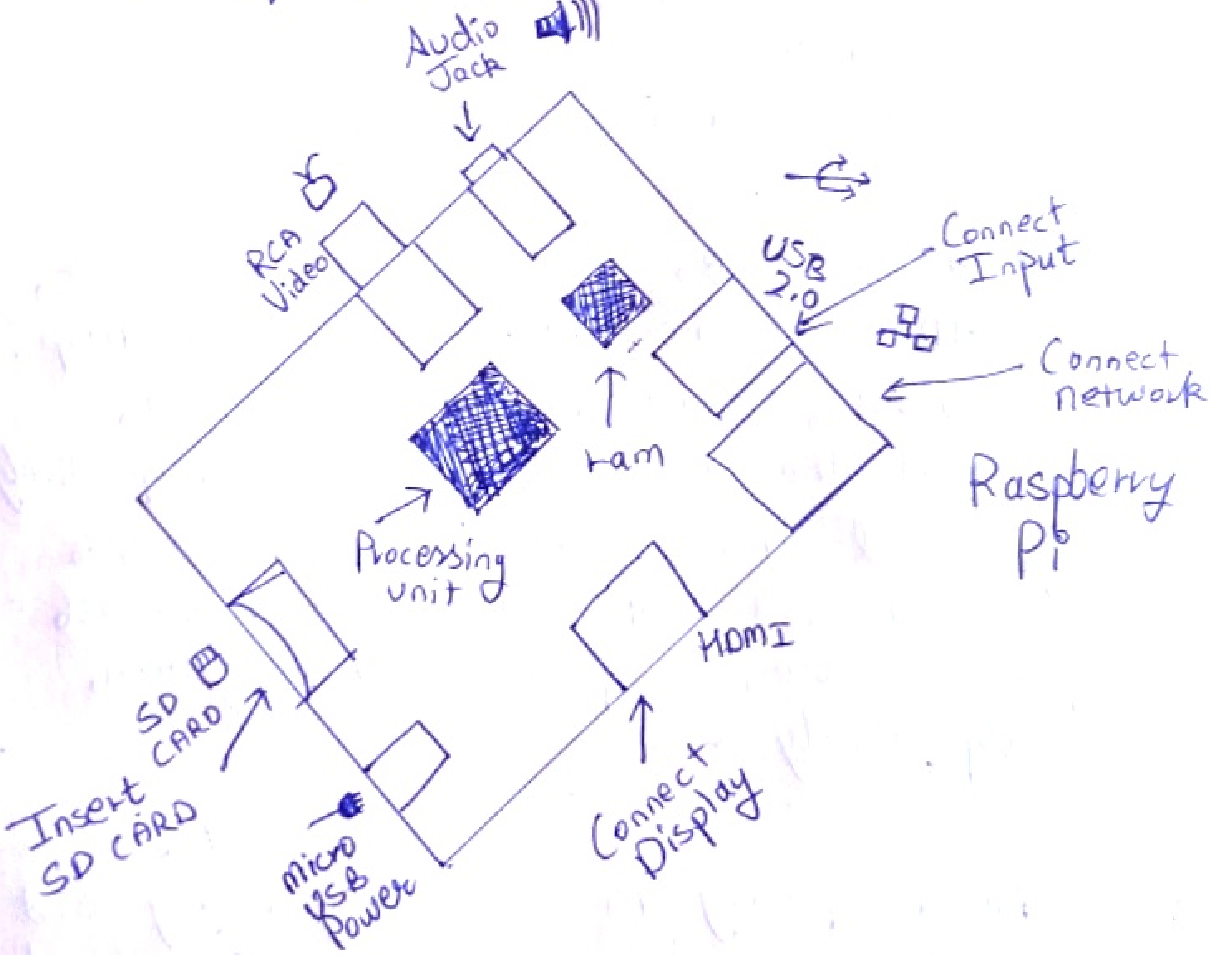
Practicals - Internet of things (IOT).
Roll No :- 8,
T.Y.B.Sc.IT

Practicals - 0.

◎ → Starting Raspbian OS, Familiarising with ~~Rasb~~ Raspberry Pi Components and interface. Connecting to ethernet, Monitor, U.S.B.



* Rasberry Pi Components.

Starting Raspbian OS.

1. Download the ~~Rasb~~ Raspberry Pi operating system.
2. Unzip the file that you just downloaded.
3. Download BalenaEtcher.
4. Burn the O·S on the flash drive.
5. Live Disk has been created.
6. Reboot, the PC, boot into bootmenu.
7. Select the flash drive.
8. Press Enter.
9. Raspbian O·S has been started.

- Using RJ-45 Jack, connect to the internet,
- Using HDMI connect the PC to the Desktop.
- Using U·S·B, connect to the U·S·B 2·0, and start the keyboard and mouse, respectively.
- Before all this, insert the S·D card into the S·D card slot and connect micro usb power cable, to power on the system, respectively.

Now the Raspberry Pi, is fully ready and developed to be tinkered with and explored, respectively.

Practicals - 1

⊙ Displaying different LED patterns with Raspberry Pi.

Requirements :-

- A Raspberry Pi with Raspbian already installed.
- You will also need to be able to access the Pi using a Monitor, Mouse, and Keyboard.
- R, G, B LEDs.
- A Solderless Prototyping BreadBoard.
- 4 x 330 ohm Resistors
- Some Male to Female

- Step 2 → Build your Circuit.
- Step 3 → Create a Script to Control and Test the LEDs.

The Program as Such been :-

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.Bcm)
GPIO.setup(17,GPIO.OUT)
GPIO.setup(18,GPIO.OUT)
GPIO.setup(22,GPIO.out)
GPIO.setup(23,GPIO.OUT)
GPIO.output(17,true)
time.sleep(3)
GPIO.output(17,False)
time.sleep(1)
GPIO.output(18,False)
time.sleep(1)
```

```
GPIO. output (22, False)
 time. sleep(1)
GPIO. output (23, True)
 time. sleep(3)
GPIO. output (23, False).
```

Step 4:- Adding Flexibility by Using Parameters and Conditional Statements.

```
import RPi.GPIO as GPIO.
import time.
from sys .import argv
whichled = argv[1]
ledaction = argv[2]
 LEDa = 17
 LEDb = 18
 LEDc = 22
 LEDd = 23
GPIO. setmode (GPIO. BCm)
GPIO. setup (LEDa, GPIO.out)
GPIO .cetmode (GPIO . BCm)
GPIO. setup (LEDb, GPIO.out)
GPIO. Setmode (GPIO . BCm)
GPIO. setup ( LEDc , GPIO. OUT)
GPIO. Setmode (GPIO. BCm)
GPIO. setup (LEDd , GPIO. out)
GPIO .
  if ledaction = “off” ;
     if whichled == “a”;
          GPIO. output (LEDa, False)
     if whichled = “b”;
          GPIO output (LEDb, False)
```

```
If whichled == "c":
     GPIO.output(LEDc, False)
 if whichled == "d"
```

⊙ The task has been completed successfully.

⊙ The required task has been configured successfully, respectfully.

Practicals - 2

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

⊙ Here are the general steps to display time over a 4 digit 7-segment display using a Raspberry Pi:

→ Choose a 4 digit 7-segment display that is compatible with the Raspberry Pi. There are many options available online, so choose one that suits your needs and budget.

→ Connect the display to Raspberry Pi according to the pinout diagram provided by the manufacturer.

→ Write a python script that will read the current time. from the Raspberry Pi's system clock and convert it to a format that can be displayed on the 7-segment display.

→ There are several libraries available for this such as RPi.GPIO and Adafruit's Python GPIO library.

→ In the script, use the GPIO library to control the pins of the 7-segment display. turning on the appropriate segments to display the desired digits.

→ Run the script on the Raspberry Pi and test the display to ensure that it is functioning as expected.

→ You may also want to add additional features, to your script, such as the ability to set alarms or change the display format, depending on your specific requirements.

Note:- It is important to be familiar with the GPIO pins on the Raspberry Pi and to exercise caution when connecting the 7- segment display to prevent damage to the hardware.

Practicals 3 :-

⊙ Raspberry Pi based Oscilloscope .

Oscilloscope :-

A device for viewing oscillations by a display on the screen , of a cathode ray tube, or similar .

Procedure :

⊙ To build a Raspberry Pi based oscilloscope, you will need the following components :-

→ Raspberry Pi board.

→ ADC (Analog to Digital Converter)

→ Signal generator or function generator.

→ Software to display the signal on the screen.

So , gather the necessary hardware Components :

→ Raspberry Pi,

→ high-speed ADC,

→ breadboard,

→ Jumper wires,

→ resistors ,

→ Capacitors,

→ and a USB oscilloscope probe .

→ Done ;

2 → Connect the ADC to the Raspberry Pi via the breadboard and jumperwires, making sure to connect the power, ground, and data pins correctly.

3 → Install the necessary software packages and libraries on the Raspberry Pi to communicate with the ADC and display the data on the screen.

4 → Connect the USB oscilloscope probe to the input of the circuit or component that you want to measure.

5 → Run the oscilloscope software on the Raspberry Pi, configure the sampling rate and voltage range, and start capturing data.

6 → Display the captured data on the screen as the two-dimensional graph, with voltage on the Y-axis and the time on the X-axis.

7 → Analyse the data, Zoom in or out of the waveform, and adjust the trigger level as needed.

· Builiding a Raspberry Pi-based oscilloscope can be fun and educational project for electronics enthusiasts and hobyists, but it require some technical knowledge and experence with electronics and programming.

Practical 4

[•] Controlling Raspberry Pi with WhatsApp?

Here,

Yes, it is possible to control a Raspberry Pi with Whatsapp using a third-party API such as Twillio or Whatsapp Business API.

→ Here are the High-level steps you need to follow.

→ Create a Twillio Account or Whatsapp business ~~account~~ API account.

→ Setup a webhook URL that will receive incoming messages from Whatsapp or Twillio.

→ Write a Python script that - listens to incoming messages and sends Commands to the Raspberry Pi GPIO pins or other connected devices.

→ Deploy the python script on the Raspberry Pi and set it to run as as a service on Startup.

→ Link your Twillio ~~an~~ or Whatsapp Business API account to your phone number and configure it to send messages to the webhook URL.

→ Keep in mind that using third-party API's may have limitations or cost associated with them. Also you need to ensure proper security, privacy and integrity.

Practical 5

[•] Setting up wireless Access Point using Raspberry Pi

→ Here are the steps to set up a wireless access point using Raspberry Pi :-

1 → Update and upgrade your Raspberry Pi:

```
'sudo . apt -get update && sudo apt-get upgrade.'
```

2 → Install the required software:

~~Arduino :-~~

```
sudo apt -get install hostapd isc-dhcp-server
sudo opt -get install dnsmasq
```

3 → Configure a static IP address for the wireless interface by editing the '/etc/dhcpcd.conf' file:

```
interface wlan0
static ip-address = 192.168.4.1/24
nohook wpa- supplicant.
```

4 → Configure the DHCP server by editing the '/etc/dhcpd.conf' file :-

```
subnet 192.168.4.0 netmask 255.255.255.0
 { range 192.168.4.10 192.168.4.250;
 option broadcast - address 192.168.4.255;
 option routers 192.168.4.1;
  default -lease- time 600;
 max - lease - time 7200;
 option domain - name "local";
 option domain - name - servers 8.8.8.8, 8.8.4.4;

 }
```

5. Configure the access point by editing the '/etc/hostapad/hostapad.conf' file

```
interface =wlan0
ssid = MyAccessPoint
hw_mode =g
channel =7
macaddr_acl =0
auth_algs =1
ignore_broadcast_ssid=0
wpa =2
wpa_passphrase = MyPassword.
wpa_key_mgmt =WPA -PSK
wpa_pairwise =TKIP
wr
rsn_pairwise = CCMP.
```

6. Configure the 'hostapd' daemon by editing the '/etc/default/hostpd' file:

7. Configure 'dnsmasq' by editing the '/etc/dnsmasq.conf' file:

8. Enable packet forwarding by editing the '/etc/sysctl.conf' file:

9. Set up IP masquerading by entering the following commands:

10. Save the IP tables configuration.

11. Configure the system to read the 'iptables.ipv4.nat' file at boot:

12. Reboot the Raspberry Pi: