

**S.Y.B.Sc**

**Computer Science**

**2019 – 2020**

**This is to certify that PRASHANT SINGH of S.Y.B.Sc Roll No. 70 has successfully completed the practical of Paper – VI (sem – III)** **Physical Computing & IOT Programming during the Academic Year 2019-2020 as specified by the MUMBAI UNIVERSITY.**

**Prof. (Ms.) Pooja Kushwaha. Prof.(Mrs.)Anita Gaikwad.**

**Professor-In-charge In-charge**

**Dept. of Computer Science Dept. of Computer Science**

**Name:**  PRASHANT SINGH

**Class:** SYBSC(COMPUTER SCIENCE)

**Subject:** Physical Computing and IOT Programming

**Roll no:** 070

**Academic Year** : 2019-2020

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **INDEX** |  |  |
| **No** | **Date** | **TITLE** | **Page**  **No** | **SIGN** |
| 1 |  | Raspberry Pi hardware preparation & installation. |  |  |
| 2 |  | Linux commands exploring the raspbian. |  |  |
| 3 |  | Turning ON an LED with Raspberry pi using GPIO pins. |  |  |
| 4 |  | 8\*8 LED matrix interfacing with Raspberry Pi |  |  |
| 5 |  | Camera connection and capturing images. |  |  |
| 6 |  | Setting up a webserver using Raspberry Pi. |  |  |
| 7 |  | Node RED connect to internet of things. |  |  |
| 8 |  | Controlling stepper motor with RPi. |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |

**Practical 1**

**Aim:** Raspberry Pi hardware preparation & installation.

**Description:**

1. **Hardware Guide:-**

For getting started with Raspberry Pi for the first time you will required the following hardware..

* Rpi3 model B
* Monitor or TV
* HDMI cable
* Ethernet cable
* USB keyboard
* USB mouse
* Micro USB power supply
* 8GB or larger micro SD-card
* SD-card reader

1. **RPi3 model B:-**

The RPi3 is the third generation of Raspberry Pi. It replace the RPi2 model B in February 2016. Compare to RPi2 it has

* 1.2GHz 64 bits quad core ARM V8 CPU.
* Wireless LAN
* Bluetooth 4.1
* Like the RPi2, it also has
* 4 USB ports
* 40 GPIO pins
* HDMI PORTS
* Ethernet port
* Combine 3.5 mm audio jack & composite video
* Camera interface
* Display Interface
* Micro SD card slot
* Video core 3D graphics core.

1. **Monitor OR TV:-**

A Monitor or TV with HDMI 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 Raspberry Pi.

**4**) **HDMI cable:-**

Connect Raspberry Pi to monitor or TV with HDMI cable.  **5)Ethernet cable:-**

Ethernet cable will allow your Pi connect with the internet.

**6) USB Keyboard & Mouse**:-

Any standard USB keyboard & mouse can be used with Raspberry Pi. This plug and play device will work without any additional driver.

7) **Power Supply**:-

It is recomanded that you used 5V power supply for all models of Raspberry Pi.

**8) SD card:-**

The latest version of Raspbian the default operating system required an 8GB micro SD card. SD card will stored operating system as well as the files & applications created by you.

**9) Installation Guide:-**

1. Raspbian comes preinstalled with plenty of software for education, programming & general use.
2. To download Raspbian, log on to raspberrypi.org click on the download, then click on Raspbian & lastly download the RASPBIAN JESSIE WITH DESKTOP file. You can choose either the torrent file or zip file.
3. The downloaded file will be in zip format. To unzip the file ,you will require an unzip tool. You can use any unzipping the file, you will file a disc image file in the unzip folder.
4. Now format the SD card before writing the disk image file on the SD card. You can used SD format tool.
5. To write the image file of the operating system on the SD card. You will require a Disc Imager tool.
6. Once the image is return on the SD card you untitled SD card will now have the name boot. Your SD card will now hold the Raspbian opportunity system required for the first time set up.

**Practical 2**

**Aim:** Linux commands exploring the raspbian.

**Discription:**

1. **File system**

* ls : The ls command list the content of the current directory.
* cd : Using cd changes the current directory to the specified.
* pwd : The pwd command displays name of the present working directory.
* mkdir : You can use mkdir to create new directory.
* rmdir : You can use rmdir to remove empty directories.
* rm : It removes the specified file.
* cp : cp makes a copy of file and places it at the specified location.
* mv : mv command moves a file and placed it at the specified location.
* cat : You can use cat to list the content of file.
* head : It displays the beginning of file.we can used –n to specified no. of lines to show(by default 10)or with –c to specified the no of bytes.
* tail : It displays the end of file. we can used –n to specified no. of lines to show(by default 10)or with –c to specified the no of bytes.
* chown : This command changes the user and/or group that owns the file.
* chmod : It used to changed permission of file.chmod command can use symbols u(user that owns the file) , g(the files group) , and o(other users) and the permissions r(read) , w(write),x(executable).
* sudo : It unables you to run command as a super user or another user.
* unzip : This command extracts the file from compress zip file.
* tree : use the tree command to show a directory and all subdirectories and files indented as a tree structure.

1. **search**

* grep : Use grep to search inside files for certain search patterns.It supports regular expressions which allows special letter combinations to be included in the search.
* awk : awk is a programming language useful for searching and manipulating text files.
* find : The find command searches a directory and subdirectories for files matching certain patterns.

1. **Networking**

* ping : The ping utility is usually 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).It can specify the number of packets to send with the –c flag.
* hostname : The hostname command displays the current hostname of the system.A privileged 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 network configuration details for the interfaces on the current system when run without any arguments.(i.e. ifconfig).

**Practical 3**

**Aim**: Turning ON an LED with Raspberry pi using GPIO pins.

**Requirement**: Breadboard, an LED, 330 ohm resistor, two male-female jumper wires, Raspberry pi kit.

**Description:**

1) Breadboard

* The breadboard is a way of connecting electronic components to eachother. They are often used to test a circuit design before creating a printed circuit board(PCB).
* In this breadboard top row of holes are connected together, the same goes for bottom two rows of the breadboard.
* In the middle, the columns of wires are connected together.

1. LED

* LED stands for Light Emitting Diode, and glows when electricity is passed through it.
* When you pick up the LED, you will notice that one leg is longer than the another.
* The longer leg(known as Anode), is always connected to the +ve supply of the circuit known as ground connection.

3) Resistor

* You must always used resistors to connect LED’s upto the GPIO pins of the RPi.
* The RPi can only supply a small current(about 60mA).

The LED’s will want to draw more, and if allowed to they will burn-out of the RPi.

Therefore, putting the resistors in the circuit will ensure that only this small current will flow and the pi will not damage.

4) Jumper wires

* Jumper wires are used on breadboards to jump from one connection to another.
* The once you will be using in this circuit have different connectors on each end.

The end with the ‘Pin’ will go in to the breadboard. And the end with the piece of plastic with a hole in it will go on to the RPi GPIO pins.

5) The Raspberry Pi GPIO pins

* + GPIO stand for General Purpose Input O/P.
  + It is a way the RPi can control and monitor the outside world by being connected to electronic circuits.
  + The Pi is able to control LED’s turning them on and off, or motor’s or many other things.
  + It is also able to detect whether a switch has been pressed, or temperature, or Light.

6)Building the circuit:

* The circuit consist of a power supply, an LED that lights when power is applied and resistor to limit the current that can flow through the circuit.
* You will be using one of the ‘Ground’(GND) pins to act like ‘Negative’ or Zero volts ends of a battery.
* The +ve end of the battery will provided by a GPIO pin, here we will be using pin 15.
* When they are ‘taken high’ which means it O/P 3.3V , the LED will light.
* Use one of the jumper wires to connect a ground pin on the breadboard.
* The female and goes on the Pi’s pin and the male end goes in to the breadboard.
* Then connect the resistor from the same row on the Breadboard to a column on the breadboard.
* Insert the LED’s leg into the breadboard, with the long leg.
* Lastly complete the circuit by connecting pin 15 to the right hand leg of the LED.
* Write the python code and run the module.

**Practical 4**

**Aim:**8\*8 LED matrix interfacing with Raspberry Pi

**Hardware Requirement:**

8x8 LED matrix module,7291 driver board ,connecting wires, Raspberry Pi3 model.

**Description:**

1. **Introduction**:-

* LED matrix display device which contains light emitting diode aligned in the matrix form.
* This LED matrix displays are used in applications where symbol ,graphics ,characters , numbers are needed to be displayed together in static as well as in scrolling motion.
* LED matrix display is manufactured in various dimensions like 5x7, 8x8, 128x32 and 128x64 where the numbers represent LED’s in row and columns respectively.
* Also these displays comes in different colors such as red, green, yellow, blue, orange, white.
* In LED matrix display , multiple LED’s are wired together in rows and columns, to minimize the number of pins required to drive them.
* The matrix pattern is made either in row anode-columns cathode or row cathode-column anode pattern.

1. **7219 Driver board:**

* Before interfacing LED matrix with Raspberry Pi , we need to connect the max7219 IC which is an LED driver to the LED matrix display.
* The reason behind using this LED driver is that it drives 64 LED’s simultaneously which in turn reduces the number of wires so that the user will find it easy to connect to the display to the Raspberry Pi.
* The max7219 has a wire SPI interface:

1. Din-MOSI(Master Output Serial Input)
2. CS(Chip Select)
3. SCK(Clock)
4. GND(Ground)
5. VCC(5 Volt Power Supply)
6. **Software guide:**

(a) Before write a code we need to enable SPI(Serial Pepripheral Interface) and we need to install library for driving LED matrix moduleusing RPi.

(b) To enable the SPI use the following steps:

(i) Open terminal and type “sudo raspi-config” and press enter.

(ii) Use the down arrow to select interfacing option.

(iii) Arrow down to SPI.

(iv) Select “yes” when it ask about automatically loading the kernel module.

1. Use the right arrow to select finish button.
2. Select “yes”when it ask to reboot.

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

(a) Select Preferences from the raspberry pi application menu.

(b) From preferences select raspi-configuration.

(c) Now from this window navigate to interface option.

(d) Select the enabled radio button in front of SPI to enable it and click on “ok”.

(e) Finally do not forget to reboot your RPi after changing this settings.

1. **Installing the library**

Open the terminal window.Install the latest version of the library directly from python by typing the following commands:

* Sudo apt-get install python3\_dev python3-pip
* Sudo pip3 install max7219

1. **Source code**

import max7219.led as led

device=led.matrix()

device.show\_message(“hello world”)

1. **Wiring up your circuit**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Board pin | Name | Remark | RPi pin | RPi function |
| 1 | VCC | +5v power | 2 | 5v |
| 2 | GND | Ground | 6 | GND |
| 3 | DIN | Data in | 19 | GPIO 10(MOSI) |
| 4 | CS | Cheap select | 24 | GPIO 8(SPI) |
| 5 | CLK | Clock | 23 | GPIO 11(SPI CLK) |

* Connect the VCC pin of 7219 driver board to pin-2 of RPi.
* connect the GND pin of 7219 driver board to pin6 of RPi.
* connect the DIN pin of 7219 driver board to pin-19 of RPi.
* connect the cs pin of 7219 driver board to pin-24 of RPi.
* connect the CLK pin of 7219 driver board to pin-23 of RPi.

1. **Conclusion**

After ensuring that the connections are done properly,power on your RPi.now open python3 and run the code.so now you have learned how to interface 8X8 matrix led module with your RPi

1. **8\*8 LED Matrix**

* There are 8+8=16 common terminals in the LED matrix module.
* Overthem we have 8 common positive terminals and 8 common negative terminals , in the form of 8 rows and 8 columns, for connecting 64 LED in matrix form.
* For 8 rows , we have 8 common positive terminals(9,14,8,12,1,7,2,5)
* Similar to we have 8 common negative terminals as columns(13,3,4,10,6,11,15,16).

For grounding any LED in any column the respective common negative terminal to be grounded.

* The connections which are done between RPi and LED matrix module are shown in below table.

|  |  |  |
| --- | --- | --- |
| **LED matrix module pin no.** | **Function** | **RPi GPIO Pin no.** |
| **9** | **Positive zero** | **GPIO 21** |
| **14** | **Positive 1** | **GPIO 20** |
| **8** | **Positive 2** | **GPIO 26** |
| **12** | **Positive 3** | **GPIO 16** |
| **1** | **Positive 4** | **GPIO 19** |
| **7** | **Positive 5** | **GPIO 13** |
| **2** | **Positive 6** | **GPIO 6** |
| **5** | **Positive 7** | **GPIO 5** |
| **13** | **Negative 13** | **GPIO 12** |
| **3** | **Negative 3** | **GPIO 22** |
| **4** | **Negative 4** | **GPIO 27** |
| **10** | **Negative 10** | **GPIO 25** |
| **6** | **Negative 6** | **GPIO 17** |
| **11** | **Negative 11** | **GPIO 24** |
| **15** | **Negative 15** | **GPIO 23** |
| **16** | **Negative 16** | **GPIO 18** |

**Practical 5**

**Aim:**  Camera connection and capturing images.

**Description:**

The camera module is a great accessory for the RPi,allowing users to take still pictures and record video in full HD.

**1) Hardware guide**

You will require the camera module along with your initial RPi set up.

* Camera module:

The RPi camera board plugs directly into the connector on the RPi.The camera is supported in the latest version of Rasbian,the RPi preffered os.

The RPi camera board features:

1. fully compatible with both model A and model B RPi.
2. 5 MP (5647 camera module)
3. 3) still picture resolution- 2592X1944
4. 4)video support 1080P and 640X480P
5. weight-3g
6. fully compatible with many RPi cases.

Connect the camera module

step1: first of all switch of RPi.

step2:locate the camera port and connect the camera.

step3:start up the pi

step4:open the “RPi configuration” tool from the main menu.

step5:ensure the camera s/w is enabled.if it’s not unable , enabled it and report your pi to begin.

**2) Software guide**

You can capture an image by just typing a single line command.open terminal windows and type the command as follows:

**$sudo raspistill –o /home/pi/Desktop/image.jpg**

This command will capture an image and store it at the specified location(Here the location is /home/pi/Desktop)with this specified name(image.jpg).You can even write a code in python to capture an image usin RPi camera.open python3,create a new file and type a code as follows:

**Source 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()**

**Practical 6**

**Aim:** Setting up a webserver using Raspberry Pi.

**Description:**

You can use a webserver on a Raspberry Pi to host a full website, or just use it to display some information you wish to share to other machines on your network.

Here we will be using Apache Webserver.

1. Hardware Guide:-

You will require the basic RPi setup, you need to connect your RPi with internet.

1. Installation Guide:-

Apache is a popular webserver application you can install on the RPi to allow it to server webpages.

* Install Apache

First install the Apache 2 package by typing the following command in to the Terminal:

sudo apt-get install apache2 –y

* Test the Webserver

Bydefault , apache puts a test HTML file in the web folder.

This default webpage is serve when you browse to <http://localhost/> on the Pi itself , or [http://XX.XX.XX.XX](http://xx.xx.xx.xx) (whatever the Pi’s IP address is) from another computer on the network.

To find the Pi’s IP address, type hostname –I at the command line.

* Storing the default webpage

This default webpage is just a html file on the file system.It is located at, /var/www/html/Index.html

**Practical 7**

**Aim:** Node RED connect to internet of things.

**Description:-**

1. **Hardware Guide:**

Along with the Raspberry Pi set up you will require the following component get started with the node RED as follows:

* LED
* Register
* Connecting Wires
* Bread Board

1. **Wiring of the circuit**:

* 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.
* Wire-up an LED to GPIO17(i.e. physical pin 11) on your Raspberry Pi.
* Connect the one end of register to physical pin 11.
* Connect the negative end of LED to ground of Raspberry Pi.

1. **Software Guide**:

* Start up your Raspberry Pi click on the Raspberry icon, then the programming menu to open node RED.
* You should see a window displaying information about node RED starting up.
* Now go to the internet menu & open chromium web browser.
* In Chromium, locate the address bar at the top & type in localhost:1880 then press Enter. This will display the Node-RED interface.

1. **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 & they can all run at the same time. For this guide, we will only need one flow.
* The coloured bocks on the left side of the interface are the nodes. Scroll right down to the bottom of the list & you will see some nodes labelled Raspberry Pi. The first one in the list, with the raspberry icon on the left, is for 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 would be an example of an output. Drag an output node onto the blank page in the middle.
* Double-click on the node & a box will appear to let your configuration the node. Change the GPIO pin to be GPIO17 & tick initialise pin state? Leave the setting for initial level of pin on low. Give the node a name -we called it Green LED because the LED we used was green, but if yours is a different colour feel free to change the name. when you are finished, click Done.
* Now scroll back up to the list of nodes. To turn LED on & off, we need an input. In Node RED we can inject message into the flow & caused things to happened as a result. Drag an inject node onto the flow.
* Double-click on the inject node. Use the drop down next to played to change the data type to string & type 1 in the payload box -this will be the message. Type on in the name box. Press Done.
* Repeat the previous steps to create another inject node, except this time add 0 as the payload message, &call this node Off.
* Now look for the grey dot on the right side of the inject nodes. Click & drag from the grey dot on the on node to grey dot on your LED node to join them up, Repeat for the off node, also joining it to the LED node.
* 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 like pressing the green flag in scratch or F5 to run your code in python.
* 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 & your LED should light up. You should able to turn the LED off by clicking the blue square on the off node, which injects the message 0.

**Practical 8**

**Aim :**  Controlling stepper motor with RPi.

**Description:**

1. **Hardware Guide:**

You will require the following things along with your initial RPi setup.

1. Stepper motor
2. Stepper motor driver
3. connecting wires
4. external 5v power supply
5. Bread board
6. Stepper motor:

It is good for small projects and experimenting with steppers.this unipolor motor has a build in mounting plate with two mounting holes.

Technical details:

* unipolor stepper with 0.1 inch spaced 5 pin cable connector.
* 513 steps per revolution.
* 5V DC.
* Weight 37 grams.
* Dimension 28mm Diameter,20mm tall not including 9mm shaft with 5mm diameter.
* 23cm long cable.

One side of board side has a 5 wire socket where the cable from the stepper motor hooks up and 4 LED’s to indicate which coil is currently powered.

The motor cable only goes in one way,which always helps.on the side,you have a motor on/off jumper.

The two pins below the 4 registers is where you provide power to the stepper.A separate 5-12V,1A power supply or battery pack should be use,as the motor may drain more current than the microcontroller can handle and could potentially damaged it.

In the middle of the board we have ULN2003 chip.at the bottom are the 4 control inputs that should be connected to 4 GPIO pins of RPi.

1. **Wiring up your circuit :**

Wire your circuit as follows:

* Connect the input 1 pin.i.e. IN 1 of driver board to physical pin of RPi.
* Connect the IN2 pin of driver board to physical pin33 of RPi.
* Connect the I/P three pin of a driver board to physical pin of 35 of RPi.
* Connect the IN4 of pin of driver board to physical pin 37 of RPi.
* Connect the positive pin of driver board to external 5V power supply.
* Connect the –ve ground pin of driver board,the ground pin of driver supply and the ground pin of RPi(physical pin 6)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.

1. **S/W guide**

Open python3,navigate to file & create new file,write the code as follows:

**source code:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.Board)

stepPin1=31

stepPin2=33

stepPin3=35

stepPin4=37

GPIO.setup(stepPin1,GPIO.OUT)

GPIO.setup(stepPin2GPIO.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);

GPIO.output(stepPin1,stepVal1)

GPIO.output(stepPin2,stepVal2)

GPIO.output(stepPin3,stepVal3)

GPIO.output(stepPin4,stepVal4)

def clockWiseRotate(delay,steps1):

for 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:

delay=input(“enter delay between steps milliseconds):”)

steps1=input(“how many steps clockwise?”)

steps2=input(“how many steps anticlockwise?”)

clockWiseRotate(int(delay)/1000,int(steps2))

anticlockWiseRotate(int(delay)/1000,int(steps2))

finally:

GPIO.cleanup()