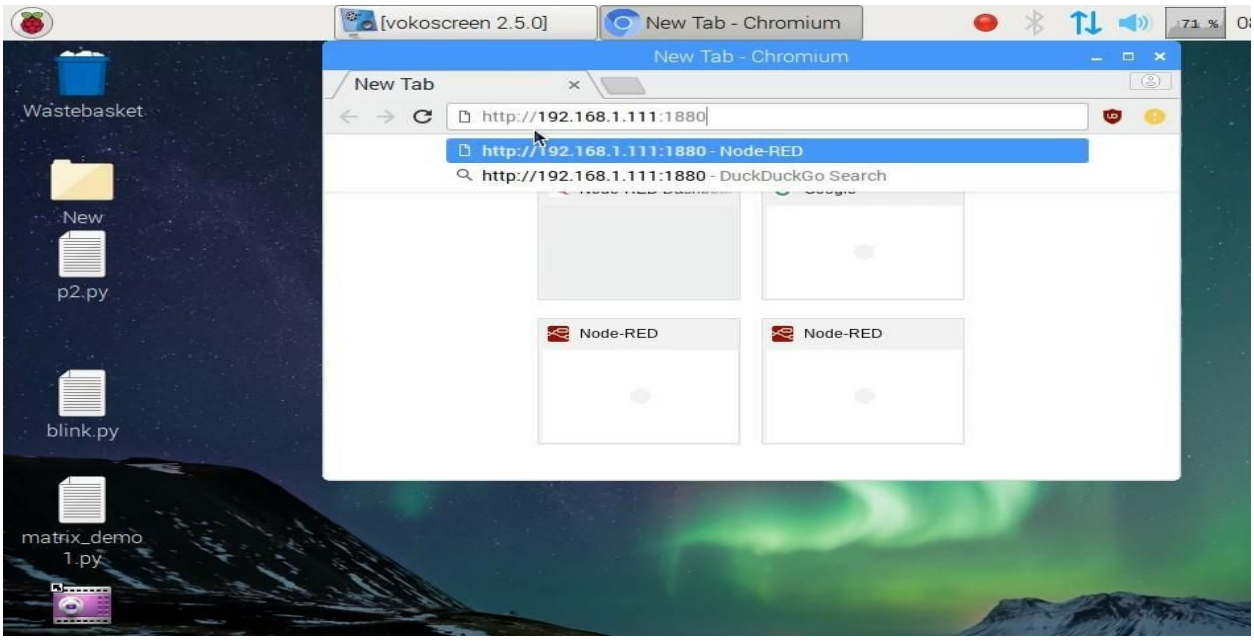
Go to Programming ➔ Node-RED



Step 2 →

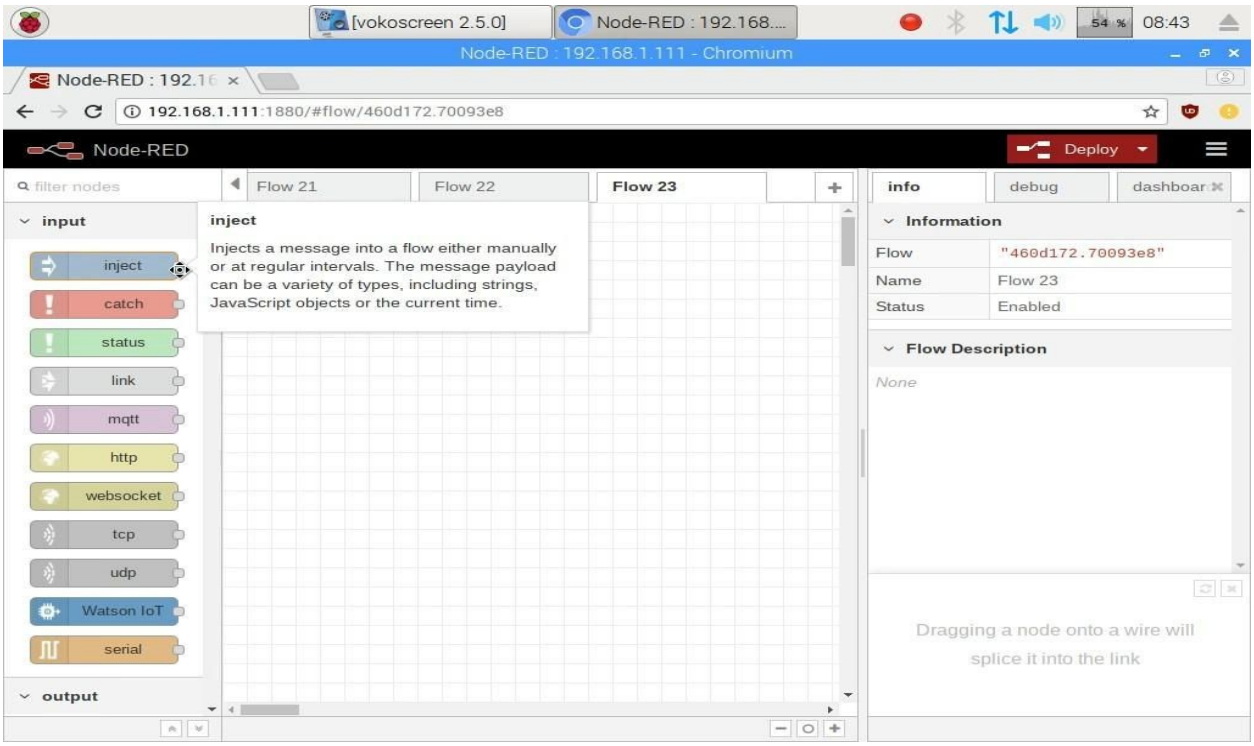
URL will display just copy the URL and past it at browser address bar then press enter.
(http://192.168.1.111:1880).

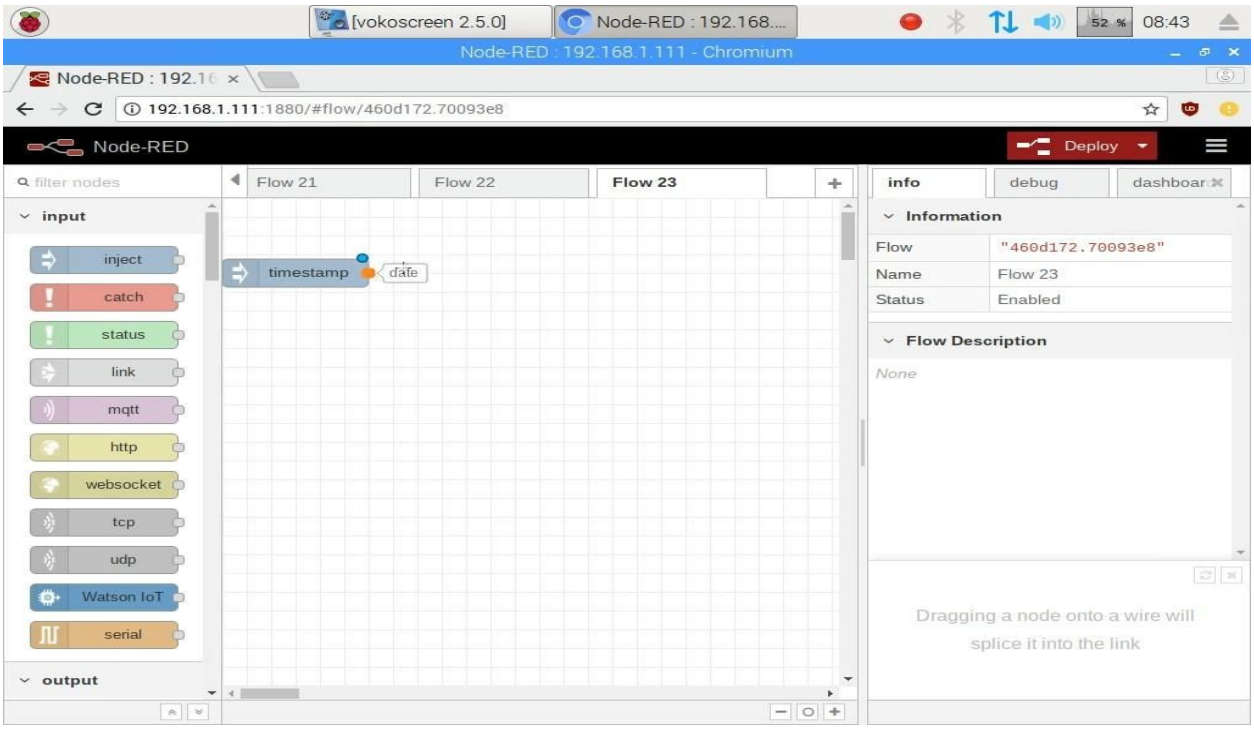




Step 3→

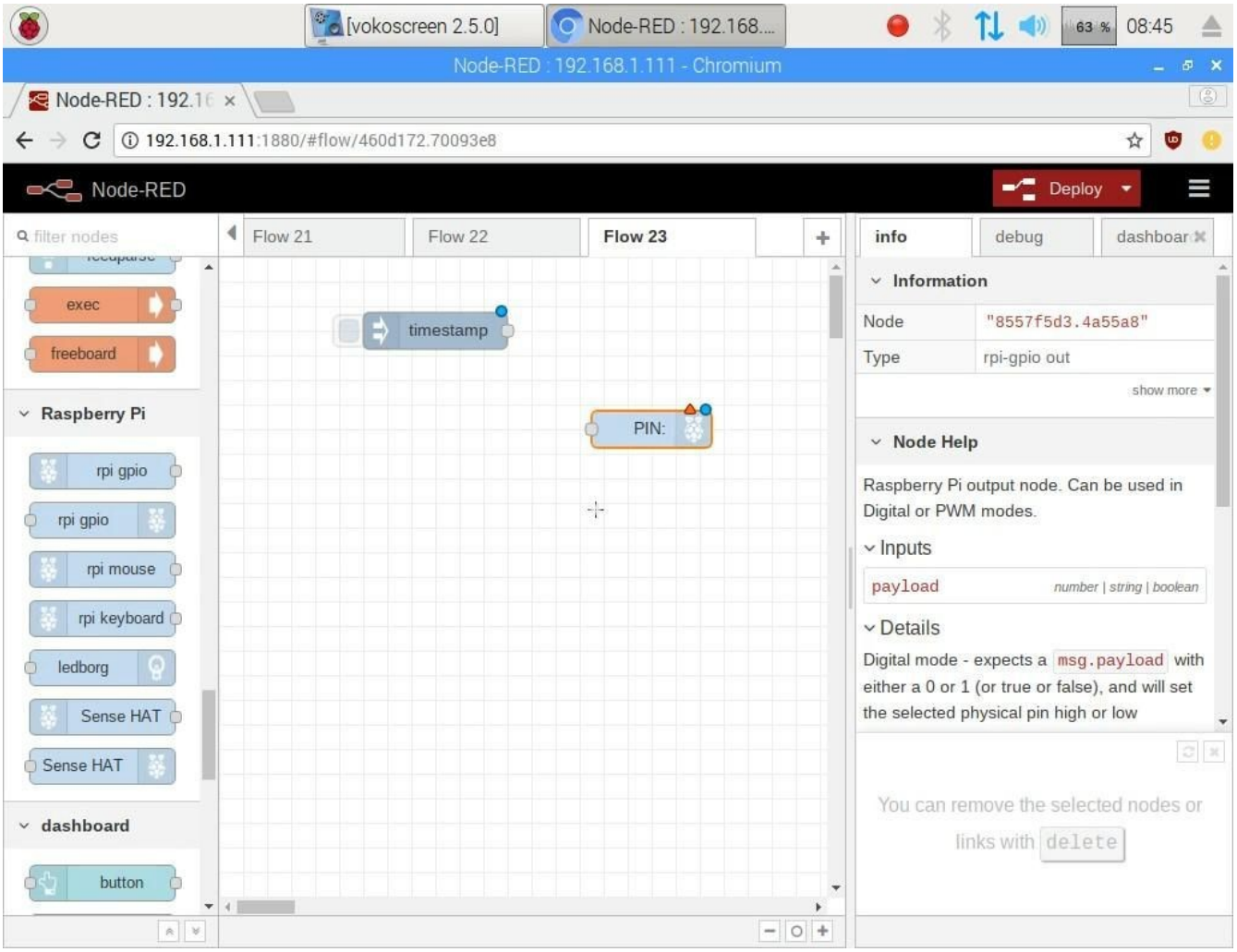
Select Inject from Left side input bar.


[vokoscreen 2.5.0]
Node-RED : 192.168....
54 %
08:43
Node-RED : 192.168.1.111 - Chromium
Node-RED : 192.16
192.168.1.111:1880/#flow/460d172.70093e8
Node-RED
Deploy
filter nodes
Flow 21
Flow 22
Flow 23
info
debug
dashboar
input
inject
Injects a message into a flow either manually or at regular intervals. The message payload can be a variety of types, including strings, JavaScript objects or the current time.
catch
status
link
mqtt
http
websocket
tcp
udp
Watson IoT
serial
output
Information
Flow
"460d172.70093e8"
Name
Flow 23
Status
Enabled
Flow Description
None
Dragging a node onto a wire will splice it into the link


[vokoscreen 2.5.0]
Node-RED : 192.168....
52 %
08:43
Node-RED : 192.168.1.111 - Chromium
Node-RED : 192.16
192.168.1.111:1880/#flow/460d172.70093e8
Node-RED
Deploy
filter nodes
Flow 21
Flow 22
Flow 23
info
debug
dashboar
input
inject
timestamp
date
catch
status
link
mqtt
http
websocket
tcp
udp
Watson IoT
serial
output
Information
Flow
"460d172.70093e8"
Name
Flow 23
Status
Enabled
Flow Description
None
Dragging a node onto a wire will splice it into the link

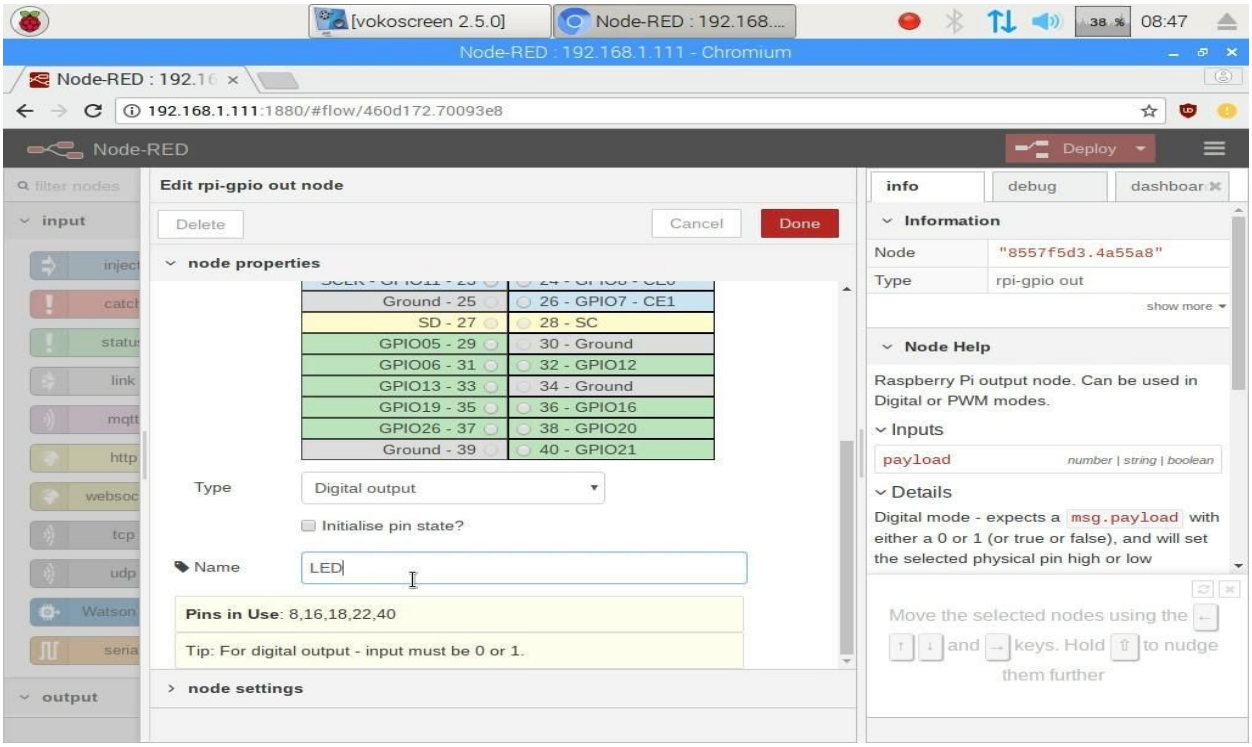
Step 4 →

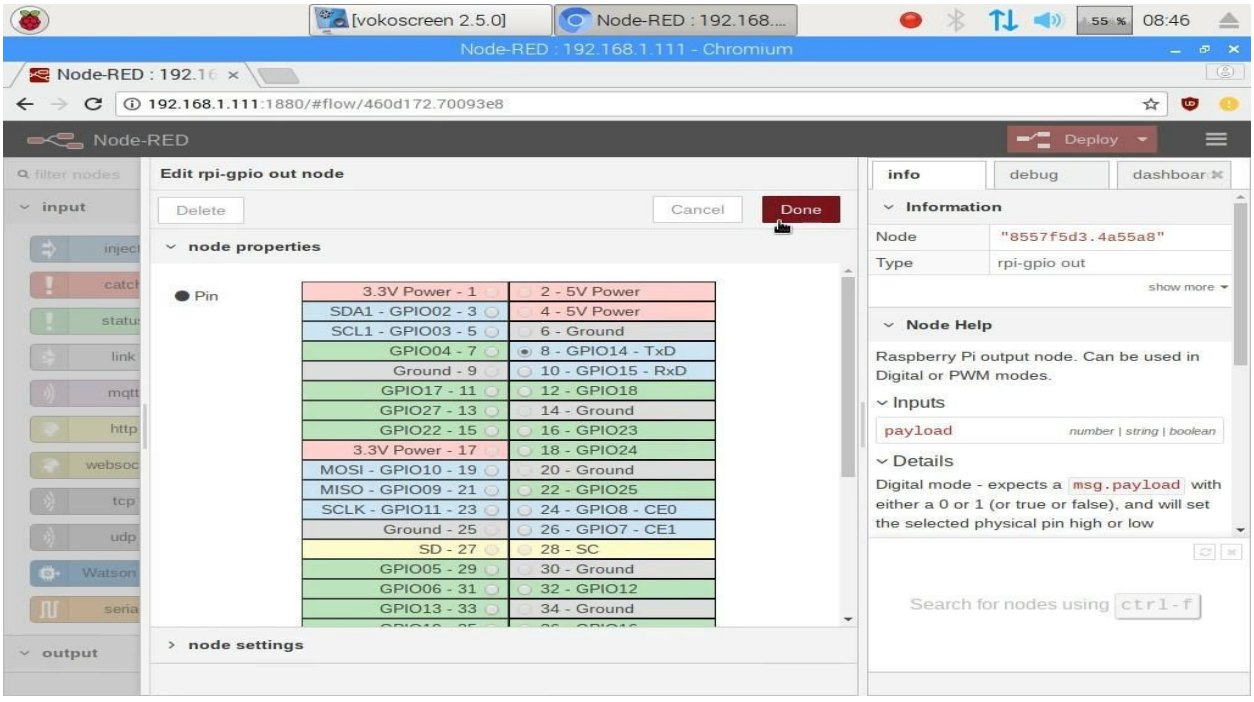
Go to Raspberry Pi bar which is available at the left side and select RP i.GPIO(Output).



Step 5 →

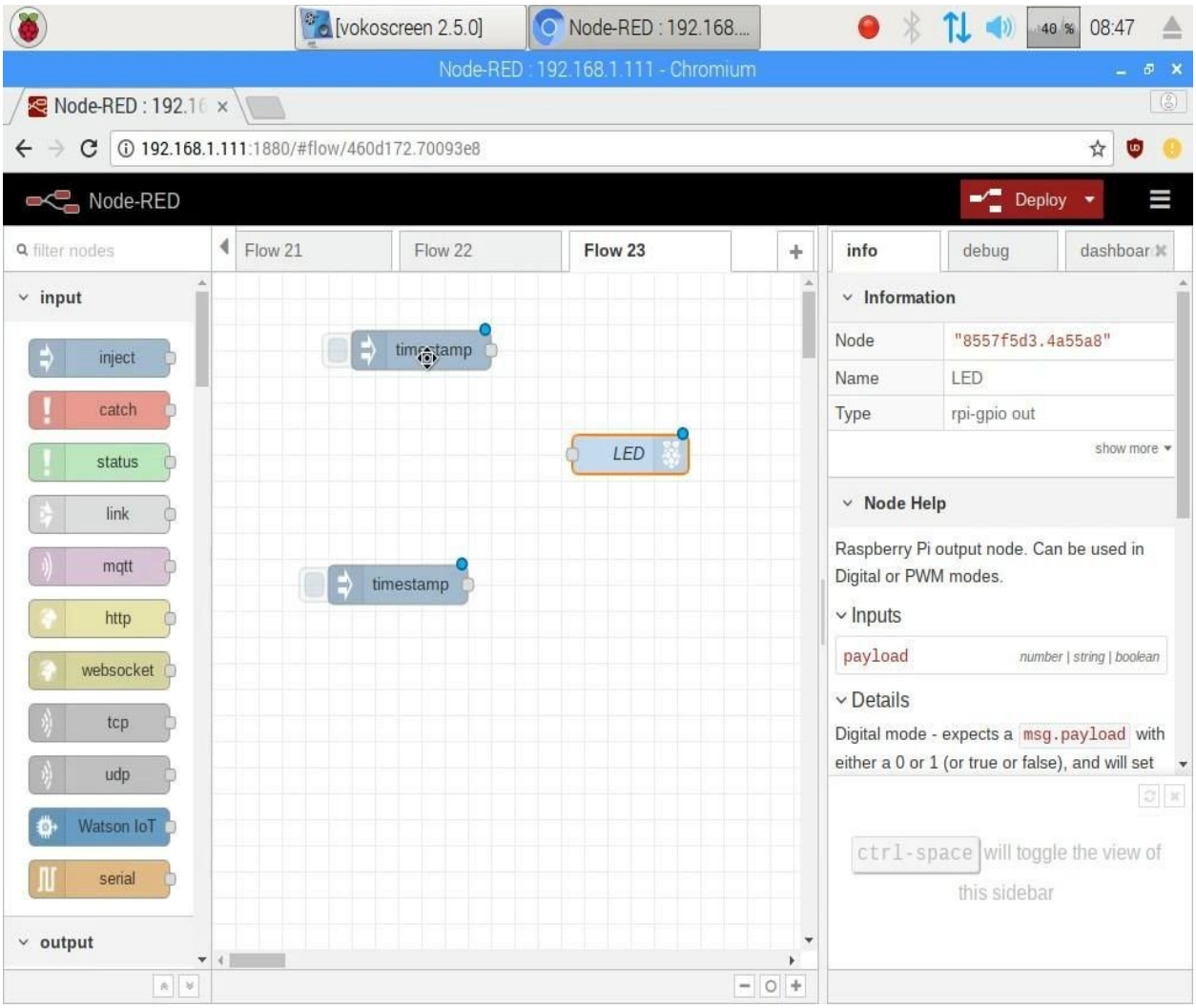
Double Click on RPi.GPIO and name it as LED and select the GPIO Pin no.8 .Type = Digital Output||Click on Done.





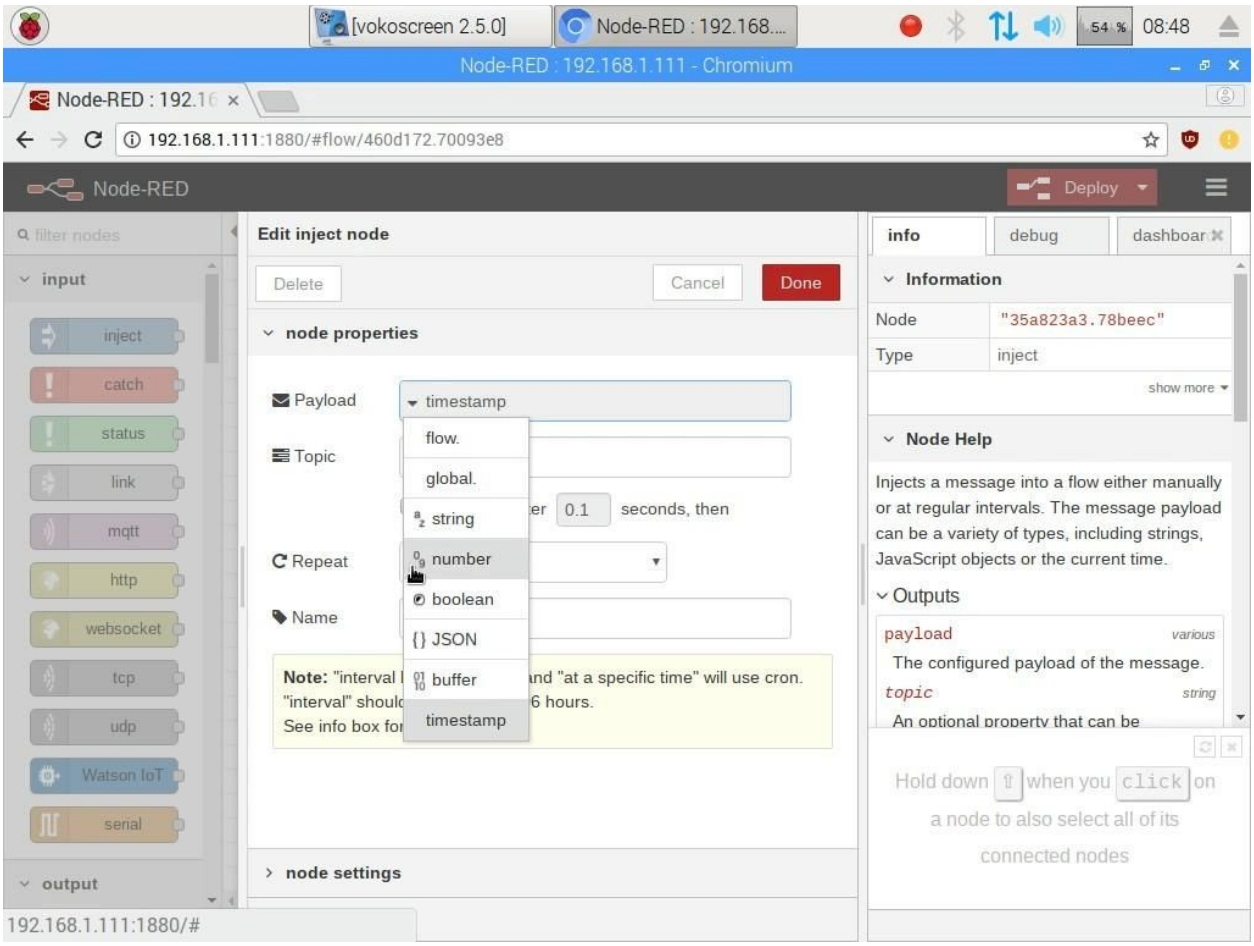
Step 6 →

Take one more Inject button on the screen.



Step 7 →

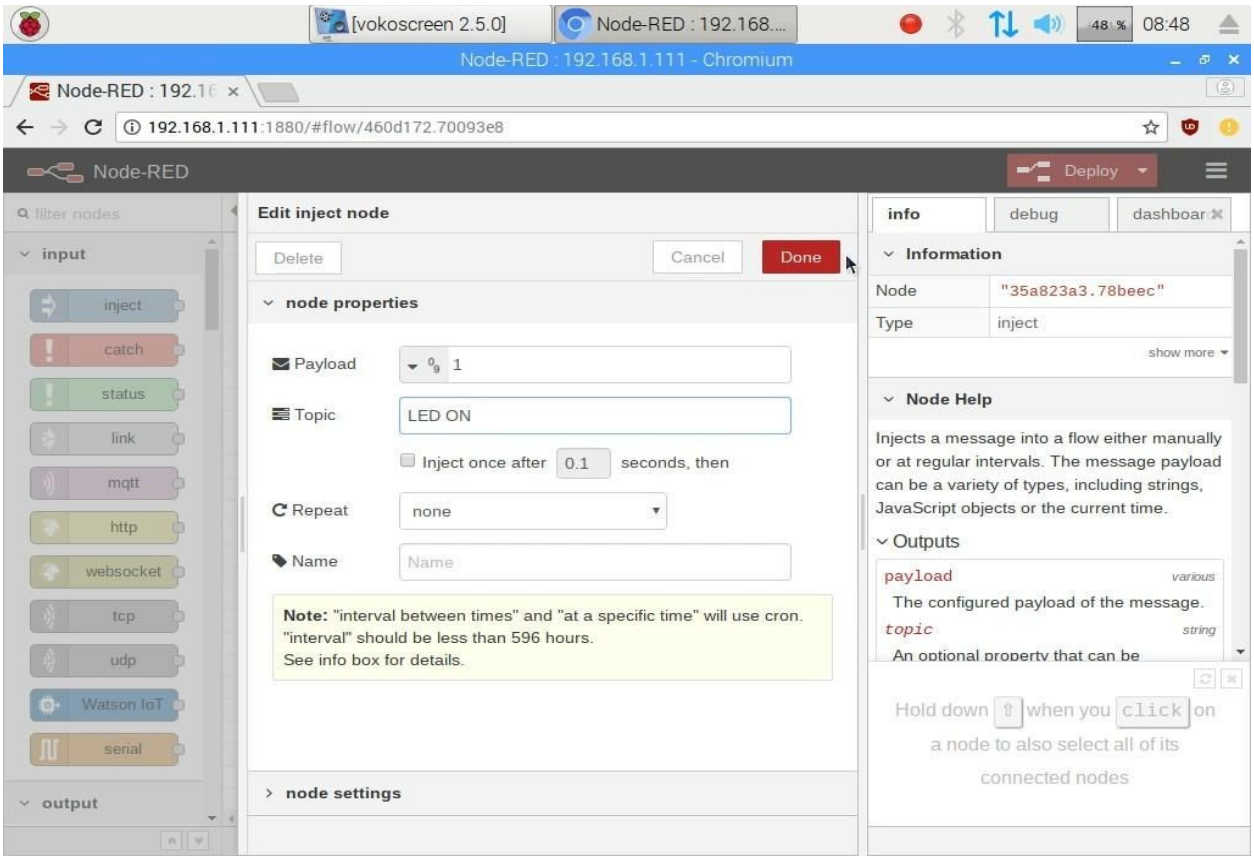
Now double click on the inject button and select time number.

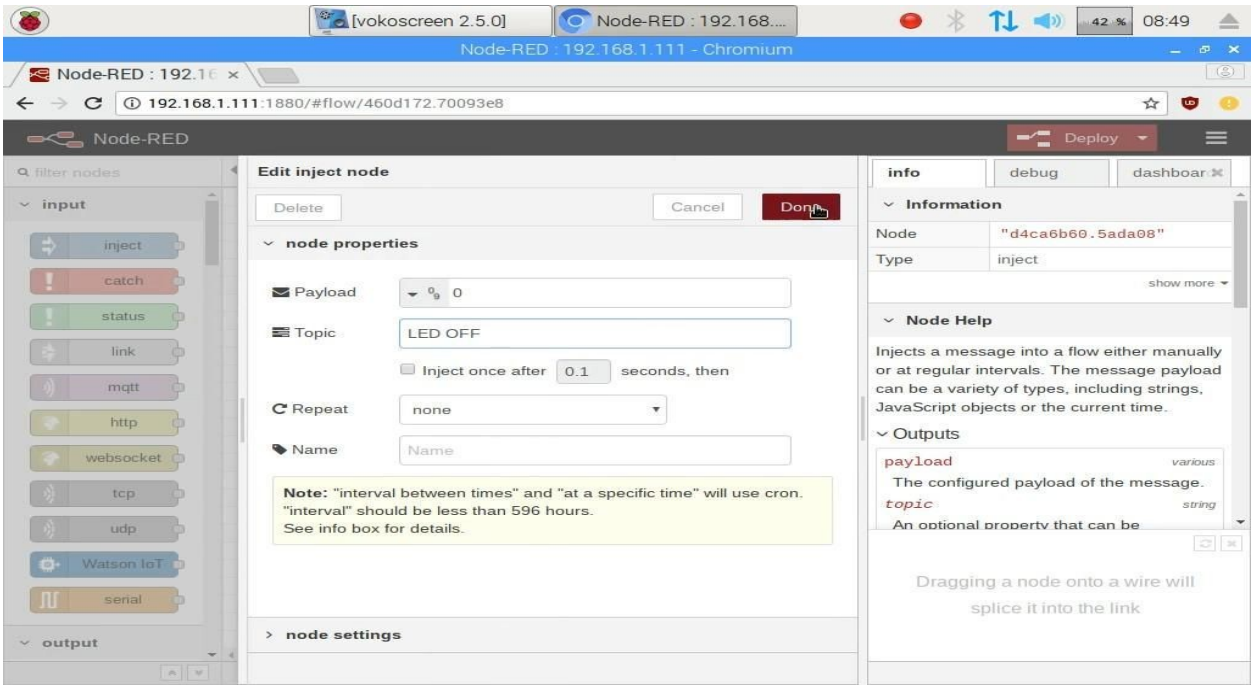


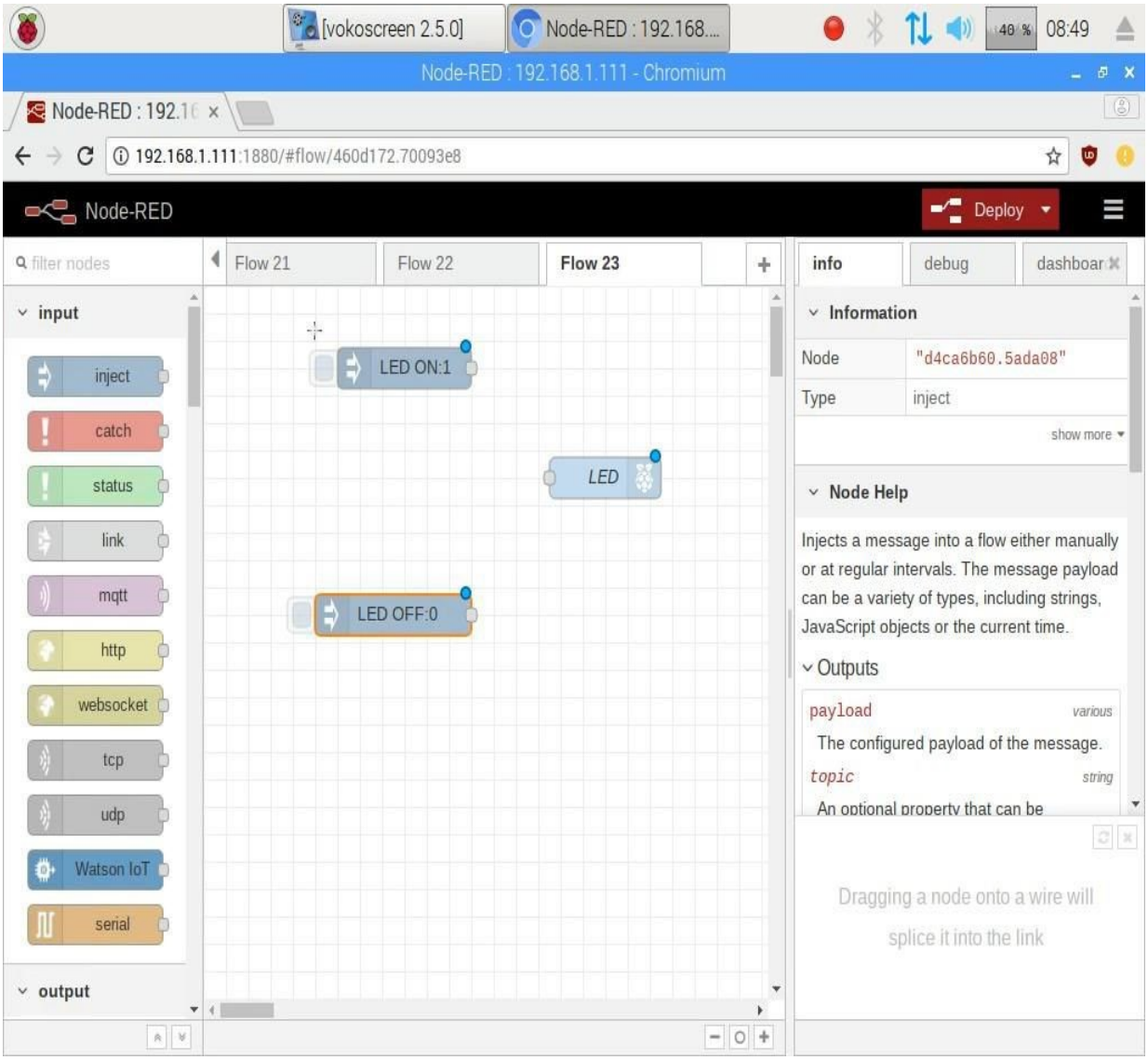
Step 8 →

Name Inject Button as the following (Button1 = 1 as ON Button 2 = 0 as OFF)

Write 1 for ON in payload and 0 for OFF payload.

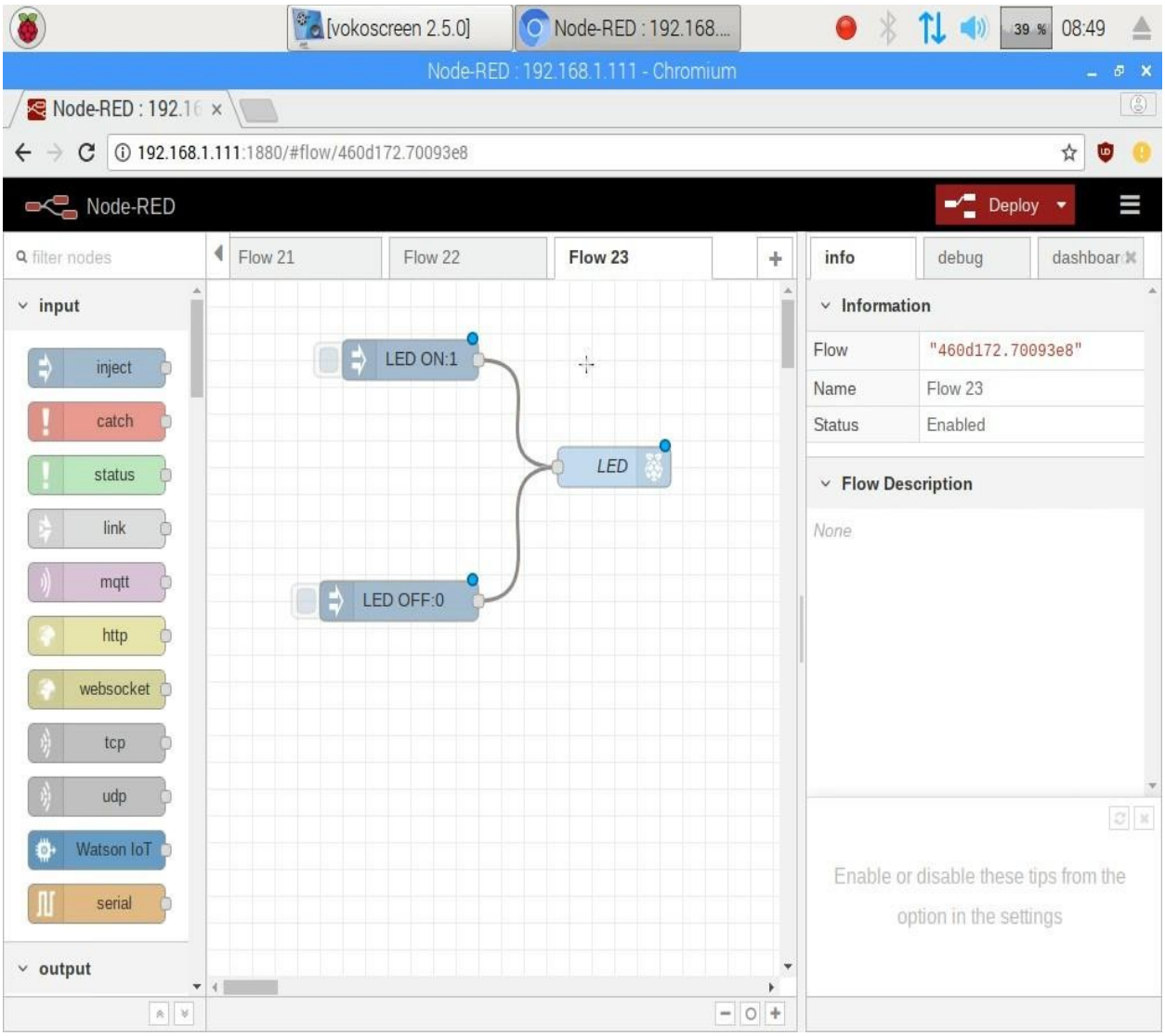

[vokoscreen 2.5.0]
Node-RED : 192.168....
48 %
08:48
Node-RED : 192.168.1.111 - Chromium
Node-RED : 192.16
192.168.1.111:1880/#flow/460d172.70093e8
Node-RED
Deploy
filter nodes
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
Watson IoT
serial
output
Edit inject node
Delete
Cancel
Done
node properties
Payload
1
Topic
LED ON
Inject once after
0.1
seconds, then
Repeat
none
Name
Name
Note: "interval between times" and "at a specific time" will use cron.
"interval" should be less than 596 hours.
See info box for details.
node settings
info
debug
dashboar
Information
Node
"35a823a3.78beec"
Type
inject
show more
Node Help
Injects a message into a flow either manually or at regular intervals. The message payload can be a variety of types, including strings, JavaScript objects or the current time.
Outputs
payload
various
The configured payload of the message.
topic
string
An optional property that can be
Hold down ⇧ when you click on a node to also select all of its connected nodes


[vokoscreen 2.5.0]
Node-RED : 192.168....
42 %
08:49
Node-RED : 192.168.1.111 - Chromium
Node-RED : 192.16
192.168.1.111:1880/#flow/460d172.70093e8
Node-RED
Deploy
filter nodes
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
Watson IoT
serial
output
Edit inject node
Delete
Cancel
Done
node properties
Payload
0
Topic
LED OFF
Inject once after
0.1
seconds, then
Repeat
none
Name
Name
Note: "interval between times" and "at a specific time" will use cron.
"interval" should be less than 596 hours.
See info box for details.
node settings
info
debug
dashboar
Information
Node
"d4ca6b60.5ada08"
Type
inject
show more
Node Help
Injects a message into a flow either manually or at regular intervals. The message payload can be a variety of types, including strings, JavaScript objects or the current time.
Outputs
payload
various
The configured payload of the message.
topic
string
An optional property that can be
Dragging a node onto a wire will splice it into the link



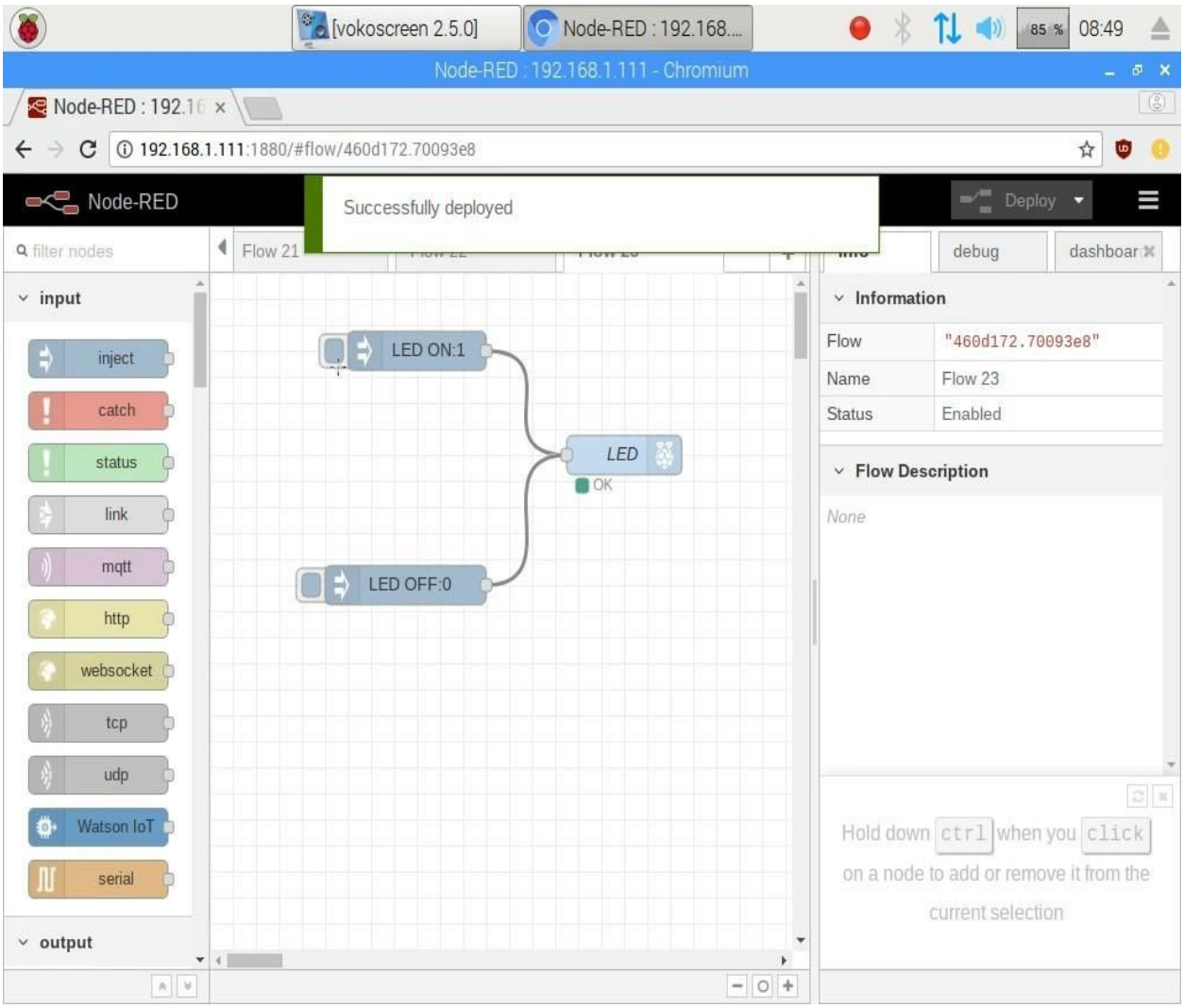
Step 9 →

Connect first and second inject button to the LED.



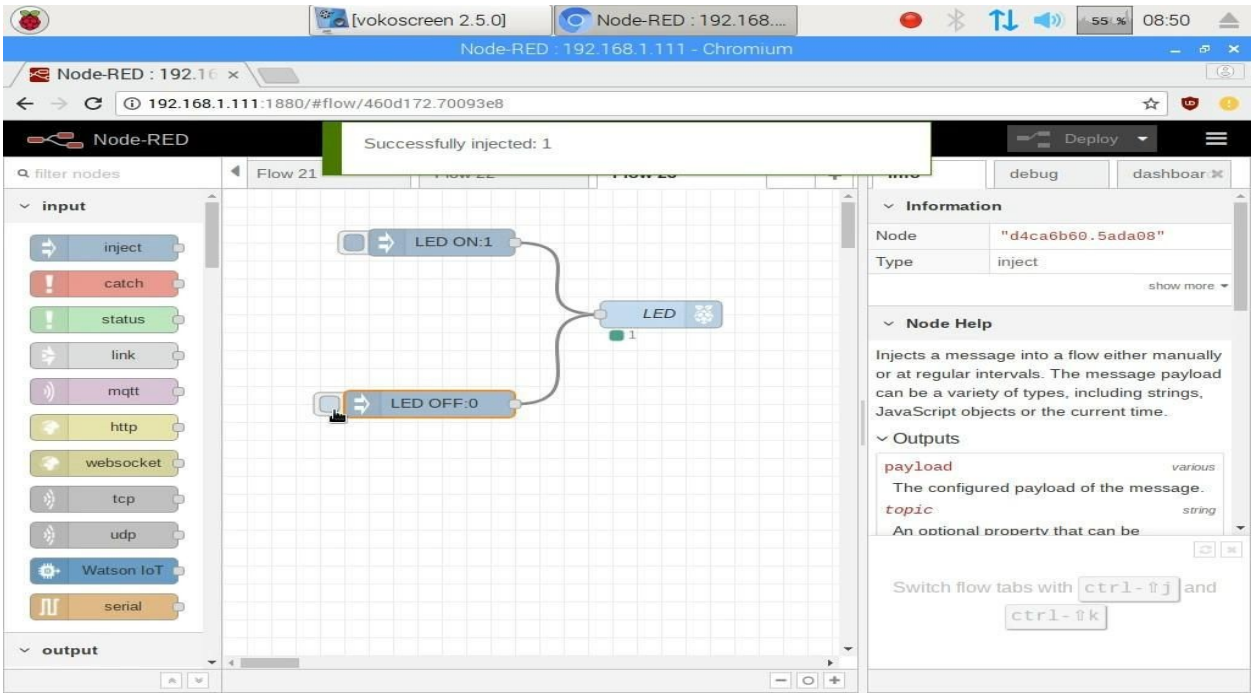
Step 10 →

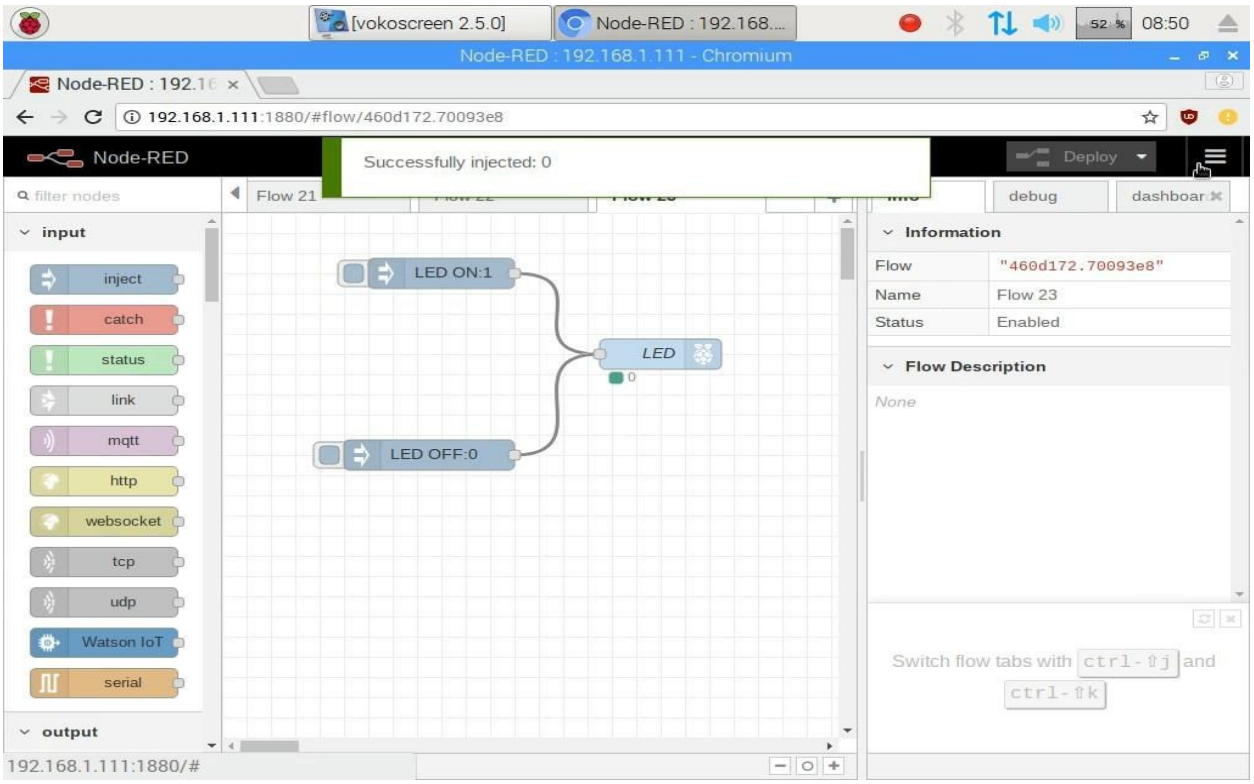
Now Click on the Deploy at the right top.



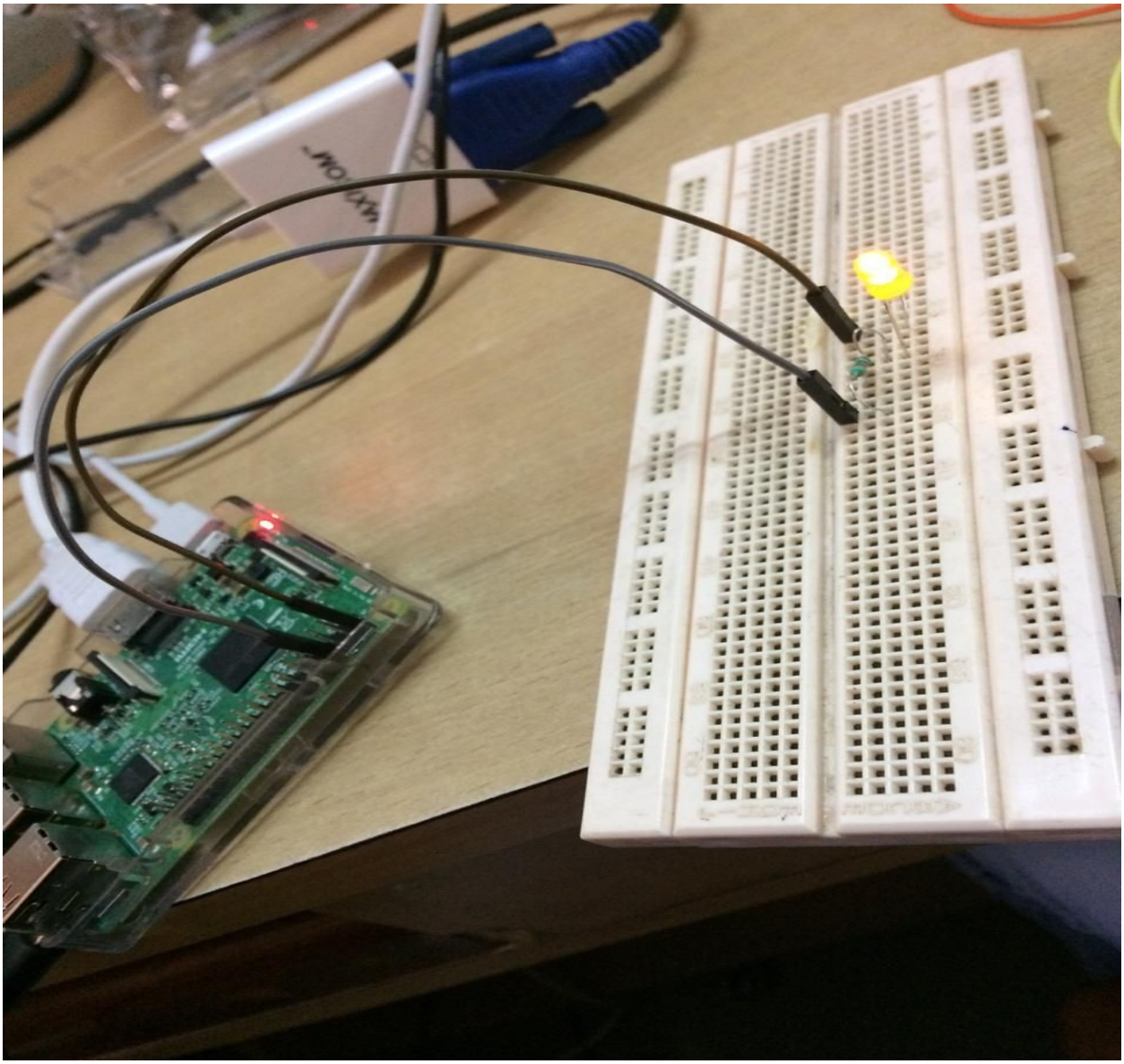
Step 11 →

Click on LED ON:1 for blow the LED .
Click on LED OFF:0 for Shut the LED .


[vokoscreen 2.5.0]
Node-RED : 192.168....
08:50
Node-RED : 192.168.1.111 - Chromium
Node-RED : 192.16
192.168.1.111:1880/#flow/460d172.70093e8
Node-RED
Successfully injected: 1
Deploy
filter nodes
Flow 21
debug
dashboar
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
Watson IoT
serial
output
LED ON:1
LED
LED OFF:0
Information
Node
"d4ca6b60.5ada08"
Type
inject
show more
Node Help
Injects a message into a flow either manually or at regular intervals. The message payload can be a variety of types, including strings, JavaScript objects or the current time.
Outputs
payload
various
The configured payload of the message.
topic
string
An optional property that can be
Switch flow tabs with ctrl-⇧j and ctrl-⇧k

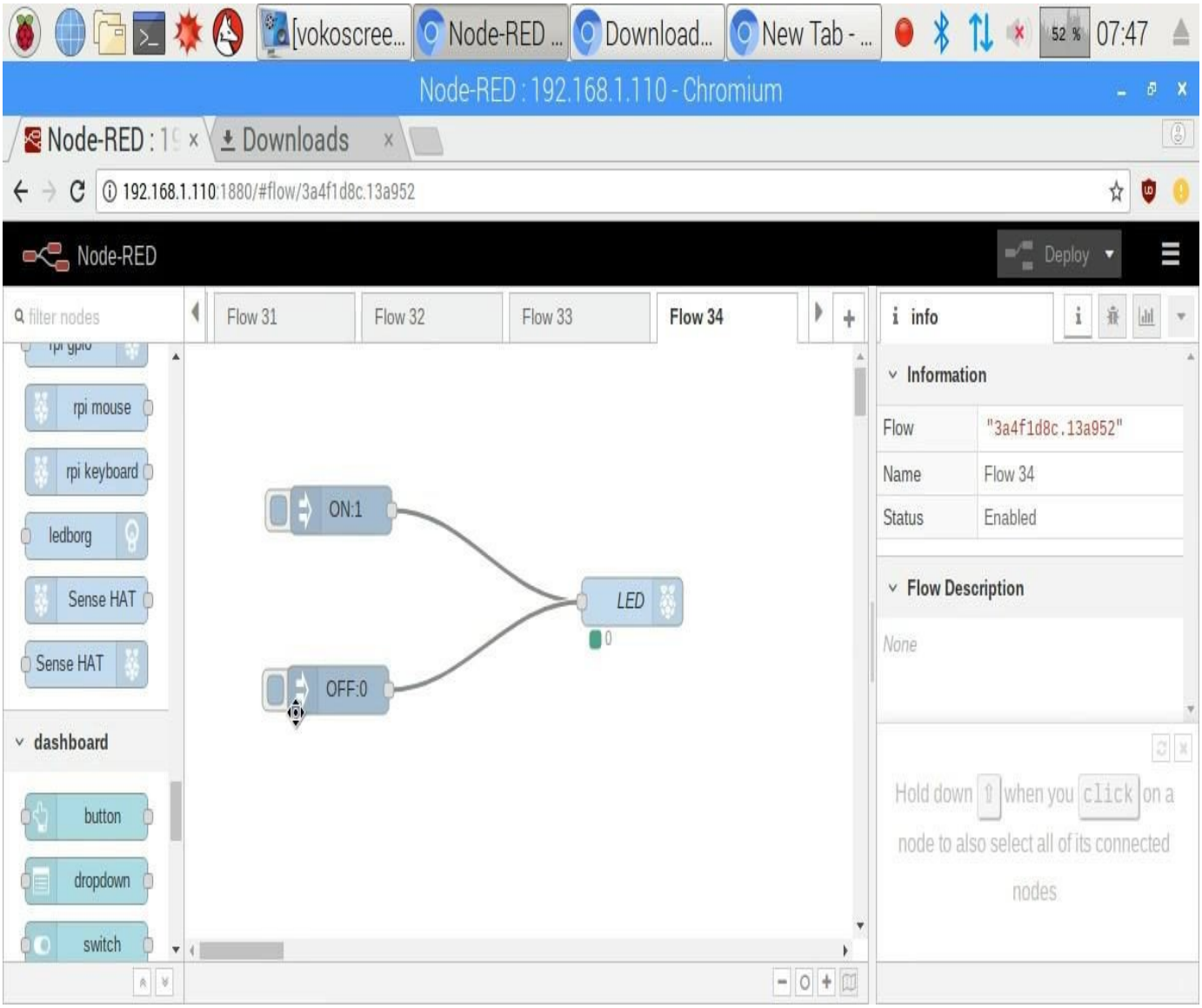

[vokoscreen 2.5.0]
Node-RED : 192.168....
08:50
Node-RED : 192.168.1.111 - Chromium
Node-RED : 192.16
192.168.1.111:1880/#flow/460d172.70093e8
Node-RED
Successfully injected: 0
Deploy
filter nodes
Flow 21
debug
dashboar
input
inject
catch
status
link
mqtt
http
websocket
tcp
udp
Watson IoT
serial
output
LED ON:1
LED
LED OFF:0
Information
Flow
"460d172.70093e8"
Name
Flow 23
Status
Enabled
Flow Description
None
Switch flow tabs with ctrl-⇧j and ctrl-⇧k
192.168.1.111:1880/#

HARDWARE COMPONENT

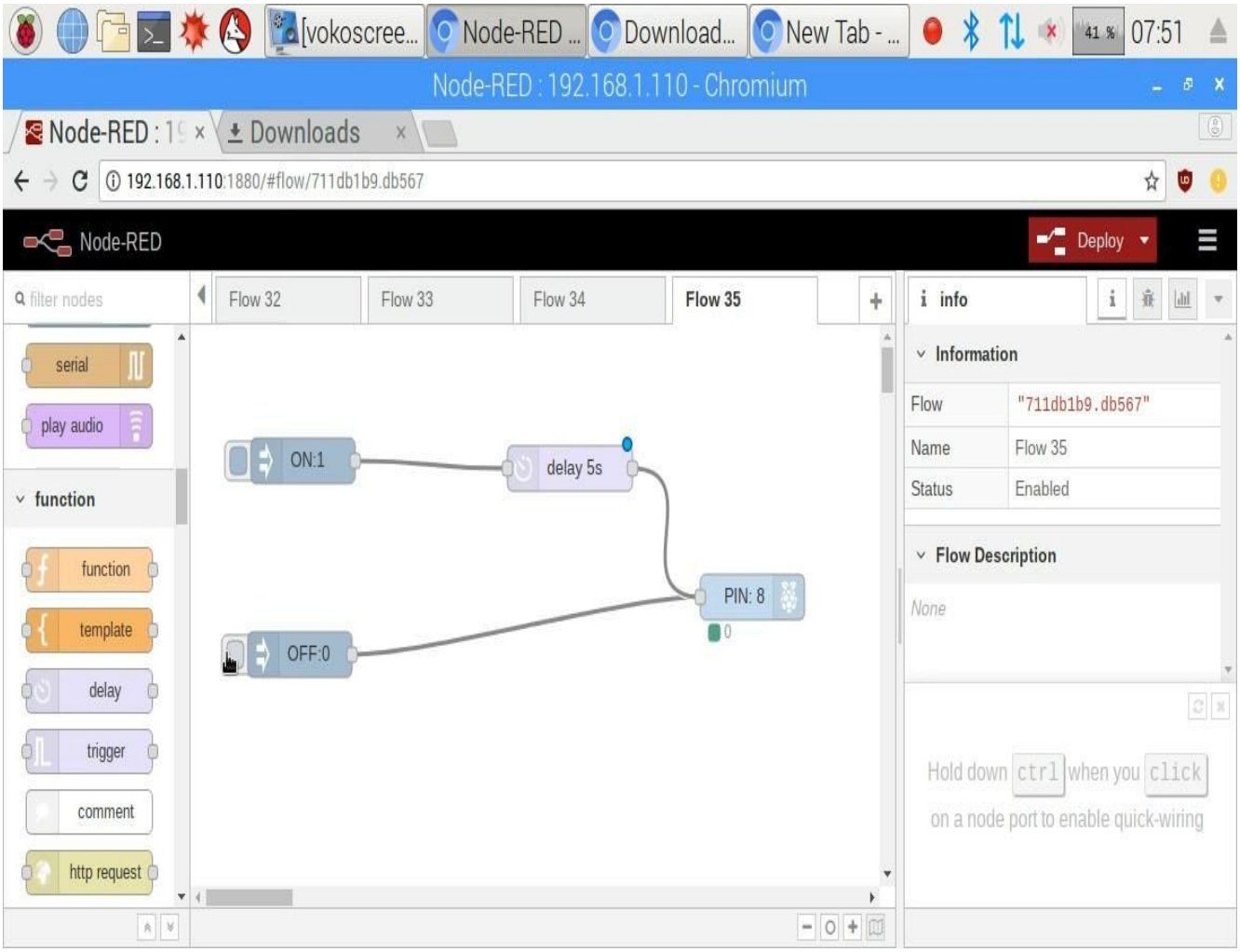


Following are the examples of the programs performed using Node-RED

1. Glowing up a LED using Node-RED.



2. Glowing up a LED with delay using Node-RED.

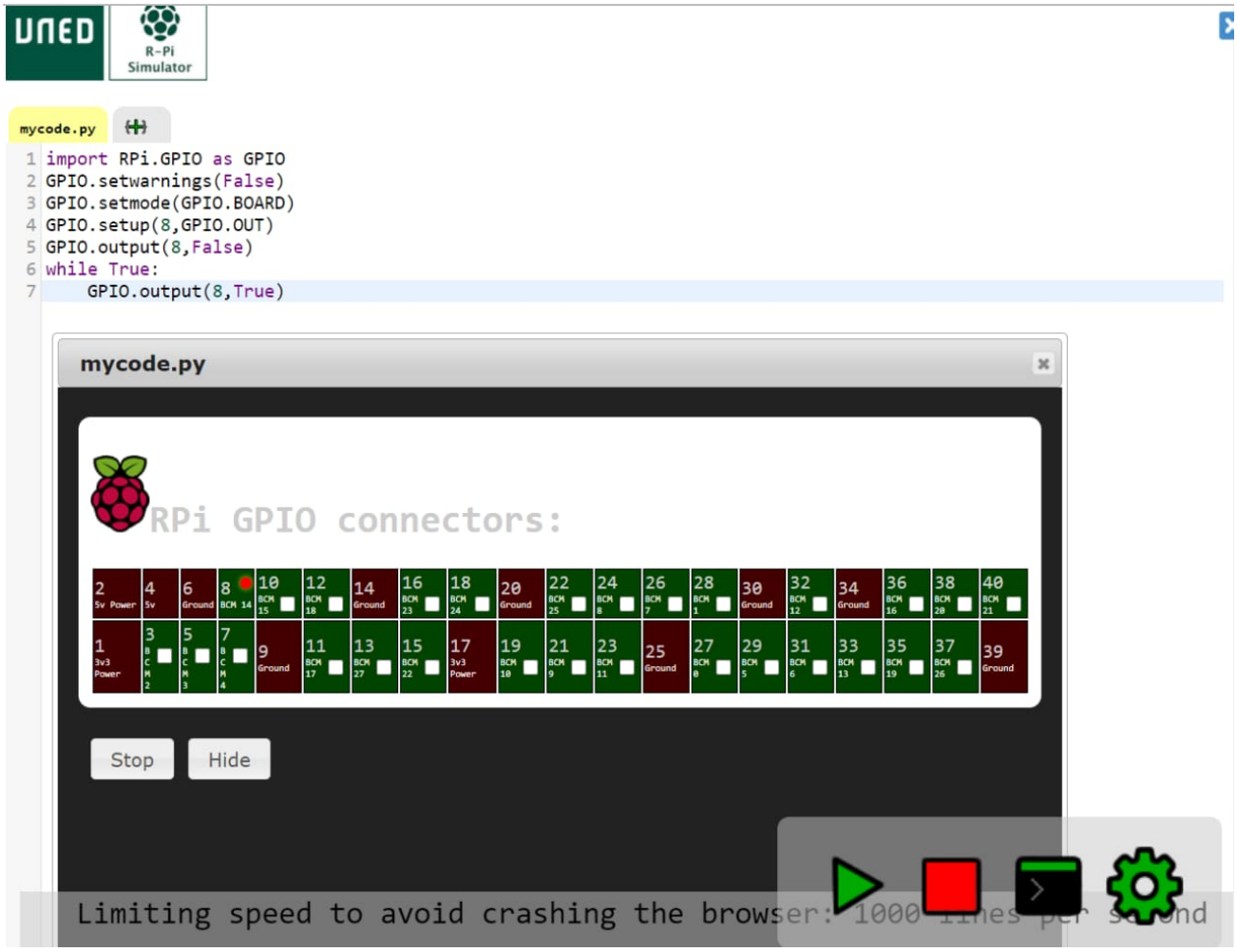


PRACTICAL NO . 8

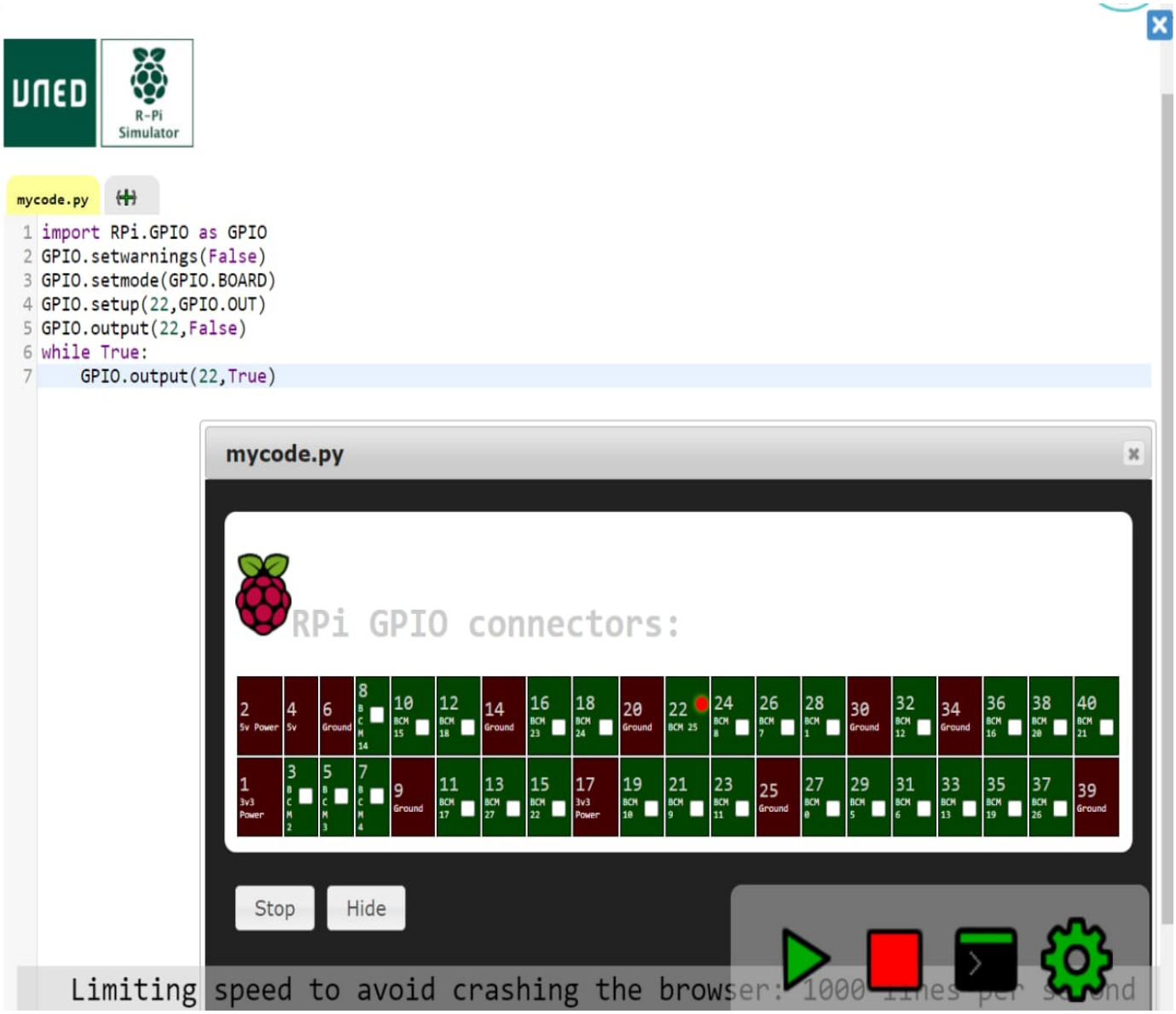
Raspberry Pi Programming Using Online Simulator .

GPIO LED WITH PYTHON PROGRAMS :

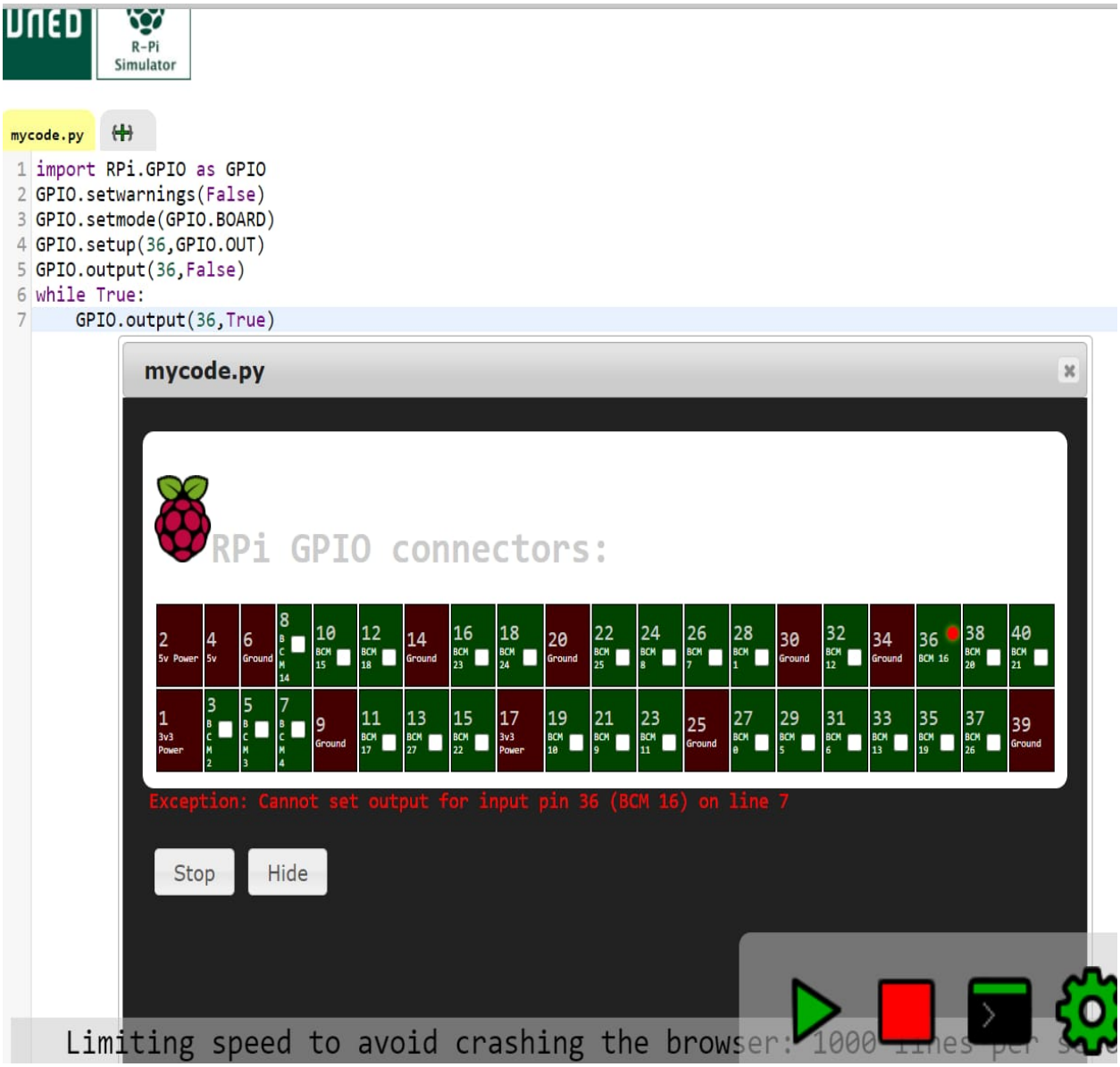
1. Write a program to glow an LED in board mode by using pin number 8.



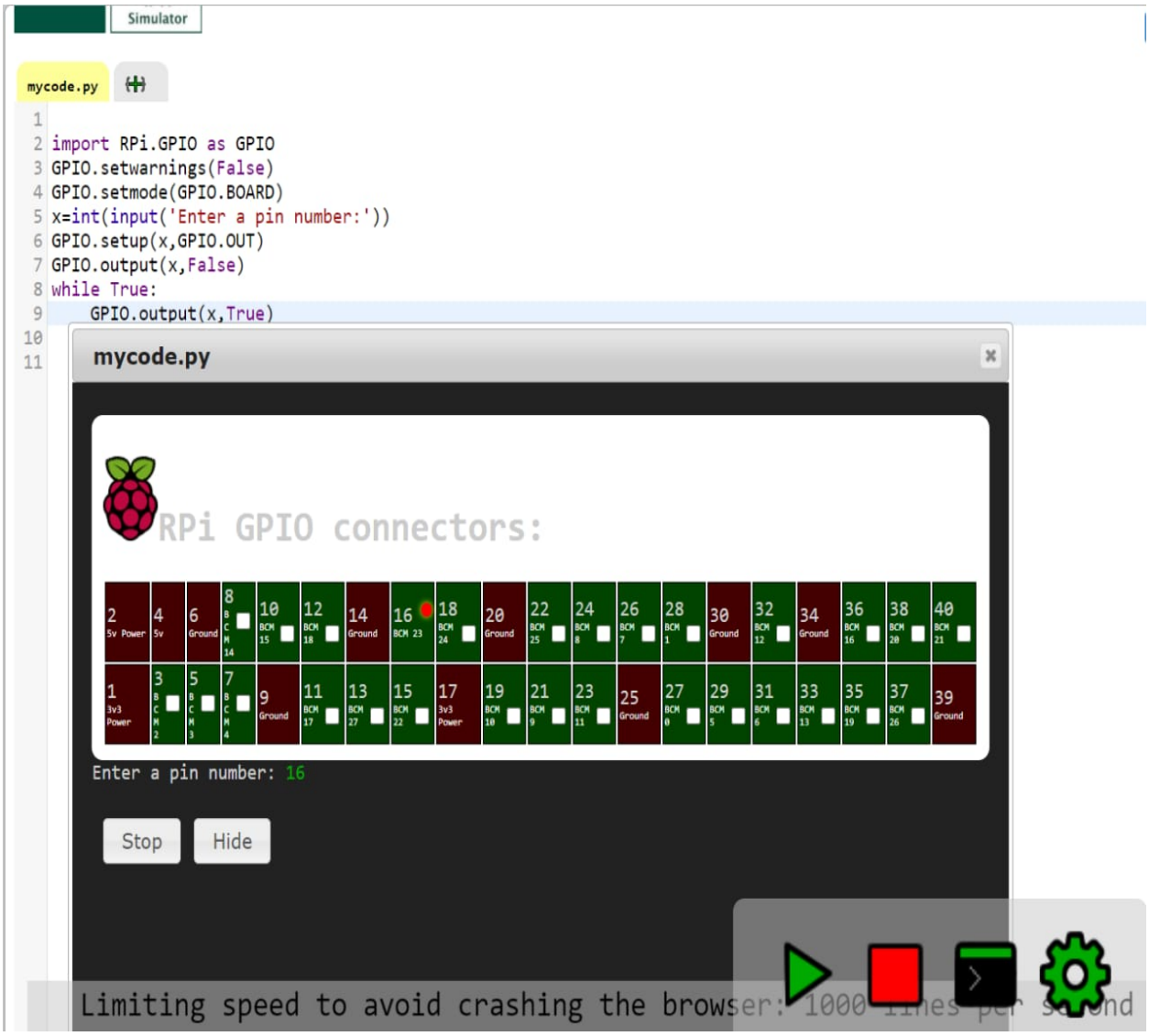
2. Write a program to glow an LED in board mode by using any pin number.



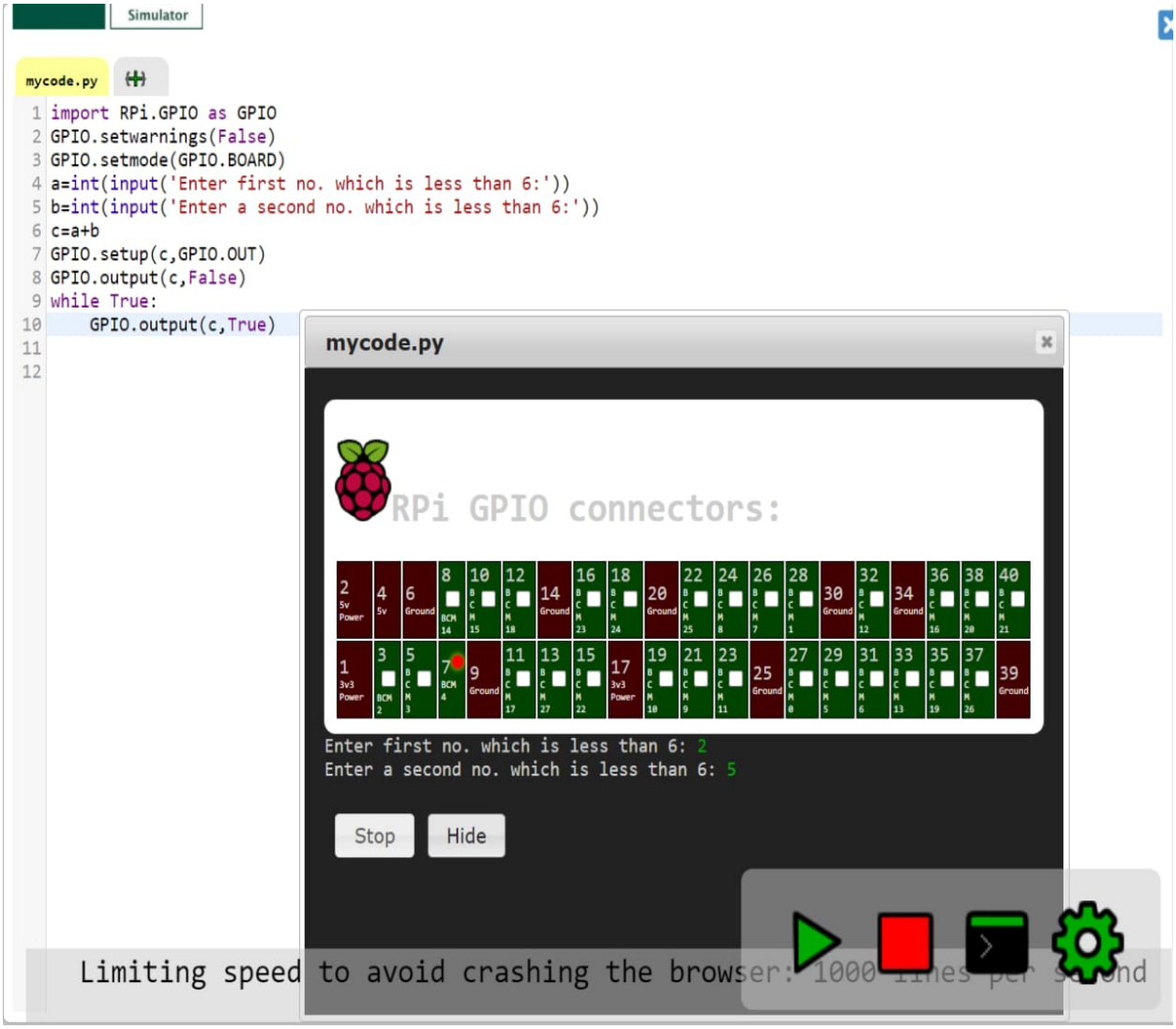
3. Write a program to glow an LED at pin number of 36 Board mode.



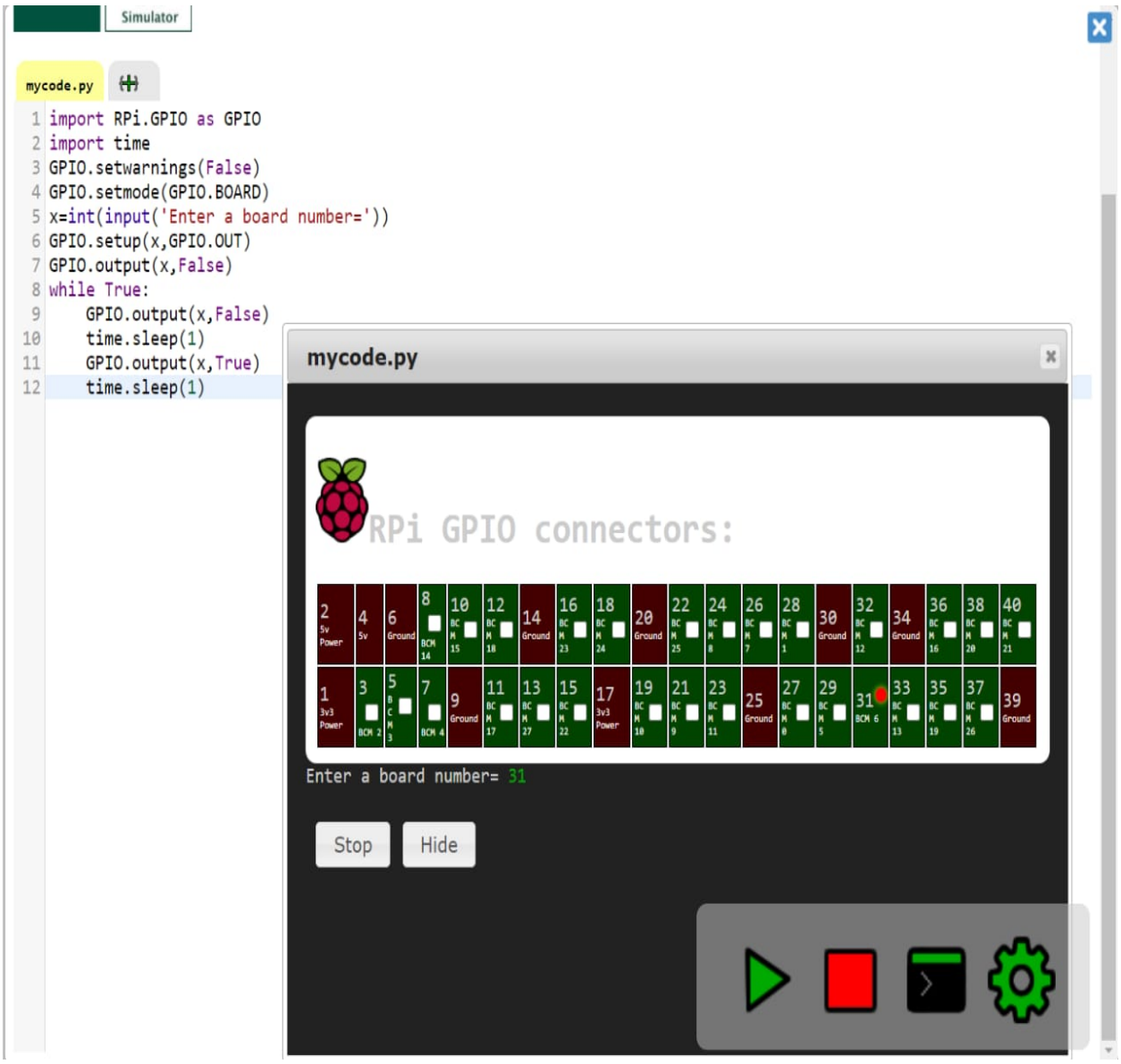
4.Write a program to glow the LED by accepting pin number from the user in BOARD mode.



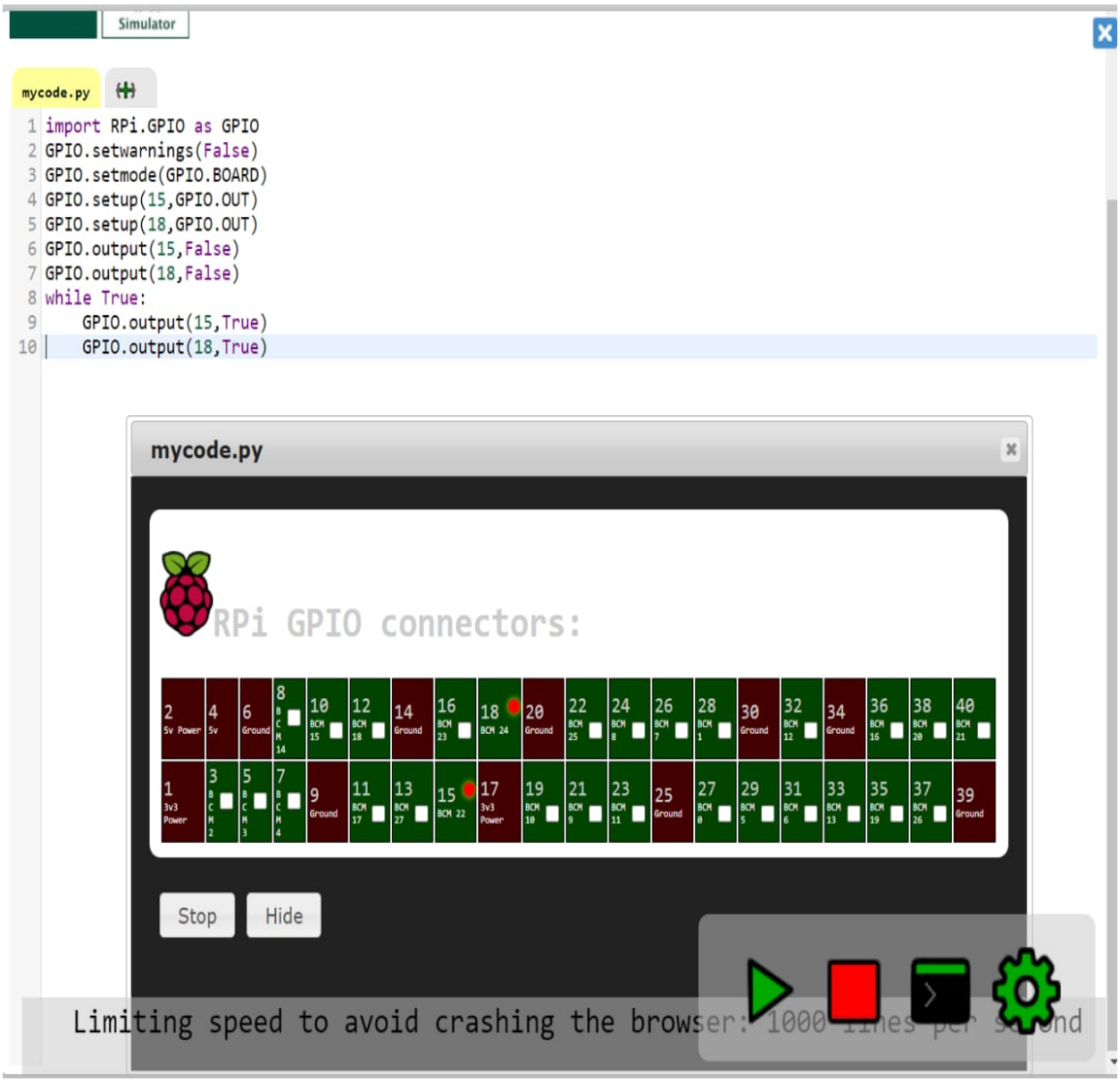
5. Write a program to glow the LED by accepting any 2 number from the user (number should be less than 6). .Add 2 nos and result of addtion should be your pin no. to glow an LED.



6.Write a program to blink an Led by accepting the pin no and sleep time from user.

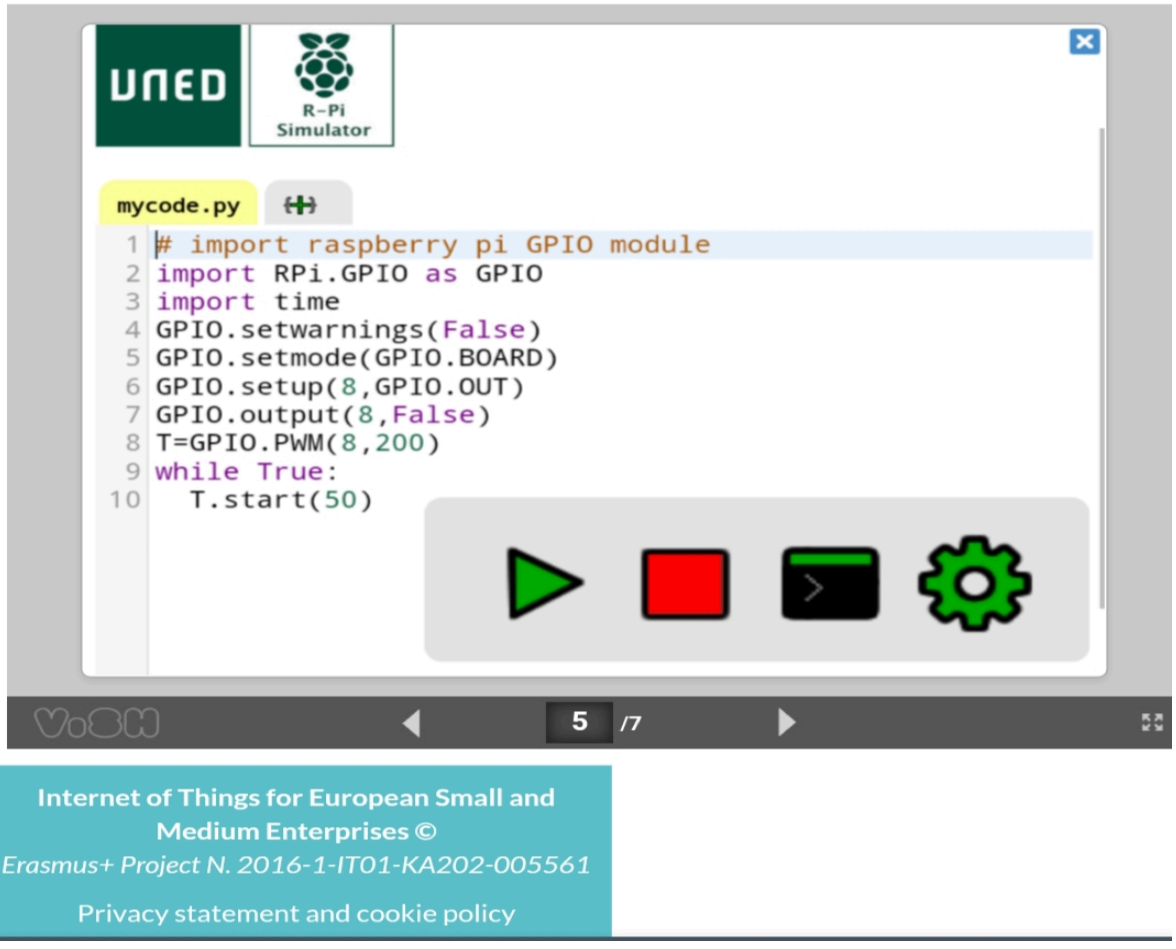


7. Write a program to glow any two LED's in BOARD mode.

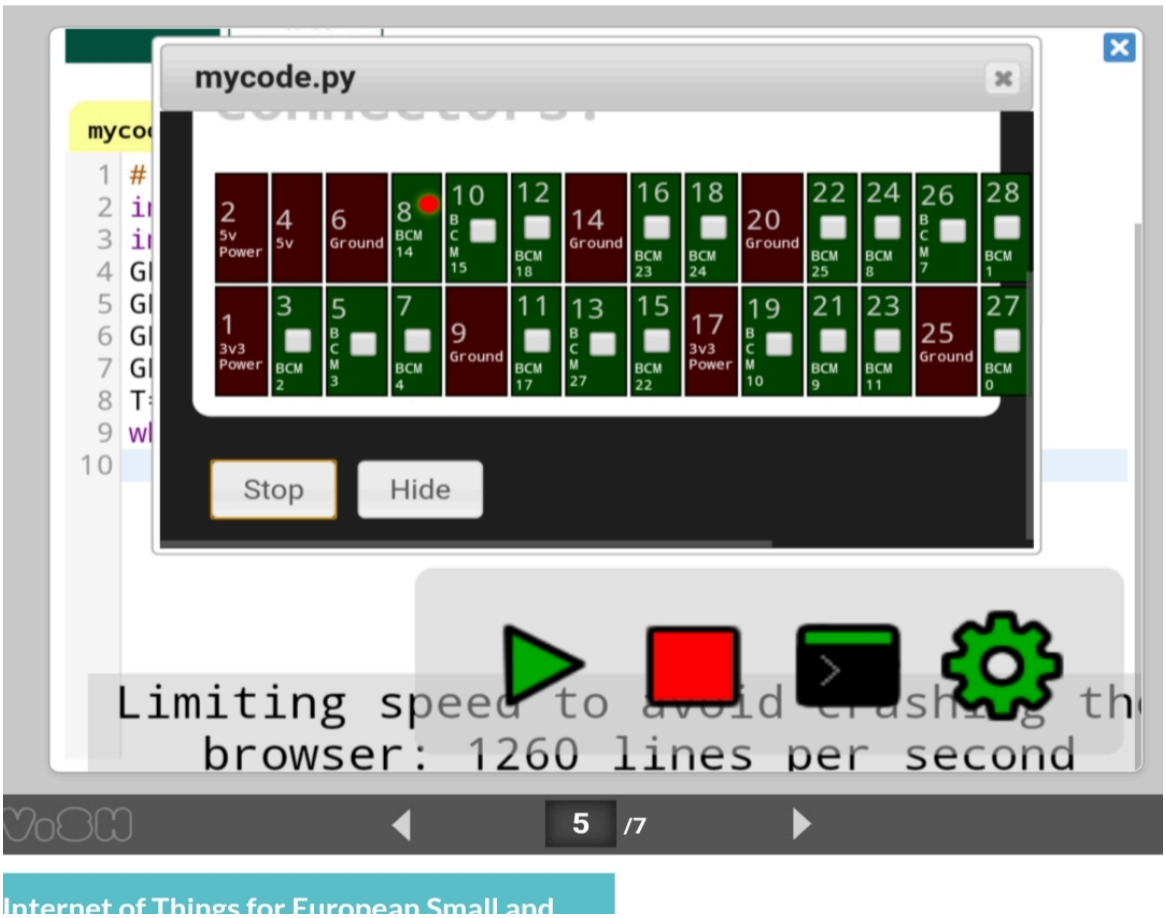


PWM PROGRAMS :

1. WRITE A PROGRAM TO INCREASE THE INTENSITY OF AN LED BY USING PWM.

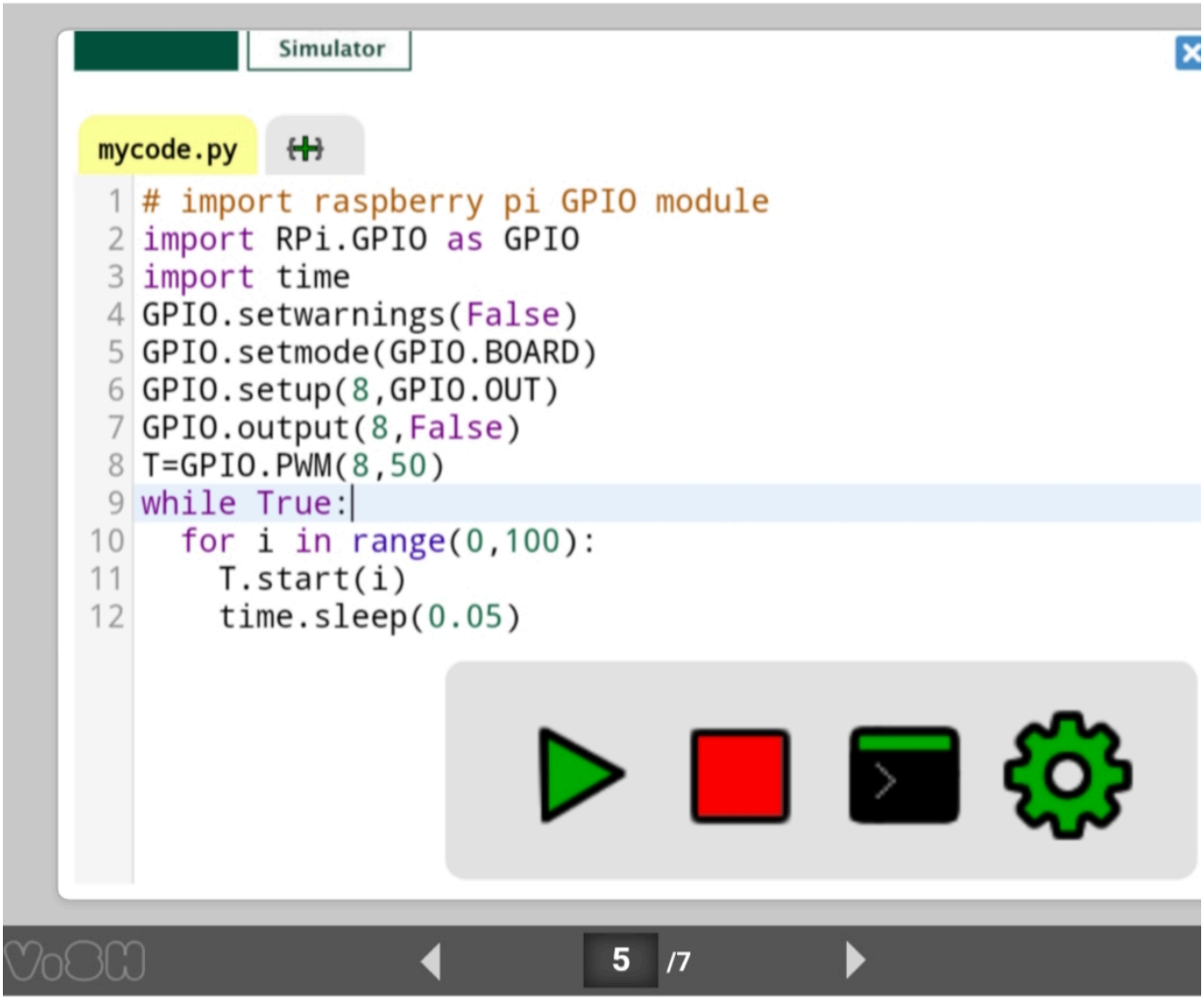


OUTPUT :



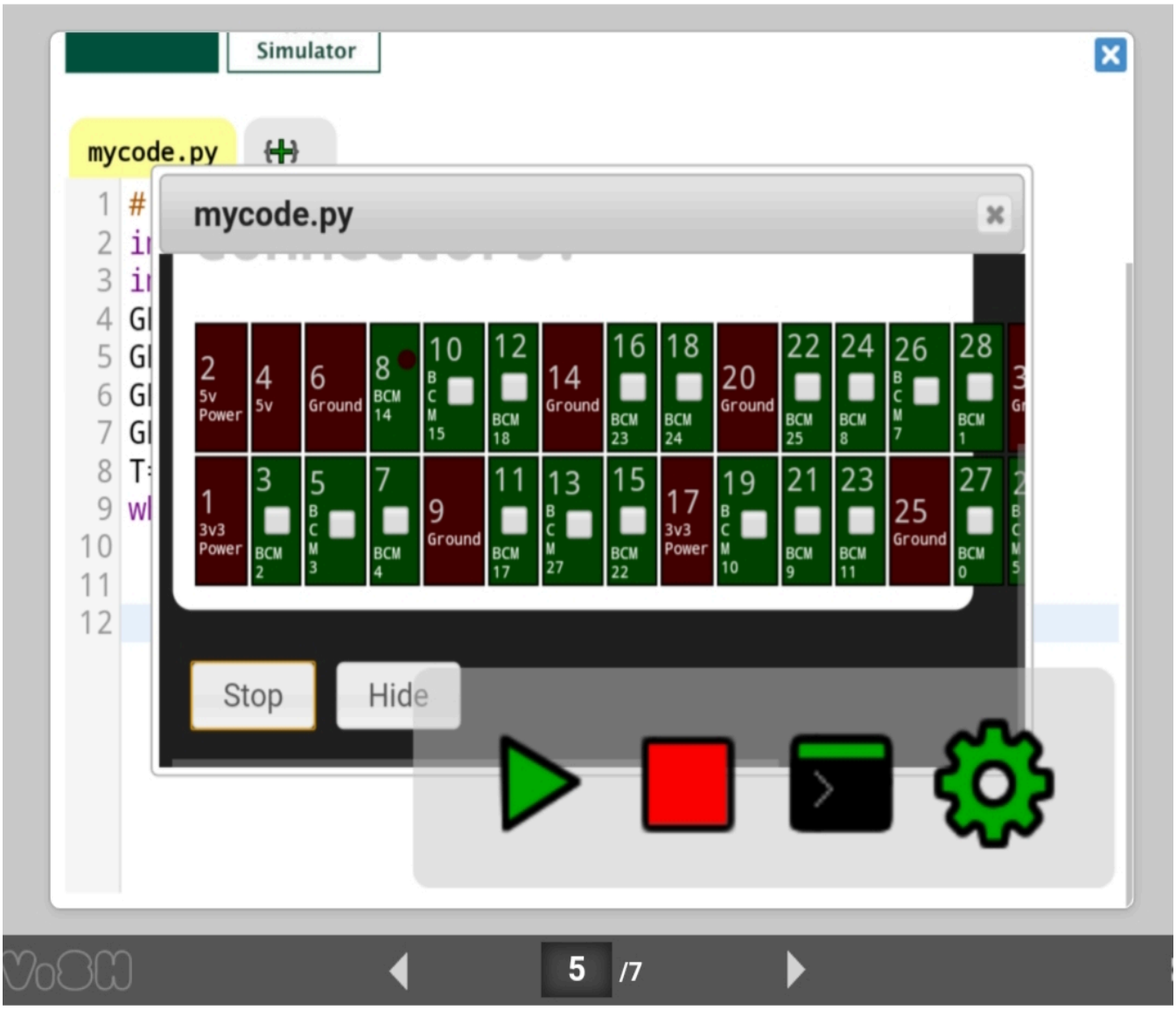
2. WRITE A PROGRAM TO INCREASE THE INTENSITY OF AN LED BY USING FOR LOOP IN PWM .

➔



Internet of Things for European Small and

OUTPUT :


Simulator
mycode.py
mycode.py
Stop
Hide
5
/7

PRACTICAL NO.9

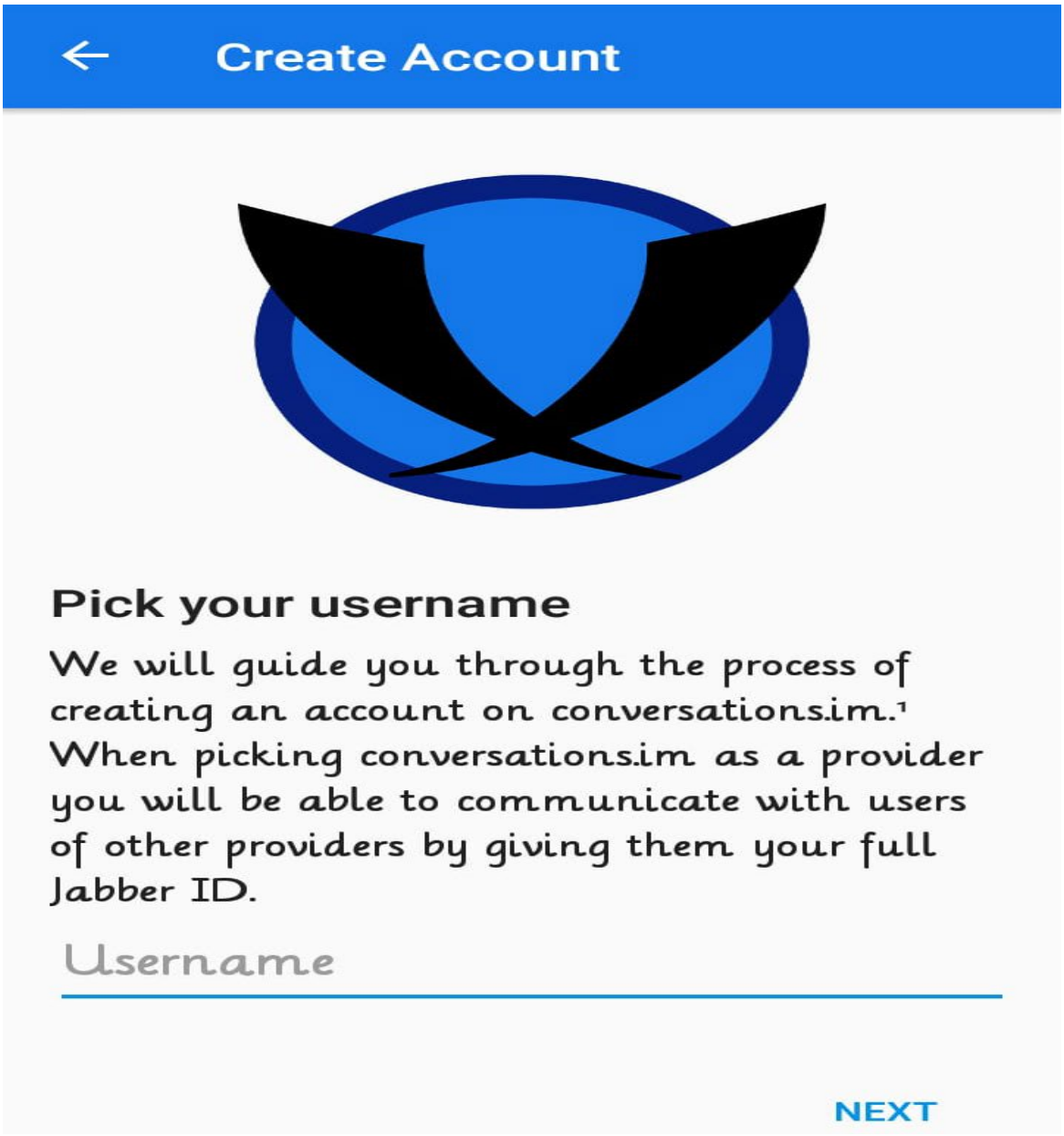
STUDY OF XMPP SERVER PROTOCOL

STEP 1 :

Download the XMPP app on your device .

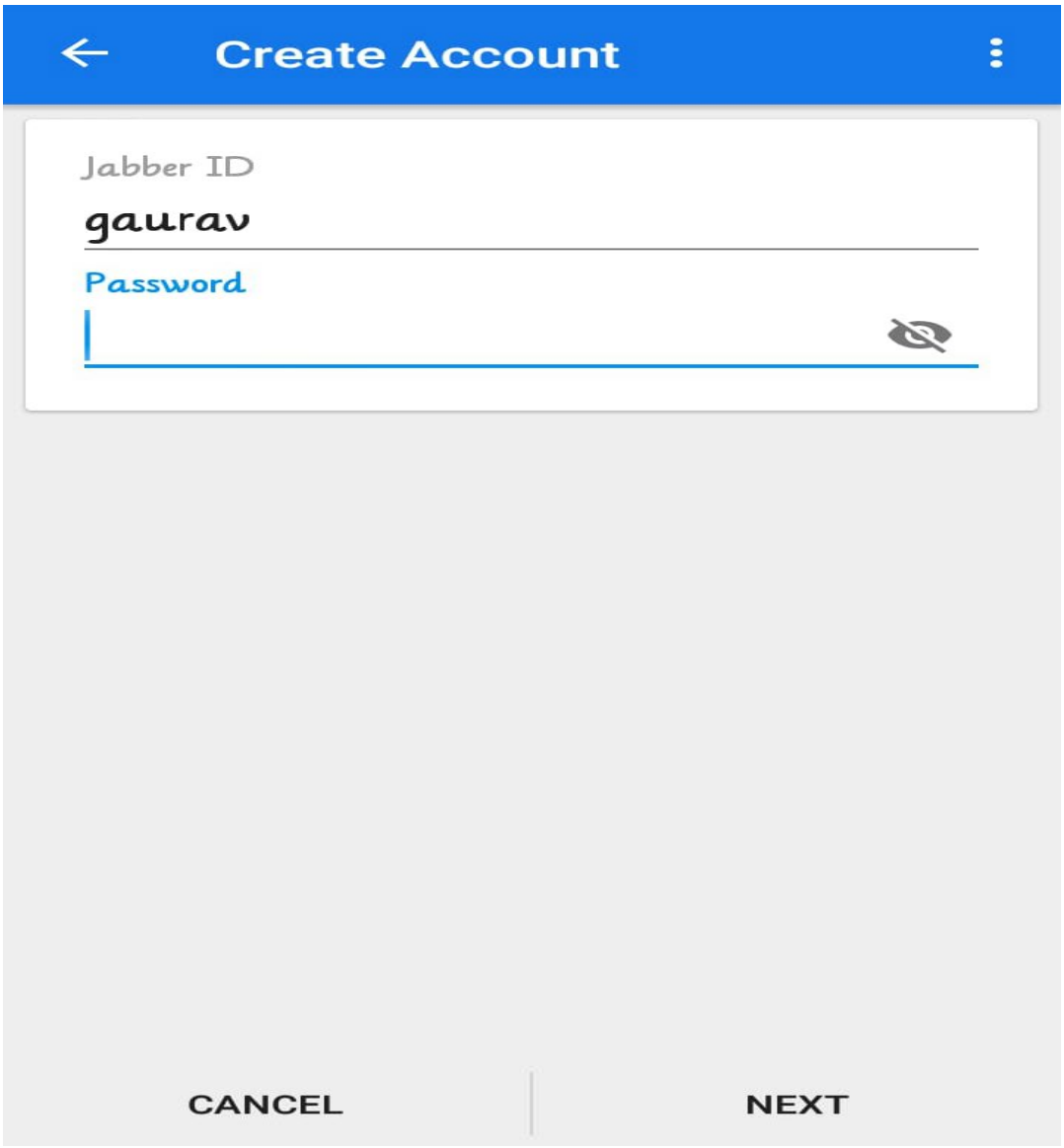
Step 2 :

Create an account on the app which is also called as JABBER ID .



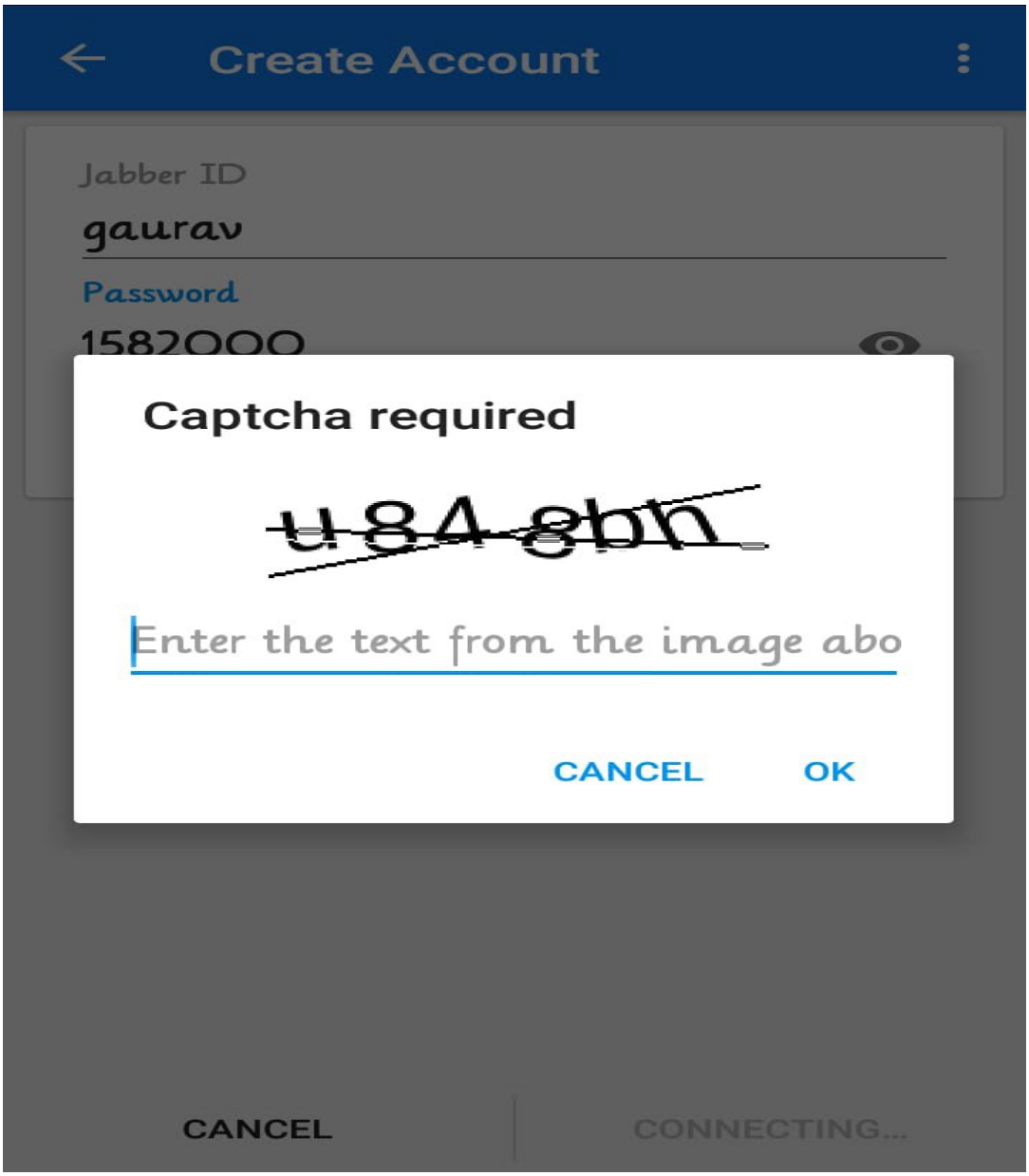
Step 3 :

Set an password for your JABBER ID .



Step 4 :

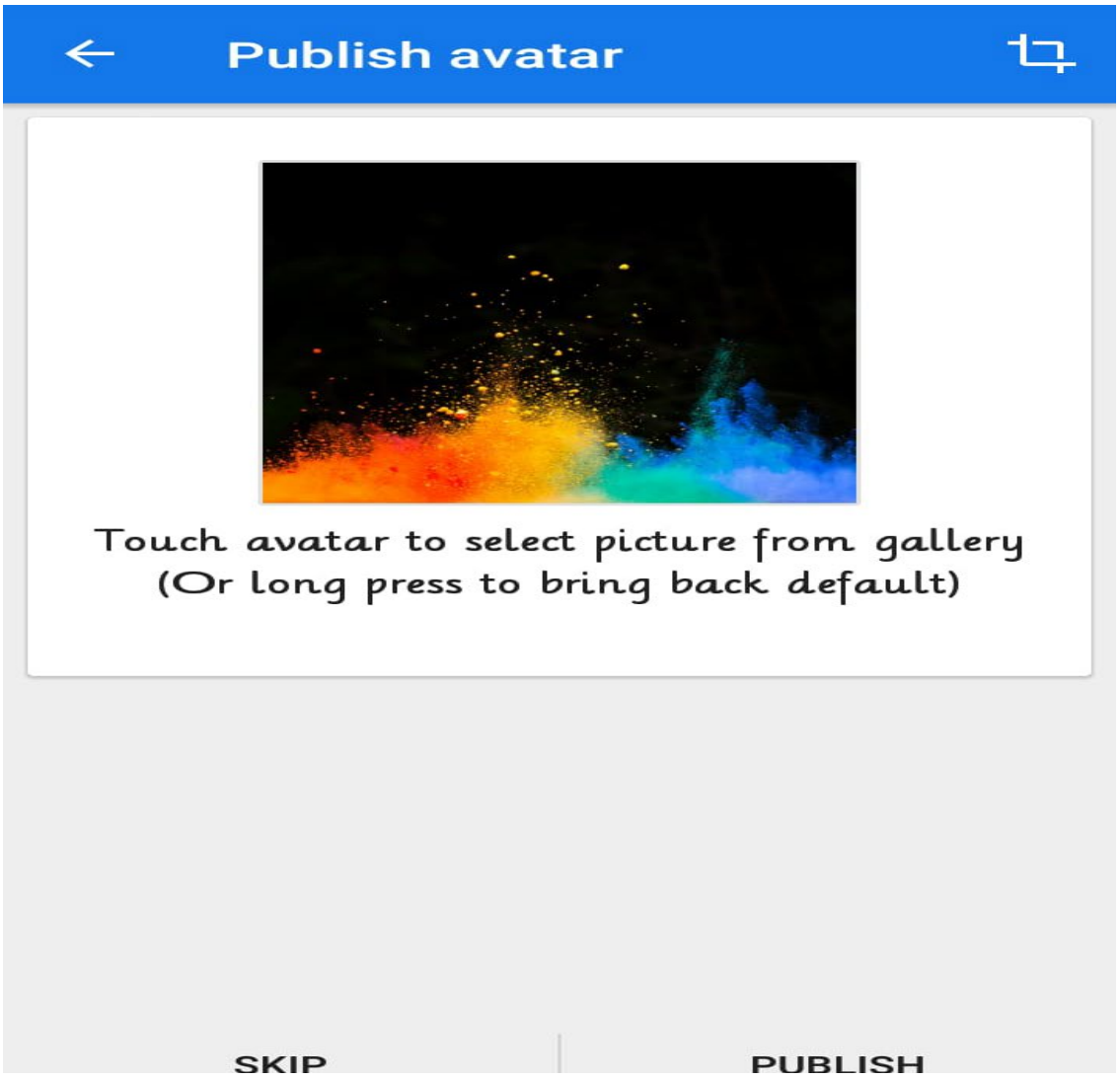
Enter the Captcha Code mentioned in the page.



Step 5 :

Set Avatar to publish of your choice .

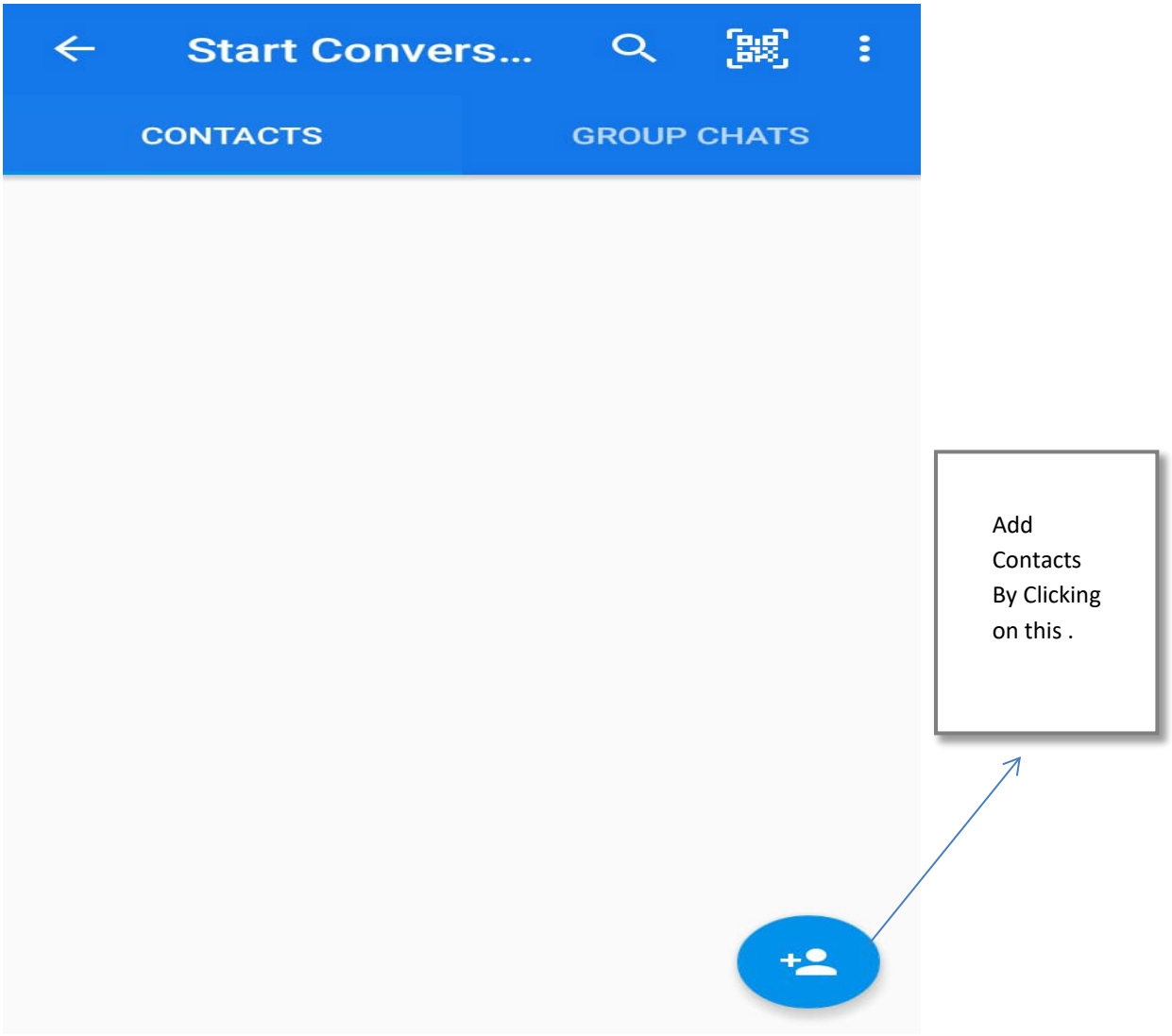
Or long press for default avatar .



Step 6 :

Your account is created and ready to use .

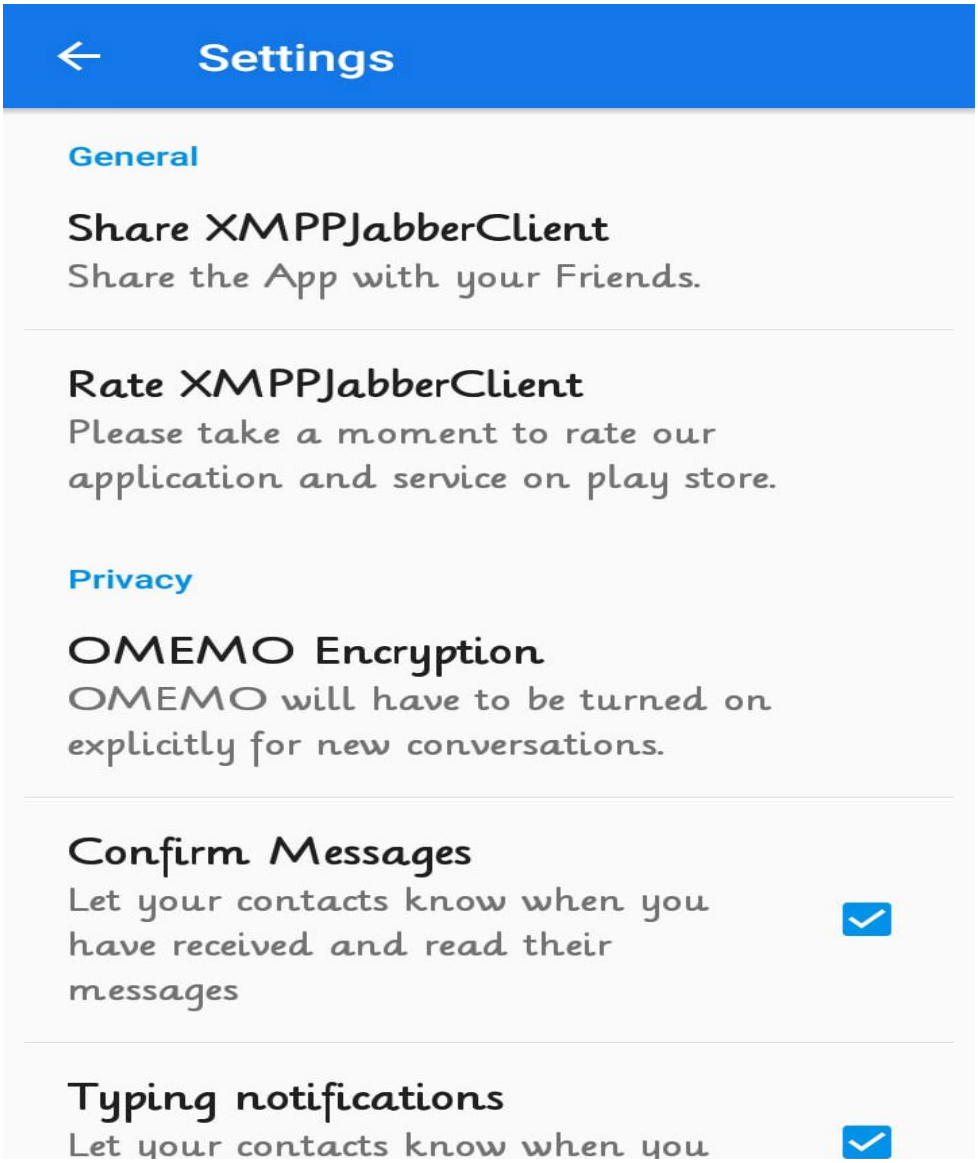
And now can add others to chat in app .



Step 7 :

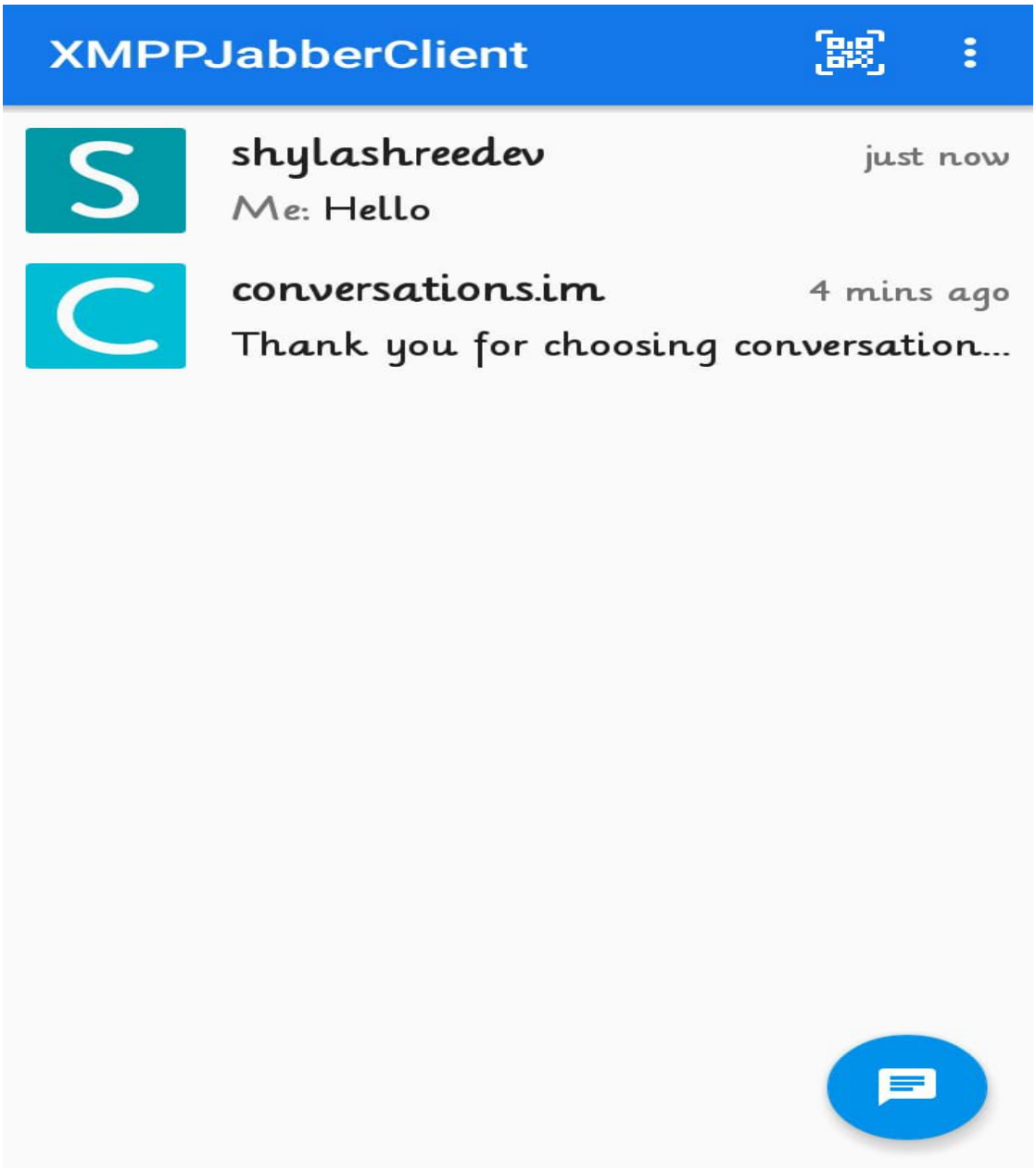
You can do various settings like encryption , notifications .

By clicking on settings in the app .



Step 8 :

Now you can message on the app by adding anyone you want to talk .

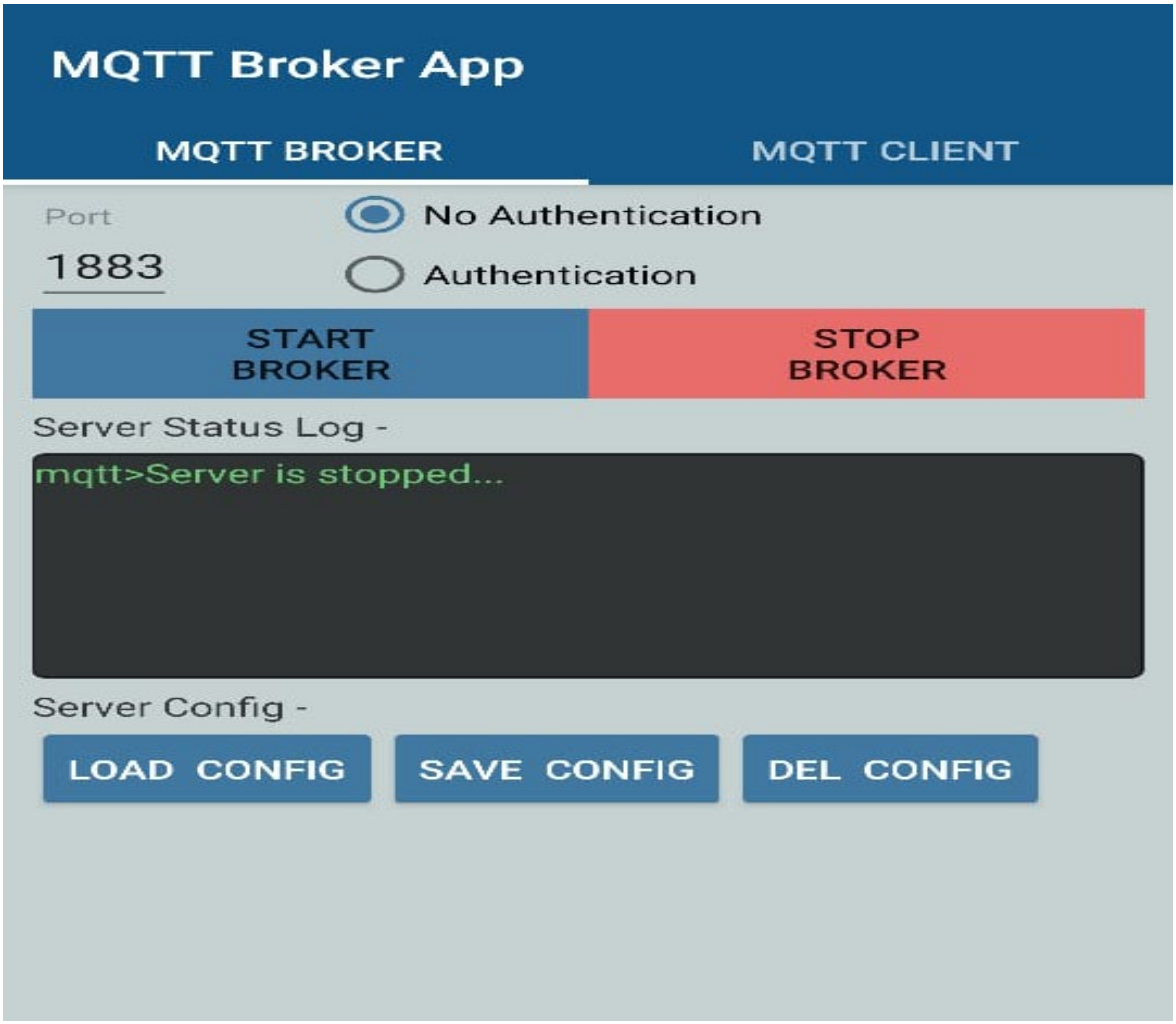

XMPPJabberClient
S
shylashreedev
just now
Me: Hello
C
conversations.im
4 mins ago
Thank you for choosing conversation...

PRACTICAL NO. 10

STUDY OF MQTT PROTOCOL

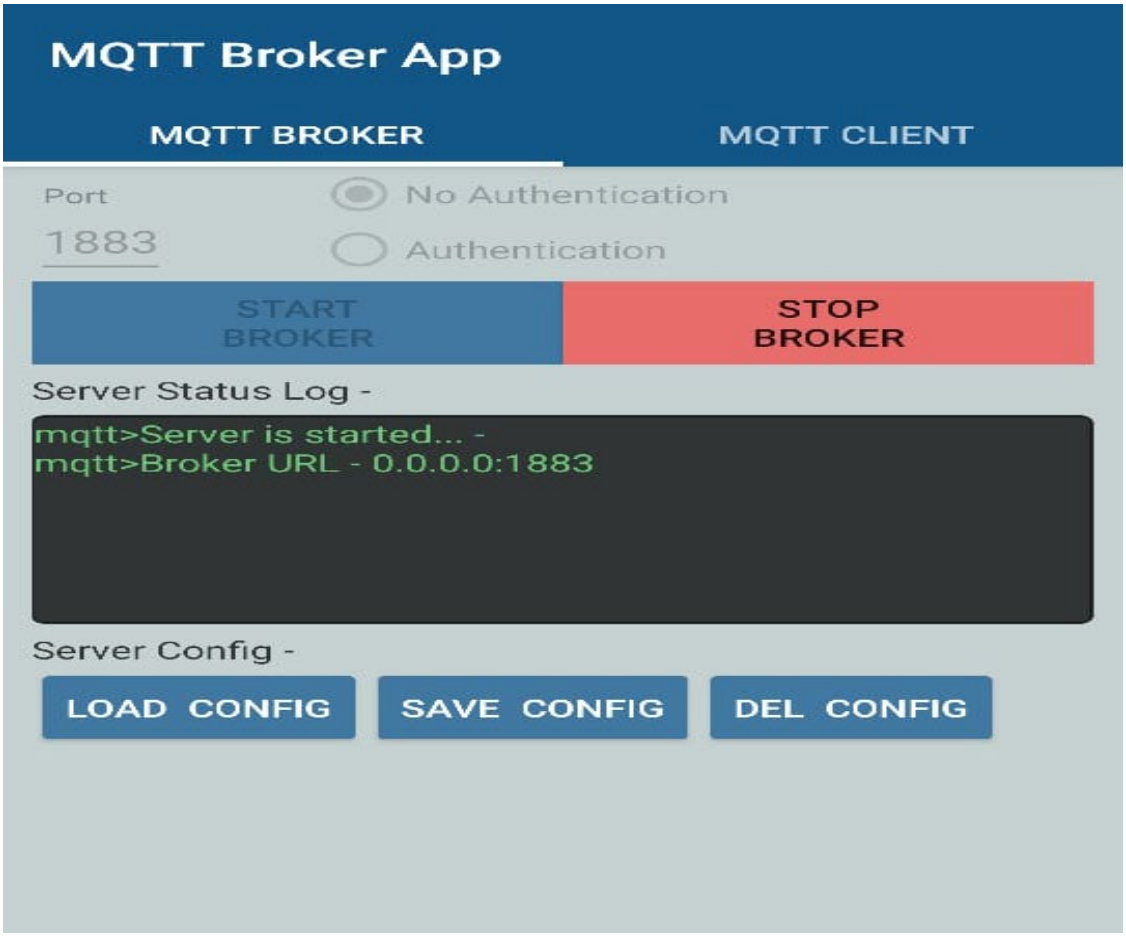
Step 1 :

Download the MQTT app on your device .



Step 2 :

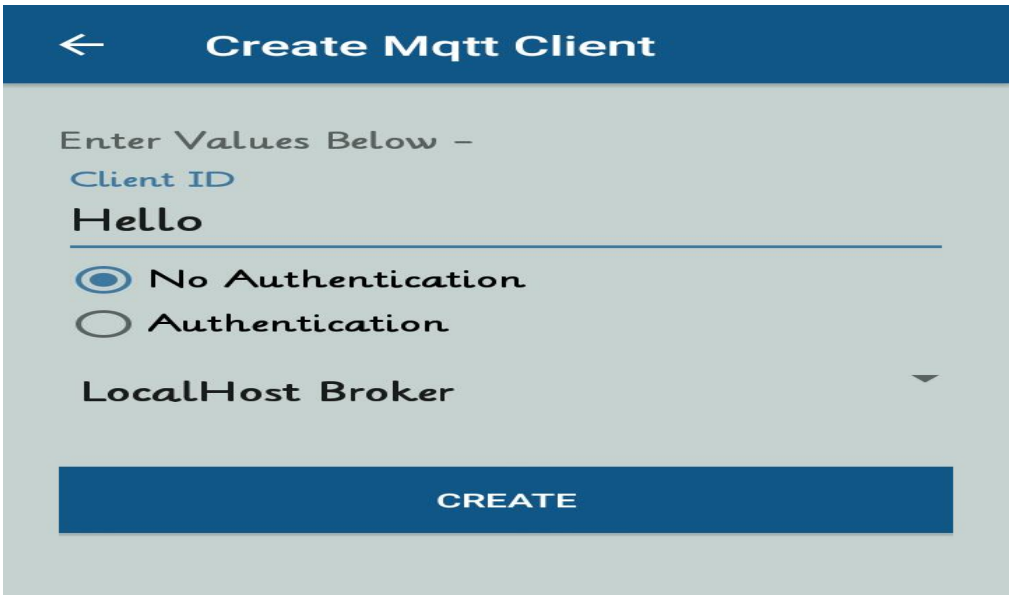
Click on **start broker** in the page .



Step 3 :

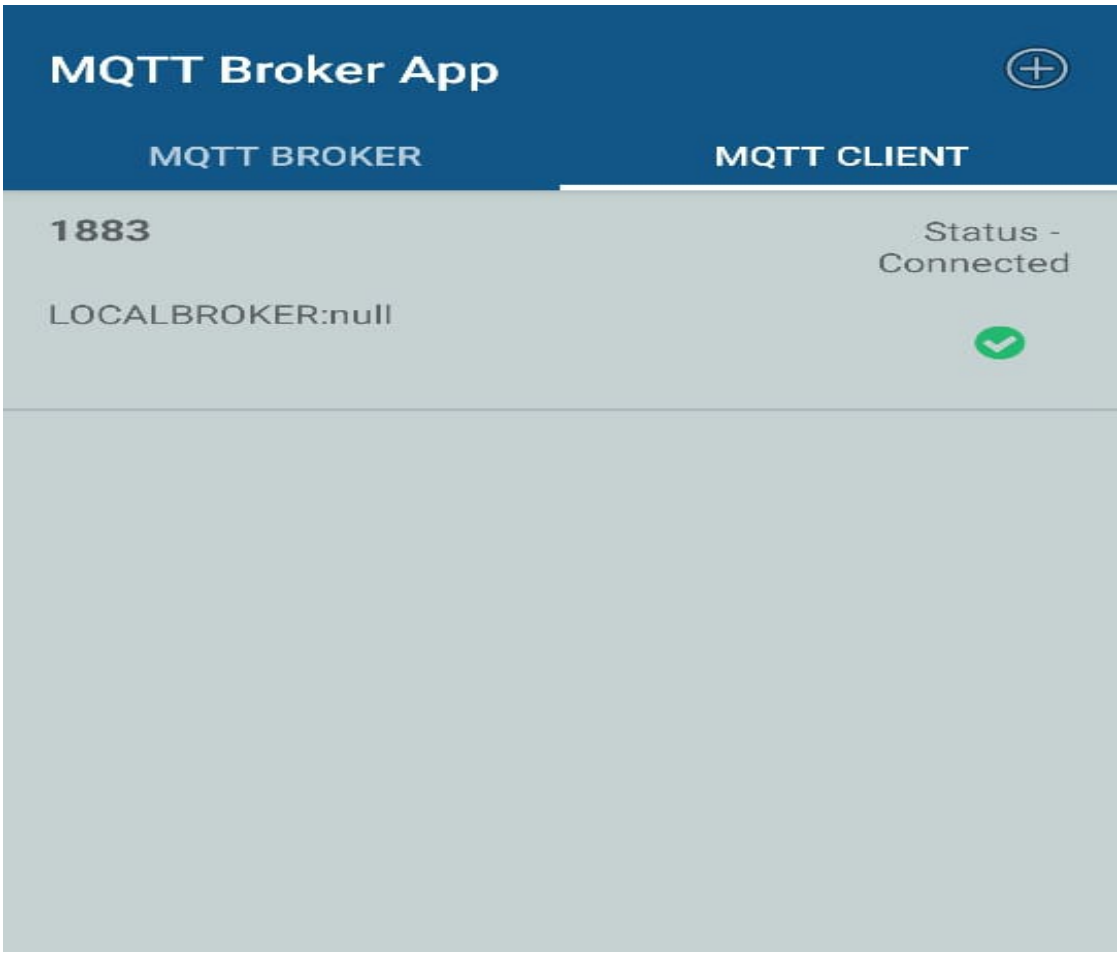
Goto **MQTT Client** and click on top corner symbol and create an **client id** for protocol

Choose Broker as **Local Host** Broker .



Step 4 :

Client Id gets create and shows status as connected .

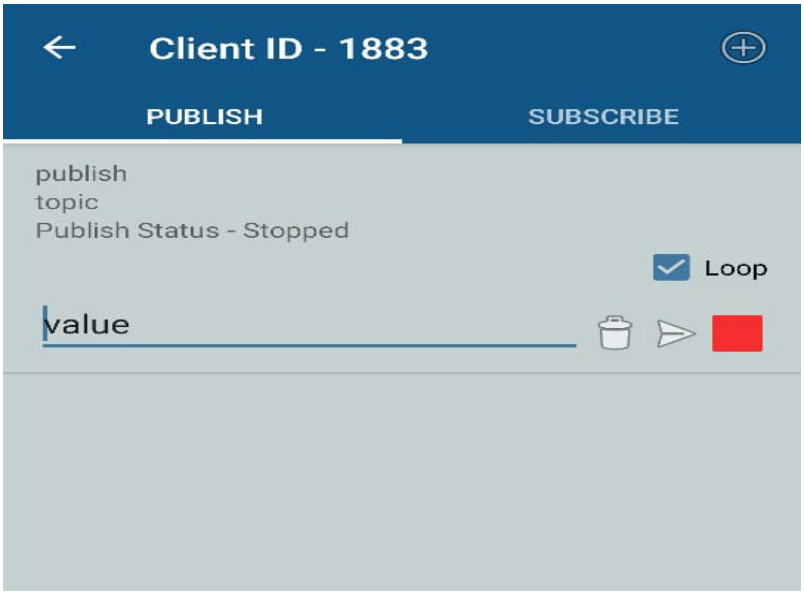


Step 5 :

Now click on it and Goto **Publish** option page .

Here publish **Name , Value , Topic** .

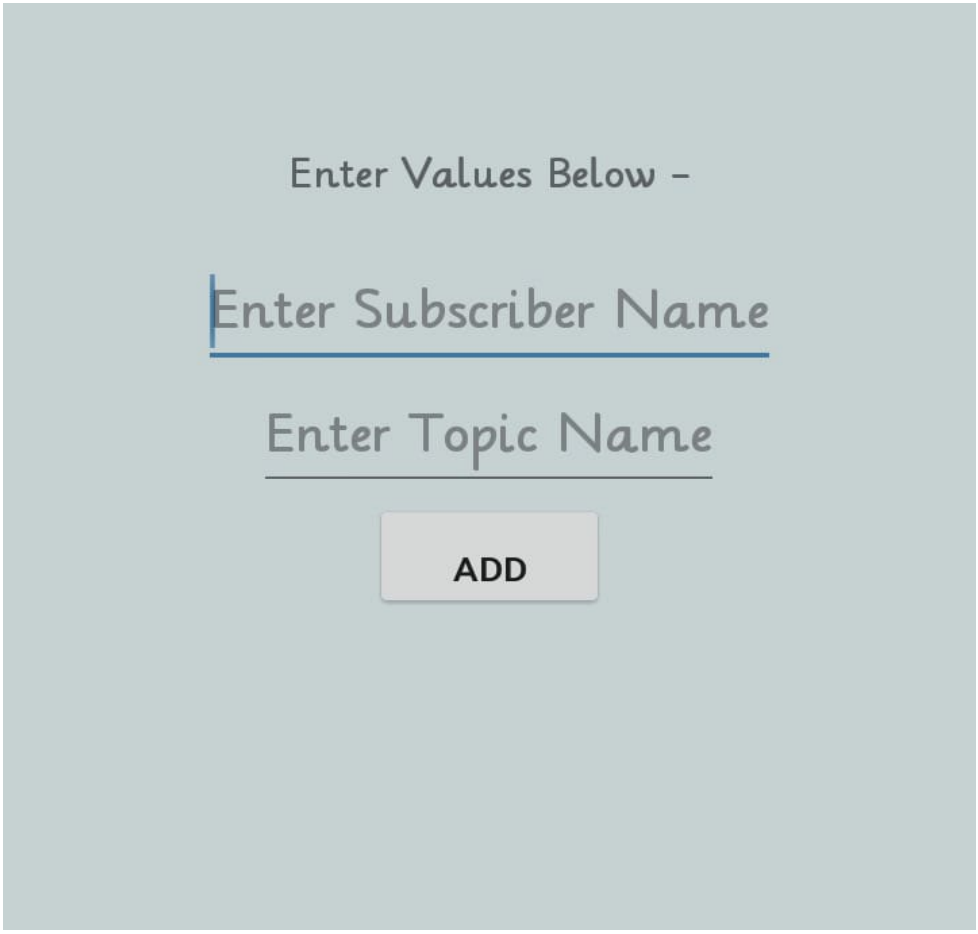
Click on add and then topic gets **published .**



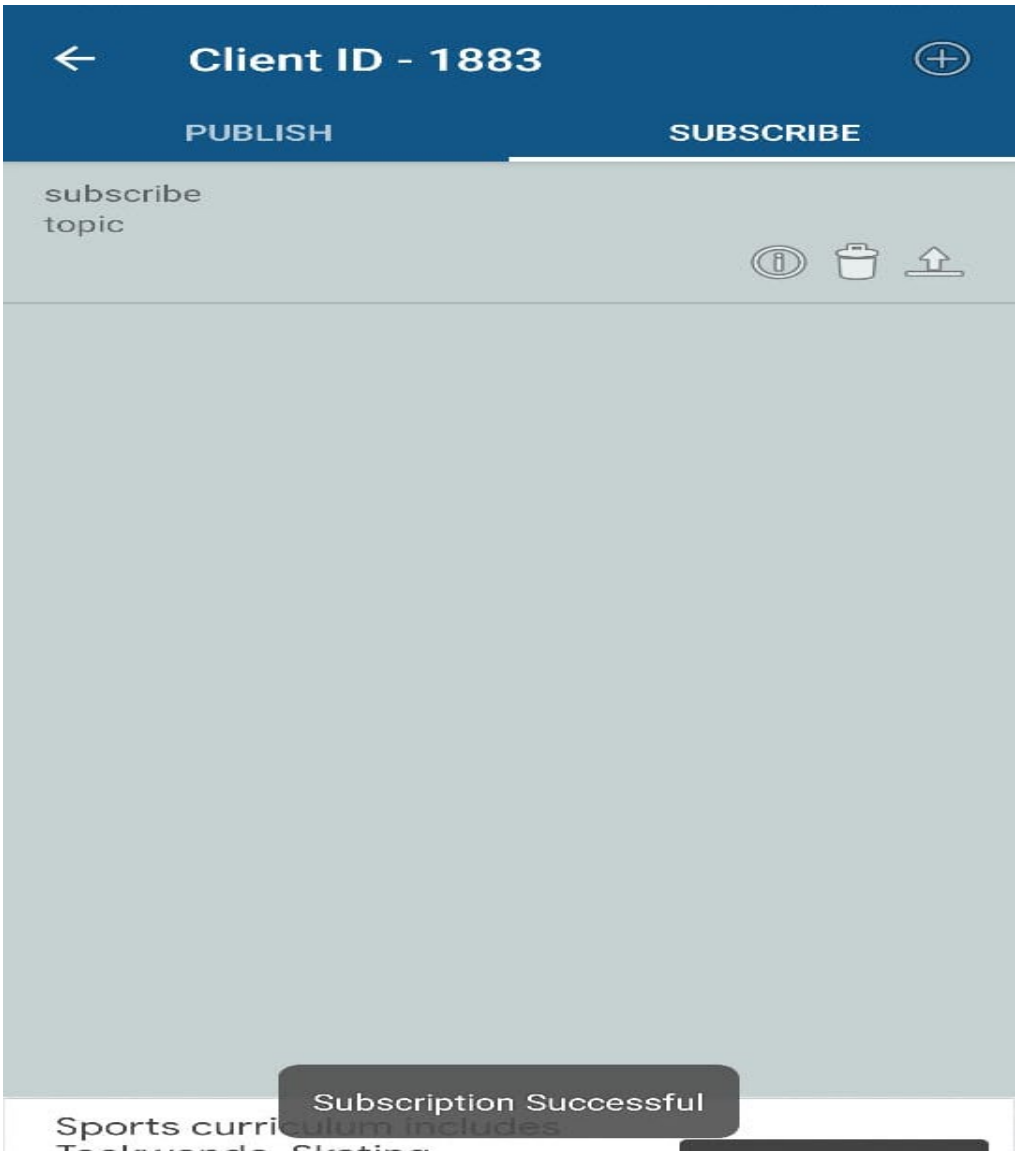
Step 6 :

Now Goto to subscribe option page .

Here enter subscriber name and topics name as given in publish .

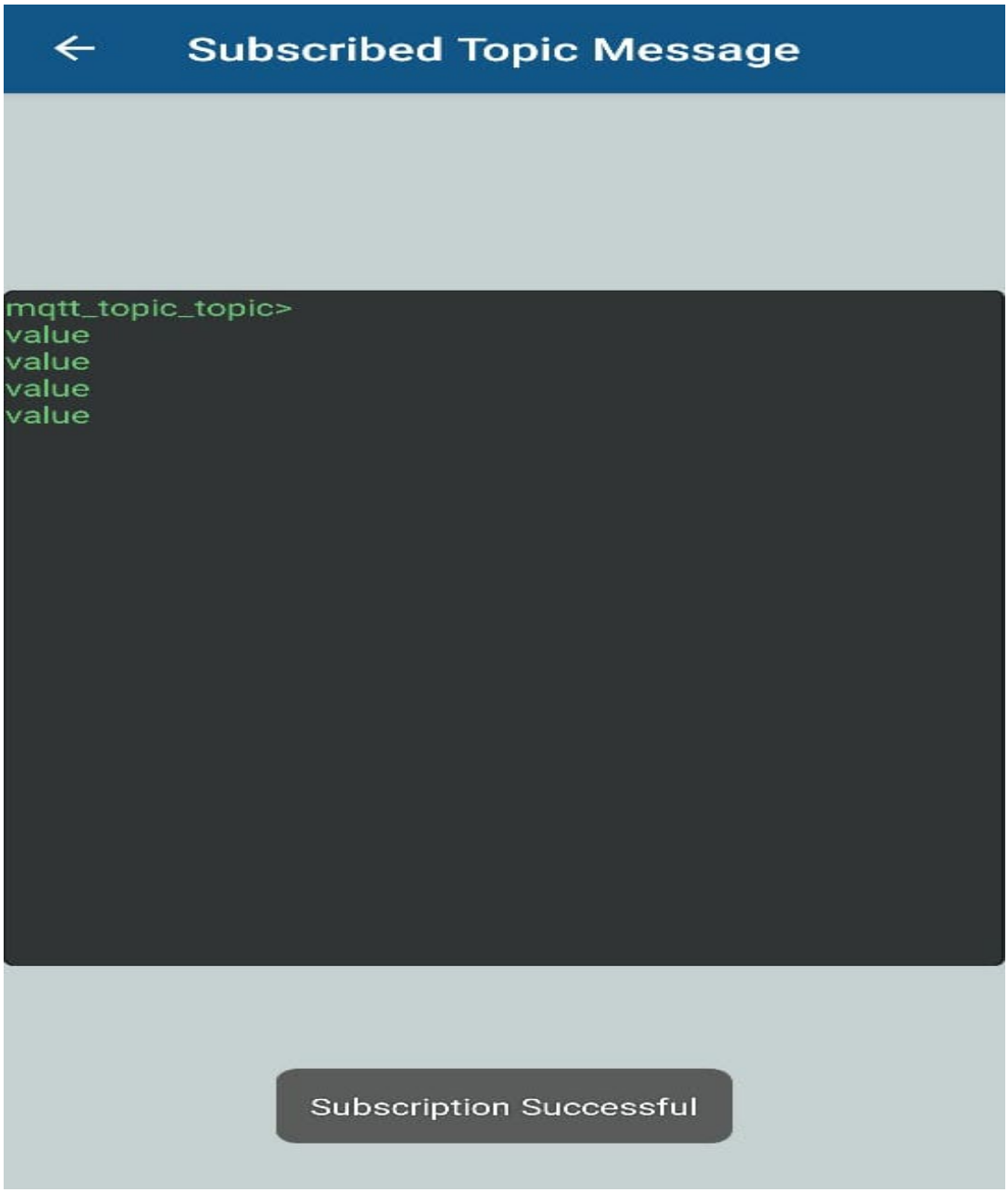


Click on add and then the topic gets **subscribed** .



Step 7 :

After subscription is successful we publish the message and it is subscribed via broker and we get an output as :


Subscribed Topic Message
mqtt_topic_topic>
value
value
value
value
Subscription Successful