RAMAIAH
College of Arts, Science
& Commerce - Autonomous

IOT LAB MANUAL

(for BSc VI SEM NEP Syllabus)

2025-26

PART B

Prepared by

Rithu R

Assistant Professor

Electronics Department

MSRCASC

RAMAIAH
College of Arts, Science
& Commerce - Autonomous

DEPARTMENT OF ELECTRONICS

LAB RULES

- **Students are not allowed to carry bags inside the lab.**
- **Loitering and throwing litter in the lab is strictly prohibited.**
- **Usage of mobile phones within the lab is restricted.**
- **The required lab equipment and components are to be used only with lab assistant supervision.**
- **The students should maintain the lab manuals and up-to-date instructions given by the lab in-charge.**
- **Use the lab equipment only after instructions given by the lab in-charge.**
- **While leaving the lab, return the electronic components used in the experiments and arrange the chairs.**
- **A student must take care of lab equipment and is expected to protect it from getting damaged/vandalized.**
- **Do not change the settings of equipment without lab in-charge permission.**
- **Students should maintain lab observation during the lab hours.**
- **Fans and lights should be switched off after use.**

RAMAIAH
College of Arts, Science
& Commerce - Autonomous

Sl.no	Date	Experiment Name	Marks	Remarks	Signature of faculty
1.					
2.					
3.					
4.					
5.					

IOT Lab (Part B)Experiments

1. **Starting Raspbian OS, Familiarization with Raspberry Pi components and interface, Connecting to Ethernet, Monitor, USB.**
2. **Displaying different LED patterns with Raspberry Pi.**
3. **Visitor Monitoring with Raspberry Pi and Pi Camera**
4. **Displaying Time over 4-Digit 7-Segment display using Raspberry Pi.**
5. **Setting up wireless Access point using Raspberry Pi.**

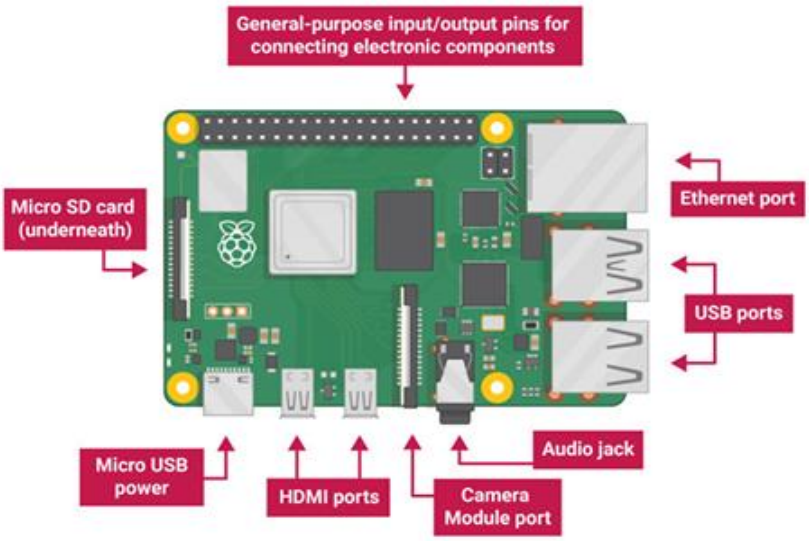
Experiment No: 1

Starting Raspbian OS, Familiarization with Raspberry Pi components and interface, Connecting to Ethernet, Monitor, USB.

Aim-:

To familiarize with Raspberry Pi components and interface, connecting to Ethernet, Monitor, USB and to learn about the installation steps of Raspian OS.

Raspberry Pi Components-:



- USB ports — these are used to connect a mouse and keyboard. You can also connect other components, such as a USB drive.
- SD card slot — you can slot the SD card in here. This is where the operating system software and your files are stored.
- Ethernet port — this is used to connect Raspberry Pi to a network with a cable. Raspberry Pi can also connect to a network via wireless LAN.
- Audio jack — you can connect headphones or speakers here.

- HDMI port — this is where you connect the monitor (or projector) that you are using to display the output from the Raspberry Pi. If your monitor has speakers, you can also use them to hear sound.
- Micro USB power connector — this is where you connect a power supply. You should always do this last, after you have connected all your other components.
- GPIO ports — these allow you to connect electronic components such as LEDs and buttons to Raspberry Pi.

Set up your SD card

If you have an SD card that doesn't have the Raspberry Pi OS operating system on it yet, or if you want to reset your Raspberry Pi, you can easily install Raspberry Pi OS yourself. To do so, you need a computer that has an SD card port — most laptop and desktop computers have one.

The Raspberry Pi OS operating system via the Raspberry Pi Imager

Using the Raspberry Pi Imager is the easiest way to install Raspberry Pi OS on your SD card.

Note: More advanced users looking to install a particular operating system should use this guide to installing operating system images.

Download and launch the Raspberry Pi Imager

RAMAIAH
College of Arts, Science
& Commerce - Autonomous

Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

Download for Windows

Download for macOS

Download for Ubuntu for x86

To install on **Raspberry Pi OS**, type
`sudo apt install rpi-imager`

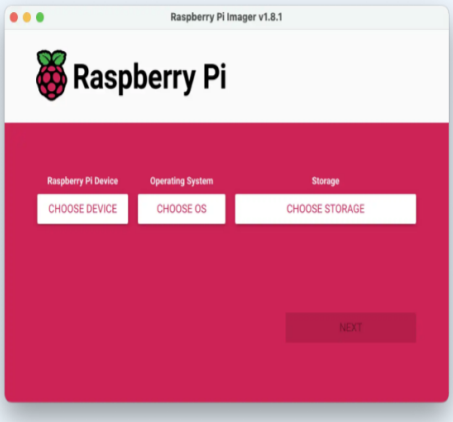
Windows Security

Windows Security

Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.



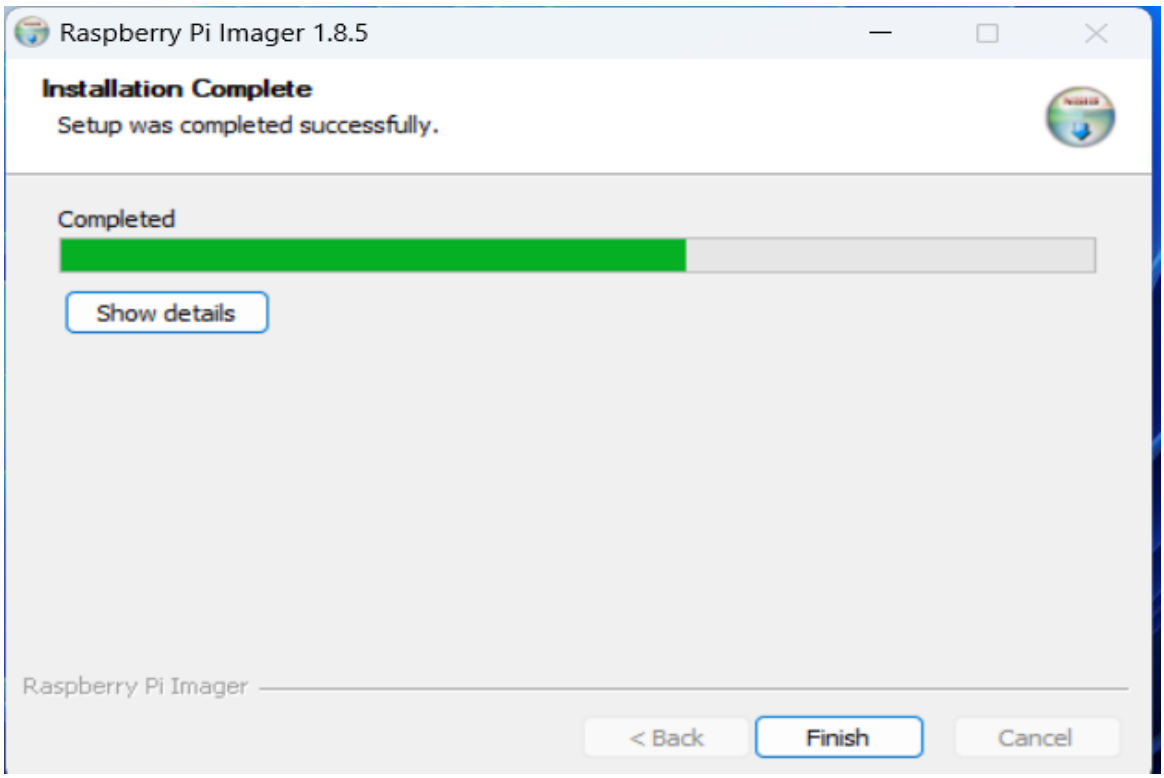
Download for Windows

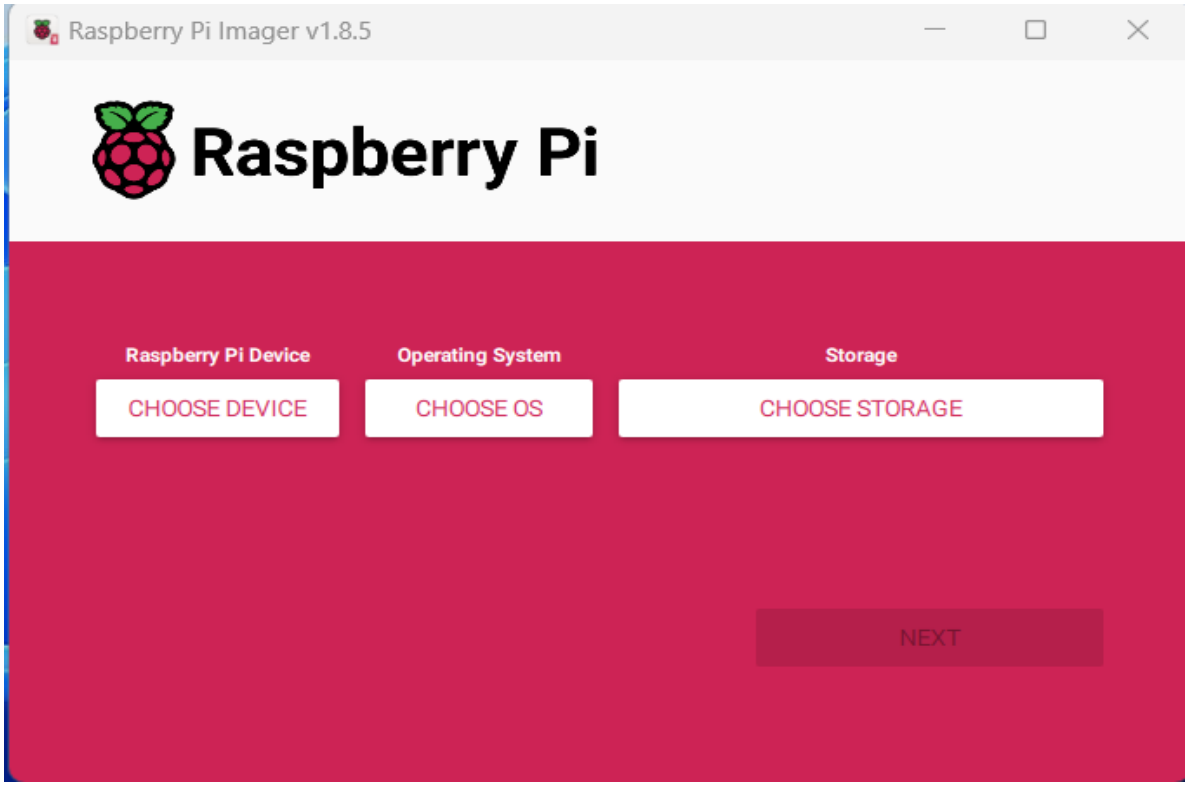
Download for macOS

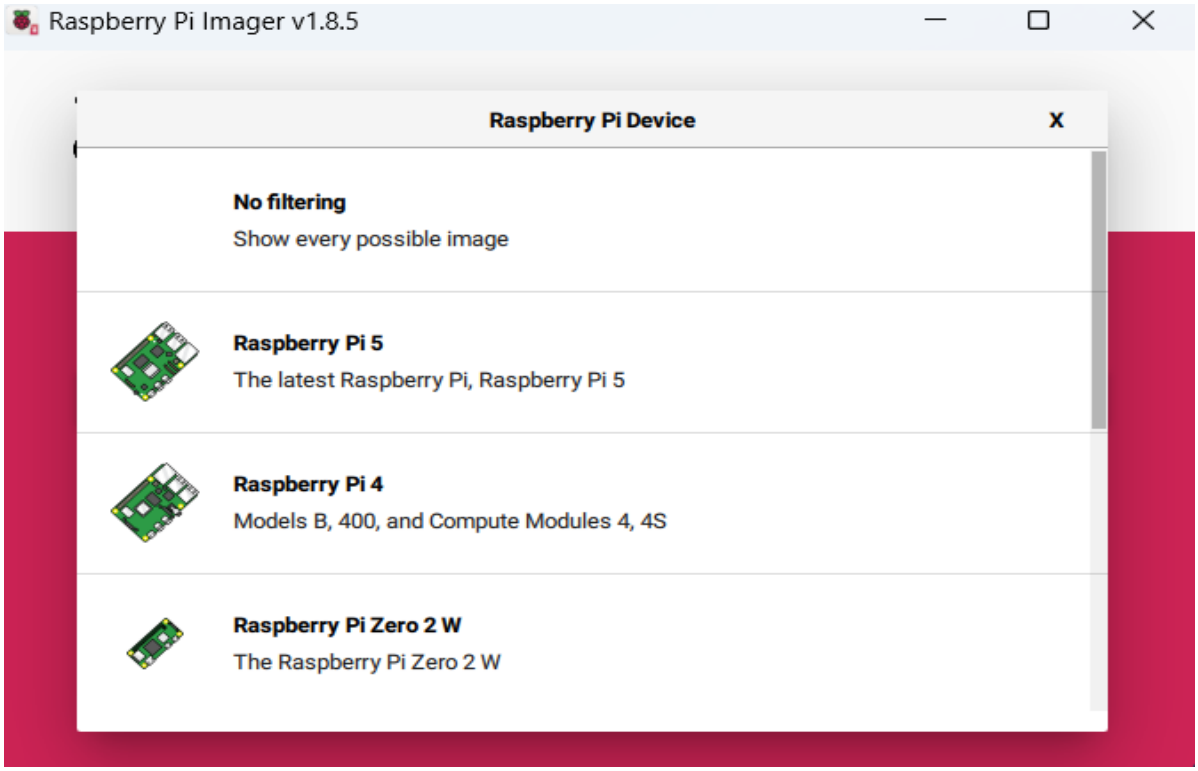
Download for Ubuntu for x86

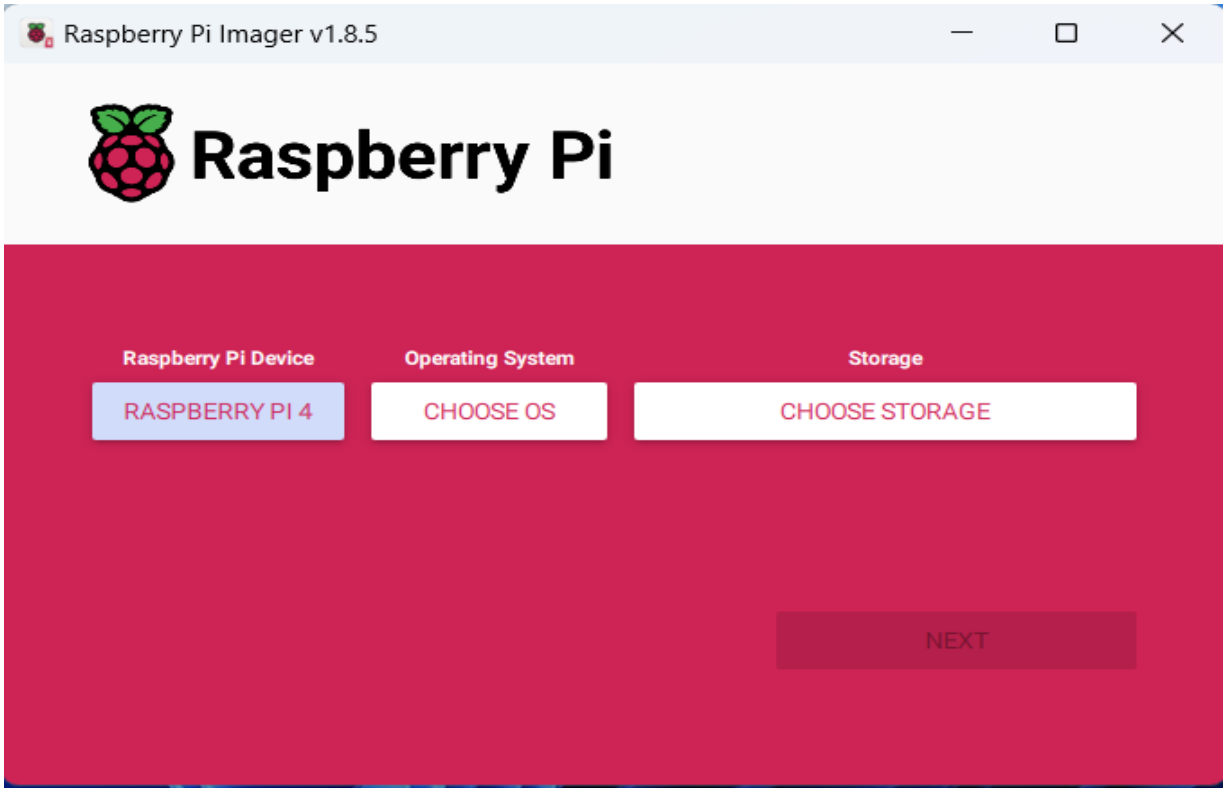
To install on **Raspberry Pi OS**, type
`sudo apt install rpi-imager`
in a Terminal window.

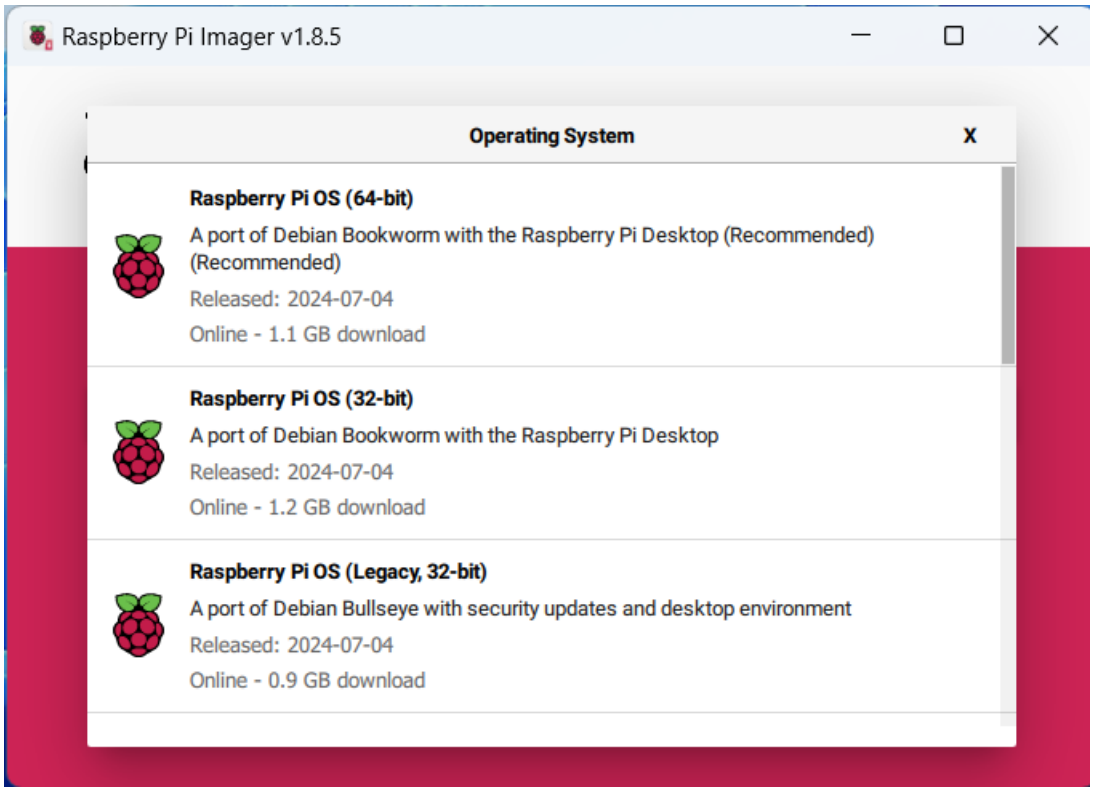

Raspberry Pi Imager 1.8.5
Welcome to Raspberry Pi Imager Setup
Setup will guide you through the installation of Raspberry Pi Imager.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Install to start the installation.
Install
Cancel

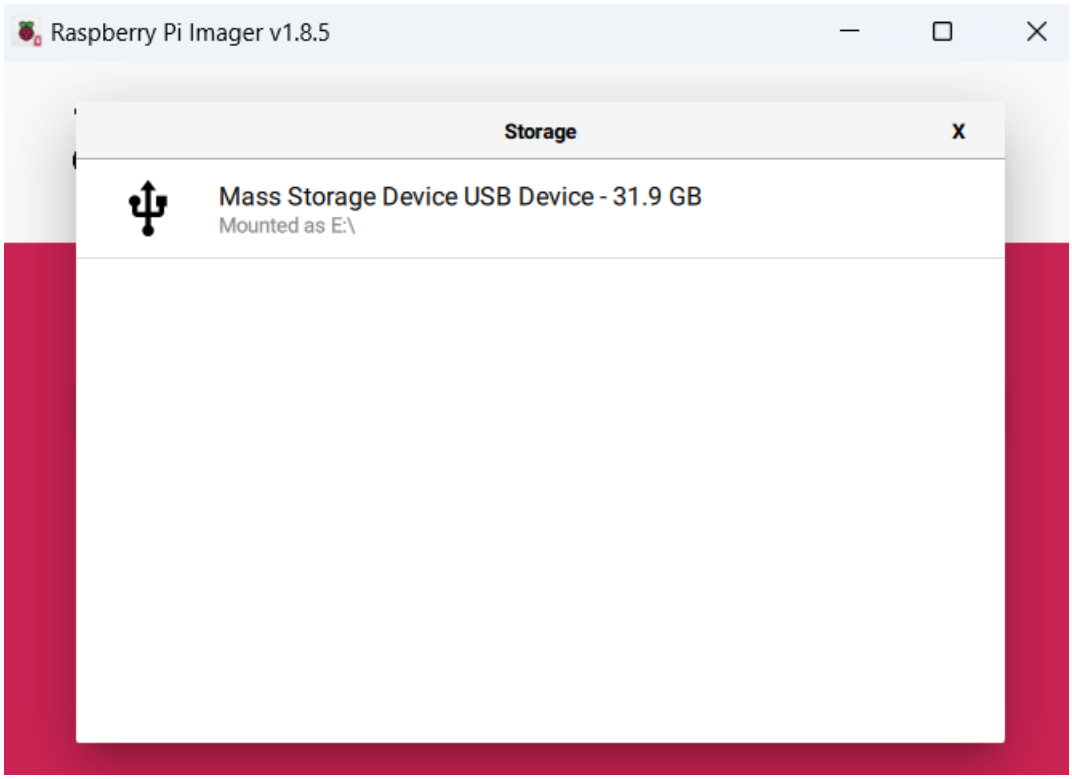

Raspberry Pi Imager 1.8.5
Installation Complete
Setup was completed successfully.
Completed
Show details
Raspberry Pi Imager
< Back
Finish
Cancel

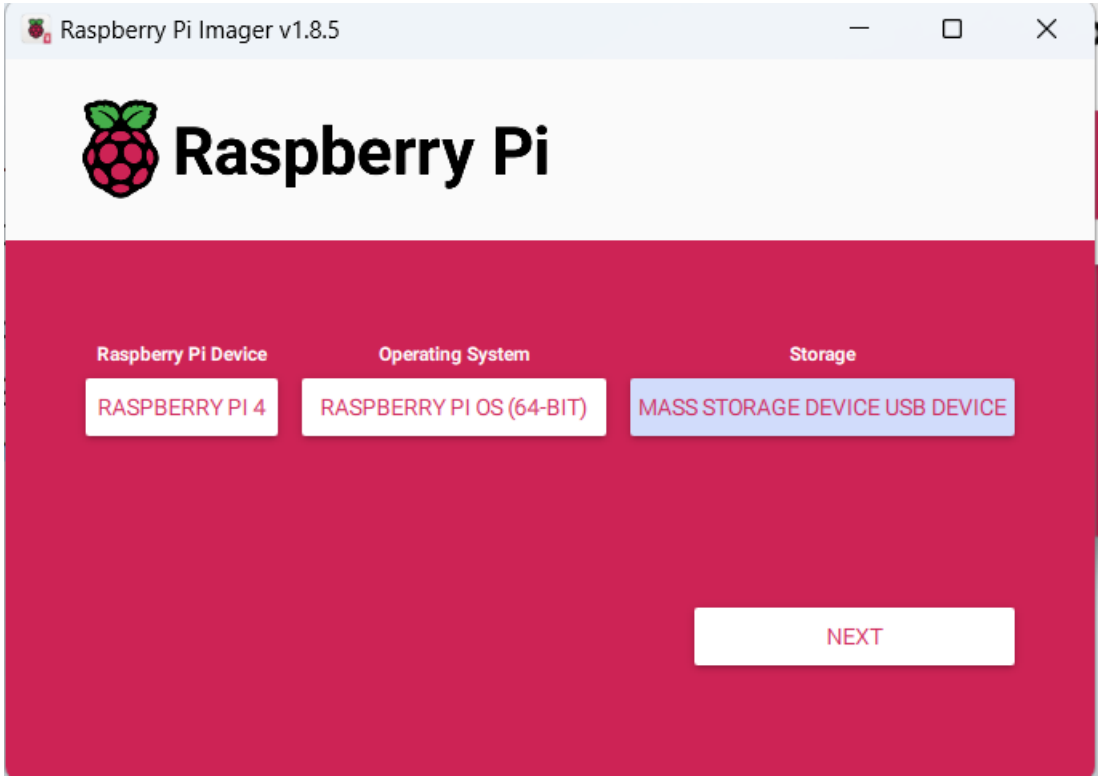

Raspberry Pi Imager v1.8.5
Raspberry Pi
Raspberry Pi Device
Operating System
Storage
CHOOSE DEVICE
CHOOSE OS
CHOOSE STORAGE
NEXT

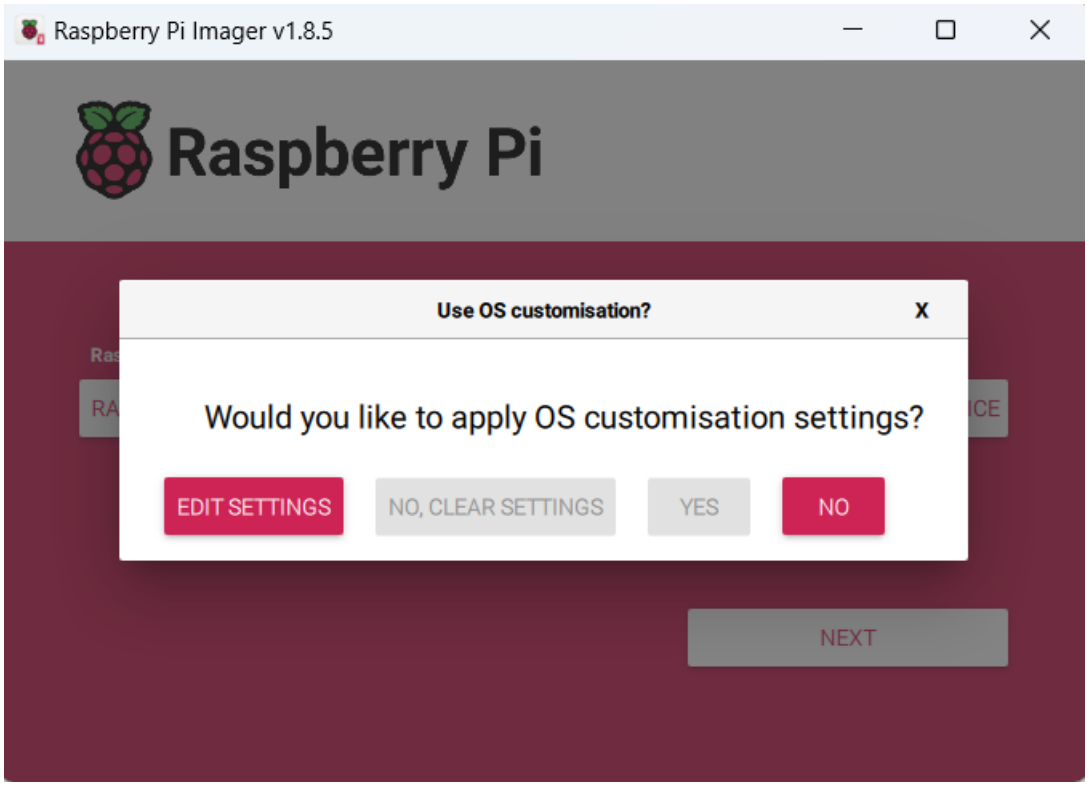

Raspberry Pi Imager v1.8.5
Raspberry Pi Device
X
No filtering
Show every possible image
Raspberry Pi 5
The latest Raspberry Pi, Raspberry Pi 5
Raspberry Pi 4
Models B, 400, and Compute Modules 4, 4S
Raspberry Pi Zero 2 W
The Raspberry Pi Zero 2 W

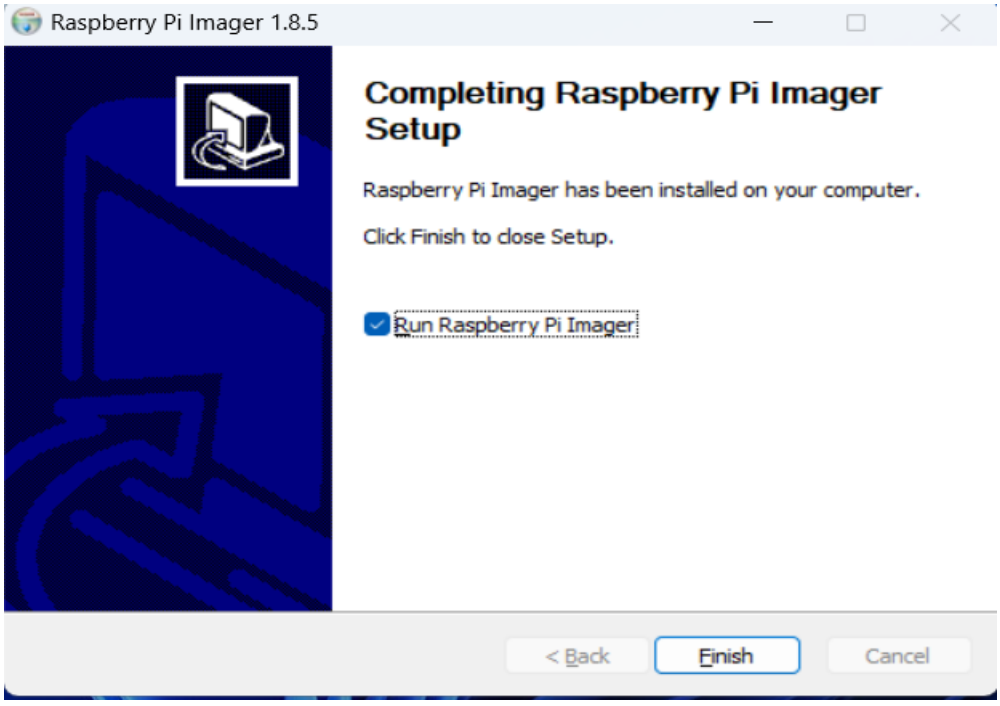

Raspberry Pi Imager v1.8.5
Raspberry Pi
Raspberry Pi Device
Operating System
Storage
RASPBERRY PI 4
CHOOSE OS
CHOOSE STORAGE
NEXT


Raspberry Pi Imager v1.8.5
Operating System
X
Raspberry Pi OS (64-bit)
A port of Debian Bookworm with the Raspberry Pi Desktop (Recommended) (Recommended)
Released: 2024-07-04
Online - 1.1 GB download
Raspberry Pi OS (32-bit)
A port of Debian Bookworm with the Raspberry Pi Desktop
Released: 2024-07-04
Online - 1.2 GB download
Raspberry Pi OS (Legacy, 32-bit)
A port of Debian Bullseye with security updates and desktop environment
Released: 2024-07-04
Online - 0.9 GB download

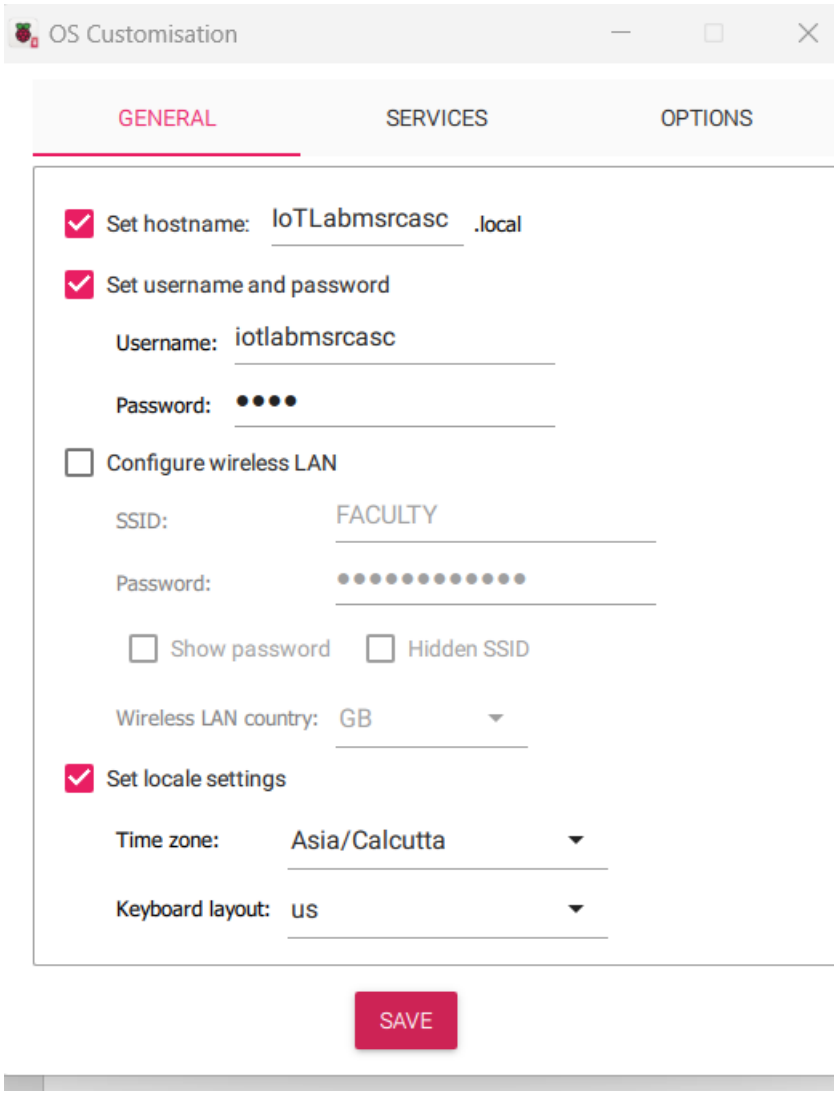

Raspberry Pi Imager v1.8.5
Storage
X
Mass Storage Device USB Device - 31.9 GB
Mounted as E:\


Raspberry Pi Imager v1.8.5
Raspberry Pi
Raspberry Pi Device
Operating System
Storage
RASPBERRY PI 4
RASPBERRY PI OS (64-BIT)
MASS STORAGE DEVICE USB DEVICE
NEXT

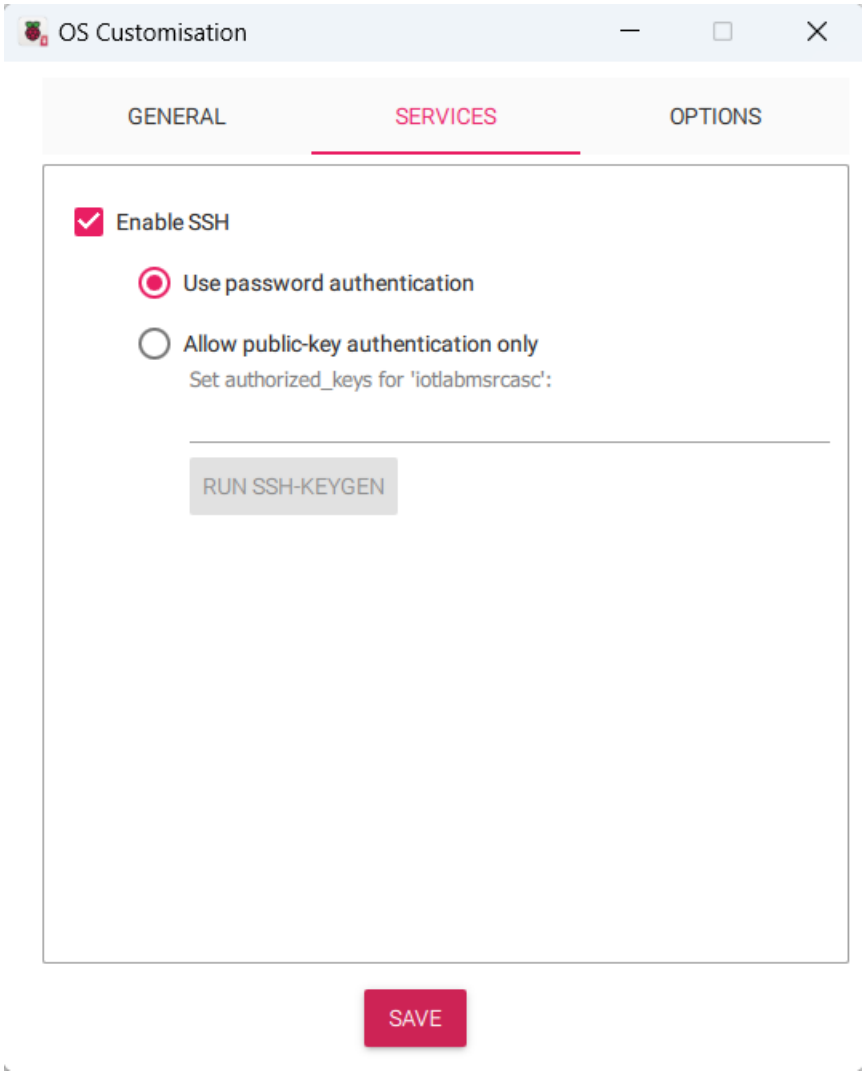

Raspberry Pi Imager v1.8.5
Raspberry Pi
Use OS customisation?
X
Would you like to apply OS customisation settings?
EDIT SETTINGS
NO, CLEAR SETTINGS
YES
NO
NEXT

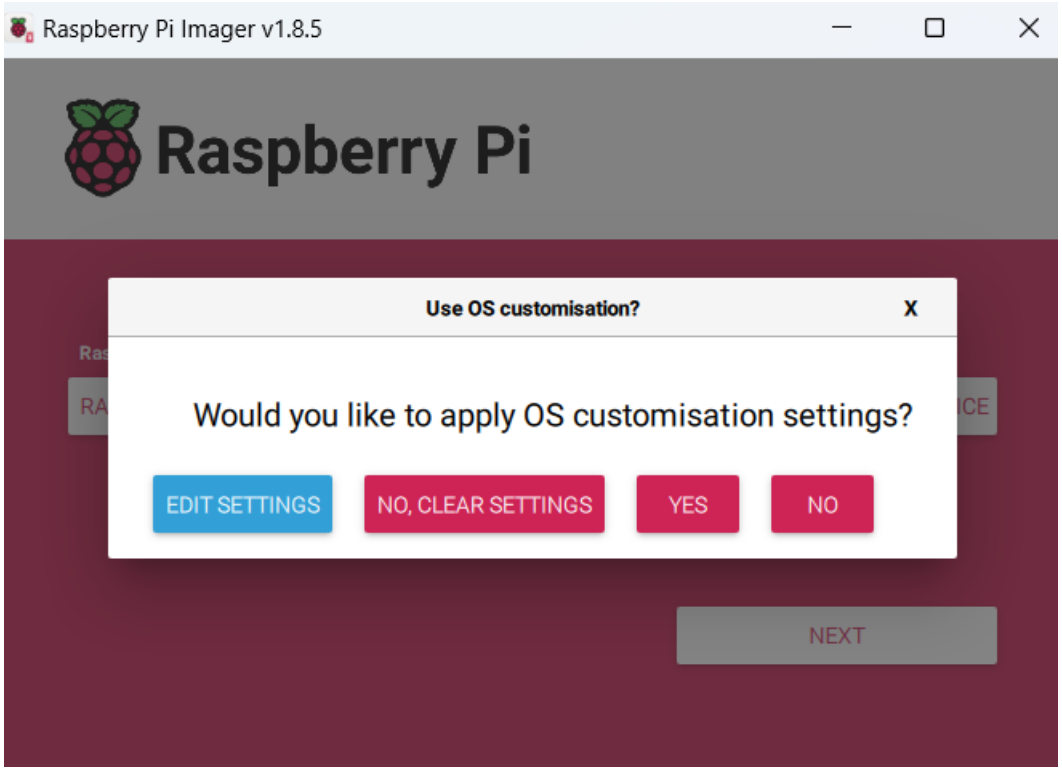

Raspberry Pi Imager 1.8.5
Completing Raspberry Pi Imager Setup
Raspberry Pi Imager has been installed on your computer.
Click Finish to close Setup.
Run Raspberry Pi Imager
< Back
Finish
Cancel

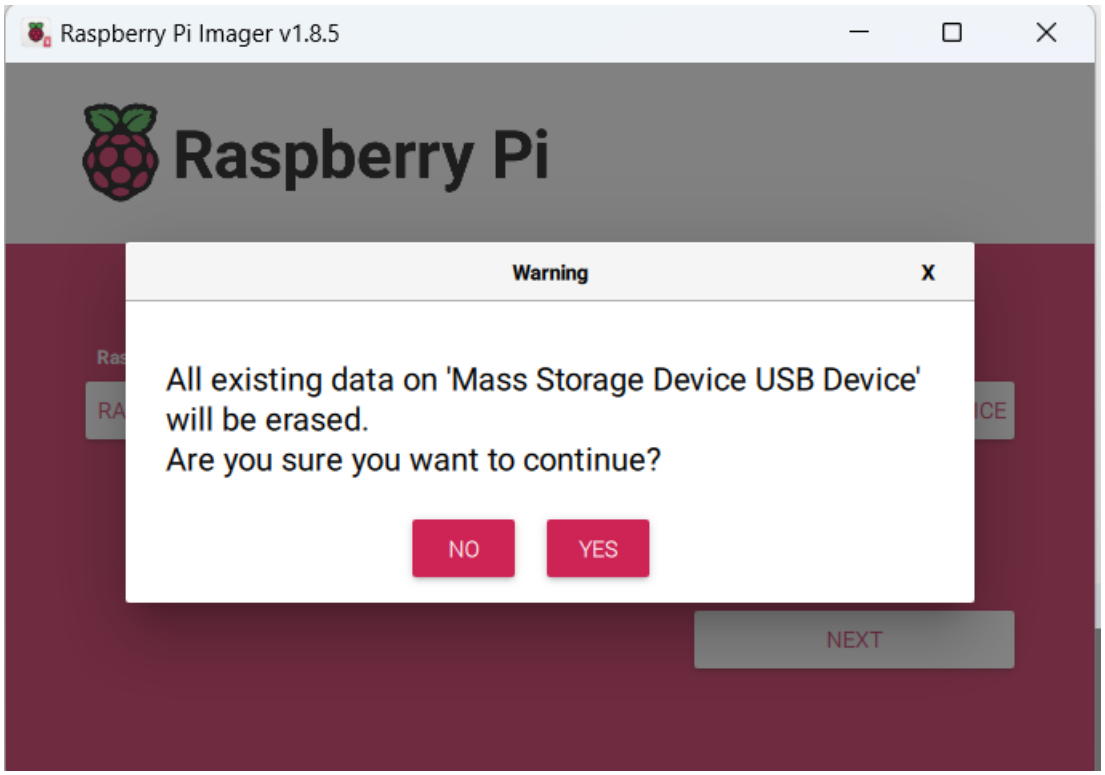
RAMAIAH
College of Arts, Science
& Commerce - Autonomous

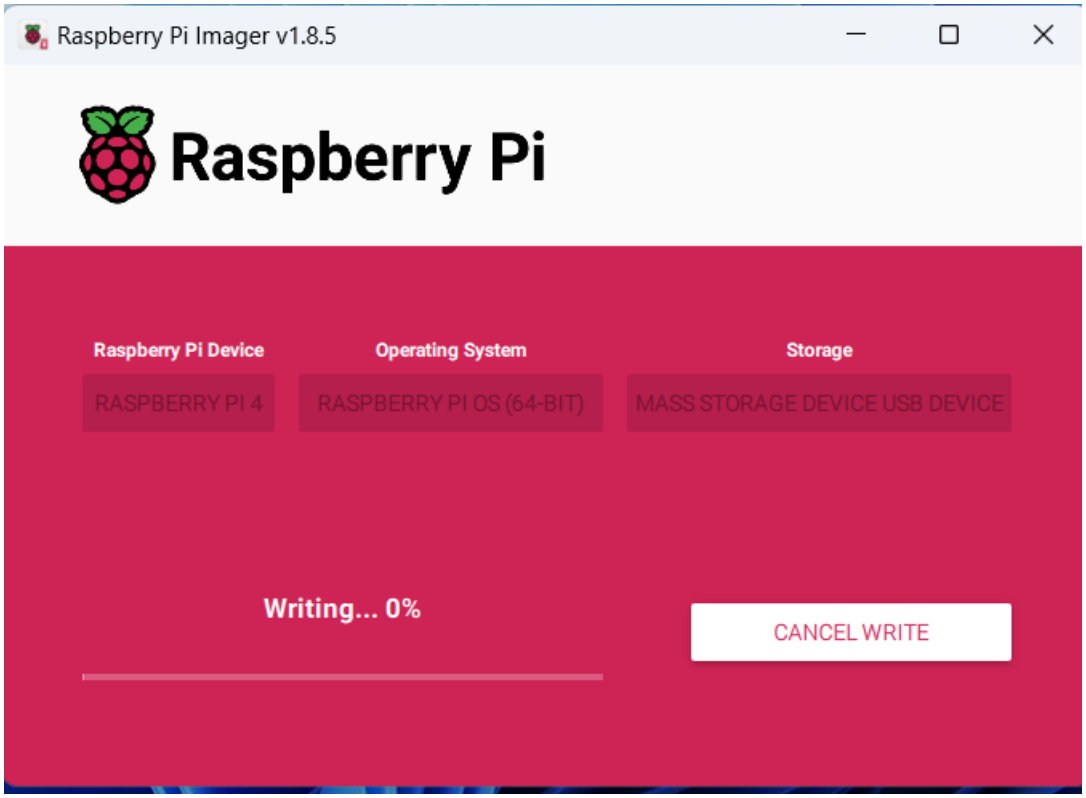


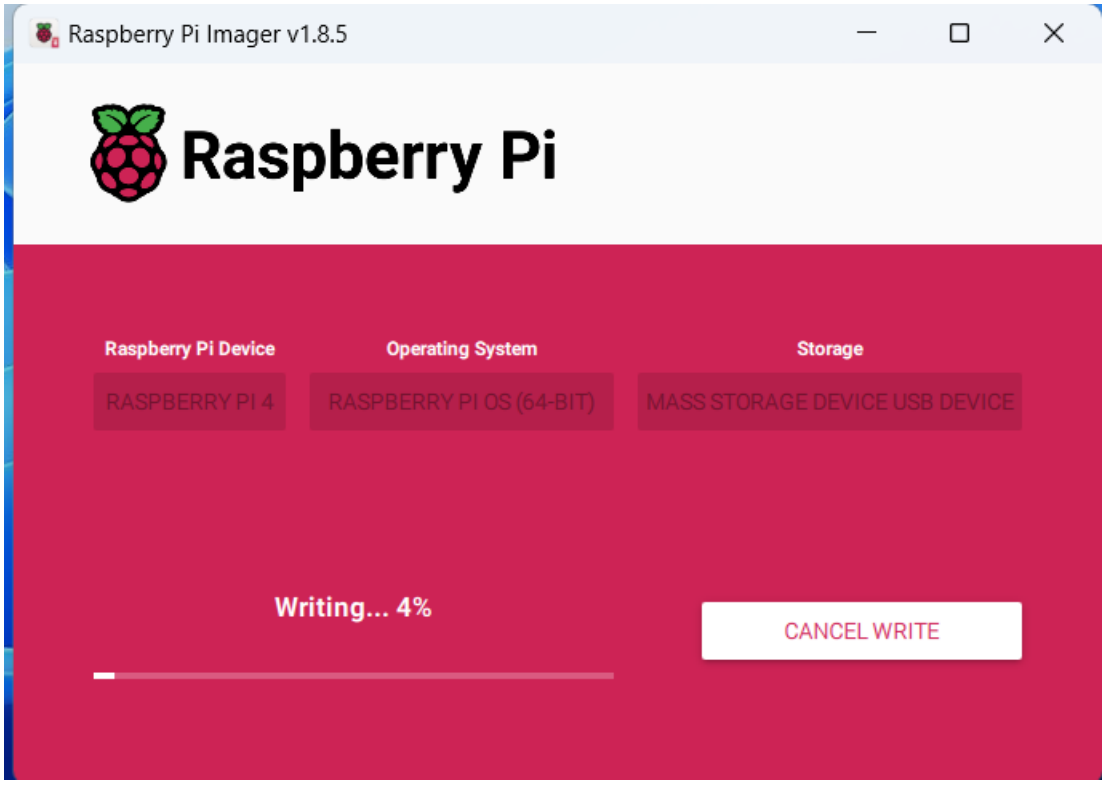
RAMAIAH
College of Arts, Science
& Commerce - Autonomous



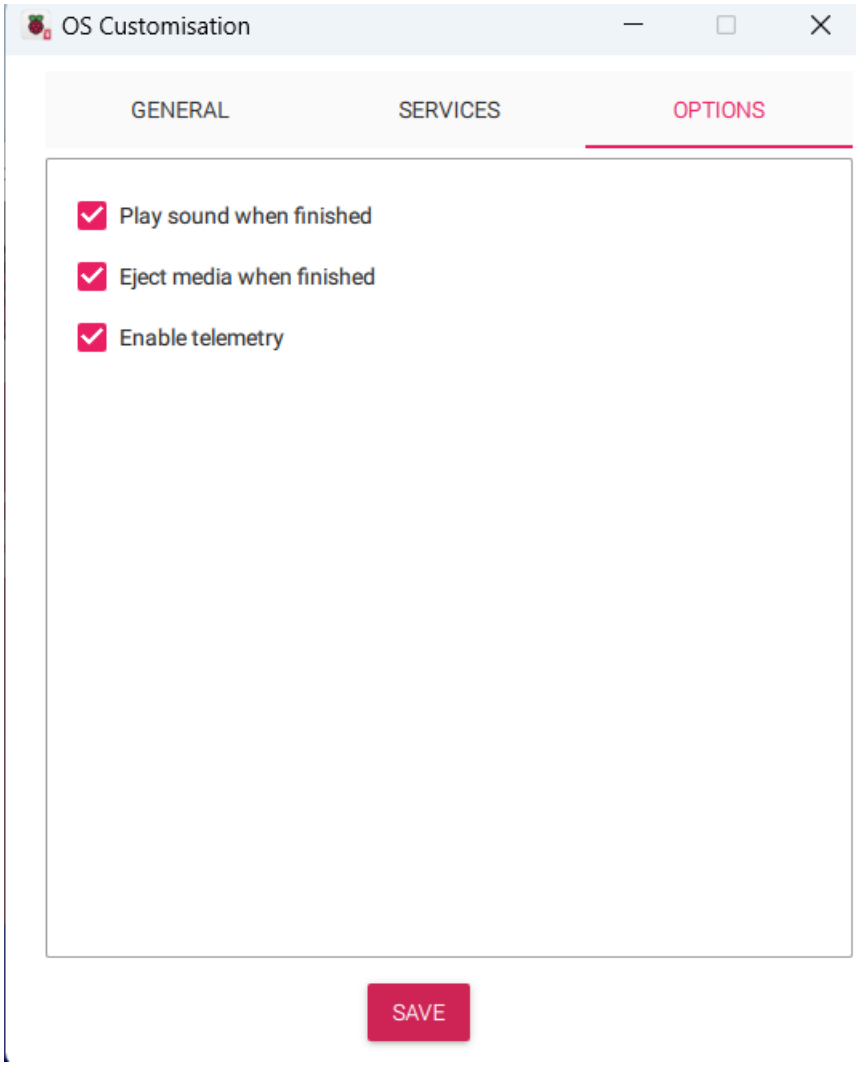

Raspberry Pi Imager v1.8.5
Raspberry Pi
Use OS customisation?
X
Would you like to apply OS customisation settings?
EDIT SETTINGS
NO, CLEAR SETTINGS
YES
NO
NEXT


Raspberry Pi Imager v1.8.5
Raspberry Pi
Warning
X
All existing data on 'Mass Storage Device USB Device' will be erased.
Are you sure you want to continue?
NO
YES
NEXT


Raspberry Pi Imager v1.8.5
Raspberry Pi
Raspberry Pi Device
Operating System
Storage
RASPBERRY PI 4
RASPBERRY PI OS (64-BIT)
MASS STORAGE DEVICE USB DEVICE
Writing... 0%
CANCEL WRITE


Raspberry Pi Imager v1.8.5
Raspberry Pi
Raspberry Pi Device
Operating System
Storage
RASPBERRY PI 4
RASPBERRY PI OS (64-BIT)
MASS STORAGE DEVICE USB DEVICE
Writing... 4%
CANCEL WRITE

RAMAIAH
College of Arts, Science
& Commerce - Autonomous



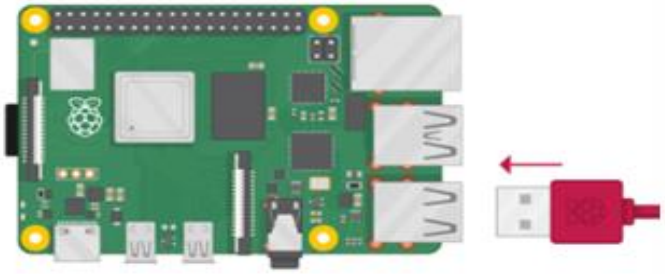
Connect your Raspberry Pi

Let's connect up your Raspberry Pi and get it running.

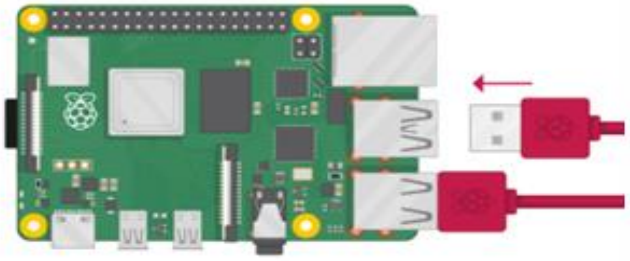
- Check the slot on the underside of your Raspberry Pi to see whether an SD card is inside. If no SD card is there, then insert an SD card with Raspbian installed (via NOOBS).

Note: Many microSD cards come inside a larger adapter — you can slide the smaller card out using the lip at the bottom.

- Find the USB connector end of your mouse's cable, and connect the mouse to a USB port on your Raspberry Pi (it doesn't matter which port you use).



- Connect the keyboard in the same way.

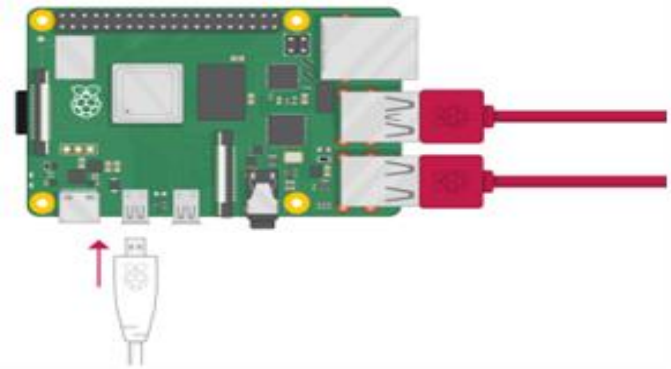


- Make sure your screen is plugged into a wall socket and switched on.
- Look at the HDMI port(s) on your Raspberry Pi — notice that they have a flat side on top.

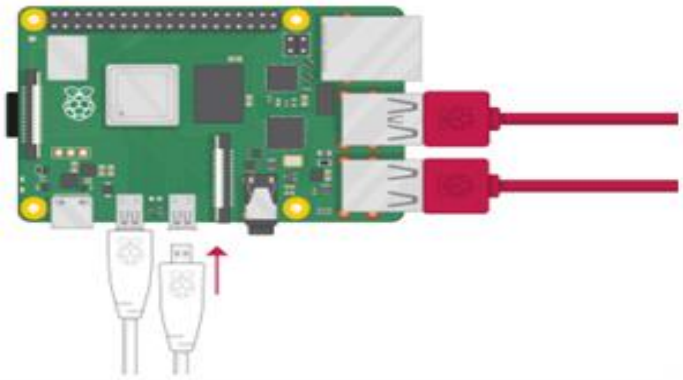
- Use a cable to connect the screen to the Raspberry Pi's HDMI port — use an adapter if necessary.

Raspberry Pi 4

Connect your screen to the first of Raspberry Pi 4's HDMI ports, labelled HDMI0.

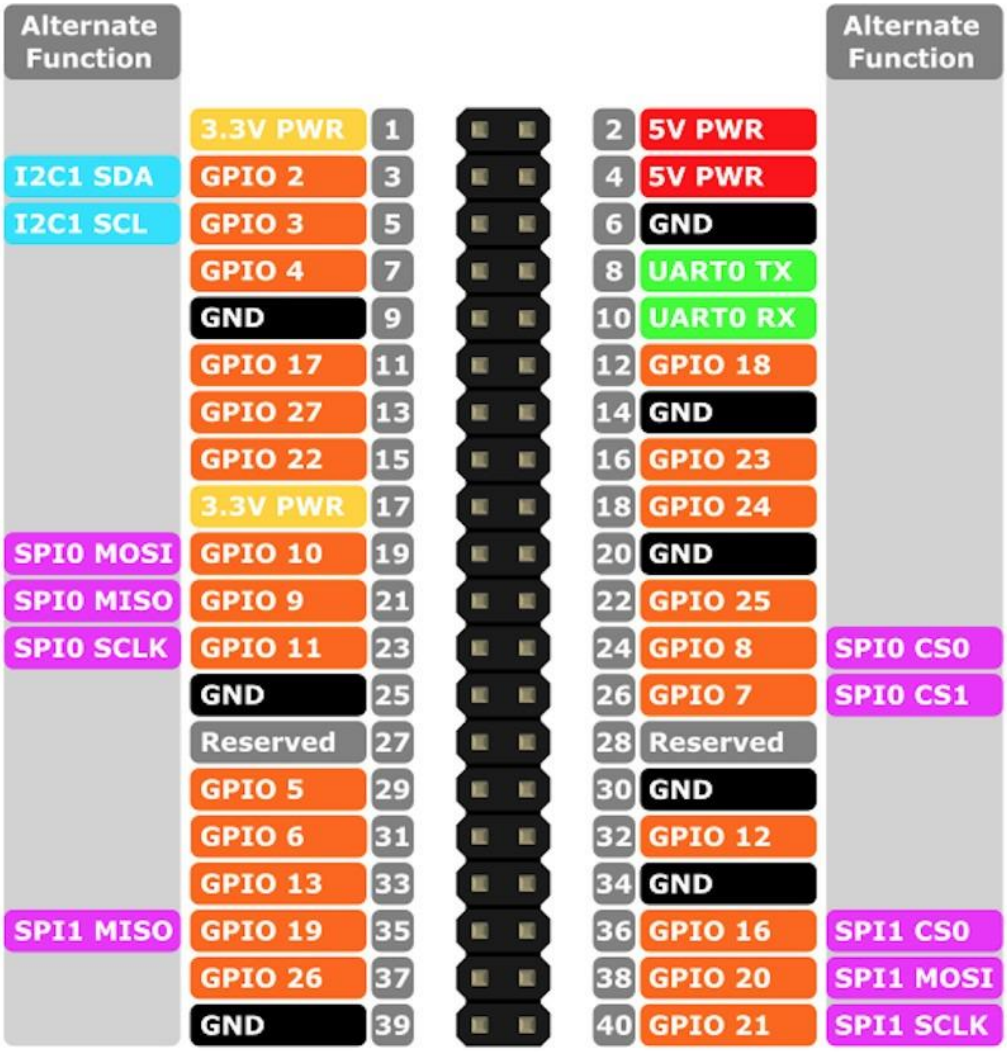


You could connect an optional second screen in the same way.



RAMAIAH
College of Arts, Science
& Commerce - Autonomous

Rasperry Pi 4 Pin Configuration



Result-:

Familiarized with Raspberry Pi components and interface, connecting to Ethernet, Monitor, USB and done the installation of Raspian OS.

Experiment No: 2

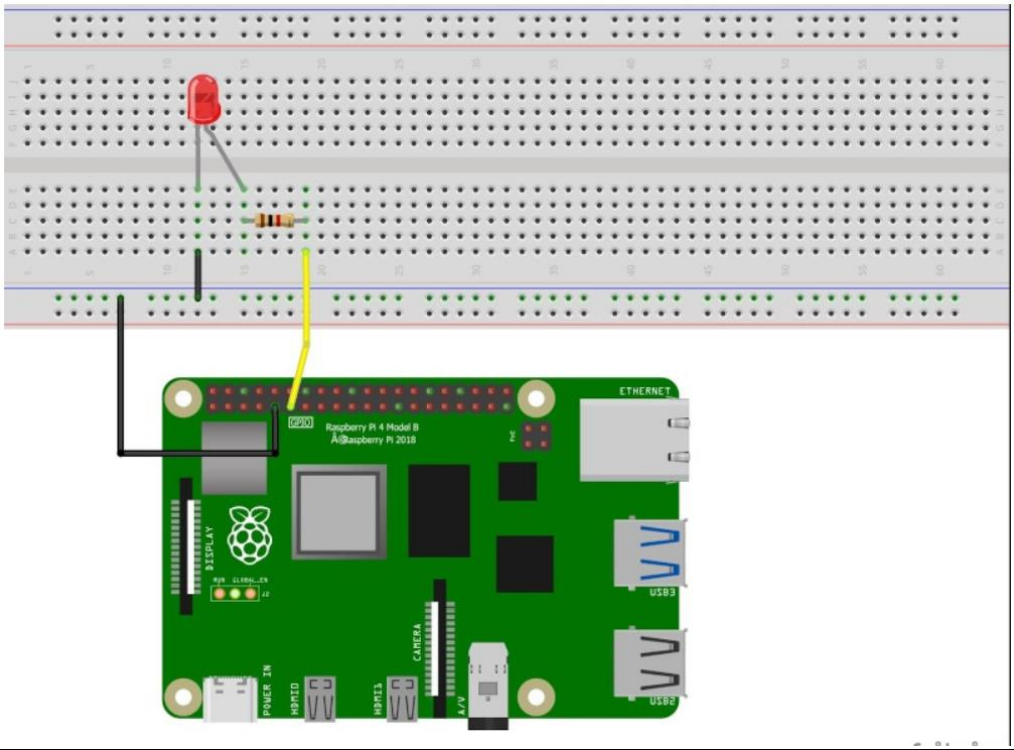
DISPLAYING DIFFERENT LED PATTERNS WITH RASPBERRY PI.

Aim-:

To display different LED patterns using Raspberry pi

1. **Flashing Single LED**

CIRCUIT DIAGRAM



Components Required-:

- breadboard
- Raspberry Pi 4
- 1 LED
- 1 resistor: 330 Ohm
  A set of male to female wires

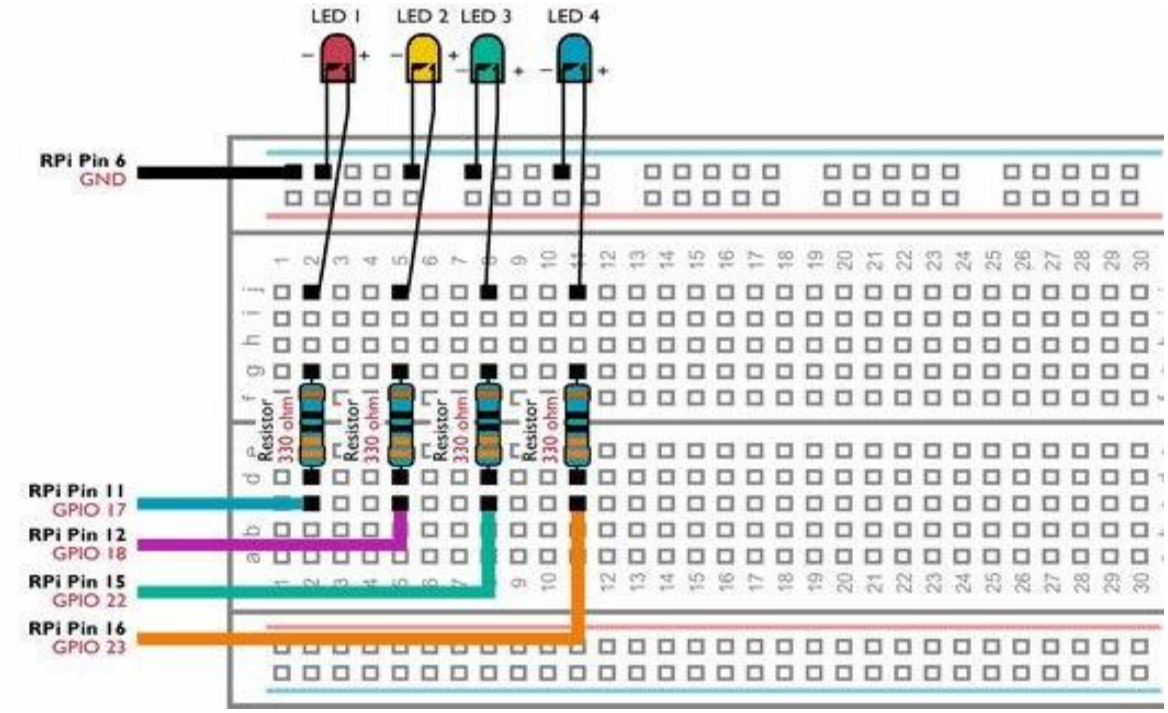
Program Code for controlling Single LED

```
import RPi.GPIO as GPIO
import time

LED_PIN = 17

GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PIN, GPIO.OUT)
```

```
GPIO.output(LED_PIN, GPIO.HIGH)
time.sleep(1)
GPIO.output(LED_PIN, GPIO.LOW)

GPIO.cleanup()
```

2. Flashing on and off four LEDs in a sequence

CIRCUIT DIAGRAM



Components Required-:

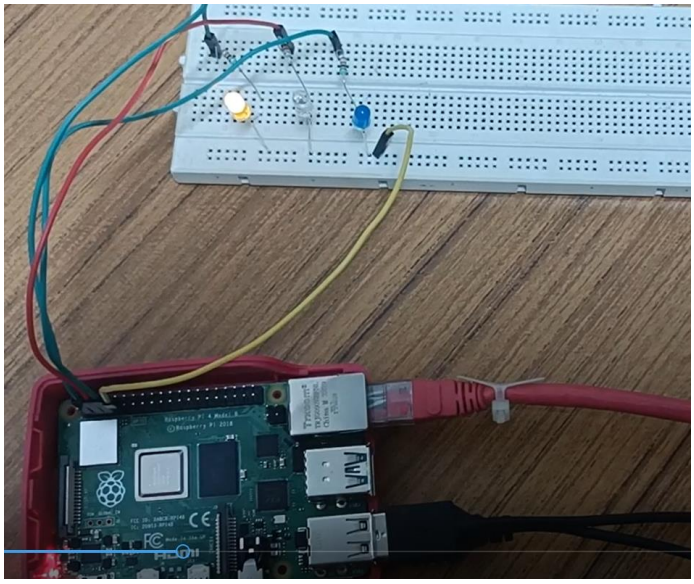
- breadboard
- Raspberry Pi 4
- 4 LEDs
- 330 Ohm Resistors (4)
  A set of male to female wires

Program Code for Flashing on and off four LEDs in a sequence

```
import RPi.GPIO as GPIO
import time
```

```
LED_PIN_1 = 17
LED_PIN_2 = 18
LED_PIN_3 = 22
LED_PIN_4 = 23

GPIO.setmode(GPIO.BCM)



while True:
    GPIO.setup(LED_PIN_1, GPIO.OUT)
    GPIO.output(LED_PIN_1, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(LED_PIN_1, GPIO.LOW)
    time.sleep(1)
    GPIO.setup(LED_PIN_2, GPIO.OUT)
    GPIO.output(LED_PIN_2, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(LED_PIN_2, GPIO.LOW)
    time.sleep(1)
    GPIO.setup(LED_PIN_3, GPIO.OUT)
    GPIO.output(LED_PIN_3, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(LED_PIN_3, GPIO.LOW)
    time.sleep(1)
    GPIO.setup(LED_PIN_4, GPIO.OUT)
    GPIO.output(LED_PIN_4, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(LED_PIN_4, GPIO.LOW)
    time.sleep(1)


GPIO.cleanup()
```

Circuit Setup



3. Display of different LED patterns

```
import RPi.GPIO as GPIO
import time

LED_PINS=[17,18,22,23]

GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PINS,GPIO.OUT)

def clear_leds():
    for pin in LED_PINS:
        GPIO.output(pin,GPIO.LOW)

def pattern_blink():
    """all leds blink together"""
    for _ in range(5):
        GPIO.output(LED_PINS,GPIO.HIGH)
```

```
        time.sleep(0.5)
        GPIO.output(LED_PINS,GPIO.LOW)
        time.sleep(0.5)


def pattern_chase():
    """LEDs turn on one by one like a running light"""
    for _ in range(5):
        for pin in LED_PINS:
            GPIO.output(pin, GPIO.HIGH)
            time.sleep(0.2)
            GPIO.output(pin, GPIO.LOW)


def pattern_alternate():
    """ Alternates even and odd LEDs """
    for _ in range(5):
        GPIO.output(LED_PINS[0::2], GPIO.HIGH) # Turn on 1st and 3rd LED
        GPIO.output(LED_PINS[1::2], GPIO.LOW) # Turn off 2nd and 4th LED
        time.sleep(0.5)

        GPIO.output(LED_PINS[0::2], GPIO.LOW) # Turn off 1st and 3rd LED
        GPIO.output(LED_PINS[1::2], GPIO.HIGH) # Turn on 2nd and 4th LED
        time.sleep(0.5)


def pattern_wave():
    """ LEDs turn on one by one, stay on, then turn off one by one """
    for pin in LED_PINS:
```

```
        GPIO.output(pin, GPIO.HIGH)
        time.sleep(0.5)

    for pin in reversed(LED_PINS):
        GPIO.output(pin, GPIO.LOW)
        time.sleep(0.5)

# Run LED patterns in sequence
try:
    while True:
        print("Pattern: Blink")
        pattern_blink()

        print("Pattern: Chase")
        pattern_chase()

        print("Pattern: Alternate")
        pattern_alternate()

        print("Pattern: Wave")
        pattern_wave()



except KeyboardInterrupt:
    print("Exiting program...")
    clear_leds() # Turn off all LEDs
    GPIO.cleanup() # Reset GPIO settings
```

Result-:
Displayed different LED patterns using Raspberry Pi 4

Experiment No:3

VISITOR MONITORING WITH RASPBERRY PI AND PI CAMERA

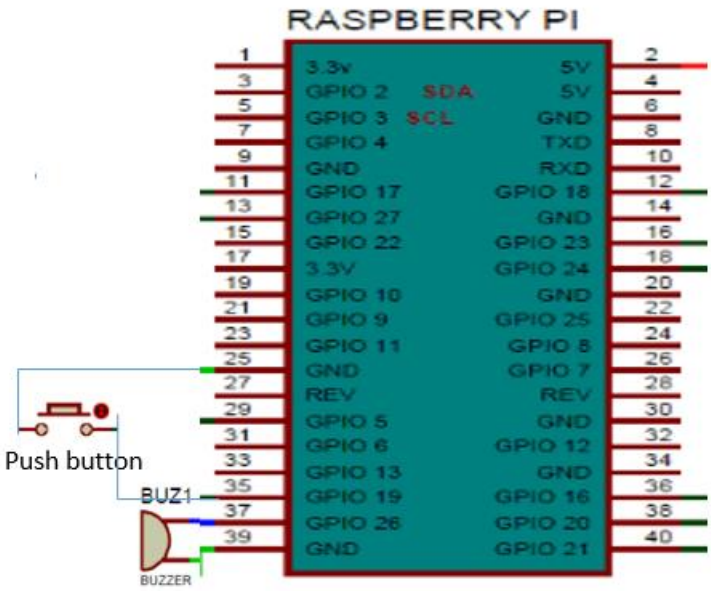
Aim-:

To monitor the visitor when push button is pressed using Raspberrpi and Pi camera module

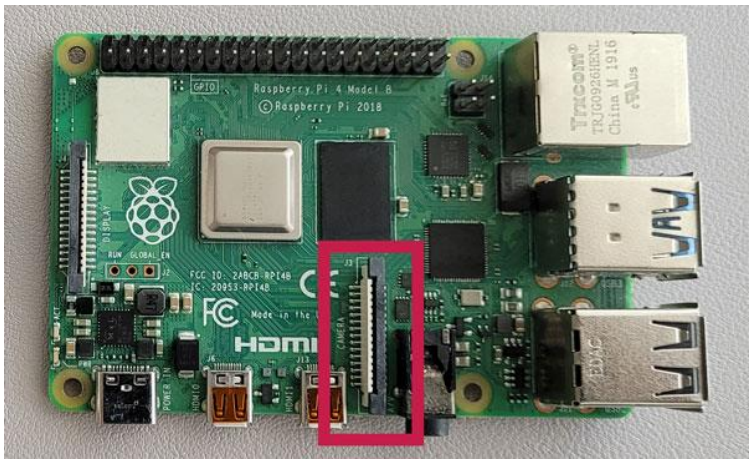
Components Required-:

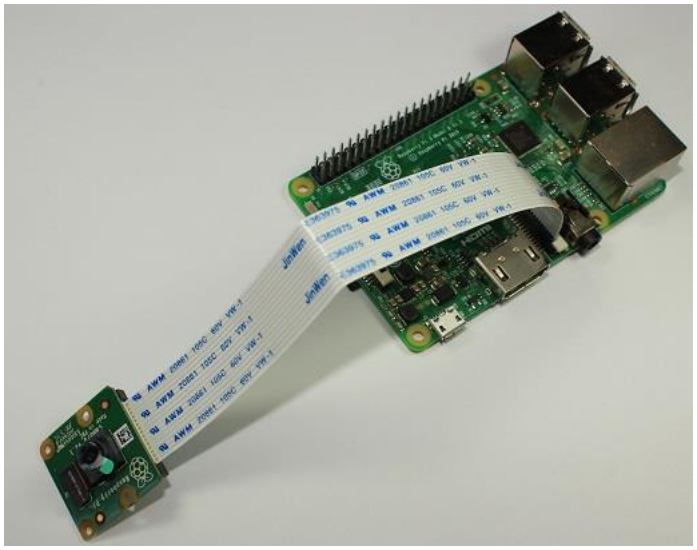
Raspberry Pi 4, Camera Module, Push button, Buzzer

Circuit Diagram



Connecting Camera Module to Raspberry pin 4





Procedure

1. Locate the Camera Module port
2. Gently pull up on the edges of the port's plastic clip
3. Insert the Camera Module ribbon cable; make sure the connectors at the bottom of the ribbon cable are facing the contacts in the port.
4. Push the plastic clip back into place
5. Connect the push button between GPIO pin 19 and ground
6. Connect the buzzer between GPIO pin 26 and ground

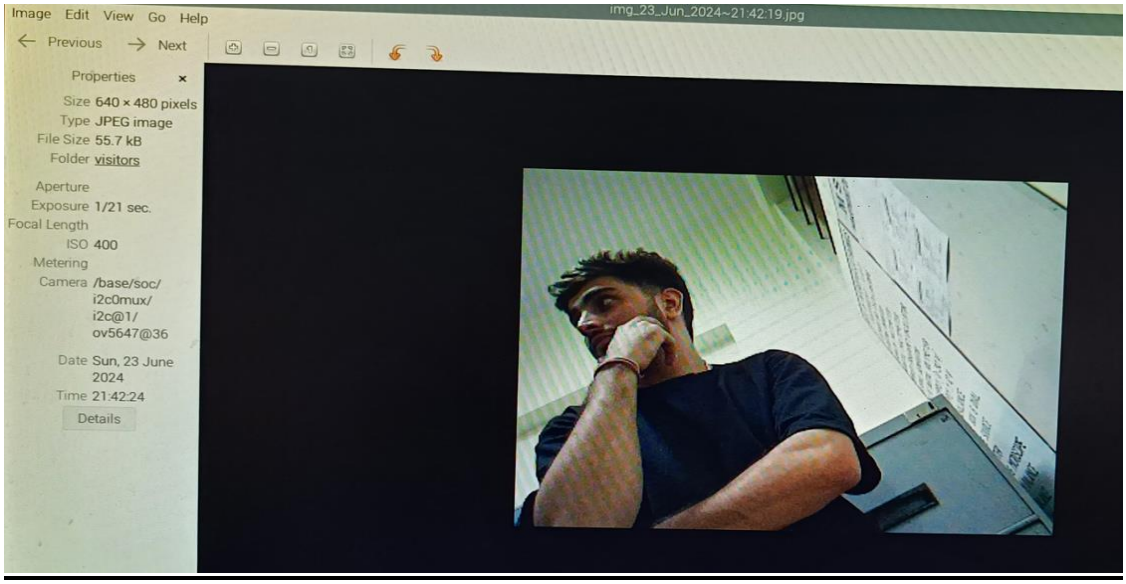
Program Code

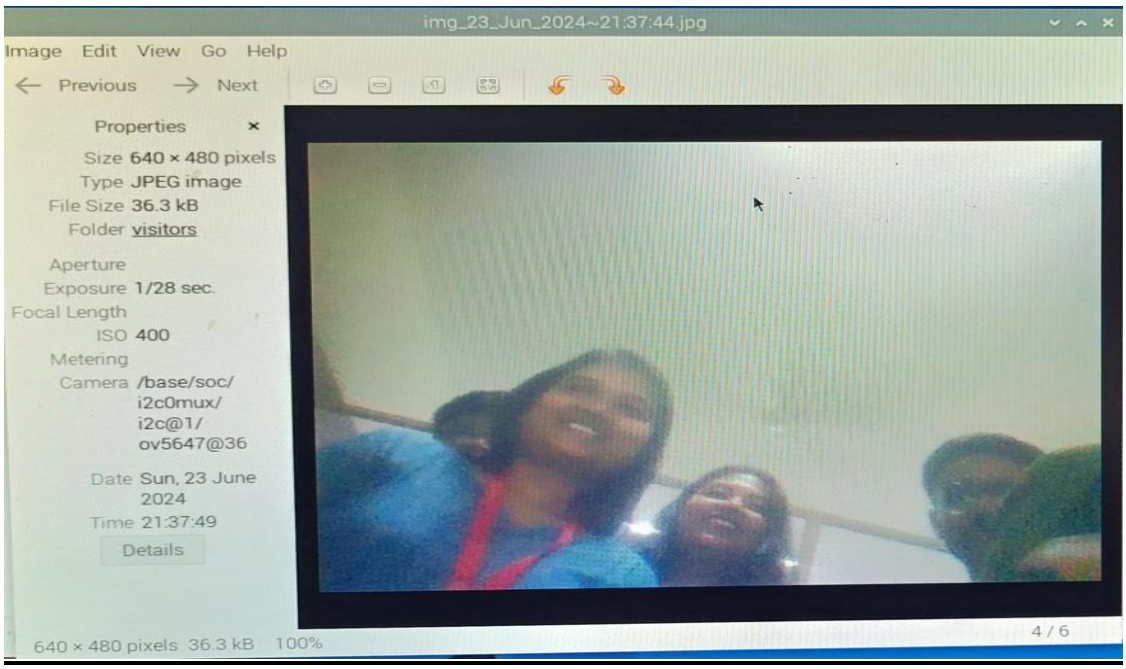
```
import RPi.GPIO as gpio
from picamera2 import Picamera2, Preview
import time


# GPIO pins
buz = 26
button = 19
# Constants
HIGH = 1
LOW = 0


# Setup GPIO
gpio.setmode(gpio.BCM)
gpio.setup(button, gpio.IN, pull_up_down=gpio.PUD_UP)  # Button with pull-up resistor
gpio.setup(buz, gpio.OUT)
gpio.output(buz, LOW)


# Initialize camera
camera = Picamera2()


def capture_image():
    try:
        data = time.strftime("%d_%b_%Y~%H:%M:%S")
```

```
        camera.start_preview()
        time.sleep(2)  # Give time for the camera to warm up
        print(data)
        filename = f"/home/Electronicsiot/Desktop/visitors/img_{data}.jpg"
        camera.capture_file(filename)
        print("Image saved:", filename)
    finally:
        camera.stop_preview()


try:
    while True:
        # Button press detection with debouncing
        while gpio.input(button) == LOW:
            time.sleep(0.1)  # Wait for button press


            gpio.output(buz, HIGH)  # Sound buzzer
            time.sleep(0.1)  # Debouncing delay
            gpio.output(buz, LOW)  # Turn off buzzer
            capture_image()


except KeyboardInterrupt:
    pass  # Exit cleanly on Ctrl+C


finally:
```

```
gpio.cleanup()  # Clean up GPIO on exit

camera.close()  # Cleanly close camera resources
```

Images Captured by Raspberry Pi Camera





Result-

The visitor image is captured through Pi camera with time, day and date when push button is pressed

Experiment No:4

DISPLAYING TIME OVER 4-DIGIT 7-SEGMENT DISPLAY USING RASPBERRY PI.

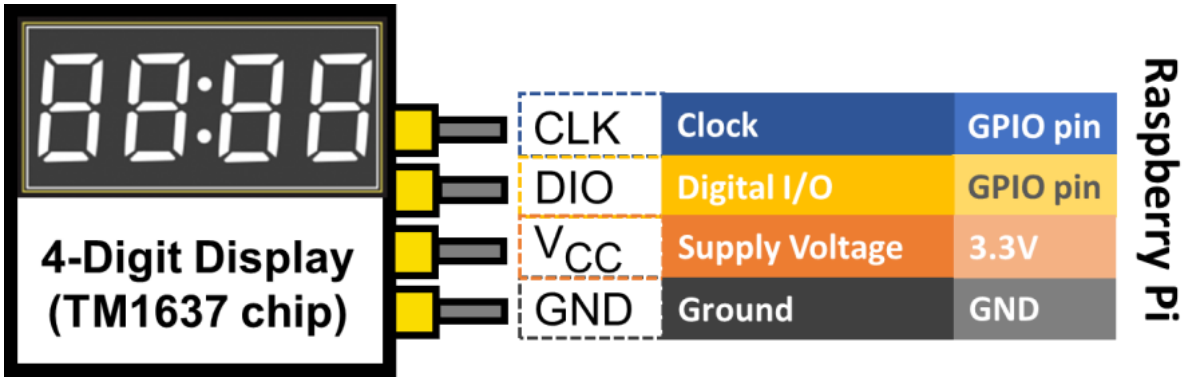
Aim-:

To display current time on 4 digit TM1637 display module.

Components Required-:

TM1637 Display Module, Female to female connecting wires

Circuit Diagram-:



Procedure-:

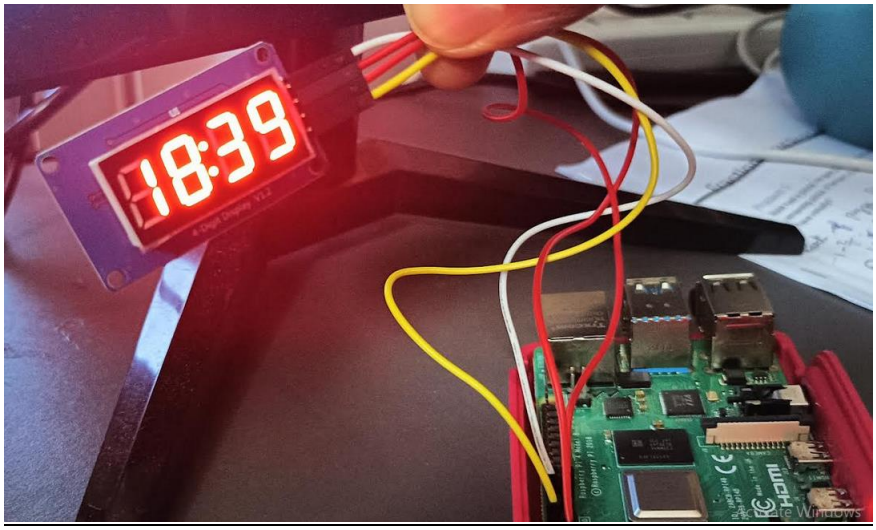
1. Connect CLK pin of TM1637 to GPIO pin18
2. Connect DIO pin of TM1637 to GPIO pin17
3. Connect Vcc to 3.3V of Raspberry Pi
4. Connect GND to raspberry pi GND pin
5. In terminal type: pip install raspberrypi-tm1637 to install tm 1637 module

Program Code-:

```
import tm1637

import time

import numpy as np

from datetime import datetime

from gpiozero import CPUTemperature


# Creating 4-digit 7-segment display object
```

```
tm = tm1637.TM1637(clk=18, dio=17)  # Using GPIO pins 18 and 17
clear = [0, 0, 0, 0]  # Defining values used to clear the display

# Displaying current time
tm.write(clear)
time.sleep(1)
now = datetime.now()
hh = int(datetime.strftime(now,'%H'))
mm = int(datetime.strftime(now,'%M'))
tm.numbers(hh, mm, colon=True)
time.sleep(60)
tm.write(clear)
```

Circuit Set up-:



Result-:

Tm 1637 display module is installed and displayed the current system time over it using python code.

Experiment No:5

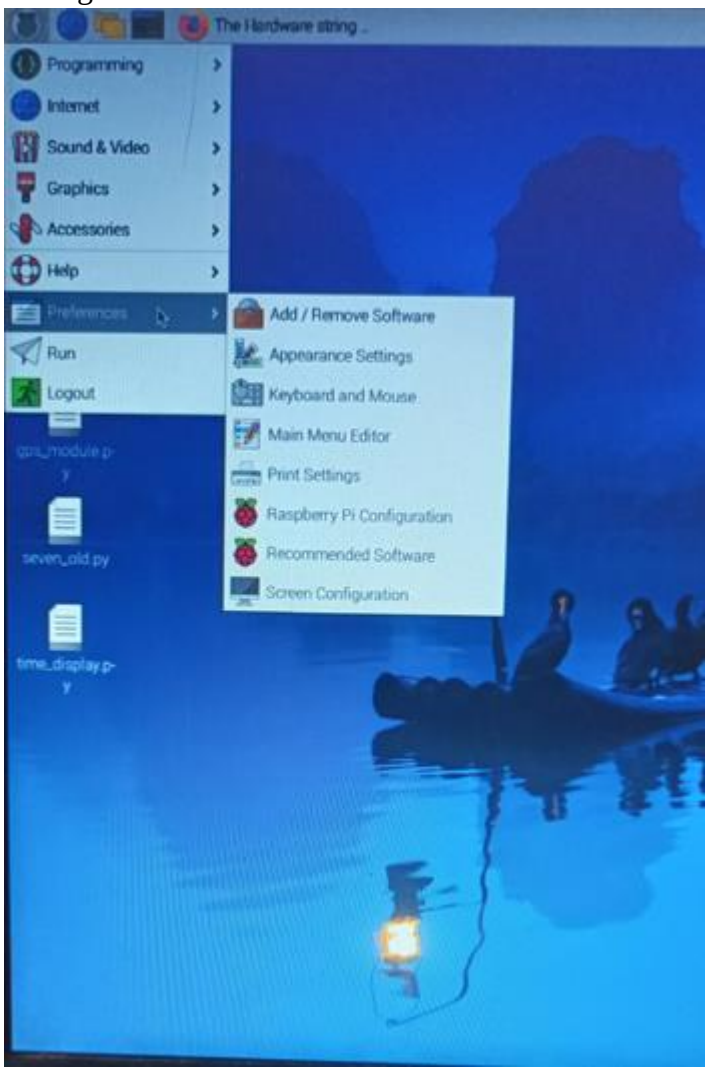
Setting up wireless Access point using Raspberry Pi.

Aim-:

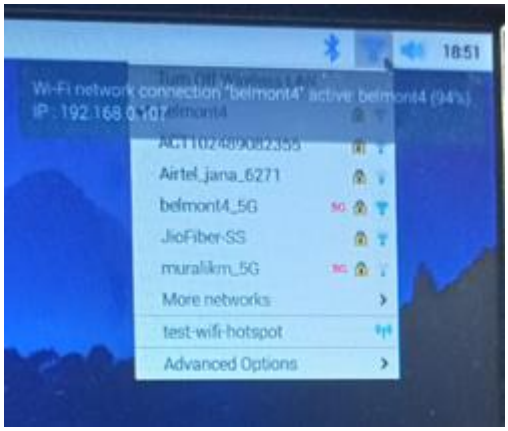
To set up a wireless Access point using Raspberry pi 4

Procedure-:

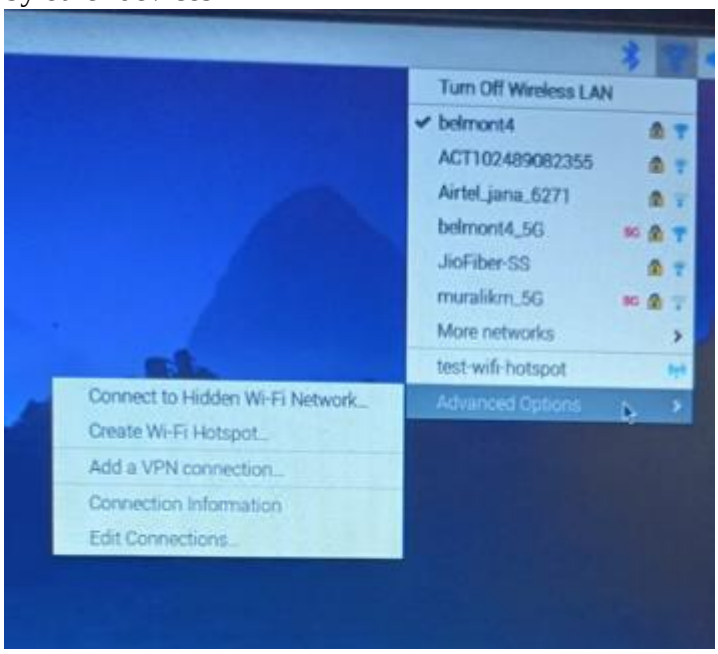
1. Open Raspberry OS window,Select Preferences and Select Raspberry pi Configuration .

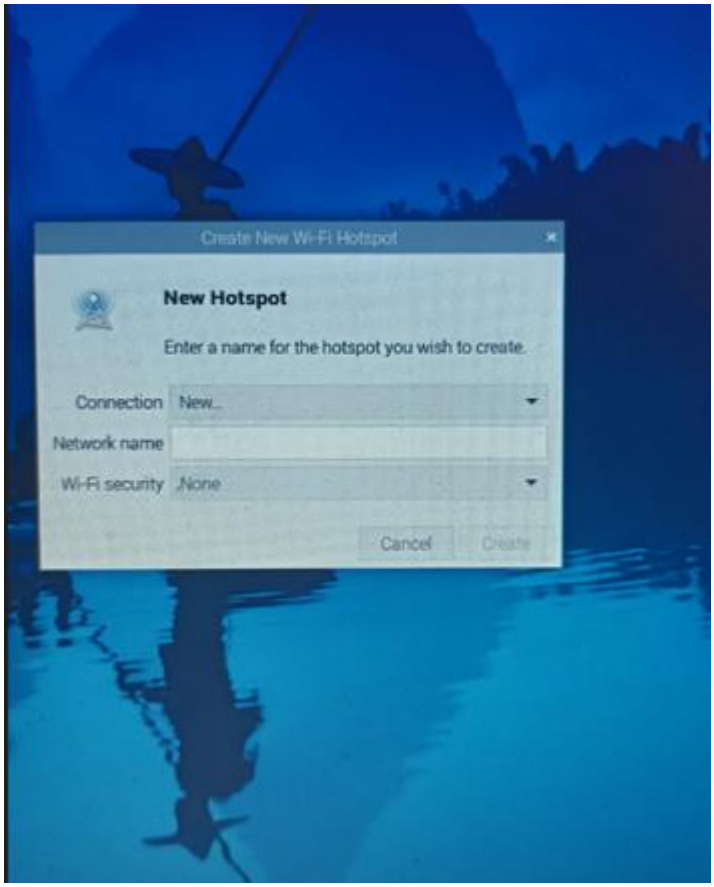


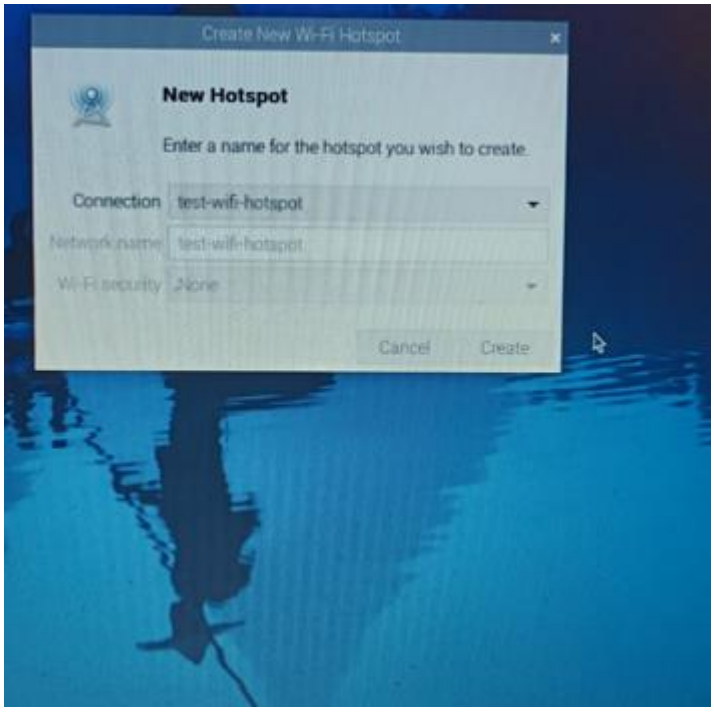
2. Select network option



3. Select Advanced options and Create Wi-Fi hotspot and give a name to be discovered by other devices







Result-:

A wireless Access had been set up using Rsapberry Pi 4.