

DEVELOPMENT KIT RASPBERRY Pi 5 DOCUMENTATION



SSG EMBEDDED SOLUTIONS

info@ssges.co.in

List of Contents

1. Description of Raspberry Pi 5

- Introduction
- Pin Diagram
- Specifications
- Application

2. Installing Raspberry pi 5 Board in Thony IDE

- Install Raspberry pi5
- Installing Thony IDE

3. Examples

- LED Blinking
- LED with Switch
- OLED
- LCD
- DHT11
- Relay
- Buzzer
- DC Motor
- Servo Motor
- Stepper Motor
- IR Sensor
- PIR Sensor
- Ultrasonic Sensor
- Bluetooth
- Seven Segment RFID (EM18)
- Finger Print Sensor Module
- GPS Sensor Module
- Fire Sensor
- MQ135 Gas Sensor
- RTC Module
- Wifi Module
- Colour Sensor
- Sound Sensor
- Camera
- Microphone

SSG EMBEDDED SOLUTIONS

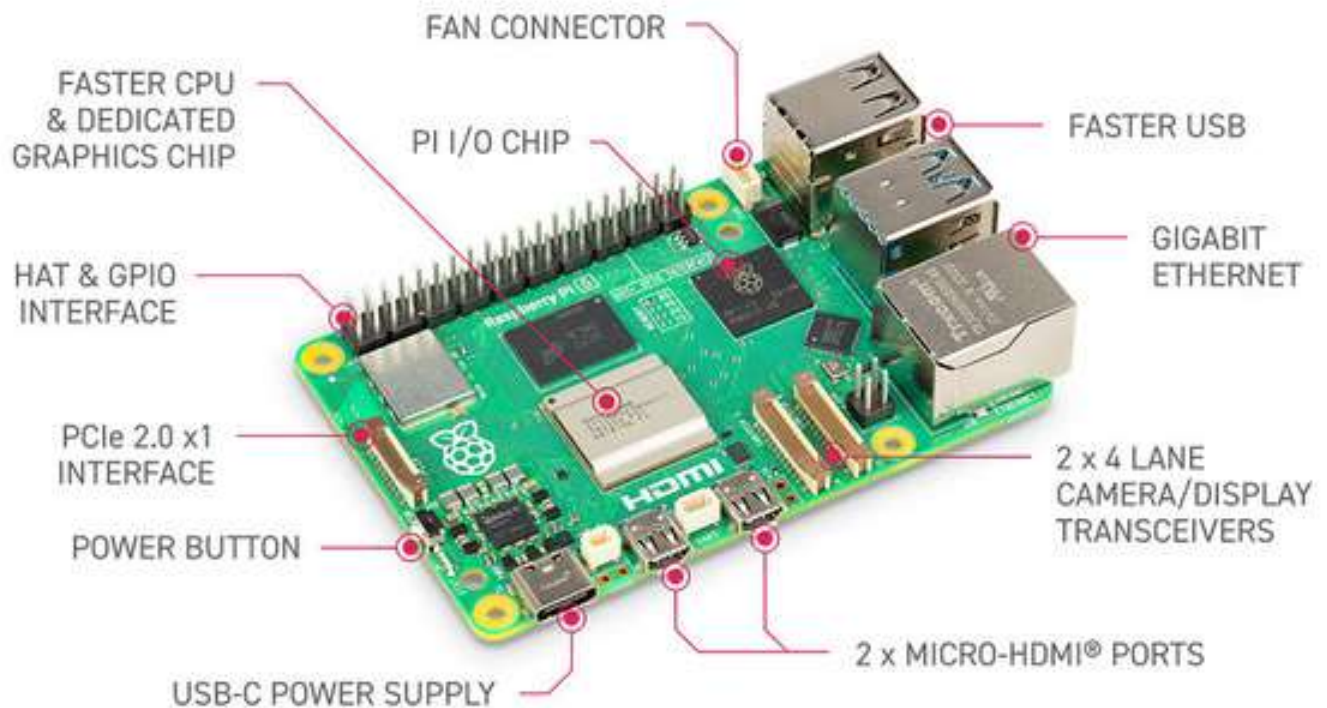
info@ssges.co.in

1. Introduction

Raspberry Pi Compute Module 5 (CM5) is a System on Module (SoM) designed to deliver the functionality of Raspberry Pi in a compact and flexible form factor suited to embedded and industrial applications. It enables developers and system designers to leverage the Raspberry Pi hardware and software stack in their own custom systems and designs. CM5 includes a processor, memory (RAM), eMMC flash storage, and supporting power circuitry. It also provides I/O interfaces beyond those available on standard Raspberry Pi boards, offering expanded options for more complex systems and designs.



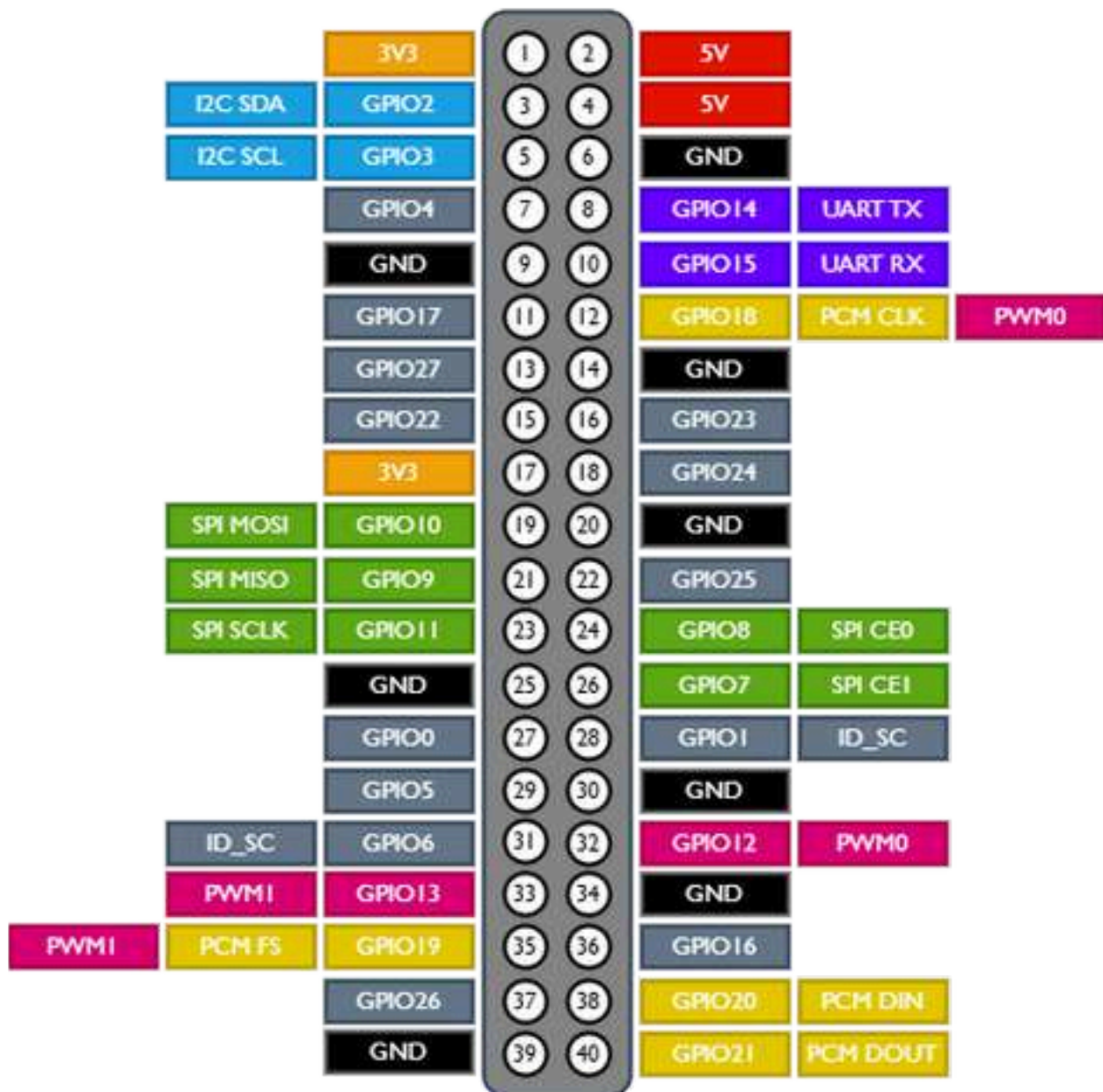
The Raspberry Pi 5 is a significant upgrade featuring a 2.4GHz quad-core Arm Cortex-A76 CPU, VideoCore VII GPU, dual 4Kp60 HDMI, RP1 I/O chip, PCIe 2.0 for fast peripherals, faster Wi-Fi/Bluetooth, and power button, offering 2-3x performance over Pi 4; it comes with 4, 8, or 16GB LPDDR4X RAM, supports up to 16GB RAM, and needs a dedicated 5V/5A USB-C power supply for full potential



One of the defining characteristics of the Raspberry Pi platform is its strong emphasis on hardware-software integration. The Raspberry Pi 5 continues this tradition by providing a 40-pin GPIO (General Purpose Input/Output) header, enabling direct interaction with external hardware such as sensors, actuators, displays, and communication modules. This makes it an ideal platform for hands-on laboratory experiments, embedded system design, and IoT development.

From a software perspective, the Raspberry Pi 5 supports a wide range of Linux-based operating systems, with Raspberry Pi OS (64-bit) being the officially recommended distribution. The operating system is optimized for the board's hardware and includes preinstalled development tools, programming environments, and system utilities. This allows users to focus on application development rather than low-level configuration.

Pin Configuration



Memory

Pin No.	Function
Pin 1	3.3V power output.
Pin 2	5V power output.
Pin 4	5V power output.
Pin 6	Ground.
Pin 9	Ground.
Pin 14	Ground.
Pin 20	Ground.
Pin 25	Ground.
Pin 30	Ground.
Pin 34	Ground.
Pin 39	Ground.

GPIO and Interfaces

Pin No.	Function
Pin 3	GPIO 2, I2C SDA. Used for I2C data communication with sensors and modules.
Pin 5	GPIO 3, I2C SCL. Used for I2C clock signal.
Pin 7	GPIO 4, general-purpose I/O. Used for custom digital input or output.
Pin 8	GPIO 14, UART TX. Serial transmit pin for communication.
Pin 10	GPIO 15, UART RX. Serial receive pin for communication.
Pin 11	GPIO 17, general-purpose I/O. Used for digital control applications.
Pin 12	GPIO 18, PWM. Used for motor or LED control.
Pin 13	GPIO 27, general-purpose I/O. Used for digital interfacing.
Pin 15	GPIO 22, general-purpose I/O. Used for control signals.

Pin 16	GPIO 23, general-purpose I/O. Used for input or output control
Pin 18	GPIO 24, general-purpose I/O. Used for digital interfacing.
Pin 19	GPIO 10, SPI MOSI. Sends data in SPI communication.
Pin 21	GPIO 9, SPI MISO. Receives data in SPI communication.
Pin 22	GPIO 25, general-purpose I/O. Used for digital control.
Pin 23	GPIO 11, SPI SCLK. Provides SPI clock signal.
Pin 24	GPIO 8, SPI CE0. Chip select for SPI devices.
Pin 26	GPIO 7, SPI CE1. Second chip select for SPI devices.
Pin 27	GPIO 0, ID_SD. Used for HAT identification EEPROM.
Pin 28	GPIO 1, ID_SC. Used for HAT identification clock.
Pin 29	GPIO 5, general-purpose I/O. Used in control circuits.
Pin 31	GPIO 6, general-purpose I/O. Used in control

	applications.
Pin 32	GPIO 12, PWM. Used for motor speed or LED dimming.
Pin 33	GPIO 13, PWM. Used for PWM-based output control.
Pin 35	GPIO 19, PWM. Used for PWM and alternate functions.
Pin 36	GPIO 16, general-purpose I/O. Used for control inputs or outputs.
Pin 37	GPIO 26, general-purpose I/O. Used for digital interfacing.
Pin 38	GPIO 20, general-purpose I/O. Used for control signals.
Pin 40	GPIO 21, general-purpose I/O. Used for digital input/output.

Important Interfaces

Pin No.	Function
Pin 3, 5	I2C interface.
Pin 8, 10	UART interface.
Pin 19, 21, 23, 24, 26	SPI interface.
Pin 12, 32, 33, 35	PWM output.

You can use raspberry pi i2c for sensors. Raspberry pi spi is good for fast data. Uart pins help with serial links.

Key Features & Upgrades:

- Processor: 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU with improved cache.
- Performance: Up to 2-3 times faster CPU performance than Raspberry Pi 4.
- Graphics: Video Core VII GPU, supporting dual 4Kp60 HDMI displays with HDR.
- I/O Controller: In-house designed RP1 chip for faster peripheral performance (USB, Ethernet, PCIe).
- Connectivity: PCIe 2.0 x1 interface, Gigabit Ethernet (PoE+ support), dual 2x4-lane MIPI (camera/display) connectors.
- Memory: Available in different RAM sizes (e.g., 1GB, 4GB, 8GB, 16GB) of LPDDR4X SDRAM.
- Power: 5V/5A DC via USB-C, with a dedicated power button and RTC battery header.
- Desktop Experience: Smooth desktop usage, ideal for low-cost computing.

Applications :

The Raspberry Pi 5 is a powerful, versatile single-board computer used as a mini desktop PC, home media center (Kodi), retro gaming console, home automation hub (Home Assistant), network-attached storage (NAS), and for AI/machine learning project.

Requirements

To get started with Raspberry Pi, You'll need the following things:

1. Display & Cable

If your Raspberry Pi has more than one HDMI port, plug your primary monitor into the port marked HDMI0



2 . Keyboard

You can use any of the USB ports on your Raspberry Pi to connect a wired keyboard.



3 Mouse

You can use any of the USB ports on your Raspberry Pi to connect a wired mouse



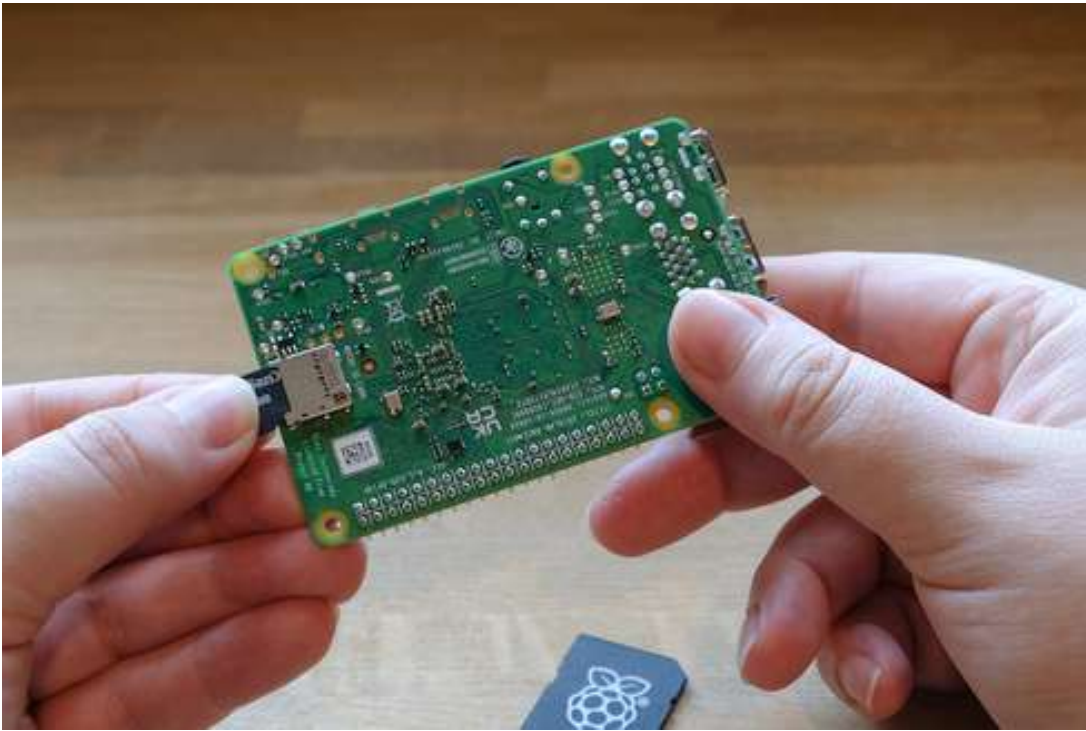
4 Power Supply

USB-PD power mode required to power various Raspberry Pi models



5 Boot Media

All Raspberry Pi consumer models since the Raspberry Pi 1 Model A+ feature a microSD slot. Your Raspberry Pi automatically boots from the microSD slot when the slot contains a card.



We recommend using an SD card with at least 32 GB of storage for Raspberry Pi OS installations.

6 Audio

All Raspberry Pi models with HDMI, micro HDMI, or mini HDMI support audio output over HDMI.

All Raspberry Pi models support audio over USB. All Raspberry Pi models equipped with Bluetooth support Bluetooth audio.

7 Networking

Used for Ethernet Connection

The "Model B" suffix indicates variants with an Ethernet port; "Model A" indicates no Ethernet port. The Keyboard models also include an Ethernet port. If your Raspberry Pi doesn't have an Ethernet port, you can still connect to a wired internet connection using a USB-to-Ethernet adapter.



Downloading and Installing Raspberry Pi OS

Once you have all the components you need, use the following steps to create the boot disk you will need to set up your Raspberry Pi. These steps should work on a using a Windows, Mac or Linux-based PC (we tried this on Windows, but it should be the same on all three).

To use your Raspberry Pi, you'll need an operating system. By default, Raspberry Pi devices check for an operating system on any SD card inserted in the SD card slot.

We recommend installing an operating system using Raspberry Pi Imager. You can use Imager to preconfigure credentials and remote access settings for your Raspberry Pi.

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 on a computer with an SD card reader. Insert the microSD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

Download for windows

https://downloads.raspberrypi.com/imager/imager_latest.exe

Download for macOS

https://downloads.raspberrypi.com/imager/imager_latest.dmg

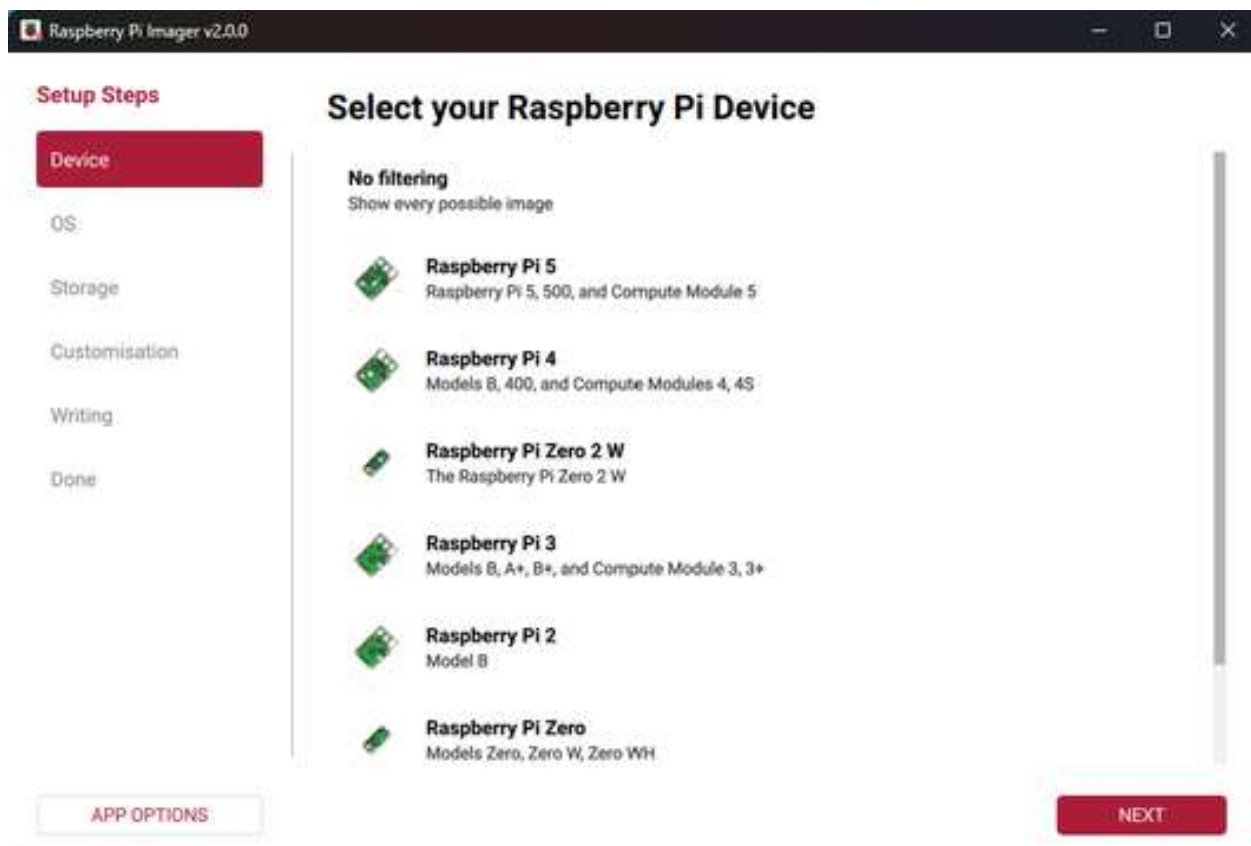
Download for Linux(x86_64)

https://downloads.raspberrypi.com/imager/imager_latest_amd64.AppImage

Step 2. Configure the fundamentals

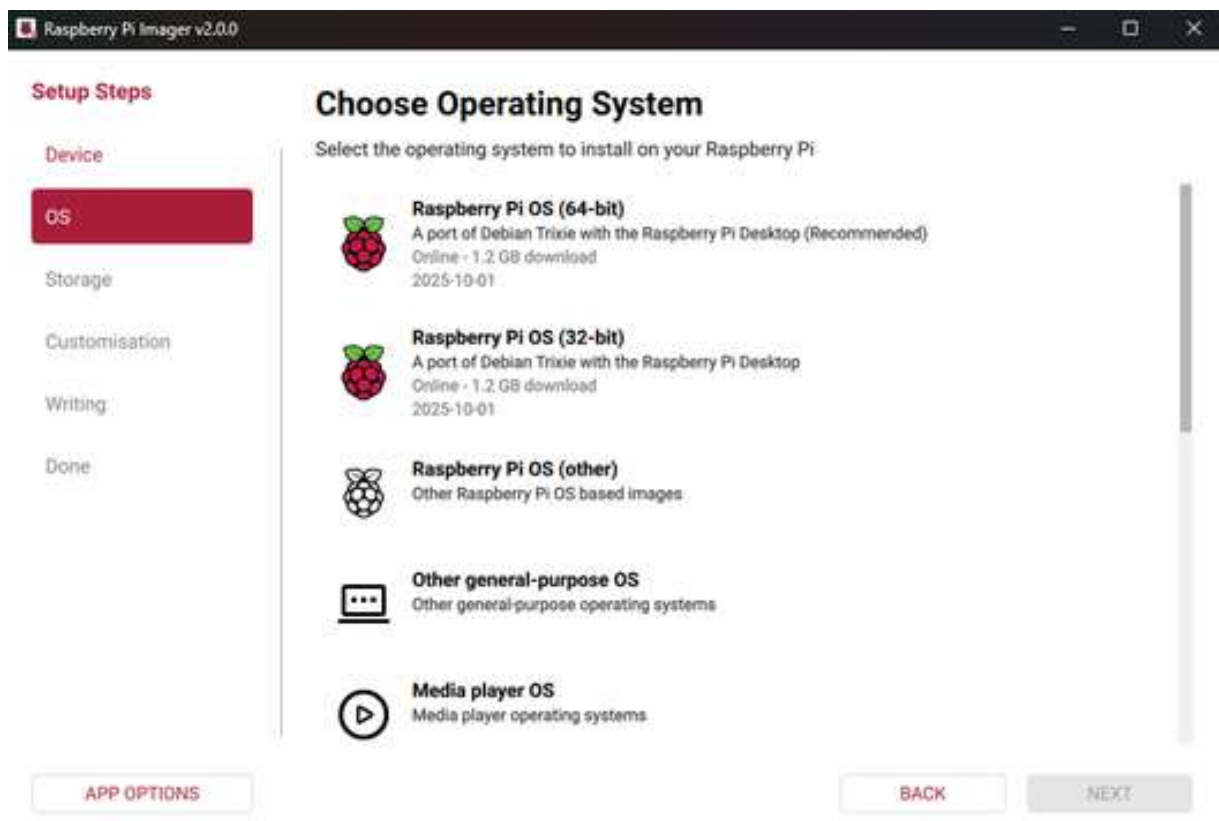
Tell Raspberry Pi Imager what combination of hardware and operating system you want to use:

1. In the Device tab, select your Raspberry Pi model from the list.
Select Next



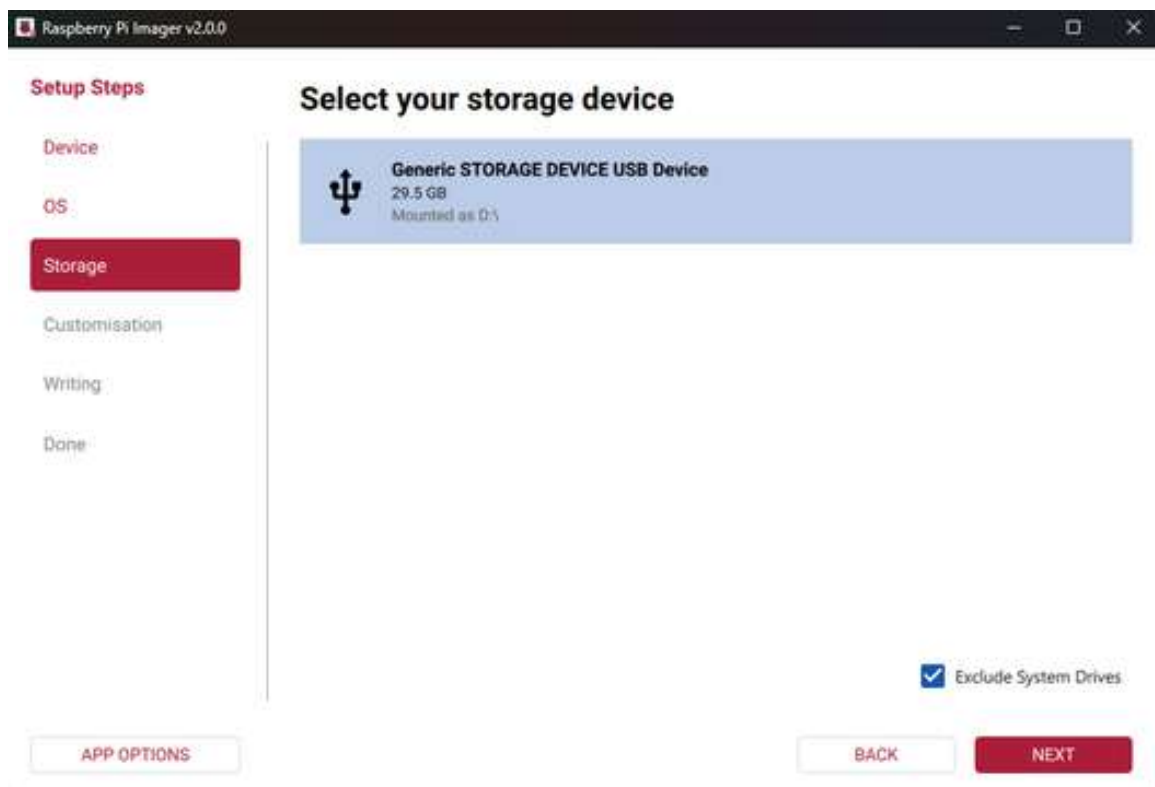
2. In the OS tab, choose from the available operating systems. Select Next.

Imager shows the recommended version of Raspberry Pi OS for your device at the top of the list. To install an operating system from an image on your computer, select the Use Custom option.



Connect your preferred storage device to your computer. For example, plug a microSD card in using an external or built-in SD card reader.

In the Storage tab, select the storage device to write the image to. Select Next.



Warning

If you have more than one storage device connected to your computer, be sure to choose the correct device! You can often identify storage devices by size. If you're unsure, disconnect other devices until you've identified the device you want to image. Imager warns you if you attempt to overwrite a system drive on your computer

Step 3. Customise your operating system

In the Customisation > Hostname tab, enter a host name for your Raspberry Pi that includes only letters, numbers, and hyphens. Select Next.

Setup steps

Device

OS

Storage

Customisation

Hostname

Localisation

User

Wi-Fi

Remote access

Raspberry Pi Connect

Writing

Done

Customisation: Choose hostname

Enter your hostname

raspberrypilog

A hostname is a unique name that identifies your Raspberry Pi on the network. It should contain only letters, numbers, and hyphens.

APP OPTIONS

SKIP CUSTOMISATION

BACK

NEXT

In the Customisation > Localisation tab, choose your capital city. Imager autocompletes the time zone and keyboard layout for that city; you can change these settings. Select Next.

Setup steps

Device

OS

Storage

Customisation

Hostname

Localisation

User

Wi-Fi

Remote access

Raspberry Pi Connect

Writing

Done

Customisation: Localisation

Select your location for suggested time zone and keyboard layout

Capital city:

Abu Dhabi (United Arab Emirates)

Time zone:

Asia/Kolkata

Keyboard layout:

us

APP OPTIONS

SKIP CUSTOMISATION

BACK

NEXT

In the Customisation > User tab, enter a username and password for the admin user of your Raspberry Pi. Select Next.

The screenshot shows the 'Customisation: Choose username' screen in the Raspberry Pi Imager. On the left is a 'Setup steps' sidebar with options: Device, OS, Storage, Customisation (highlighted), Hostname, Localisation, User, Wi-Fi, Remote access, Raspberry Pi Connect, Writing, and Done. The main area has the title 'Customisation: Choose username' and the subtitle 'Create a user account for your Raspberry Pi'. It contains three input fields: 'Username' with the value 'pi@1', 'Password' with masked characters '*****', and 'Confirm password' with masked characters '*****'. A red error message 'No enter to change password' is shown below the password fields. A note at the bottom states: 'The username must be lowercase and contain only letters, numbers, underscores, and hyphens.' At the bottom of the screen are buttons for 'APP OPTIONS', 'SKIP CUSTOMISATION', 'BACK', and 'NEXT'.

In the Customisation > Wi-Fi tab, on first use, Imager pre-fills the SSID (name) and password of the Wi-Fi network you're currently connected to.

- a. Select whether to connect to a Secure network or an Open network.
 - b. If there's no pre-filled information or if you want to change it, enter the SSID (name) of your wireless network. For secure networks, enter and confirm the password for the network.
-
- a. If your network doesn't broadcast an SSID publicly, enable the Hidden SSID setting.
 - b. Select Next.

Setup steps

Device

OS

Storage

Customisation

Hostname

Localisation

User

Wi-Fi

Remote access

Raspberry Pi Connect

Writing

Done

Customisation: Choose Wi-Fi

SECURE NETWORK

OPEN NETWORK

SSID:

Password:

Confirm password:

☐ Hidden SSID

APP OPTIONS

SKIP CUSTOMISATION

BACK

NEXT

Storage

Customisation

Hostname

Localisation

User

Wi-Fi

Remote access

Raspberry Pi Connect

Interfaces & Features

Writing

Done

Customisation: Choose Wi-Fi

SECURE NETWORK

OPEN NETWORK

SSID:

☒ Hidden SSID

APP OPTIONS

SKIP CUSTOMISATION

BACK

NEXT

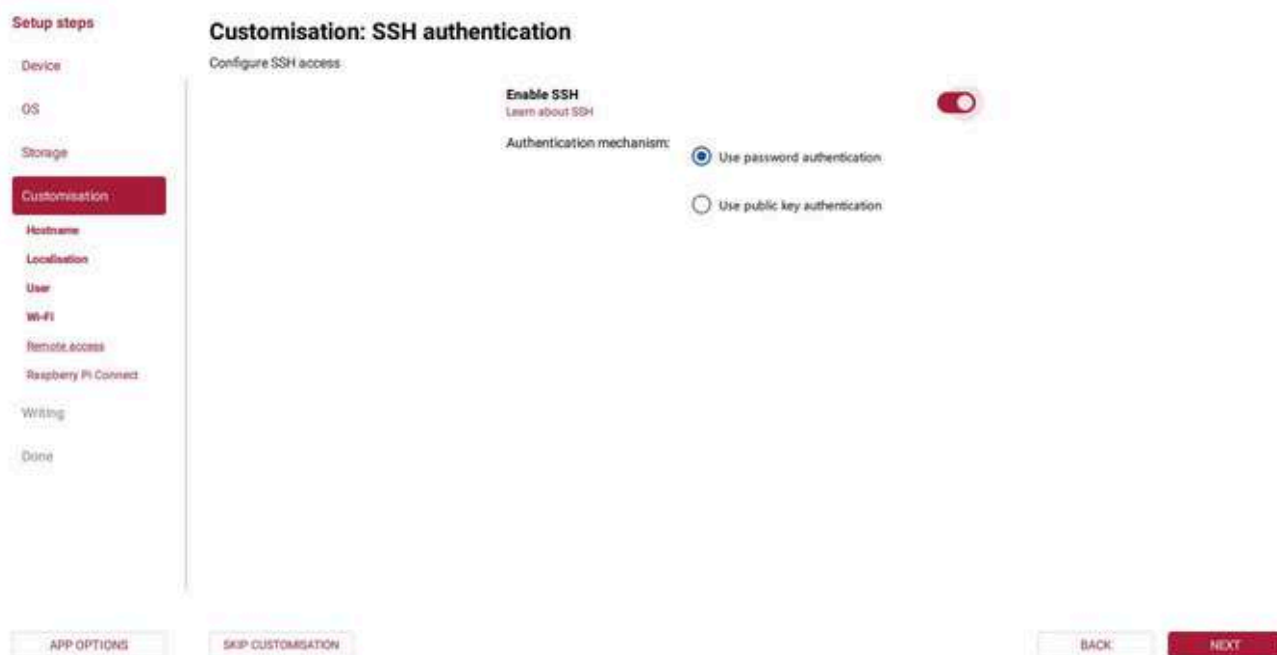
In the Customisation > Remote Access tab, configure your SSH settings.

To use your Raspberry Pi remotely over your network, toggle the Enable SSH switch to the active position.

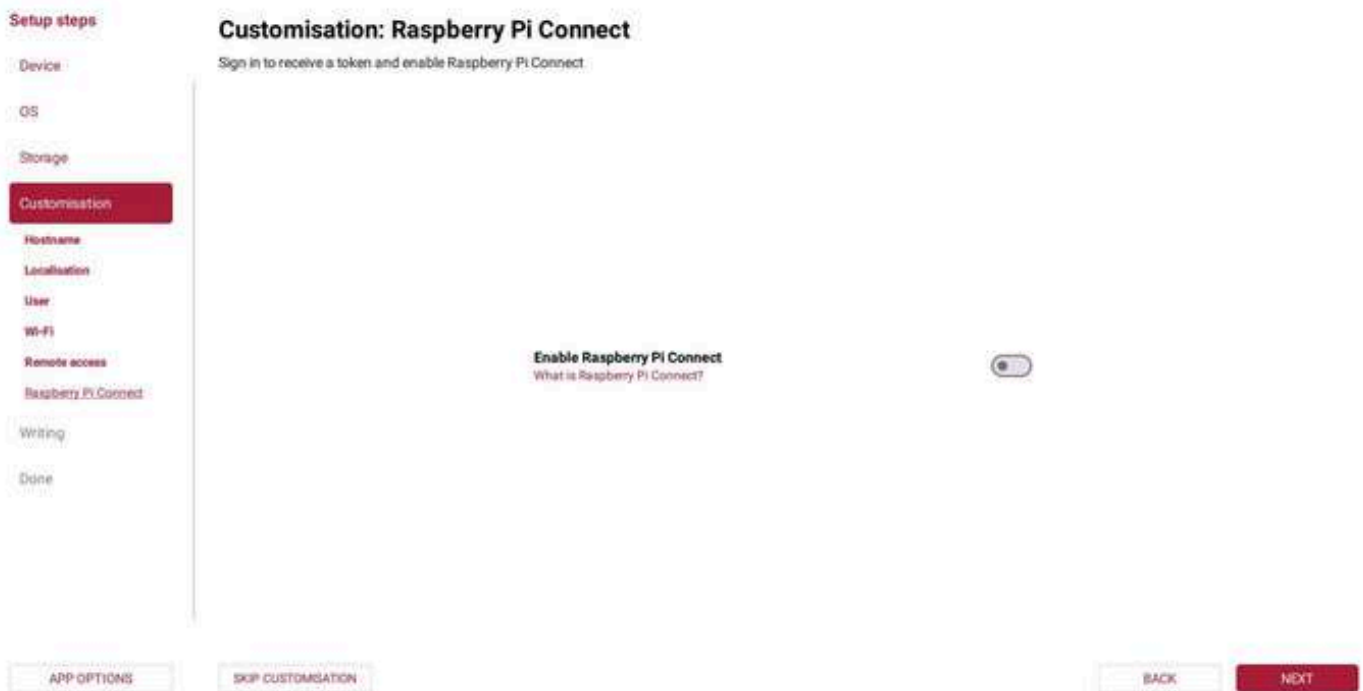
If you plan to run a headless Raspberry Pi, either enable this option or configure Raspberry Pi Connect for your device.

If you enable SSH, choose the authentication method to use:

Choose Use password authentication to log in to your Raspberry Pi over the network using the username and password you provided in the Customisation > User tab.



Some operating systems now show the Customisation > Raspberry Pi Connect tab. To link your device to your Raspberry Pi Connect account, complete the following steps: (its optional)



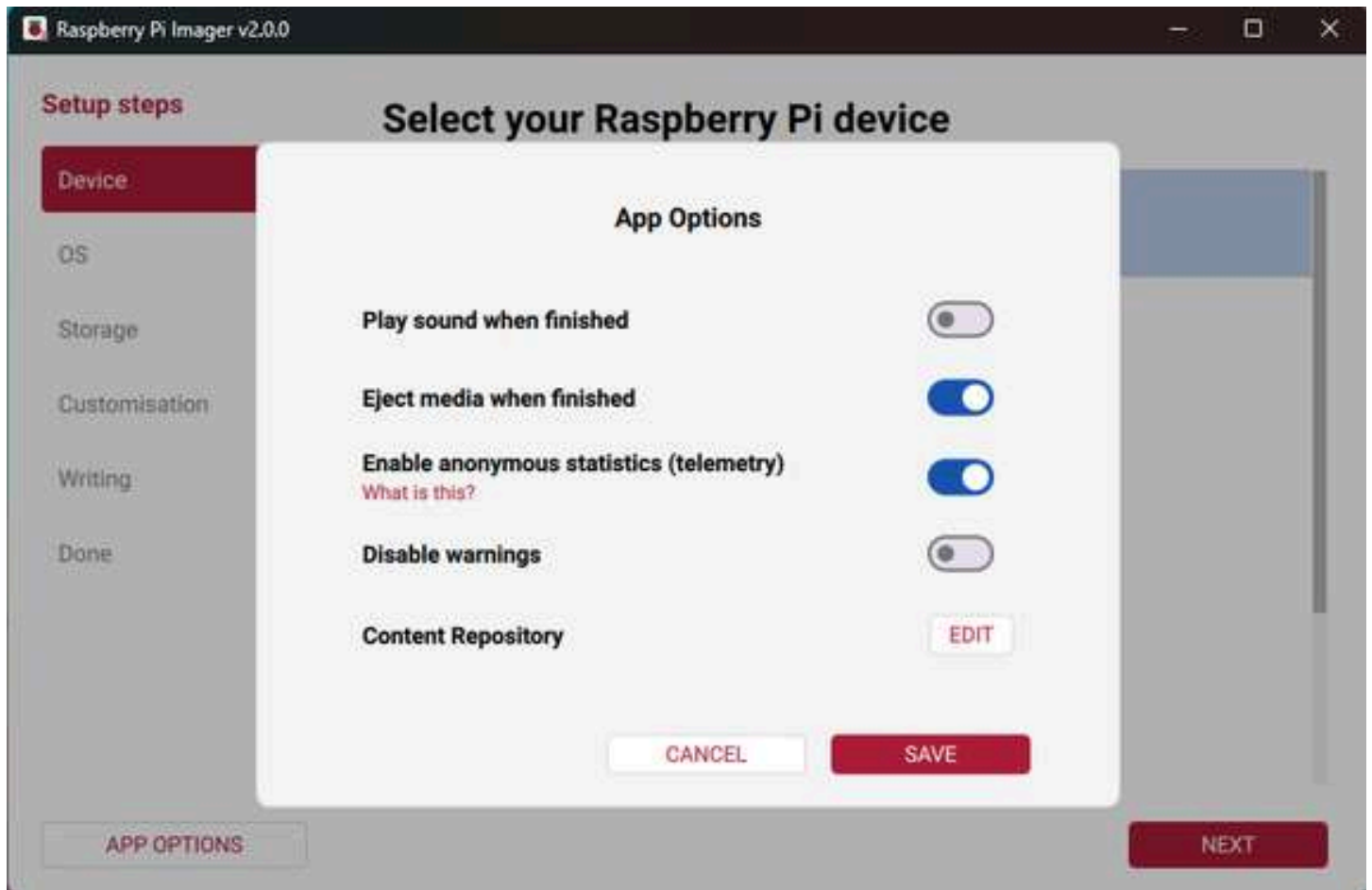
Step 4. Set the Imager options

You can choose to configure the behaviour of Imager during a write. Do this at any point before you write the image by selecting App Options.

In the App Options dialog box you can configure the following options:

- Whether Imager plays an alert sound when the write finishes.
- Whether Imager ejects the storage media when the write finishes.
- Whether Imager sends anonymous telemetry to us about your device and operating system. For more information, see [Anonymous metrics \(telemetry\)](#).
- Whether Imager shows warnings during the write process.

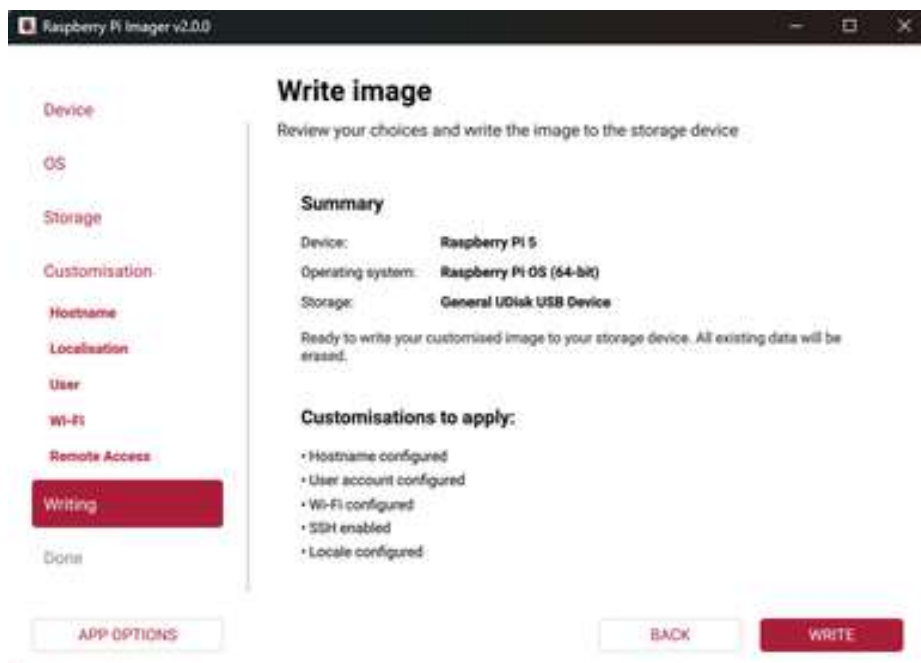
(Advanced users) The App Options dialog box also lets you set a custom repository to retrieve the operating system and devices list from. For more information, see [Imager repo URL](#).



Step 5. Write the image to the storage device

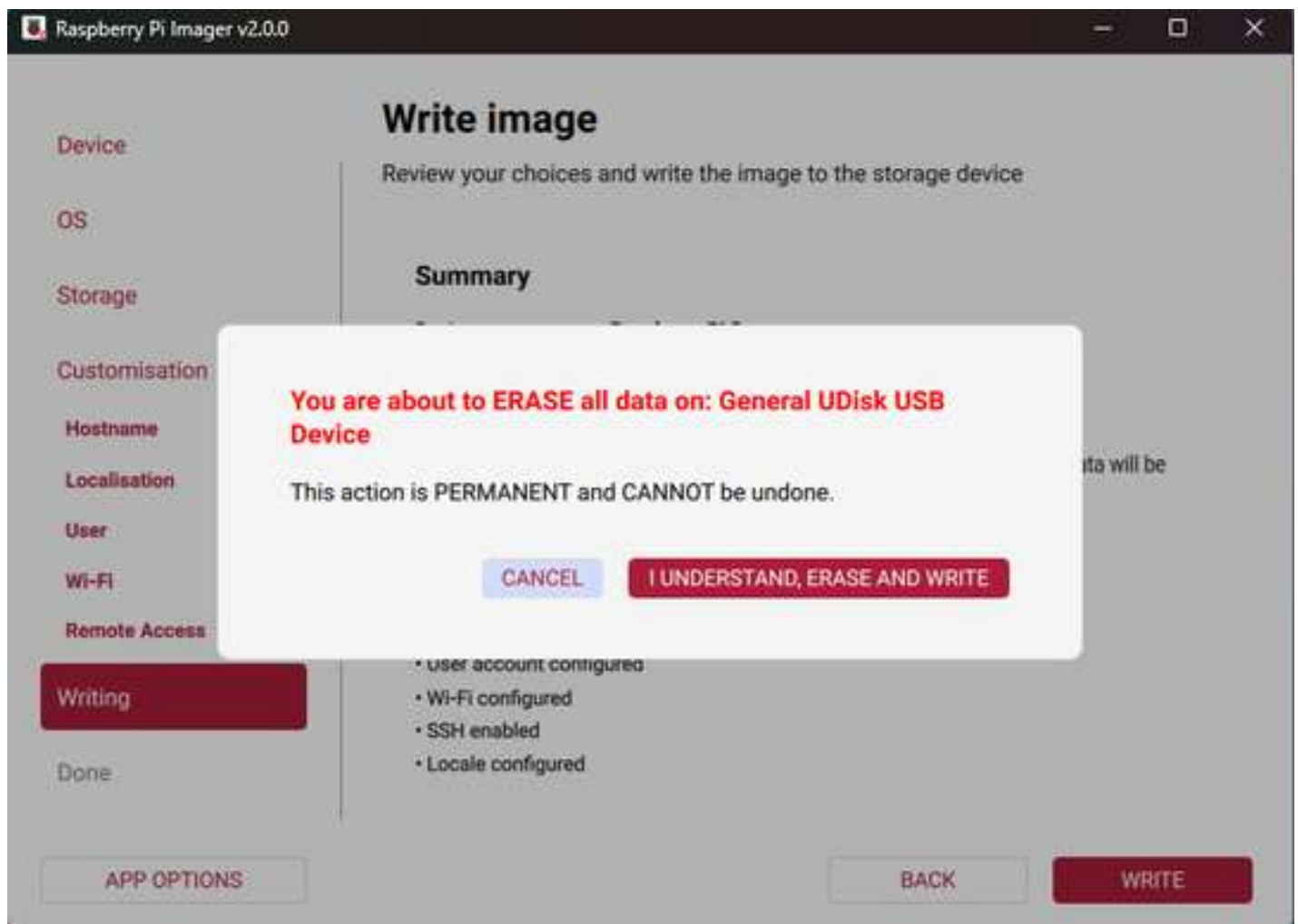
After you've gone through the tabs in Imager, it shows a summary page where you can review your choices.

- If you're happy with your choices and want to proceed, select Write.

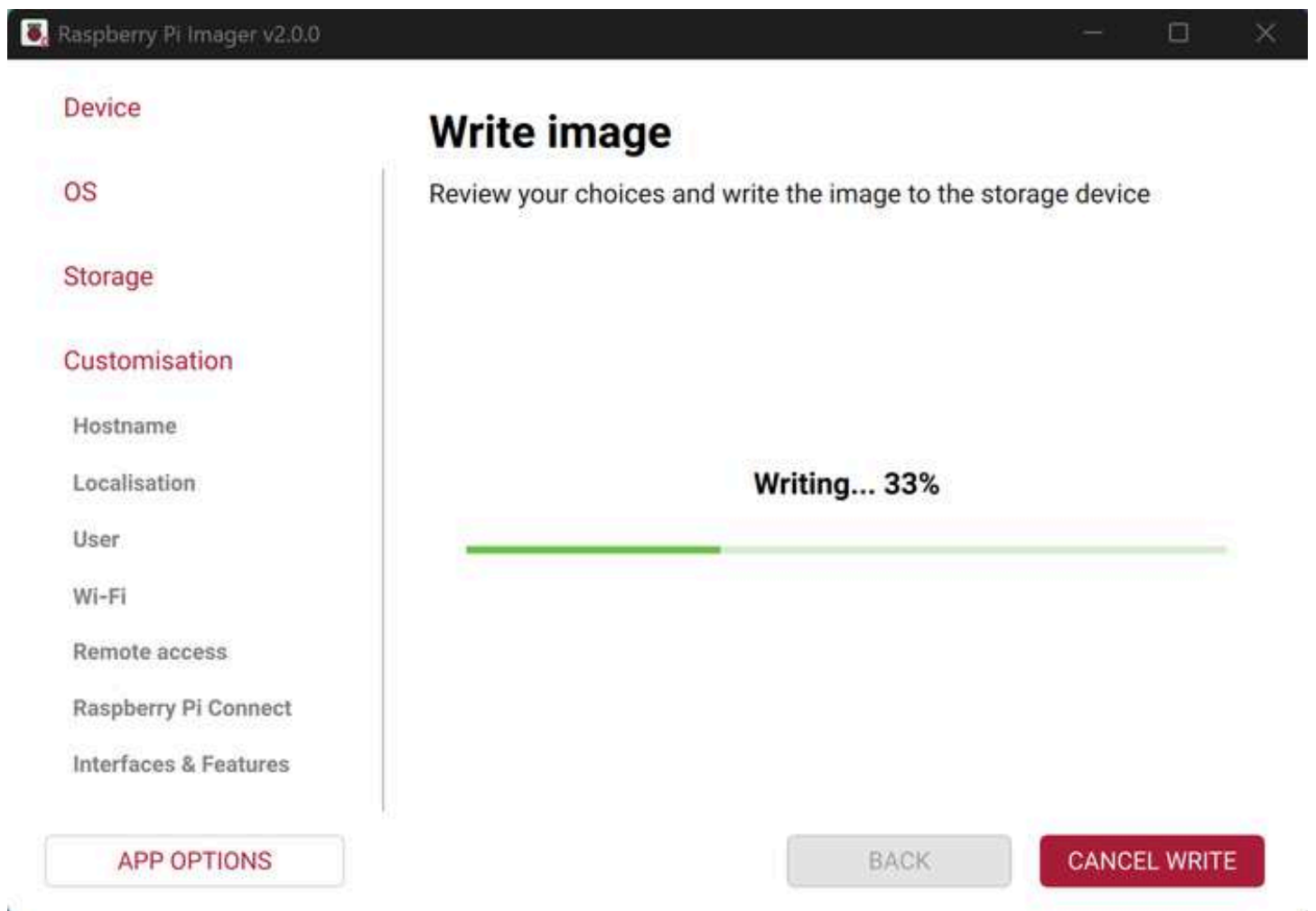


- Imager shows a prompt warning you that you're about to overwrite all data on your storage media. To continue, select I understand, erase and write.

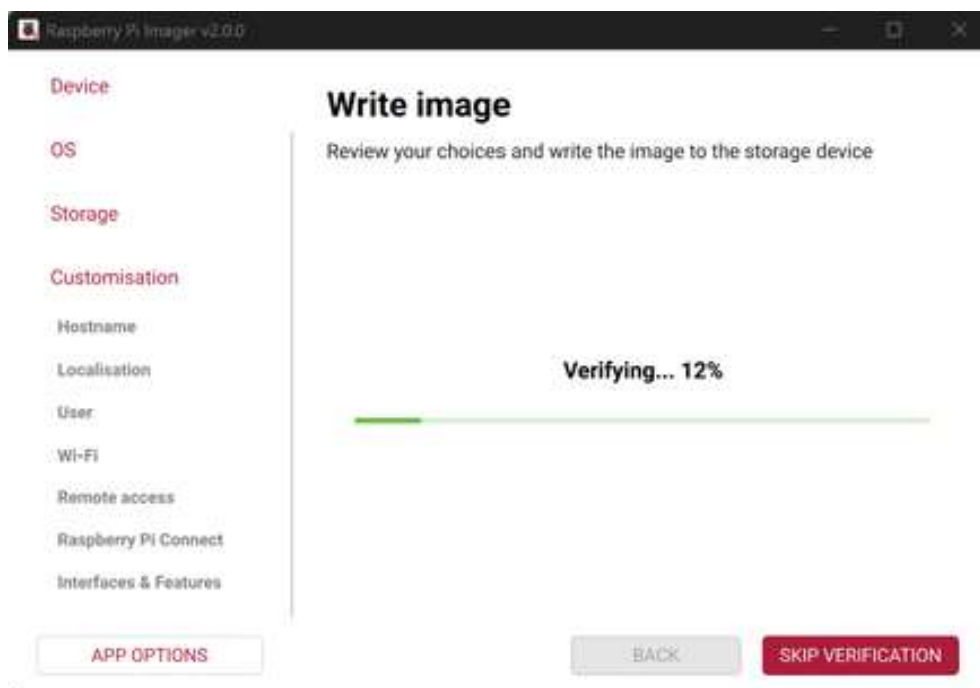
If you see an admin prompt asking for permissions to read and write to your storage medium, grant Imager the permissions to proceed.



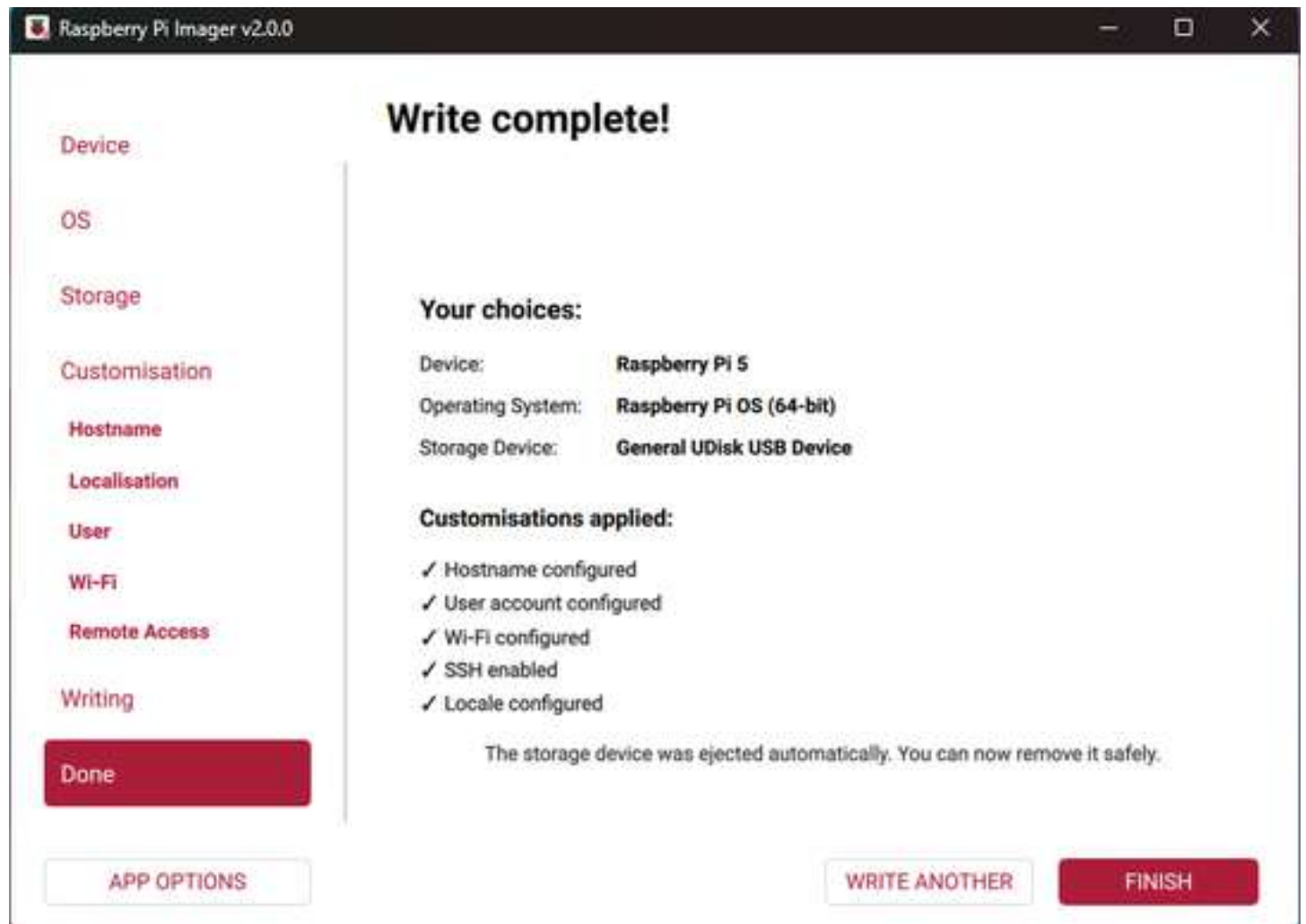
Imager begins writing your image to the storage media. This can take a few minutes; grab a cup of coffee while you wait.



After writing the image, Imager verifies that the image is correctly written. We recommend that you complete the verification step. However, if you want to cancel verification, select Skip verification.

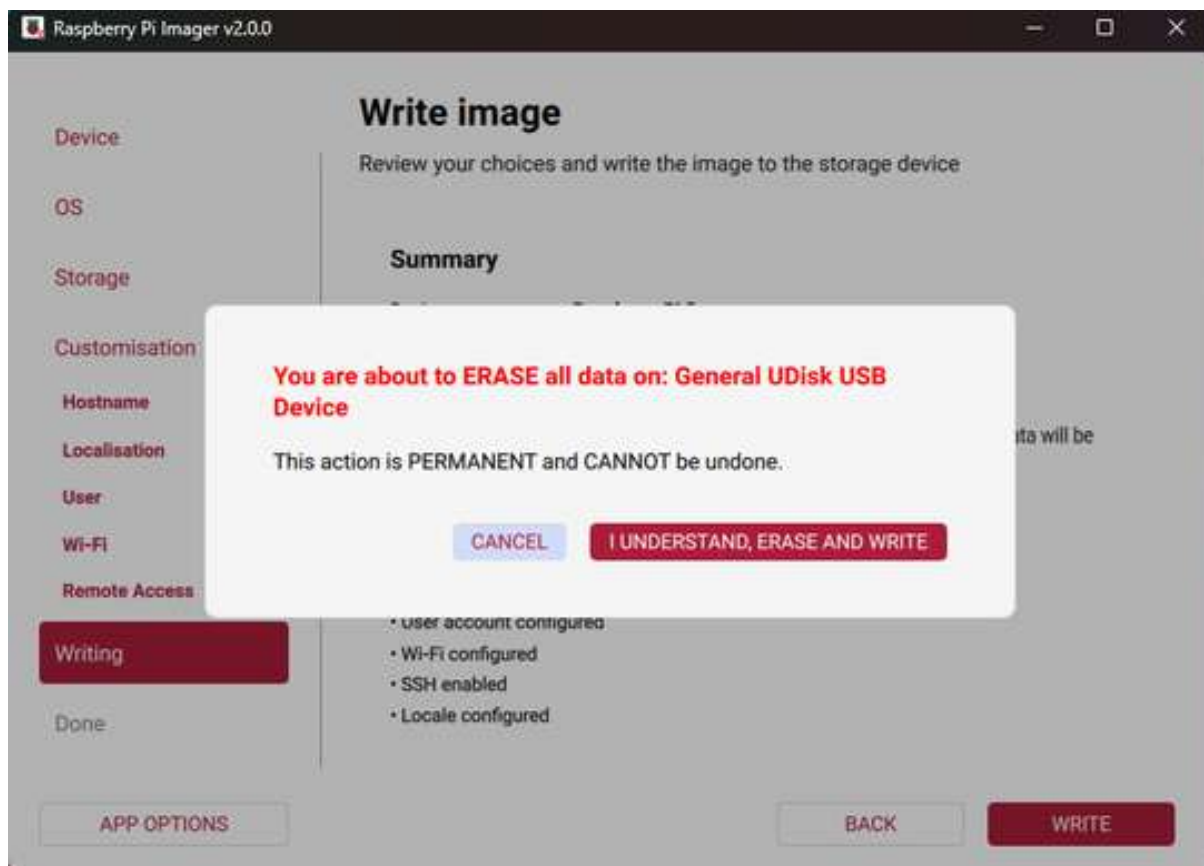


After the image is written and verified, Imager shows the Done tab. Select Finish to exit Imager.



SSG EMBEDDED SOLUTIONS

info@ssges.co.in



After the image is written and verified, Imager shows the Done tab. Select Finish to exit Imager.

Set up your Raspberry Pi

After installing an operating system image, connect your storage device to your Raspberry Pi.

First, unplug your Raspberry Pi's power supply to ensure that the Raspberry Pi is powered down while you connect peripherals. If you installed the operating system on a micro SD card, you can plug it into your Raspberry Pi's card slot now.



Then, plug in any other peripherals, such as your mouse, keyboard, and monitor.



Finally, connect the power supply to your Raspberry Pi. You should see the status LED light up when your Pi powers on. If your Pi is connected to a display, you should see the boot screen within minutes.

You can see These kind of Desktop for raspberrypi 5

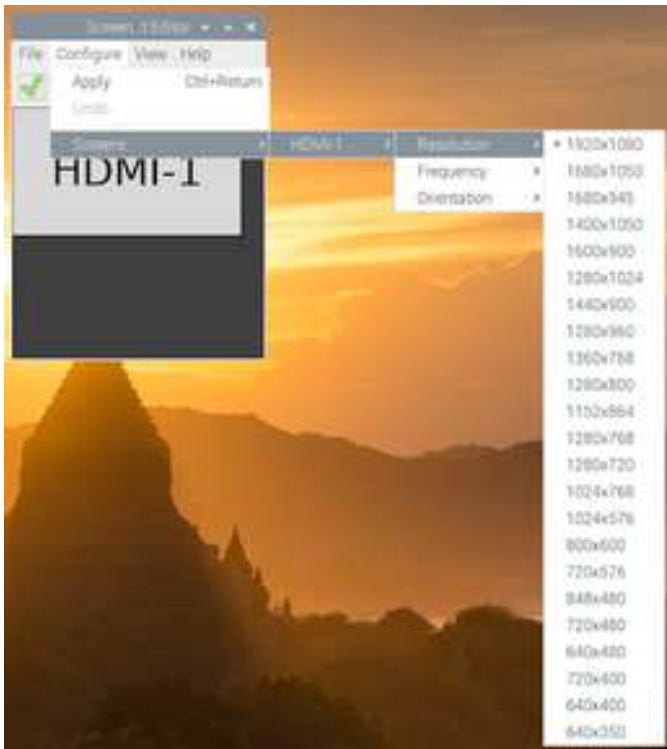


Changing Your Screen Resolution on Raspberry Pi

If you don't have enough desktop real estate, you may want to change your screen resolution to ensure that it matches what your display is capable of. If you are using a headless Pi and accessing it via VNC, you still probably want at least a 720p screen.

To change the Raspberry Pi resolution:

1. Open the Screen configuration menu by clicking on the Pi icon then selecting Preferences -> Screen Configuration.
2. Right Click on the HDMI box and select your Resolution from the Resolution menu.



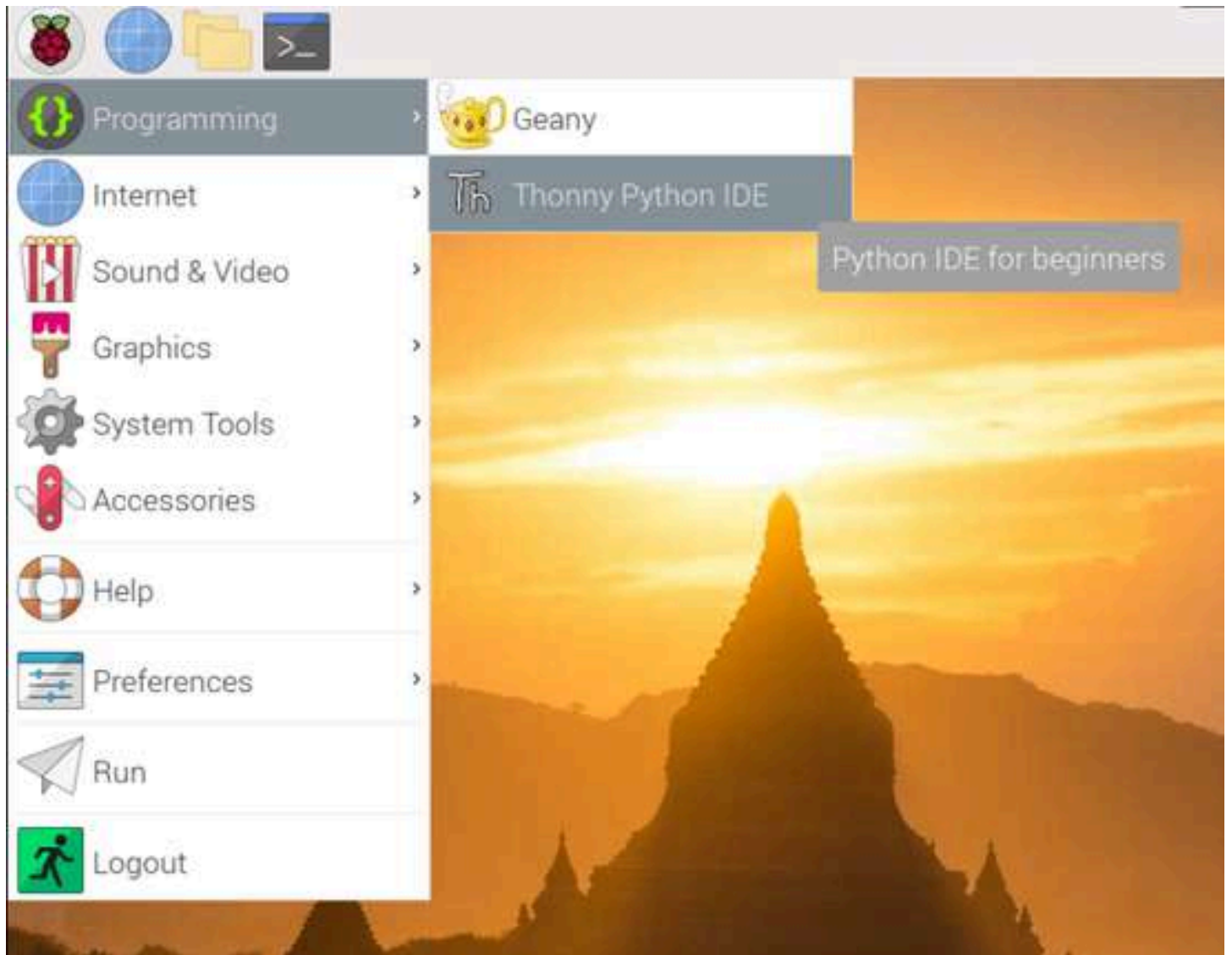
1. Click the Check box. The screen resolution will update.



1. Select yes and reboot.

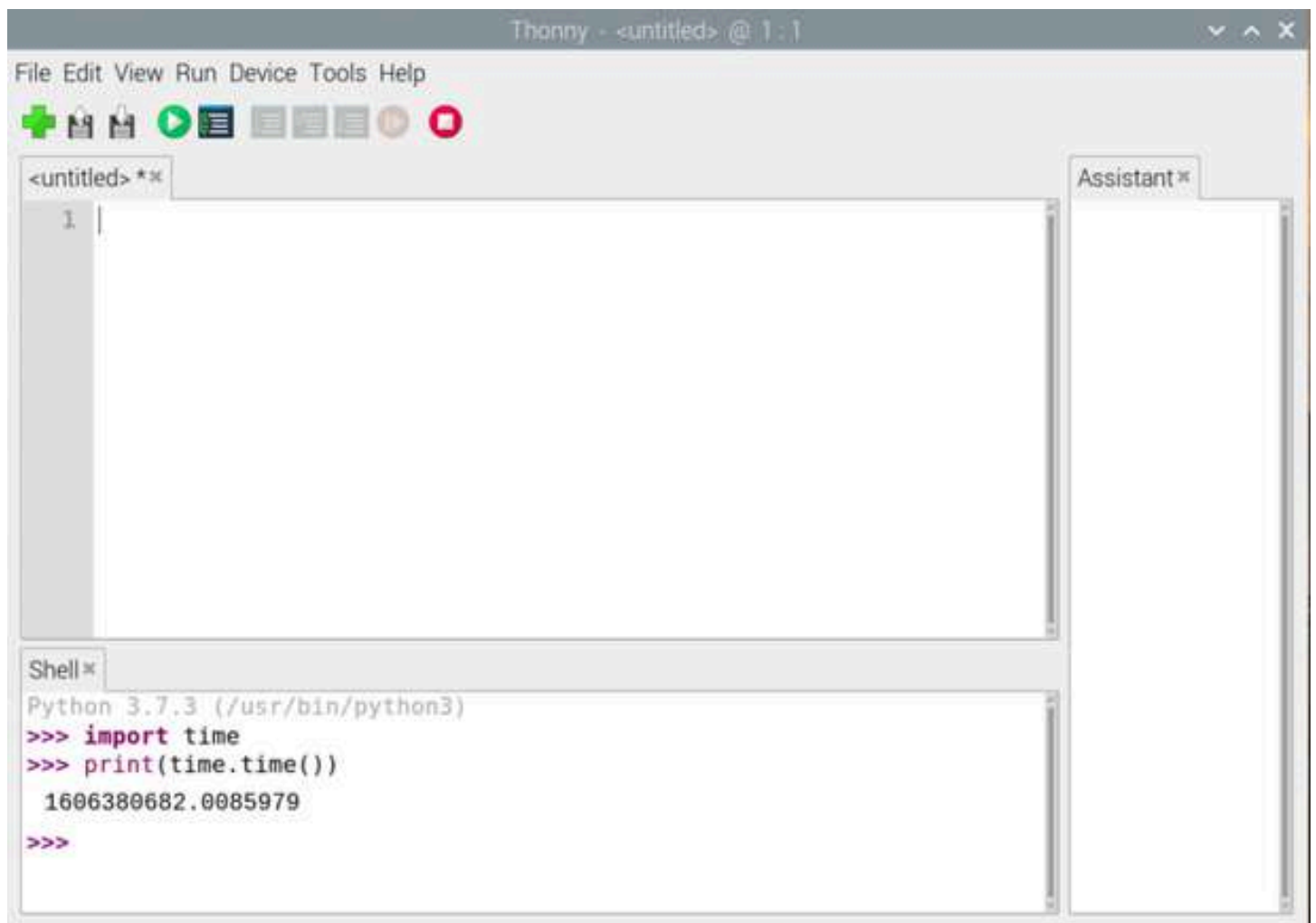
How to start Thonny IDE on Raspberry Pi

Click on the top left icon representing a raspberry > Programming > Thonny Python IDE.



You don't need to configure anything related to Python. Just click on Thonny Python IDE and you can start programming.

As Python is an interpreted language, you can directly run Python commands one by one. To do that you'll need to open a special environment (named Python shell, or Python interpreter). But in Thonny IDE, nothing to do, the shell is already here.

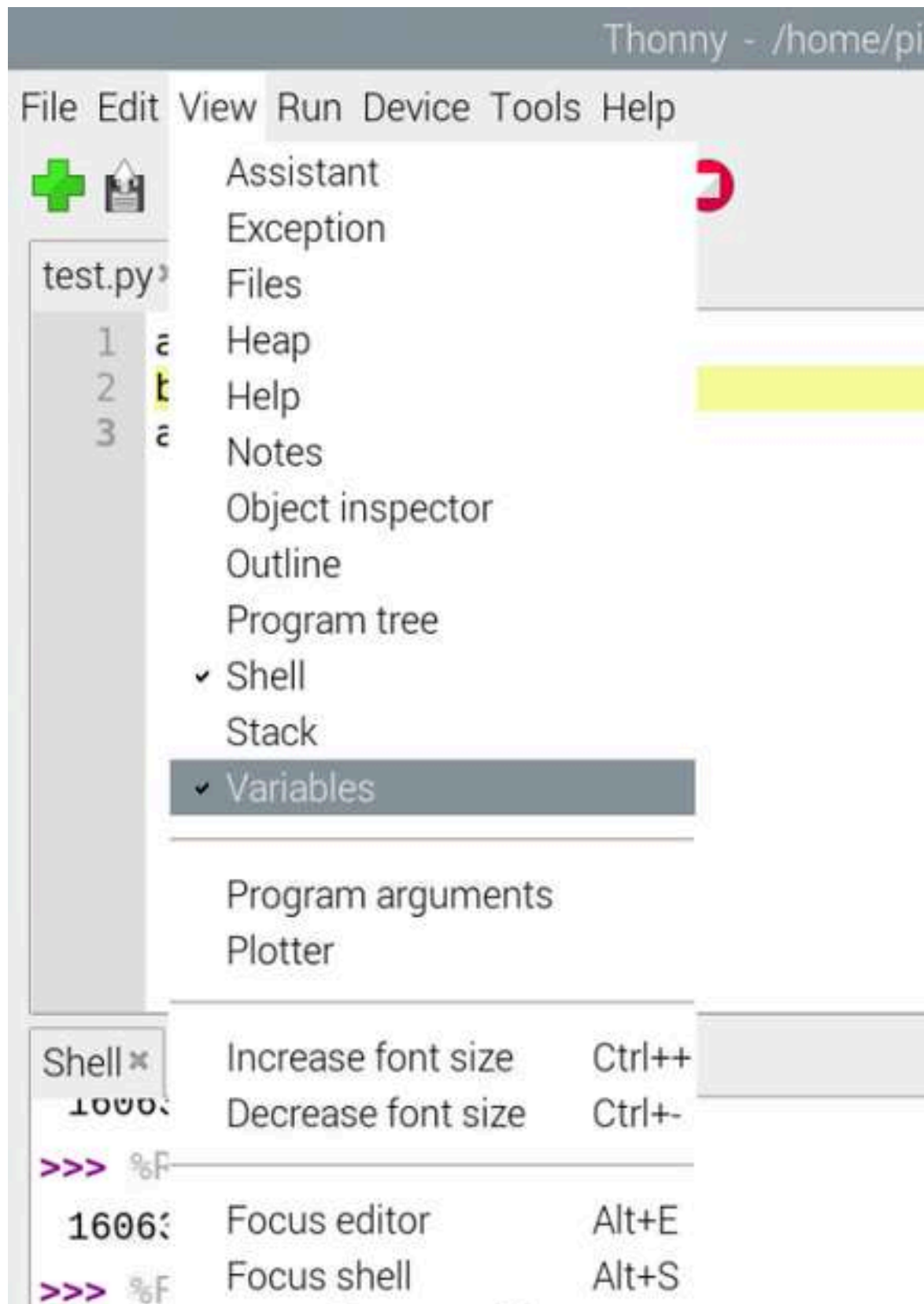


SSG EMBEDDED SOLUTIONS

info@ssges.co.in

Thonny IDE debugger

First, click on “View”. If you want to be able to see what are the values inside the variables of your program, make sure to click on “Variables”.



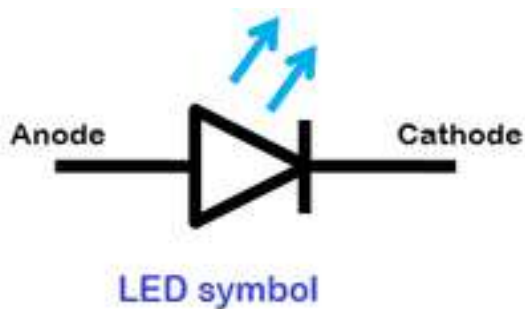
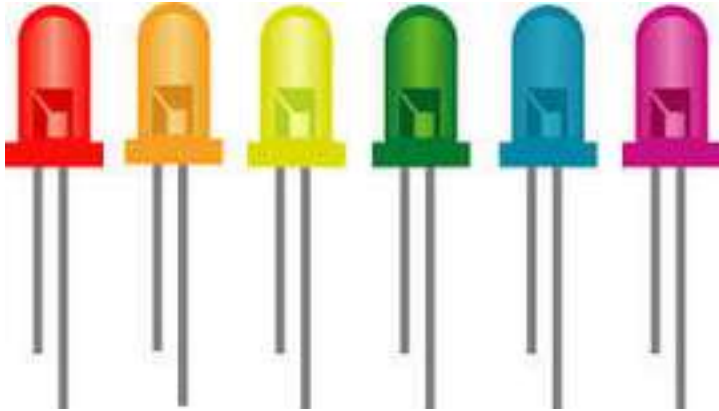
Then, click on the icon next to the “play” button to start the execution of your program

SSG EMBEDDED SOLUTIONS

info@ssges.co.in

1. LED

It is most widely used semiconductor which emit either visible light or invisible infrared light when forward biased. Remote controls generate invisible light. A Light emitting diodes (LED) is an optical electrical energy into light energy when voltage is applied.



These are the applications of LEDs:

- Digital computers and calculators.
- Traffic signals and Burglar alarms systems.
- Camera flashes and automotive heat lamps
- Picture phones and digital watches.

Connections Of Raspberry pi with LED

Raspberry Pi 5	LED
GPIO 17	Y
GPIO 18	R
GPIO 27	G
GPIO 22	B

Prepare the Software Environment

On the Raspberry Pi 5, you must use a Virtual Environment (venv) to run Python scripts that interact with hardware. Open the **Terminal** and run:

1. Create a virtual environment named 'env'

```
python -m venv env
```

2. Activate the virtual environment

```
source env/bin/activate
```

3. Install the necessary libraries for Pi 5 hardware

```
pip install gpiozero lgpio
```

CODE

```
from gpiozero import LED
```

```
from time import sleep
```

```
# Re-defining pins clearly
```

```
# If your Blue LED is on Pin 15, that is GPIO 22
```

```
led_y = LED(18)
```

```
led_r = LED(17)
```

```
led_g = LED(27)
```

```
led_b = LED(22)
```

```
leds = [led_y, led_r, led_g, led_b]
```

```
names = ["YELLOW", "RED", "GREEN", "BLUE"]
```

```
print("Starting Sequential Test. Each LED should blink 3 times.")
```

```
try:
```

```
    for i in range(len(leds)):
```

```
        print(f"Testing: {names[i]}")
```

```
        for _ in range(3):
```

```
            leds[i].on()
```

```
            sleep(0.5)
```

```
            leds[i].off()
```

```
            sleep(0.5)
```

```
print("Sequential test complete. Now cycling all...")
```

```
while True:
```

```
    for l in leds:
```

```
        l.on()
```

```
        sleep(0.2)
```

```
        l.off()
```

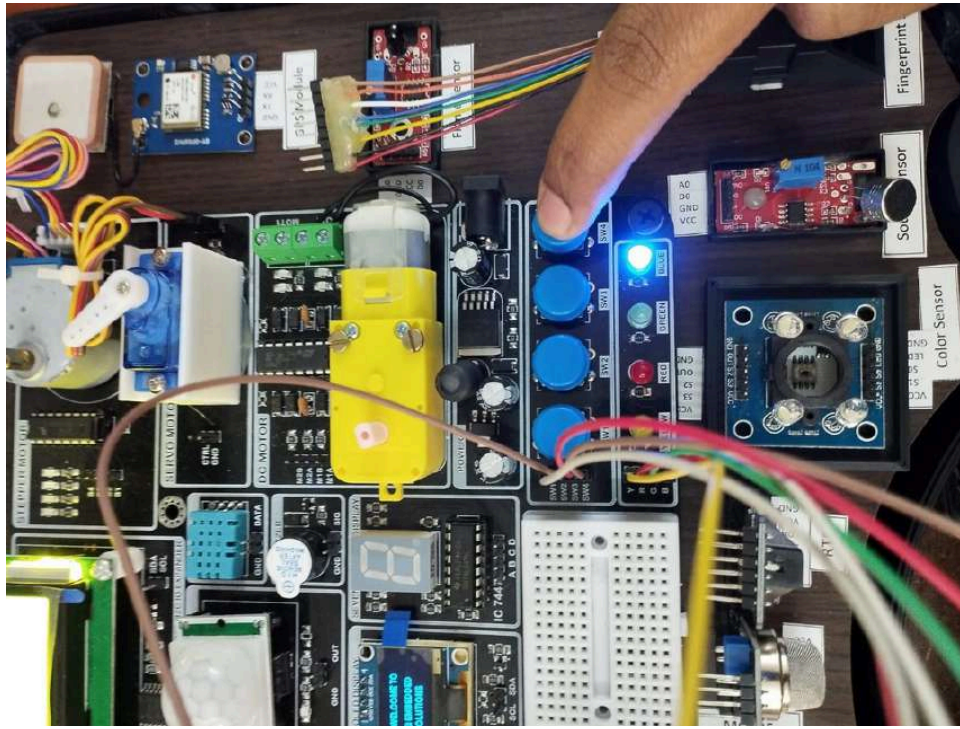
```
except KeyboardInterrupt:
```

```
    print("\nTest stopped.")
```

```
    # Ensure everything is off when we exit
```

```
    for l in leds:
```

l.off()



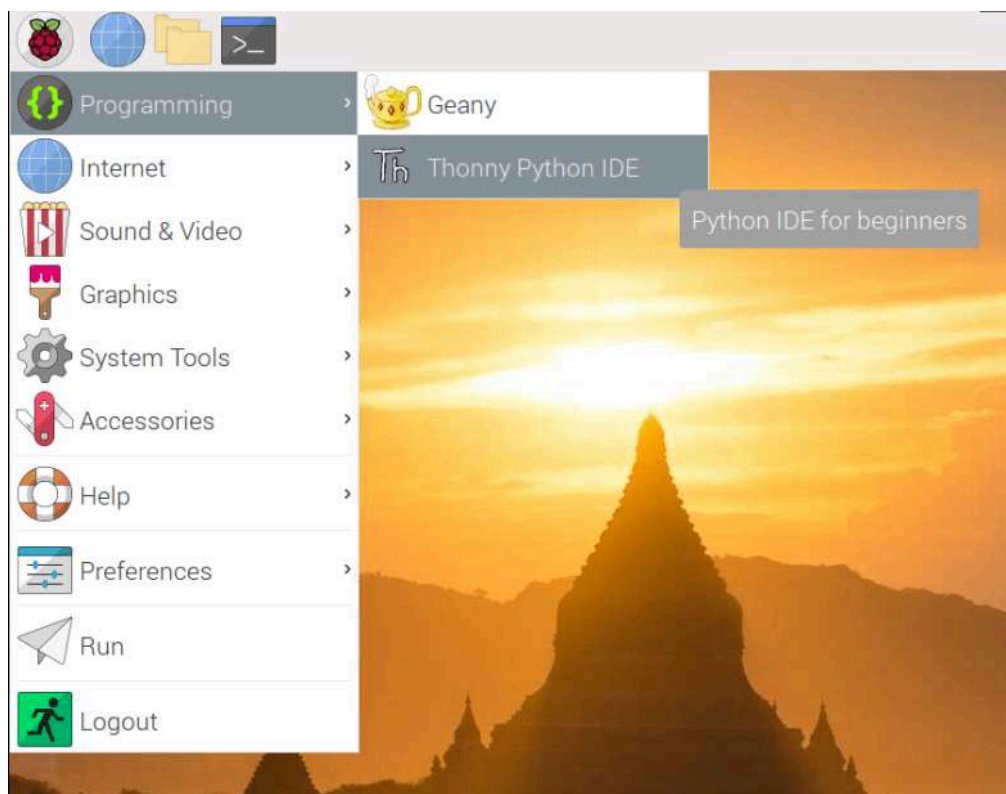
SSG EMBEDDED SOLUTIONS

info@ssges.co.in

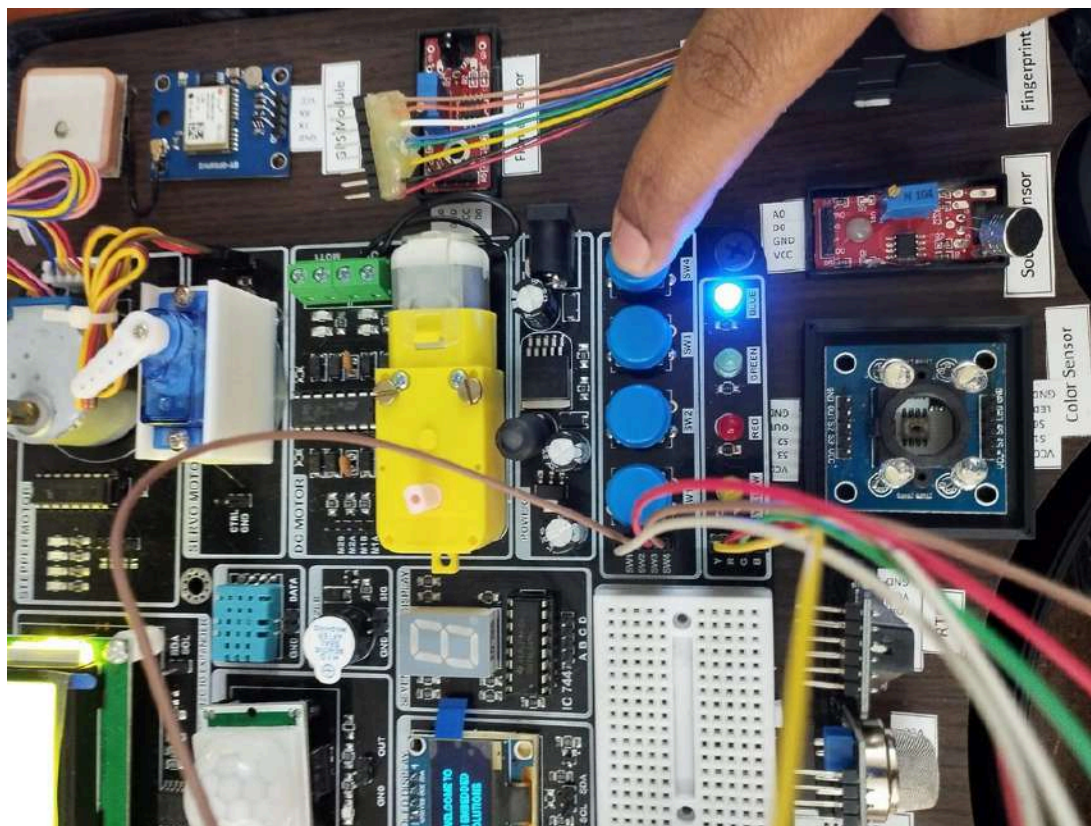
2. LED WITH SWITCH

RASP BERRY PI	SWITCH	LED
GPIO 17	23	4
GPIO 18	24	17
GPIO 22	25	27
GPIO 27	5	22

Click on the top left icon representing a raspberry > Programming > Thonny Python IDE.



Write complete Python programs on Thonny IDE or your codes



SSG EMBEDDED SOLUTIONS

info@ssges.co.in

3. OLED

An OLED (Organic Light-Emitting Diode) display is a screen technology where each pixel produces its own light using carbon-based organic materials, eliminating the need for a separate backlight, resulting in thinner screens, true blacks, superior contrast, vibrant colors, and faster response times, making them ideal for TVs, smartphones, and monitors.



OLED displays consist of organic semiconductor compounds that emit a bright light on the passage of electric current through them, and hence it is termed as OLED. Since, OLED displays can emit light on their own, thus they are considered as self-emissive types of display. There is no need of backlight panel with LEDs to illuminate the screen.

The primary advantages of OLED displays include better picture quality, relatively wider viewing angles, greater flexibility in design, compact size, faster response time, and low power consumption.

Connections Of Raspberry Pi with OLED

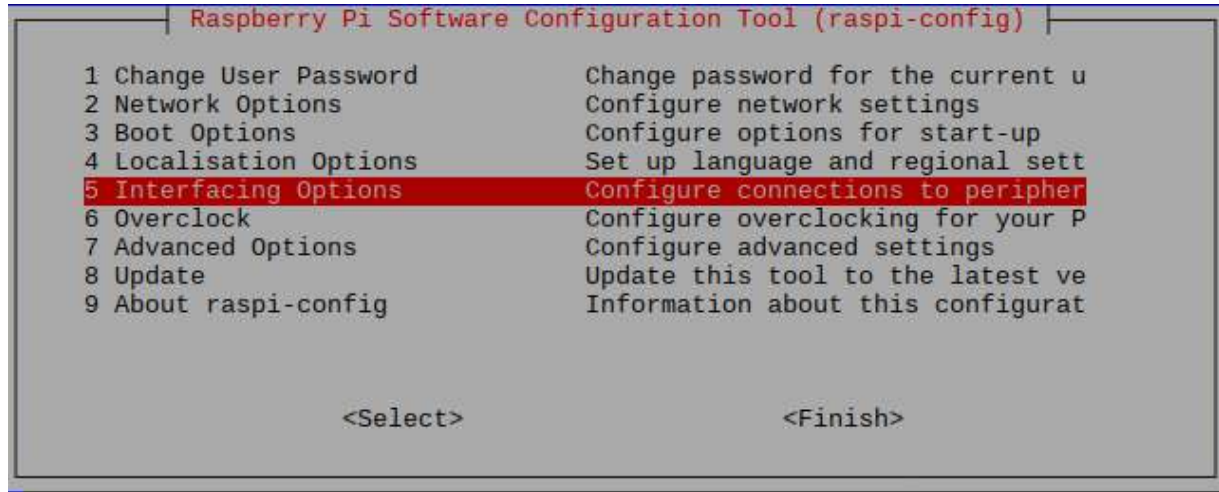
RASPBERRY PI	OLED
GPIO 3 SDA	SDA
GPIO 3 SCL	SCL

Step 1: Enable the I2C Interface

The Raspberry Pi 5 has the I2C interface disabled by default. You must enable it to communicate with the display.

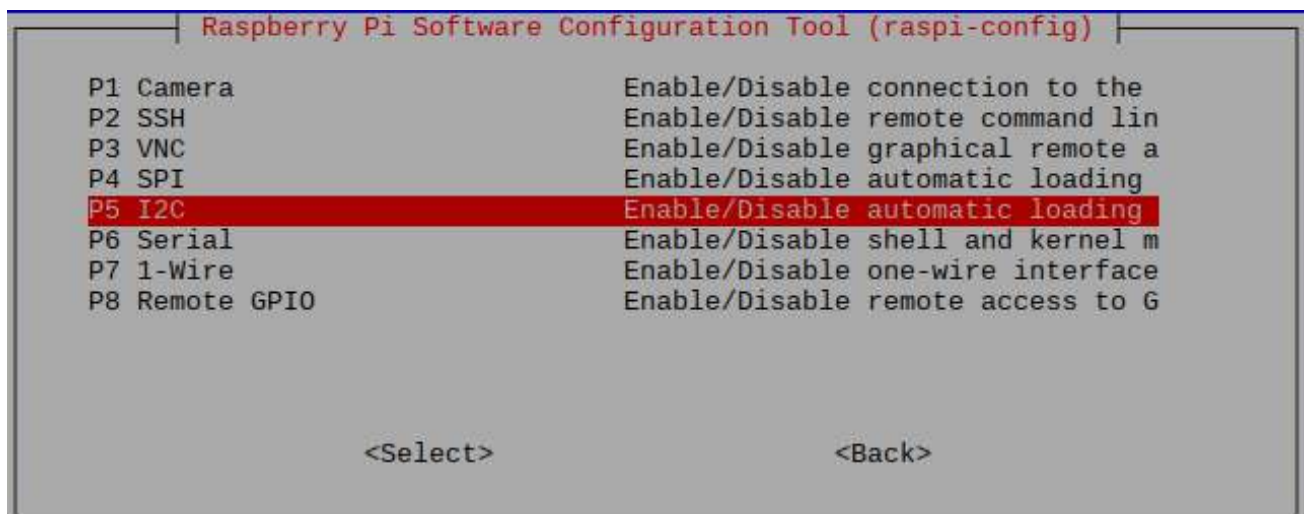
- **Open the terminal and run:**

```
sudo raspi-config
```

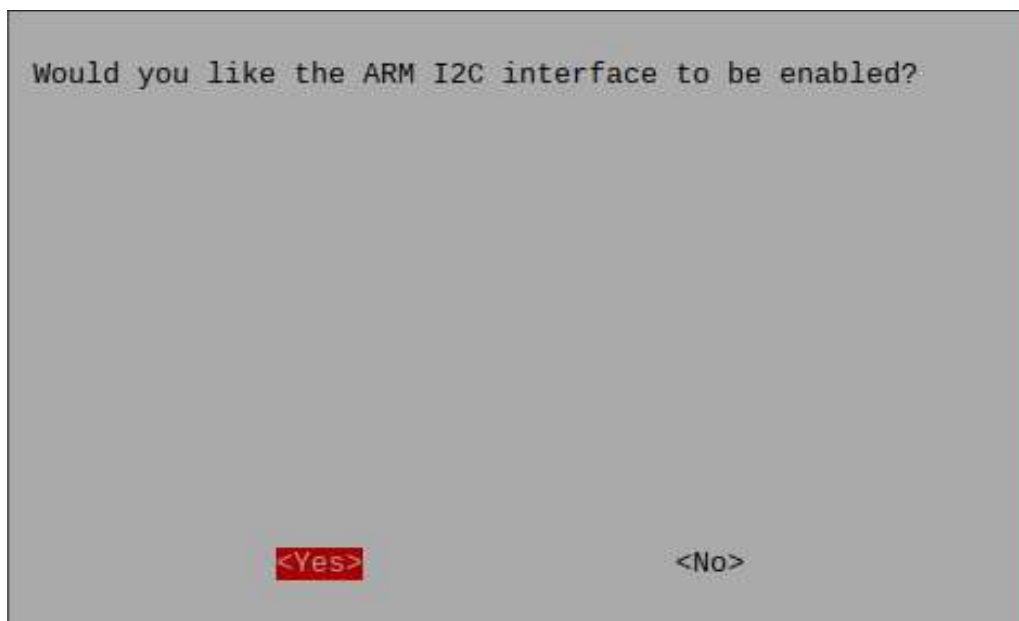


Navigate to Interface Options.

Highlight the “I2C” option and activate “<Select>”.



Select and activate “<Yes>” :



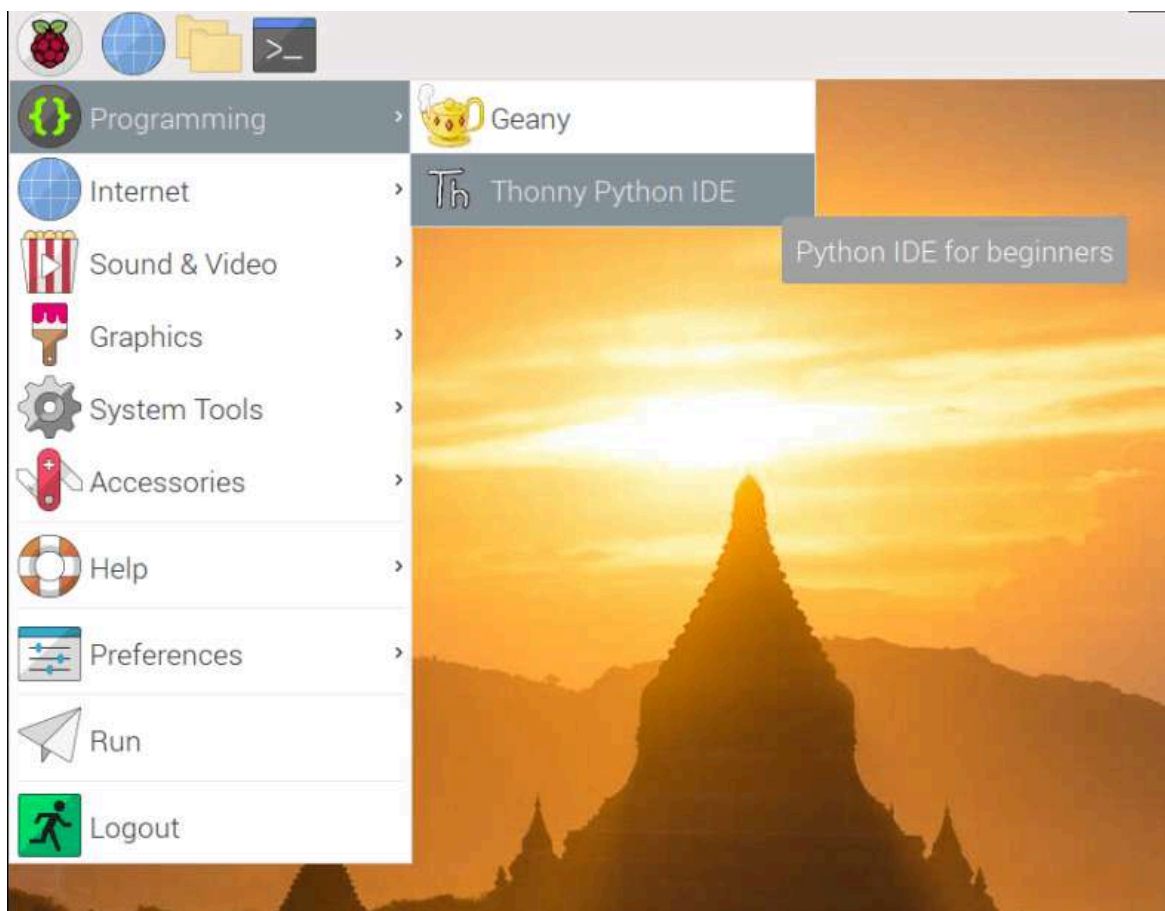
Highlight and activate “<Ok>” :

When prompted to reboot highlight and activate “<Yes>” :

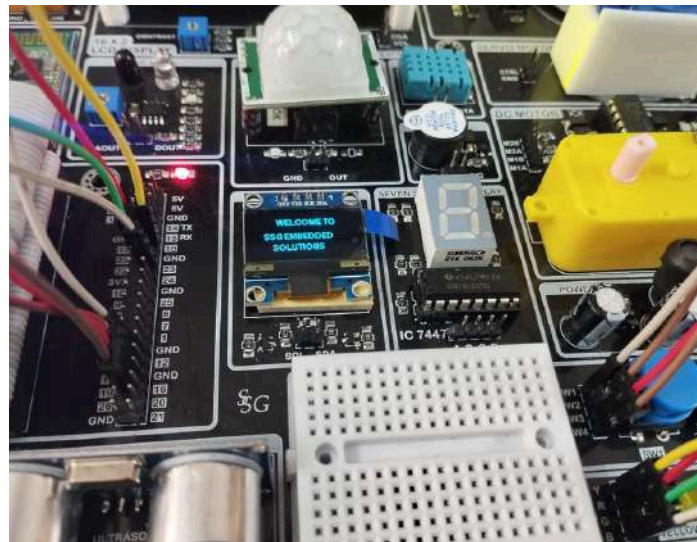


The Raspberry Pi will reboot and the interface will be enabled.

- Click on the top left icon representing a raspberry > Programming > Thonny Python IDE.

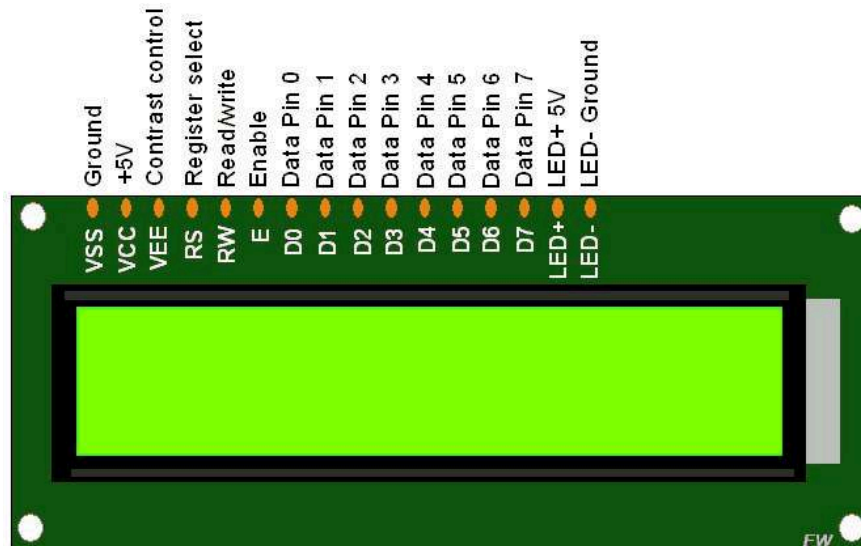


- Now Run your code in the Terminal Window



3. LCD (Liquid Crystal Display)

An LCD (Liquid Crystal Display) is a flat-panel screen technology that uses liquid crystals to control light and create images, found in TVs, monitors, phones, and calculators, known for being thin, lightweight, and energy-efficient compared to older CRT screens.



The features of this LCD mainly include the following.

- The operating voltage of this LCD is 4.7V-5.3V
- It includes two rows where each row can produce 16-characters.
- The utilization of current is 1mA with no backlight
- Every character can be built with a 5×8 pixel box
- The alphanumeric LCDs alphabets & numbers
- Is display can work on two modes like 4-bit & 8-bit

Connections Of Raspberry Pi 5 with LCD

RASPBERRY PI 5	LCD
GPIO 3 SDA	SDA
GPIO 2 SCL	SCL

Step 1: Enable the I2C Interface

- Open the terminal: `sudo raspi-config`

- Go to Interface Options -> I2C.
- Select Yes to enable it.
- Finish and exit.

Step 2: Set Up the Environment & Library

On the Raspberry Pi 5, you must install the rpi-lcd library within a virtual environment.

Create and activate a virtual environment python

```
-m venv env
```

Install the specific LCD library

```
pip install rpi-lcd
```

Step 3: Verify the I2C Address

Before running the code, confirm that your LCD is detected at the address 0x27

```
sudo i2cdetect -y 1
```

If you see "27" in the grid, you are ready to proceed. If you see "3F", you must change the address in your code to 0x3F.

Step 4: Create and Run the Script.

Write code and run it.



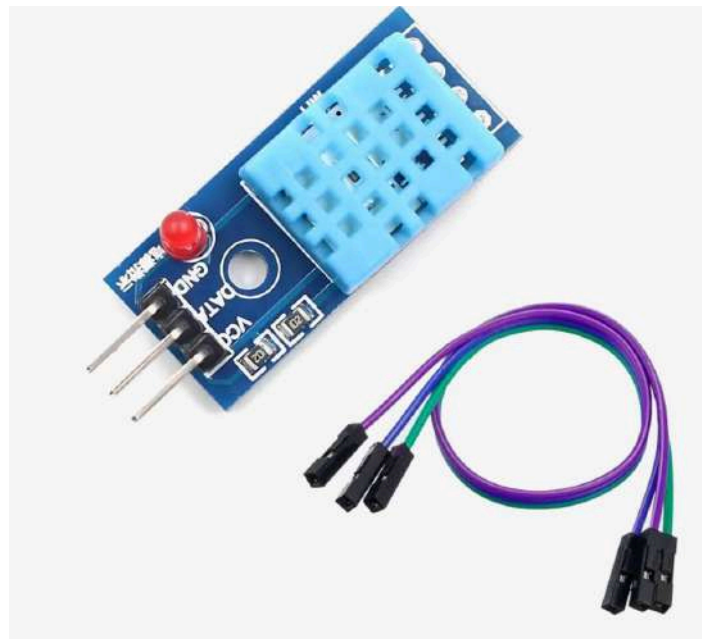
Shell x

```
>>> %Run lcd.py
```

```
LCD Connected Successfully!  
Message displayed on LCD.
```

4. DHT-11

The DHT11 is a low-cost sensor that measures both temperature and humidity. It uses a capacitive humidity sensor and a thermistor to take readings and outputs a single-wire digital signal for easy integration with microcontrollers like the [Arduino](#).



The DHT22 sensor has a better resolution and a wider temperature and humidity measurement range. However, it is a bit more expensive, and you can only request readings with 2 seconds interval. The DHT33 has a smaller range and it's less accurate. However, you can request sensor readings every second. It's also a bit cheaper.

Connections of raspberry pi 5 with DHT11

RASPBERRY PI 5	DHT11
GPIO 17	SIGNAL

Step 2: Install System Dependencies

The DHT11 library on Raspberry Pi 5 relies on a system tool called libgpiod. You must install this via the terminal first.

```
sudo apt-get update
```

```
sudo apt-get install libgpiod2
```

Step 3: Set Up the Virtual Environment

On the Pi 5, all Adafruit libraries must be installed in a virtual environment.

Create and activate a virtual environment

```
python -m venv env
```

Install the DHT library and its requirements

```
pip install adafruit-circuitpython-dht psutil
```

Step 4: Create and Run the Script

5. RELAY

Relays are widely used in control panels, automation systems, and electronic circuits. They allow a low-voltage signal to control higher voltages and currents, making them useful for switching heavy loads safely.

Connections of RaspberryPi with STM32

Raspberry Pi	Relay
GPIO 23	SIGNAL

Step 2: Set Up the Virtual Environment

On the Raspberry Pi 5, the gpiozero library needs the lgpio backend to communicate with the hardware properly.

- Create and activate a virtual environment

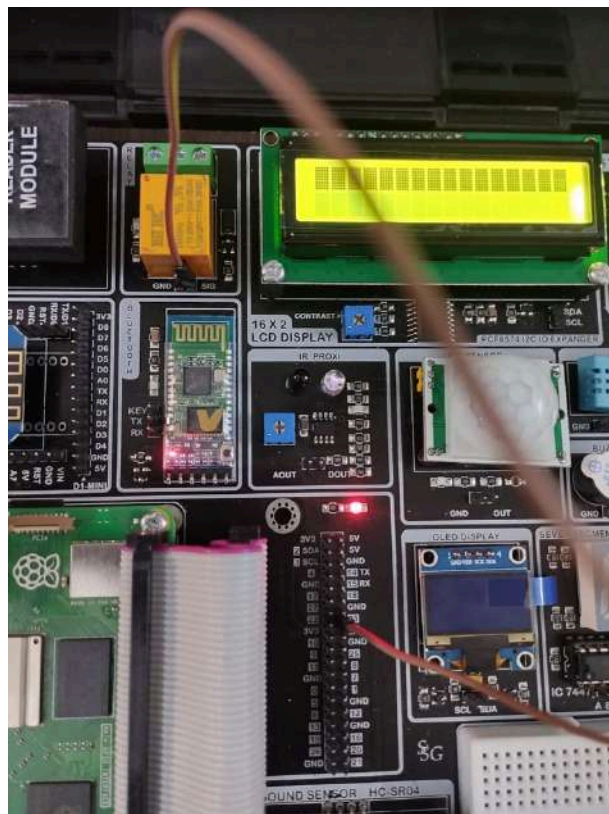
```
python -m venv env
```

- Install the required libraries

```
pip install gpiozero lgpio
```

Step 3: Create and Run the Script

Write Save & Run the code



```
28 relay.off()
29 relay.close() # Important for Pi 5 to release the pi

Shell x

>>> Run relay.py

Relay Control Initialized...
Relay ON - Device Activated
Relay OFF - Device Deactivated
Relay ON - Device Activated
```

6. BUZZER

A buzzer is an electronic device that generates sound by converting electrical energy into sound energy. It typically consists of a piezoelectric crystal, which expands and contracts when an alternating current is applied to it, creating sound waves.

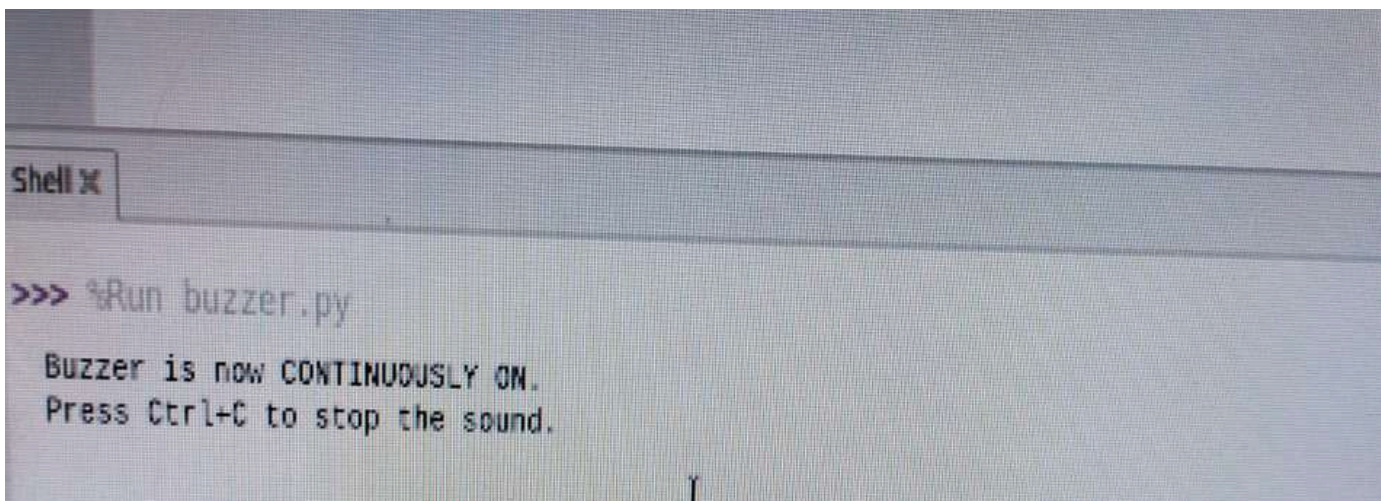
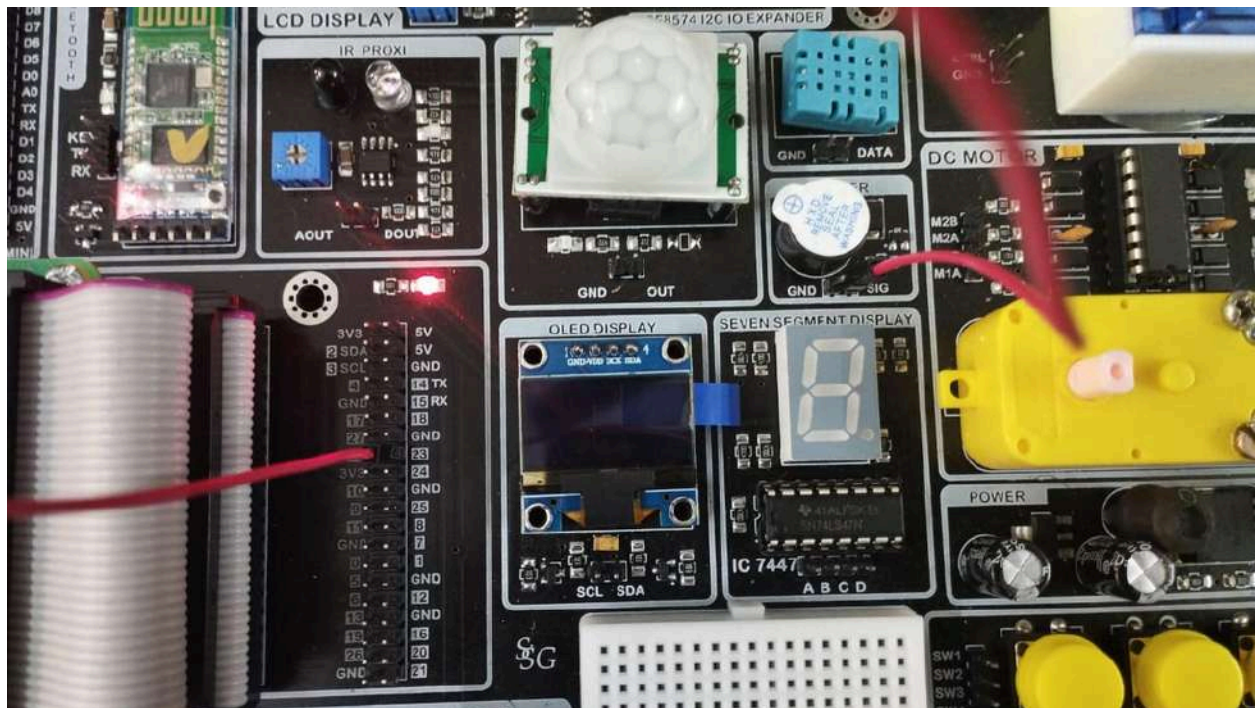


Buzzers are commonly used in a wide range of applications such as alarms, timers, and warning systems. They can also be used in electronic devices such as mobile phones, computers, and other electronic devices to generate different sounds and tones.

Connections of BUZZER with Raspberry Pi

RASPBERRY PI	BUZZER
GPIO 23	SIG

"Please follow the same environment setup and terminal commands (activating the venv, and installing lgpio) as you did for the Relay section earlier."

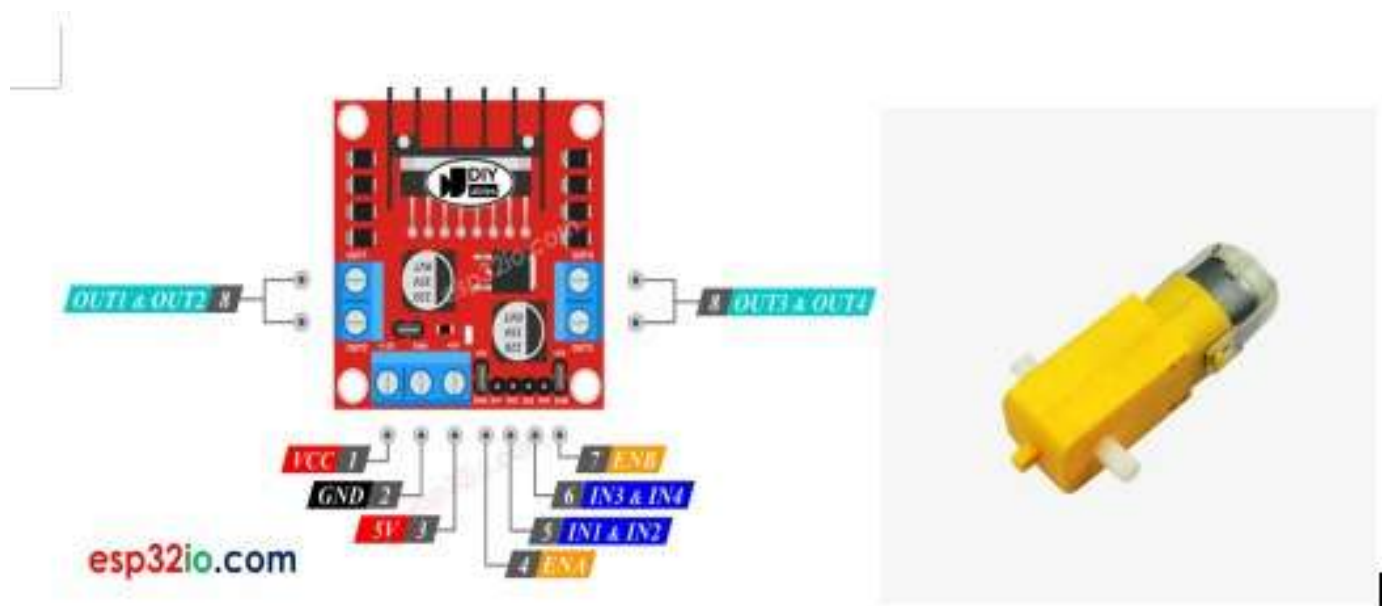


7. DC MOTOR

A DC Motor is an electromechanical device that converts direct current (DC) electrical energy into mechanical rotational energy.

Working Principle

When a current-carrying conductor is placed in a magnetic field, it experiences a mechanical force that makes it rotate. The direction of rotation is determined using Fleming's Left-Hand Rule.



Applications

- Robotics (movement control) Toys and hobby projects
- Quadcopters and drones
- CD/DVD drives in computers and laptops
- Small appliances and automation systems

Connections of DC MOTOR with Raspberry Pi

RASPBERRY Pi	DC MOTOR
GPIO 20	M1B
GPIO 21	M1A

"Please follow the same environment setup and terminal commands (activating the venv, and installing lgpio) as you did for the Relay section earlier."

```
Shell X
Spinning Backward...
Stopping...
Spinning Forward...
Spinning Backward...
Stopping...
Spinning Forward...
Spinning Backward...
Stopping...
Spinning Forward...
Spinning Backward...
Stopping...
Spinning Forward...
Spinning Backward...
Stopping...
Spinning Forward...
Spinning Backward...
Stopping...
```

8. SERVO MOTOR

A Servo Motor is an electrical device used to rotate and position objects precisely, such as in **robotic arms**, **cameras**, and **automation systems**.

It consists of a **DC motor** (sometimes AC motor) combined with a feedback

control system. The built-in error-sensing feedback mechanism continuously

monitors the output position and corrects it, ensuring accurate control of

angular velocity and position.



It is a closed loop system where it uses negative feedback to control motion and final

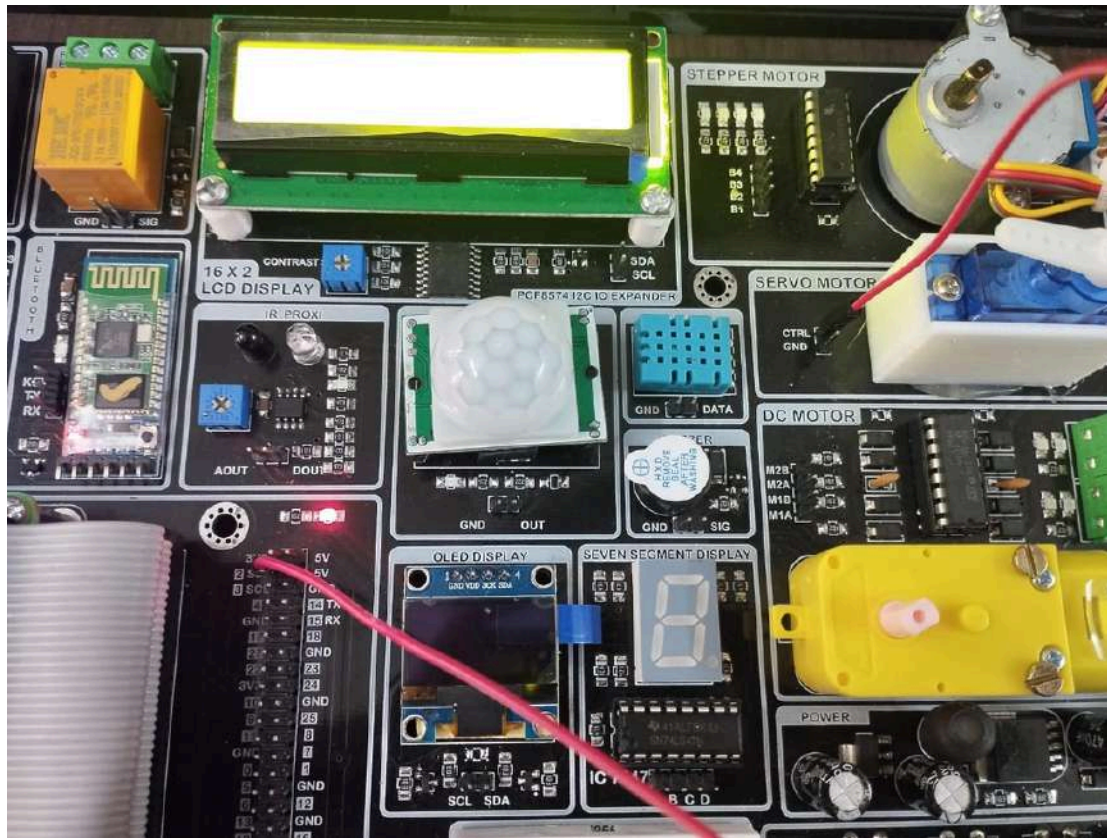
position of the shaft. It is not used for continuous rotation like conventional

AC/DC motors. It has rotation angle that varies from 0° to 360° .

Connections of SERVO MOTOR with RASPBERRY PI

RASPBERRY PI	SERVO MOTOR
GPIO 2	CTRL

"Please follow the same environment setup and terminal commands (activating the venv, and installing lgpio) as you did for the Relay section earlier."



Shell ✕

```
Moving to -1 (0 degrees)
Moving to 0 (90 degrees)
Moving to 1 (180 degrees)
Moving to -1 (0 degrees)
Moving to 0 (90 degrees)
Moving to 1 (180 degrees)
Moving to -1 (0 degrees)
Moving to 0 (90 degrees)
Moving to 1 (180 degrees)
Moving to -1 (0 degrees)
Moving to 0 (90 degrees)
Moving to 1 (180 degrees)
Moving to -1 (0 degrees)
Moving to 0 (90 degrees)
Moving to 1 (180 degrees)
Moving to -1 (0 degrees)
```


9. STEPPER - MOTOR

A Servo Motor is a special type of motor used for precise control of angular position, velocity, and acceleration. It is a closed-loop system that uses negative feedback to continuously adjust and control the motion of its shaft.

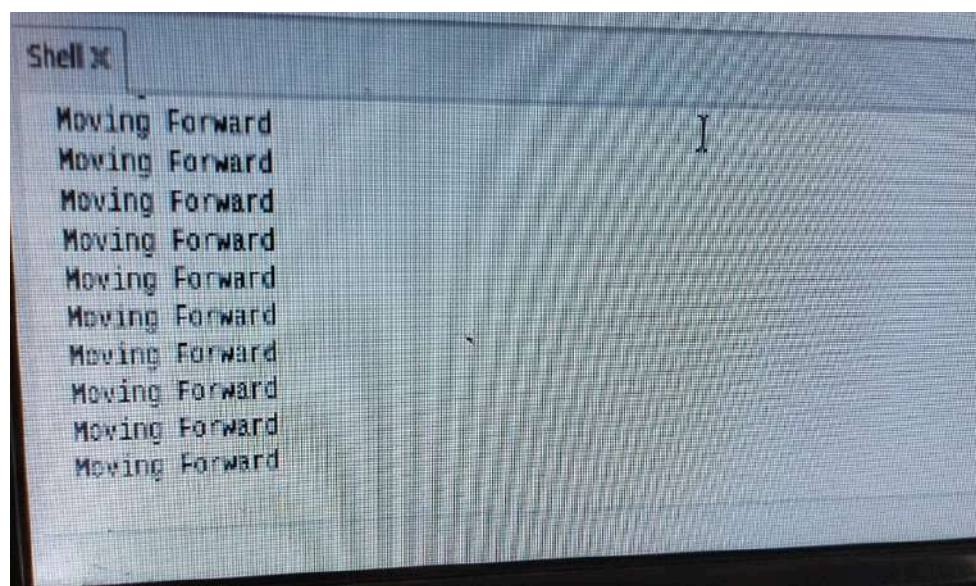
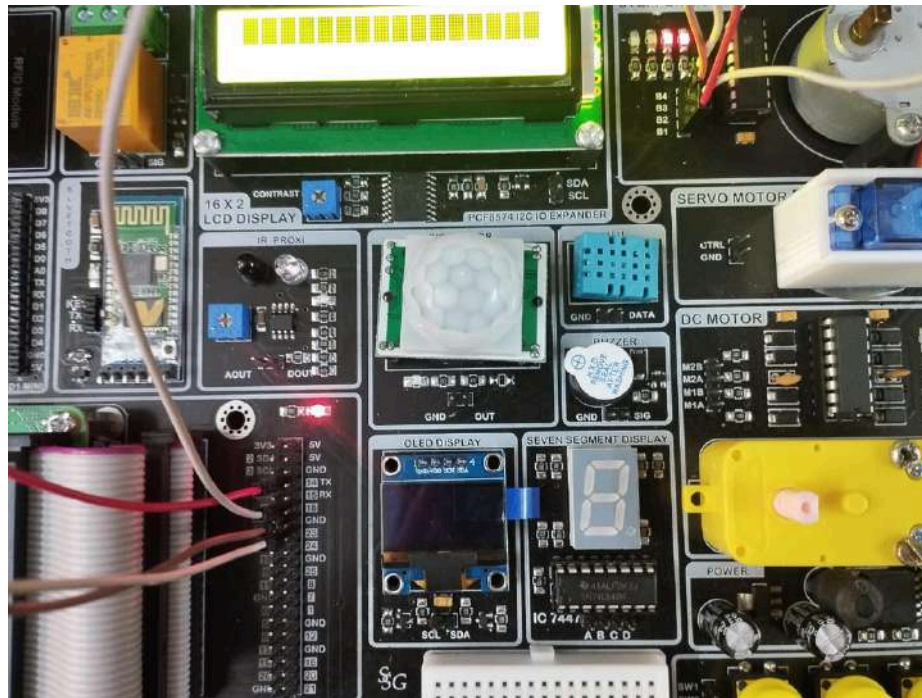


Unlike conventional AC/DC motors, a servo motor is not designed for continuous rotation. Instead, it operates within a limited angle range, typically 0° to 180° (some extended types up to 360° or more).

Connections of STEPPER MOTOR with RASPBERRY PI

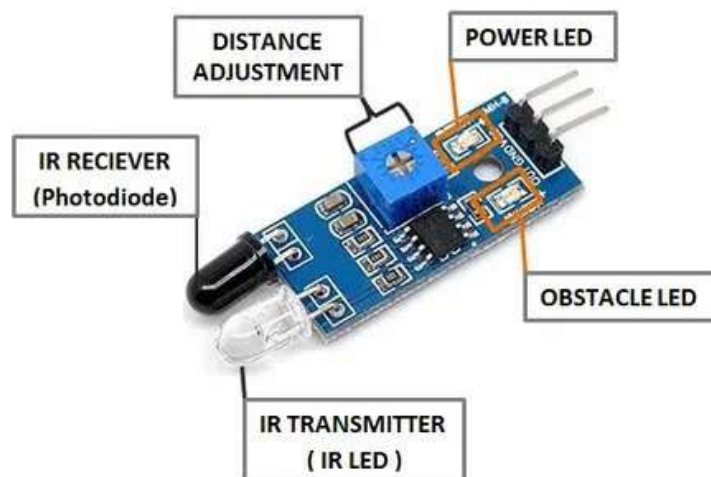
RASPBERRY PI	STEPPER MOTOR
GPIO 4	B1A
GPIO 17	B2A
GPIO 27	B3A
GPIO 22	B4A

"Please follow the same environment setup and terminal commands (activating the venv, and installing lgpio) as you did for the Relay section earlier."



10. IR SENSOR

An Infrared (IR) Sensor is an electronic device that uses infrared light to detect objects, measure heat, or sense motion in its surroundings. All objects emit some form of thermal radiation in the infrared spectrum, which is invisible to the human eye but can be detected by IR sensors.



An IR sensor mainly consists of:

IR LED (Emitter): Emits infrared light.

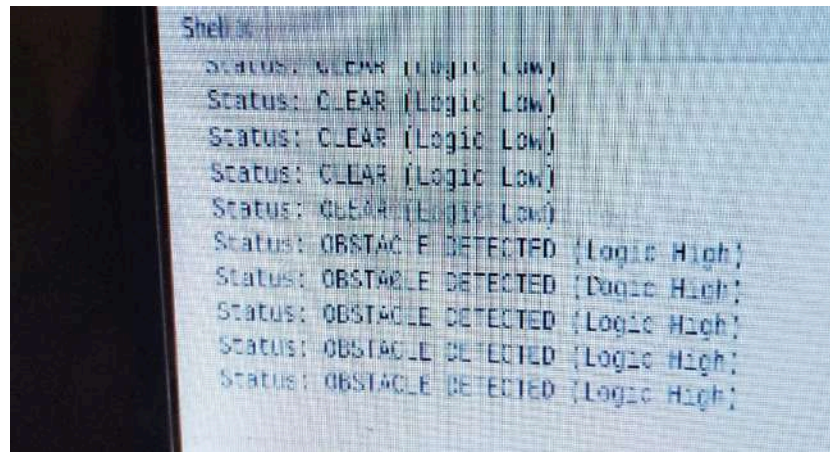
- IR Photodiode (Detector): Detects the reflected IR light. Its resistance and
- output voltage change depending on the amount of IR light received.

This combination allows IR sensors to detect objects, measure distance (basic level), and sense motion.

Connections of IR SENSOR with RASPBERRY PI

RASPBERRY PI	IR SENSOR
GPIO 4	DOUT

"Please follow the same environment setup and terminal commands (activating the venv, and installing lgpio) as you did for the Relay section earlier."

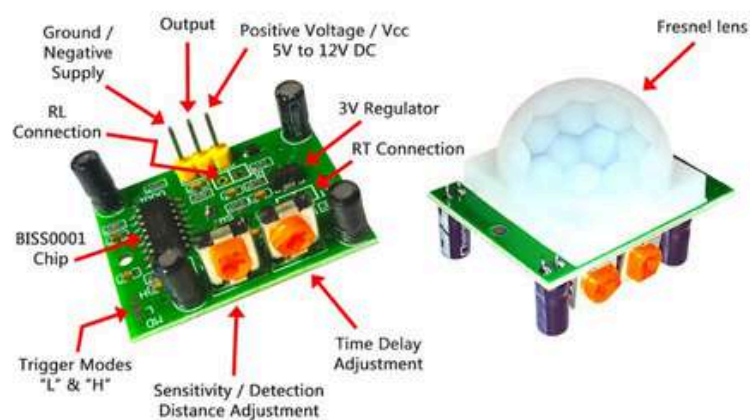


```
Shell %  
Status: CLEAR (Logic Low)  
Status: CLEAR (Logic Low)  
Status: CLEAR (Logic Low)  
Status: CLEAR (Logic Low)  
Status: CLEAR (Logic Low)  
Status: OBSTACLE DETECTED (Logic High)  
Status: OBSTACLE DETECTED (Logic High)  
Status: OBSTACLE DETECTED (Logic High)  
Status: OBSTACLE DETECTED (Logic High)  
Status: OBSTACLE DETECTED (Logic High)
```

11. PIR SENSOR

A PIR sensor is an electronic device that detects infrared (IR) radiation emitted by objects. It is mainly used in motion detection systems, security alarms, and automatic lighting applications. PIR sensors are popular because they are

simple, low-cost, and reliable for detecting human presence or movement.



Pin Configuration (3 Pins)

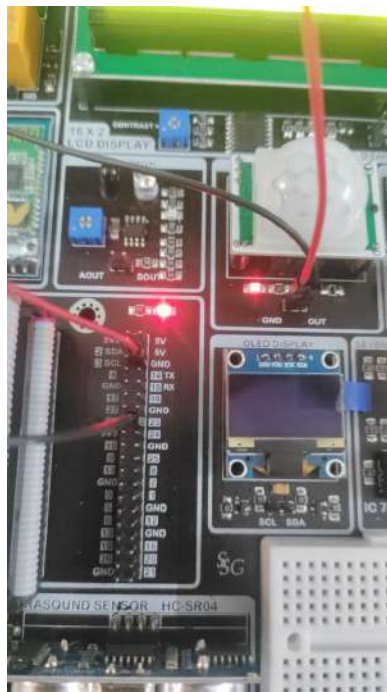
- **Pin 1 (VCC):** Connects to the **+5V DC supply**
- **Pin 2 (OUT):** Output pin, carries the detected IR signal to a microcontroller or amplifier
- **Pin 3 (GND):** Connects to **ground**

(In some PIR sensors, resistors like 100K or 47K are internally connected for stability.)

Connections of PIR SENSOR with Raspberry Pi

RASPBERRY PI	PIR SENSOR
GPIO 23	DOUT
GPIO 3	LED

"Please follow the same environment setup and terminal commands (activating the venv, and installing lgpio) as you did for the Relay section earlier."



```

Shell x
motion detected!
No motion detected
Motion detected!
No motion detected
Motion detected!
No motion detected
Motion detected!
No motion detected
Motion detected!
No motion detected
Motion detected!
No motion detected

```


12. ULTRASONIC SENSOR

The HC-SR04 ultrasonic module works on the principle of SONAR and RADAR to measure distance. It can detect objects from 2 cm up to 400 cm. The sensor consists of an ultrasonic transmitter, receiver, and a control circuit. When a trigger pulse is applied, the transmitter generates eight pulses of ultrasound at

40 kHz, which travel through the air and get reflected back as an echo when

they hit an object.



The Echo pin of the sensor goes high until the reflected signal is received. The duration of this high state corresponds to the total travel time of the sound wave. Since the wave travels to the object and back, the time is divided by two, and using the known speed of sound, the exact distance of the object can be calculated.

Connections of ULTRASONIC SENSOR with RASPBERRY PI

RASPBERRY PI	ULTRASONIC SENSOR
GPIO 21	TRIG
GPIO 20	ECO

"Please follow the same environment setup and terminal commands (activating the venv, and installing lgpio) as you did for the Relay section earlier."

```
Distance Monitor Running... (Press Ctrl+C to stop)
```

```
-----  
Distance: 52.0 cm | Status: OBJECT DETECTED  
Distance: 51.5 cm | Status: OBJECT DETECTED  
Distance: 52.2 cm | Status: OBJECT DETECTED  
Distance: 52.2 cm | Status: OBJECT DETECTED  
Distance: 51.8 cm | Status: OBJECT DETECTED  
Distance: 49.8 cm | Status: OBJECT DETECTED  
Distance: 16.3 cm | Status: OBJECT DETECTED  
Distance: 10.8 cm | Status: OBJECT DETECTED  
Distance: 6.9 cm | Status: OBJECT DETECTED  
Distance: 6.8 cm | Status: OBJECT DETECTED  
Distance: 44.8 cm | Status: OBJECT DETECTED  
Distance: 0.0 cm | Status: CLEAR  
Distance: 5.9 cm | Status: OBJECT DETECTED  
Distance: 5.9 cm | Status: OBJECT DETECTED  
Distance: 0.0 cm | Status: CLEAR  
Distance: 0.0 cm | Status: CLEAR  
Distance: 0.0 cm | Status: CLEAR  
Distance: 0.0 cm | Status: CLEAR  
Distance: 0.0 cm | Status: CLEAR  
Distance: 0.0 cm | Status: CLEAR  
Distance: 0.0 cm | Status: CLEAR  
Distance: 0.0 cm | Status: CLEAR  
Distance: 0.0 cm | Status: CLEAR  
Distance: 3.9 cm | Status: OBJECT DETECTED  
Distance: 3.9 cm | Status: OBJECT DETECTED  
Distance: 4.8 cm | Status: OBJECT DETECTED  
Distance: 4.3 cm | Status: OBJECT DETECTED  
-----
```

13. BLUETOOTH

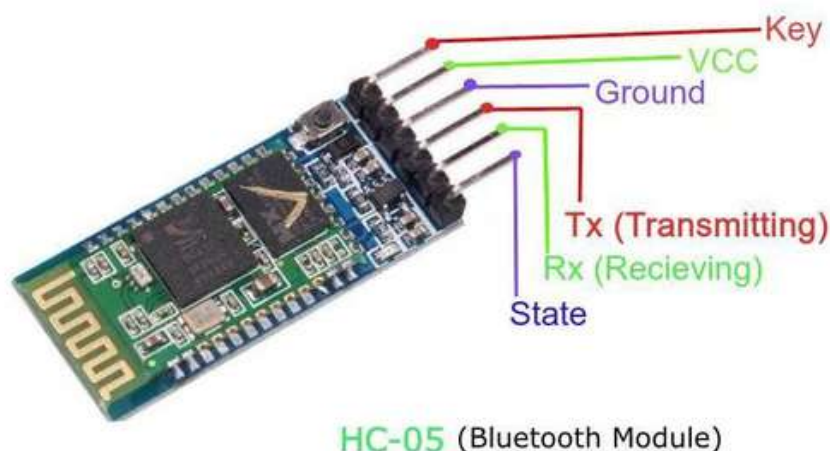
Bluetooth is a wireless communication technology used in devices like headsets, game controllers, keyboards, mice, and other consumer electronics. It works within a range of up to 100 meters, depending on environment and

hardware conditions.

It follows the IEEE 802.15.1 standard for building Wireless Personal Area

Networks (PANs) and uses Frequency-Hopping Spread Spectrum (FHSS) to send data securely.

The module communicates with microcontrollers through serial communication (USART), making it simple to use with Arduino or STM32.



Connecting Bluetooth Module (HC-05/HC-06)

1. Install a Bluetooth Terminal app on your smartphone.
2. Turn on the Bluetooth of your phone.
3. Search for available devices and select the module name (e.g., HC-05 or HC-06).
4. If prompted for a password, enter 1234 (default).
5. Once paired, open the Bluetooth Terminal app.

6. Go to the list of paired devices and select the same module (HC-05/HC-06).
7. After a successful connection, you can start sending and receiving data

Connections of BLUETOOTH with RASPBERRY PI

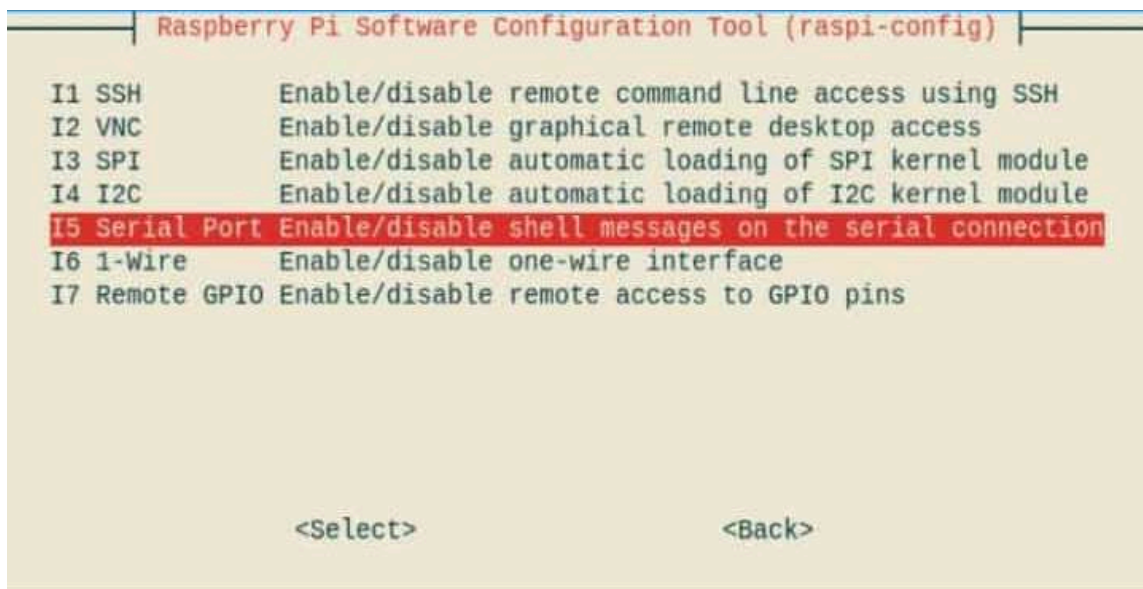
RASPBERRY PI	BLUETOOTH
GPIO 14 - TX	RX
GPIO 15 - RX	TX
GPIO 27	LED

Step 1: Enable Serial Hardware (UART)

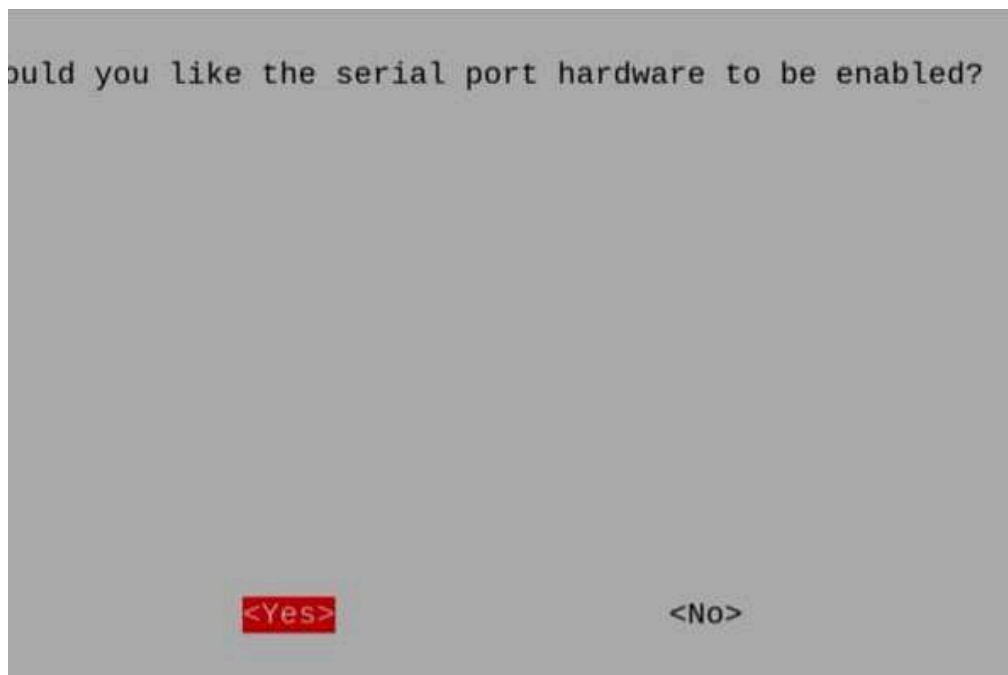
- Click the shortcut for the terminal on the left side of the taskbar and type the shell command:
`sudo raspi-config`
- This will immediately launch the Raspberry Pi Software Configuration Tool – showing the following window inside your terminal:



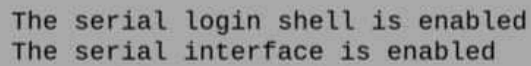
- Select **Interface Options > Serial Port** to enable UART. Unlike the Raspberry Pi Configuration application, both Serial Port and Serial Console are located within a single “Interface Options” group.



- Would you like a login shell to be accessible over serial? Select **No**.
- Would you like the serial port hardware to be enabled? Select **Yes**.



- Finally, the terminal will display what you had chosen as a way to present the interface changes.

A screenshot of a terminal window with a grey background. It displays two lines of text: "The serial login shell is enabled" and "The serial interface is enabled". At the bottom center, there is a red button with the text "<Ok>".

```
The serial login shell is enabled
The serial interface is enabled
```

- Finish and **Reboot** your Pi.

Step 2: Hardware Connections

As mentioned above in the table

Step 3: Set Up and Execution

Do follow the same terminal setup steps as you did earlier for the Buzzer to activate your environment, then install the serial library.

- **Activate Environment:** `source env/bin/activate`
- **Install Serial Library:**

pip install pyserial gpiozero lgpio

- **Paste your code.**
- **Run the script.**

```
15:20:42.233 Connecting to HC-05 ...  
15:20:48.354 Connection failed: read failed, socket mig  
ht closed or timeout, read ret: -1  
15:21:09.100 Connecting to HC-05 ...  
15:21:10.934 Connected  
15:22:20.477 0  
15:22:25.096 1
```

M1

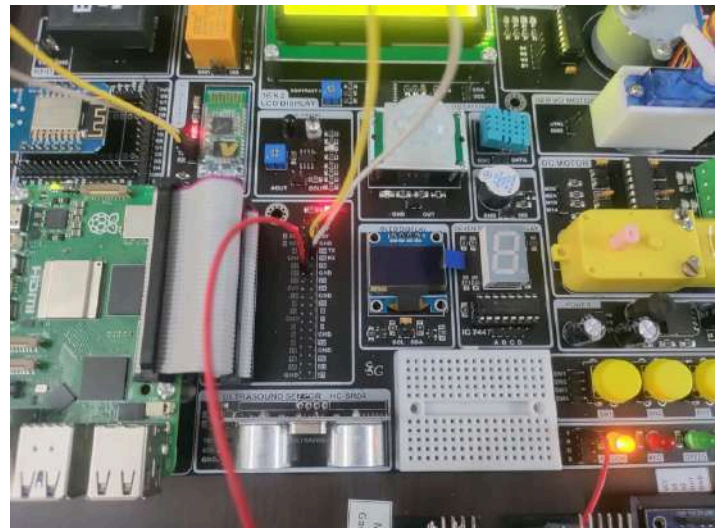
M2

M3

M4

M5

M6



Shell x

```
>>> %Run bluetoothled.py  
Raspberry Pi 5 Bluetooth Receiver Started...  
Received: 0  
LED1: OFF  
Received: 1  
LED1: ON
```

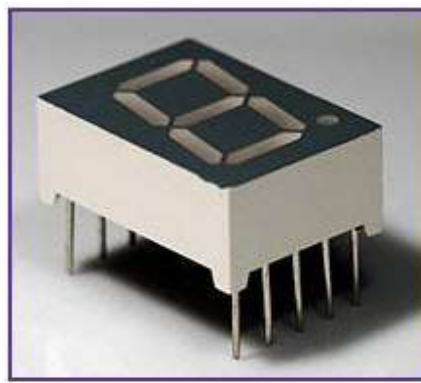
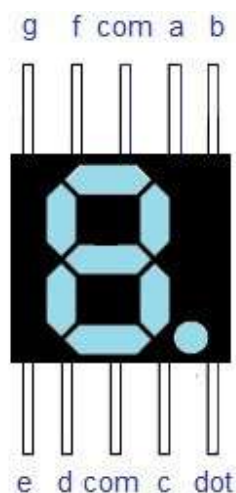
14. SEVEN SEGMENT DISPLAY

A **Seven Segment Display** (SSD) is a simple electronic display device used to show numerical digits (**0–9**) and a few characters like **A, B, C, E, F, H**. It is made up of 8 LEDs — seven LEDs form the segments labeled a–g, and one LED is used for the decimal point (dp).

By turning **ON** or **OFF** different segments, digits and characters can be displayed.

Each SSD has 10 pins:

8 pins for the segments (a–g + dp) and 2 pins as common terminals (COM). The COM pins are internally shorted, so you only need to use one of them.



Types of Seven Segment Displays

Common Anode (CA): All anodes are connected together at the COM pin.

Segments light up when their cathode is connected to **GND**. **Common Cathode (CC):** All cathodes are connected together at the COM pin. Segments light up

when their anode is connected to **VCC**.

Connections of SEVEN SEGMENT DISPLAY with RASPBERRY PI

RASPBERRY PI	SEVEN SEGMENT DISPLAY
GPIO 4	A
GPIO 17	B
GPIO 27	C
GPIO 22	D

"Please follow the same environment setup and terminal commands (activating the venv, and installing lgpio) as you did for the Relay section earlier."

```

Shell x
>>> %Run sevensegmentfinal.py
Decimal 0 → BCD [0, 0, 0, 0]
Decimal 1 → BCD [1, 0, 0, 0]
Decimal 2 → BCD [0, 1, 0, 0]
Decimal 3 → BCD [1, 1, 0, 0]
Decimal 4 → BCD [0, 0, 1, 0]
Decimal 5 → BCD [1, 0, 1, 0]
Decimal 6 → BCD [0, 1, 1, 0]
Decimal 7 → BCD [1, 1, 1, 0]
Decimal 8 → BCD [0, 0, 0, 1]
Decimal 9 → BCD [1, 0, 0, 1]
Decimal 0 → BCD [0, 0, 0, 0]
Decimal 1 → BCD [1, 0, 0, 0]
Decimal 2 → BCD [0, 1, 0, 0]

```

15. RFID (EM-18)

RFID (Radio Frequency Identification) is a wireless identification technology that uses radio waves to detect and identify the presence of RFID tags. Similar to a barcode reader, it is used to identify people, objects, or goods. However, unlike barcodes, RFID does not require direct optical scanning.

The RFID tag only needs to be within the reader's range to be detected.

Unlike barcodes that can get damaged or unreadable, RFID tags are generally more durable and reliable. Each RFID tag carries a unique ID, making the system highly secure and efficient.



Applications of RFID

- Attendance systems: Each person is assigned a unique RFID tag for identification.
- Access control: Used in companies to allow only authorized employees.
- Asset and goods tracking: Helpful in inventory management and logistics. Automated toll collection: RFID tags on vehicles enable quick toll payments without stopping.

Connections of RFID with RASPBERRY PI

RASPBERRY PI	RFID
GPIO 15 - RX	TX

Step 1: Enable Serial Port (UART)

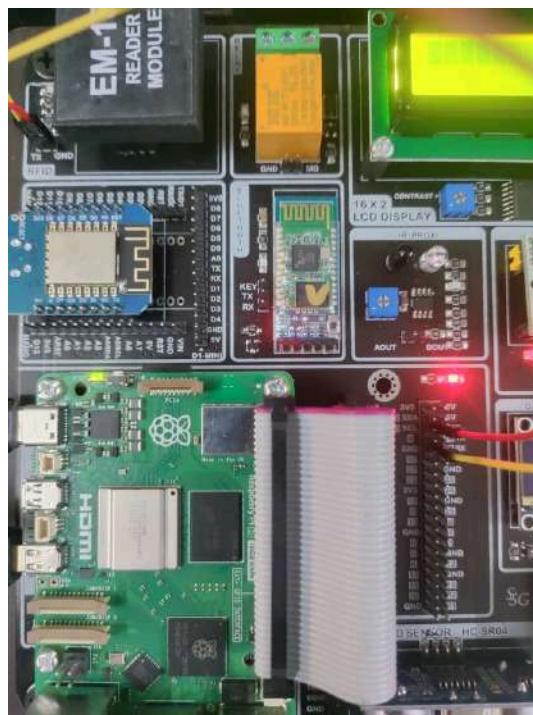
The EM-18 module communicates via Serial. You must enable this interface in the system settings.

1. Open the terminal: **sudo raspi-config**
2. Navigate to **Interface Options -> Serial Port**.
3. **Login shell over serial?** Select **No**.
4. **Serial port hardware enabled?** Select **Yes**.
5. **Reboot** your Raspberry Pi.

Step 2: Set Up and Execution

Do follow the same terminal setup steps as you did earlier for the Relay to activate your environment, then install the serial library.

1. **Activate Environment:** `source env/bin/activate`
2. **Install Serial Library:** `pip install pyserial`
3. Paste your code.
4. Run the script



Shell X

--- Raspberry Pi 5 RFID Scanner Online ---

Swipe your card.

Card ID Found: \$0013685136

Card ID Found: \$0013685136

Card ID Found: \$0013685136

Card ID Found: \$0013685136

16. ESP8266 MODULE

An ESP8266 module is a low-cost, integrated Wi-Fi chip and module popular for Internet of Things (IoT) projects, offering a complete Wi-Fi solution with a built-in TCP/IP stack, allowing it to host applications or offload networking from other microcontrollers via serial communication (AT commands) or direct programming with an SDK. Key features include 32-bit processing, GPIO pins for sensors, 2.4 GHz Wi-Fi (802.11 b/g/n), and support for protocols like HTTP and MQTT, making it versatile for connecting devices to the internet.



Connections of ESP8266 MODULE with RASPBERRY PI

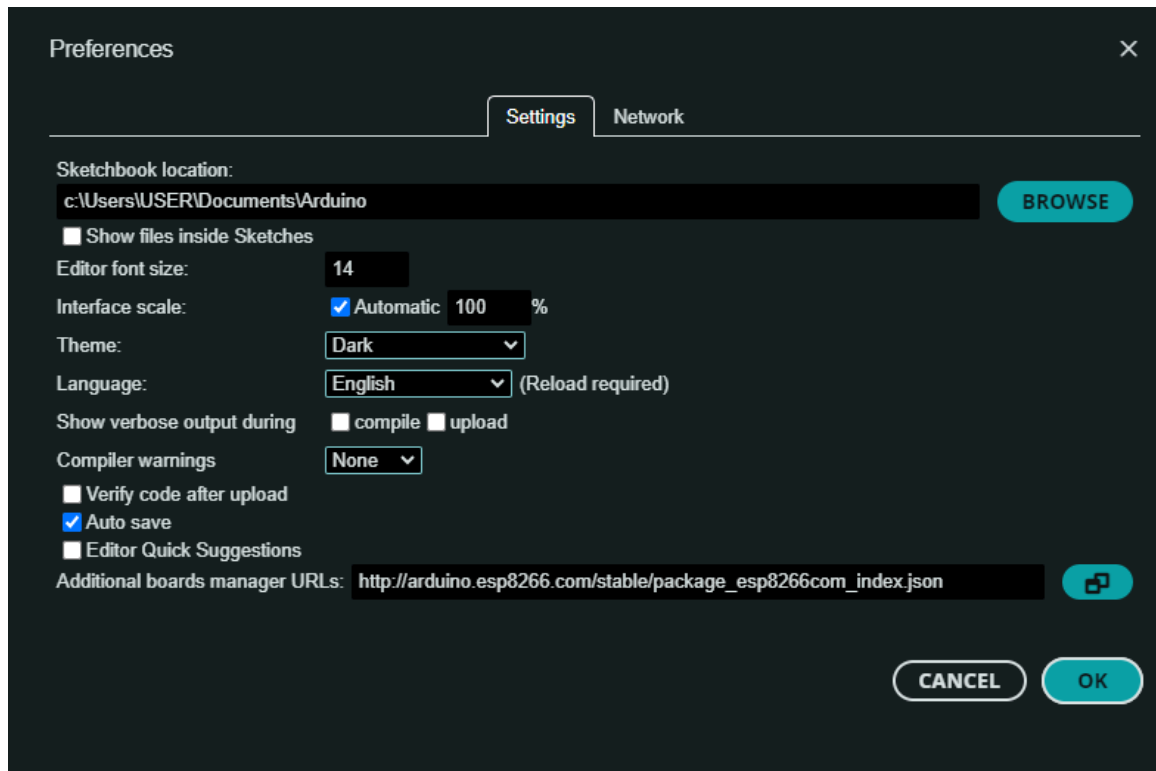
ESP8266	LED
D4	RED

The **Node MCU (ESP8266)** to create a local web server. Instead of the **Raspberry Pi terminal**, we use the **Arduino IDE** to write and upload the code.

Step 1: Preparing Arduino IDE

- **Install Arduino IDE:** Download it from the official Arduino website.
- **Add ESP8266 Board:** * Go to **File > Preferences**.

- In "Additional Boards Manager URLs", paste:
http://arduino.esp8266.com/stable/package_esp8266com_index.json



Install Board Package:

- Go to **Tools > Board > Boards Manager**.
- Search for "esp8266" and click **Install**.

Select your Board:

- Go to **Tools > Board > ESP8266 Boards > Select NodeMCU 1.0 (ESP-12E Module)**.
- Select Port: Plug in your NodeMCU and select the correct COM port under **Tools > Port**.

Step 2: Uploading the Code

- Write the Code
- Change the **ssid** and **password** variables to match your home **WiFi name and password**.
- Click the Upload button (**the right-pointing arrow**) in the Arduino IDE.

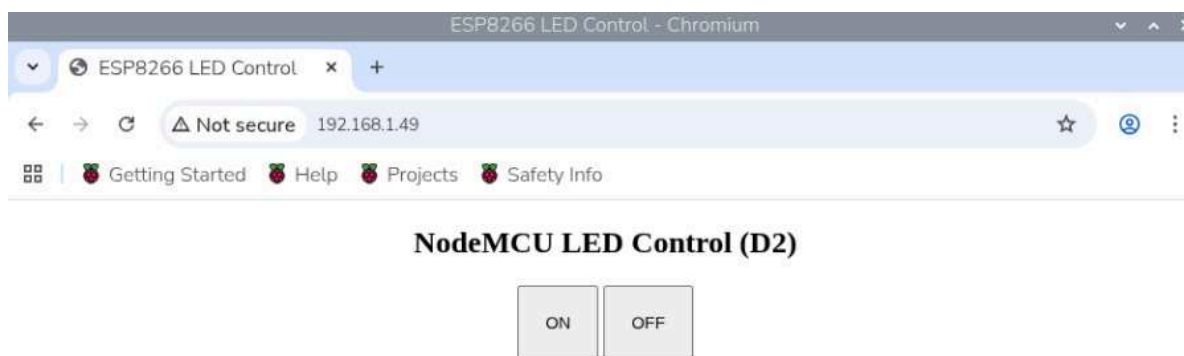
Step 3: Finding the IP Address and Controlling the LED

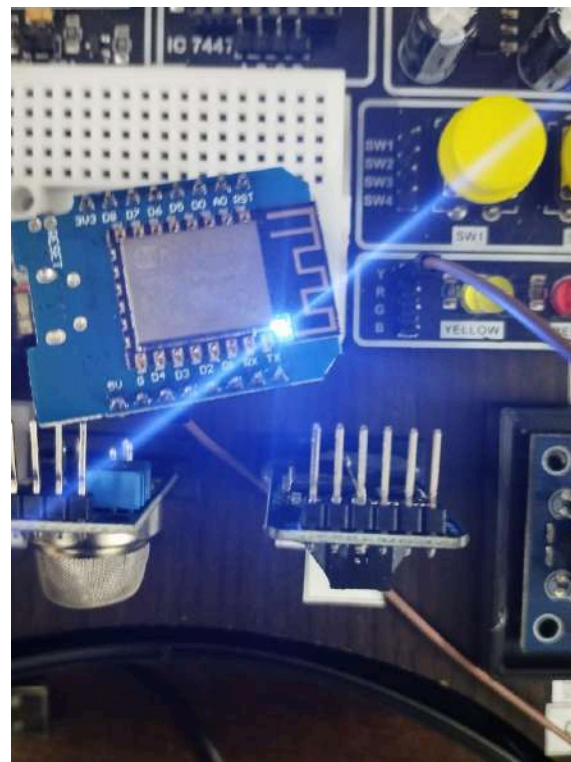
- Once the upload is complete, open the **Serial Monitor (Tools > Serial Monitor)**.
- Set the baud rate to **115200**.
- Press the **RST** (Reset) button on your NodeMCU.
- The Serial Monitor will display: WiFi connected! followed by an IP address (e.g., 192.168.1.15).

Open this IP in browser: 192.168.1.49

Step 4: Accessing the Web Interface

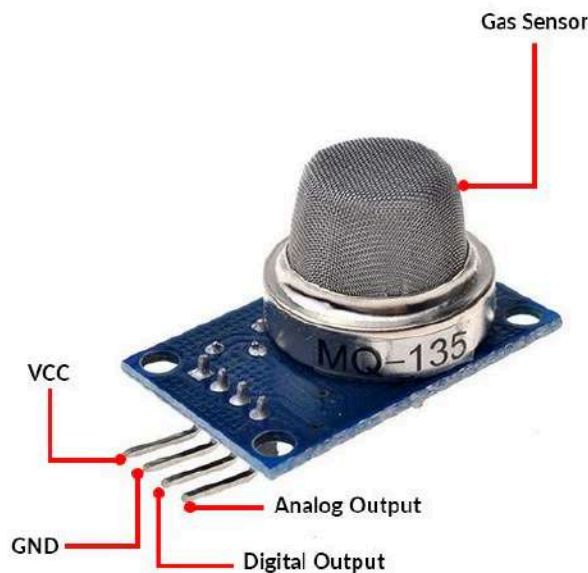
1. Ensure your Phone or Laptop is connected to the **same WiFi network** as the NodeMCU.
2. **Open Raspberry Pi** browser .
3. Type the **IP address** you saw in the Serial Monitor into the address bar and press Enter.
4. You will see two buttons: **ON** and **OFF**. Clicking them will send a request to the NodeMCU to toggle the LED.





17. MQ135 GAS SENSOR

An MQ135 air quality sensor is one type of MQ gas sensor used to detect, measure, and monitor a wide range of gases present in air like ammonia, alcohol, benzene, smoke, carbon dioxide, etc. It operates at a 5V supply with 150mA consumption. Preheating of 20 seconds is required before the operation, to obtain the accurate output.



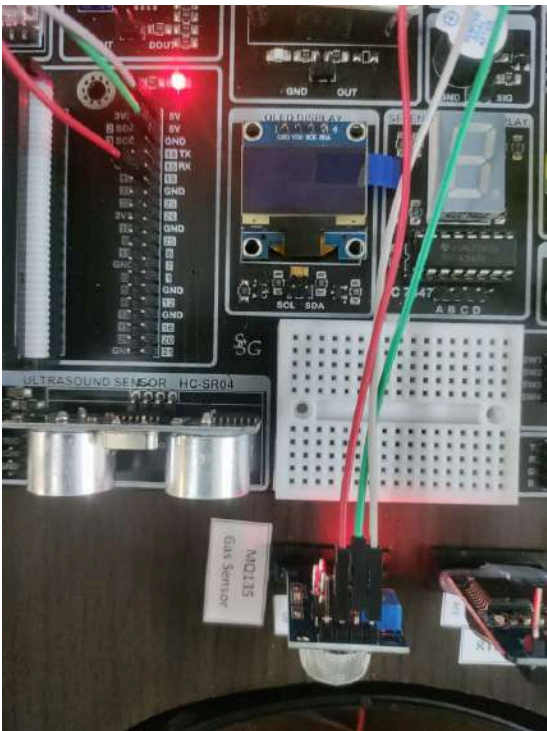
Applications

- **Industrial Safety:** Detecting gas leaks and explosive hazards in factories and mines.
- **Environmental Monitoring:** Tracking outdoor air pollution and greenhouse gas emissions.
- **Smart Homes:** Monitoring indoor air quality (CO_2) and detecting kitchen gas leaks.
- **Healthcare:** Using breath analysis to non-invasively diagnose medical conditions.
- **Food Safety:** Detecting spoilage gases to monitor the freshness of stored produce.

Connections of MQ135 GAS SENSOR with RASPBERRY PI

RASPBERRY PI	GAS SENSOR
5V	VCC
GND	GND
--	A0
GPIO 17	D0
GPIO 16	BUZZER

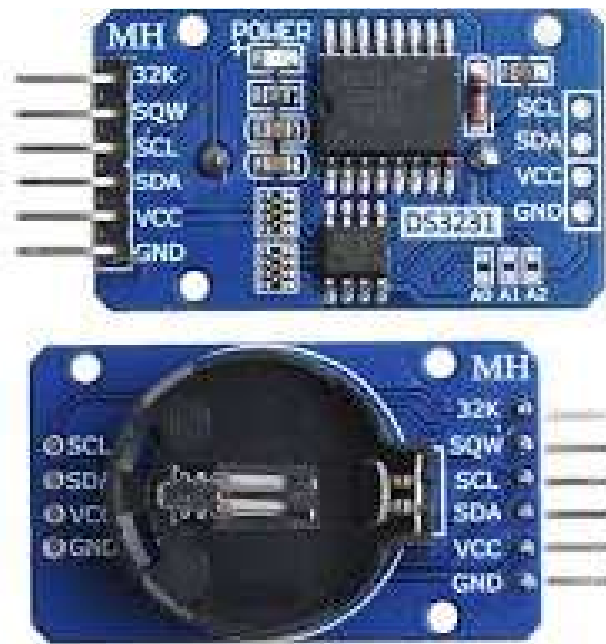
"Please follow the same environment setup and terminal commands (activating the venv, and installing lgpio) as you did for the Relay section earlier."



[illegible]

18. RTC (REAL TIME CLOCK)

DS3231 Real-Time Clock (RTC) module, a highly accurate, low-cost timekeeping component popular in electronics projects for keeping track of time (seconds, minutes, hours, date, etc.) even when power is lost, thanks to its integrated temperature-compensated crystal oscillator (TCXO) and backup battery support, communicating via the I2C protocol with microcontrollers like Arduino and Raspberry Pi.



Connections of RTC with RASPBERRY PI

RASPBERRY PI	RTC
--	32K
--	SQW
GPIO 3 - SCL	SCL
GPIO2 - SDA	SDA
5V	VCC
GND	GND

Terminal Configuration for DS3231 RTC

- **Identify the Device**

After wiring the module, check if the Pi sees it on the I2C bus:

```
sudo i2cdetect -y 1
```

You should see 68 at the intersection of row 60 and column 08. If you see UU instead of 68, it means the system driver is already controlling it

- **Add the RTC Overlay**

To make the Raspberry Pi 5 recognize the DS3231 as a system clock at boot, you need to edit the config file:

1. Open the config file:

```
sudo nano /boot/firmware/config.txt
```

2. Scroll to the bottom and add this line:

```
dtoverlay=i2c-rtc,ds3231
```

3. Save and exit (**Ctrl+O**, **Enter**, **Ctrl+X**)

- **Remove the Fake HW Clock**

Raspberry Pi OS uses a "fake-hwclock" by default, which saves the time to a file when you shut down. Since you now have a real hardware clock, you should disable the fake one:

```
sudo apt-get -y remove fake-hwclock
```

```
sudo update-rc.d -f fake-hwclock remove
```

```
sudo systemctl disable fake-hwclock
```

- **Sync the System Time to the RTC**

Now that the system knows about the DS3231, you can use these terminal commands to manage it:

- **Read the time from the RTC hardware:**

```
sudo hwclock -r
```

- **Write the current System Time (from internet) to the RTC:**

```
sudo hwclock -w
```

- **Set the System Time from the RTC (useful when no internet is available):**

```
sudo hwclock -s
```

5. Final Step: Automatic Sync

To ensure the Pi pulls the time from the RTC every time it turns on, edit the following file:

```
sudo nano /lib/udev/hwclock-set
```

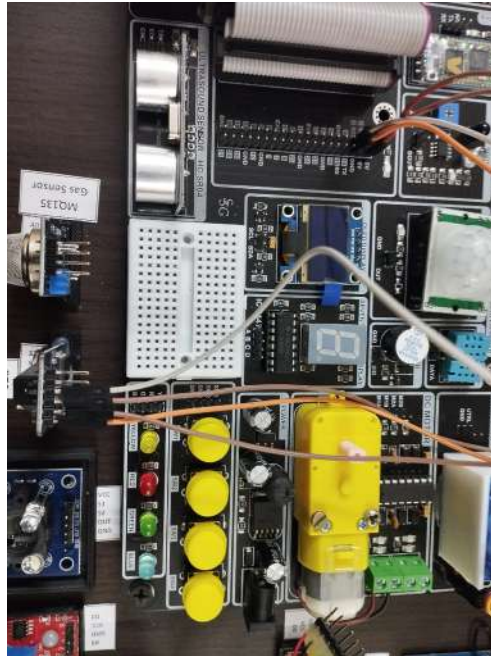
Locate these lines and comment them out by adding a # at the start, otherwise the system might overwrite your RTC time with the wrong data:

```
# if [ -e /run/systemd/system ] ; then
```

```
# exit 0
```

```
# fi
```

```
Save
```



```
python3 rtc-test.py
```

```
Setting RTC to system time: 2024-26 103515.5.123456
```

```
RTC current time: 2025-26 3:35:15
```

19. COLOUR SENSOR

A color sensor is a device that detects and measures colors by analyzing light reflected or emitted from an object, typically identifying the proportions of red, green, and blue (RGB) to determine the object's color, similar to human vision but for automated tasks like sorting, quality control, and matching in manufacturing, robotics, and consumer electronics.



These sensors use a light source (like LEDs) to illuminate a surface and photodiodes with filters (or separate R, G, B sensors) to measure the intensity of reflected light in specific wavelengths, then output data based on these ratios.

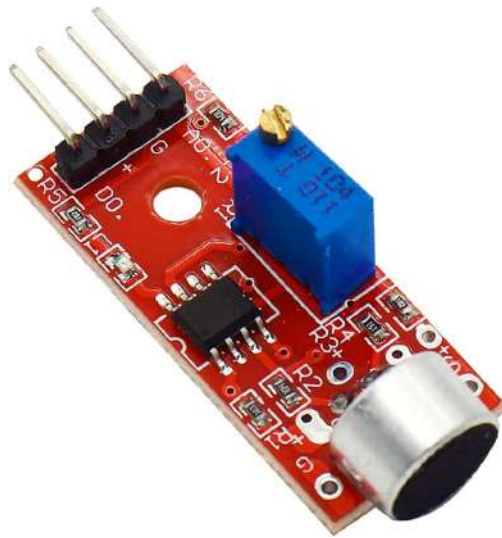
Connections of COLOUR SENSOR with RASPBERRY PI

RASPBERRY PI	COLOR SENSOR
GND	GND
GPIO 4	OUT
GPIO 24	S2
GPIO 25	S3
V5	VCC
GND	GND
GPIO 17	LED
GPIO 22	S0
GPIO 23	S1
V5	VCC

"Please follow the same environment setup and terminal commands (activating the venv, and installing lgpio) as you did for the Relay section earlier."

20. SOUND SENSOR

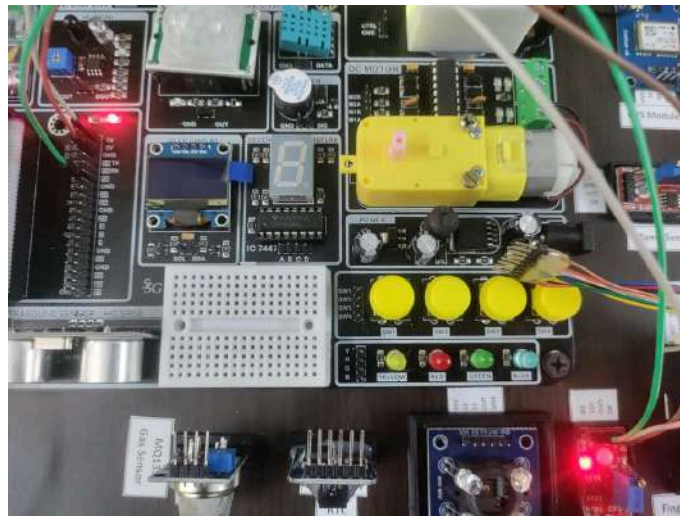
A sound detection sensor detects sound waves (vibrations) using a microphone, converts them into electrical signals, and outputs data indicating sound intensity, often with adjustable sensitivity via a potentiometer.



These modules typically connect to microcontrollers like Arduino, trigger actions when sound exceeds a set threshold and are used in security systems, voice recognition, and noise monitoring.

Connections of SOUND SENSOR with RASPBERRY PI

RASPBERRY PI	SOUND SENSOR
GPIO 17	DOUT
5V	VCC
GND	GND
--	AOUT



Shell x

```
>>> %Run soundsensor.py
```

```
--- Sound Monitor Active ---
```

```
Listening for changes (Press Ctrl+C to stop)...
```

```
[14:36:57] □ Quiet...
```

```
[14:36:57] ◀ Sound Detected!
```

```
[14:36:59] □ Quiet...
```

```
[14:36:59] ◀ Sound Detected!
```

21. FINGERPRINT SENSOR

A fingerprint sensor module is a compact biometric device that captures, processes, and matches unique fingerprint patterns for identity verification, combining a scanner (optical or capacitive) with an integrated processor (DSP/MCU) and algorithms to enroll, store, and compare fingerprints, often communicating via UART to microcontrollers for security, access control, and time attendance.



Applications

- **Biometric Security:** Used in door locks, safes, and lockers to replace traditional keys with high-security personal access.
- **User Authentication:** Integrated into smartphones, laptops, and tablets for secure screen unlocking and identity verification for banking apps.
- **Time & Attendance:** Employed by businesses to accurately track employee clock-in/out times and prevent "buddy punching" (fraudulent check-ins).
- **Authorization Systems:** Used in vehicle ignition systems or high-end machinery to ensure only authorized operators can start the engine or equipment.

Connections of Fingerprint SENSOR with RASPBERRY PI

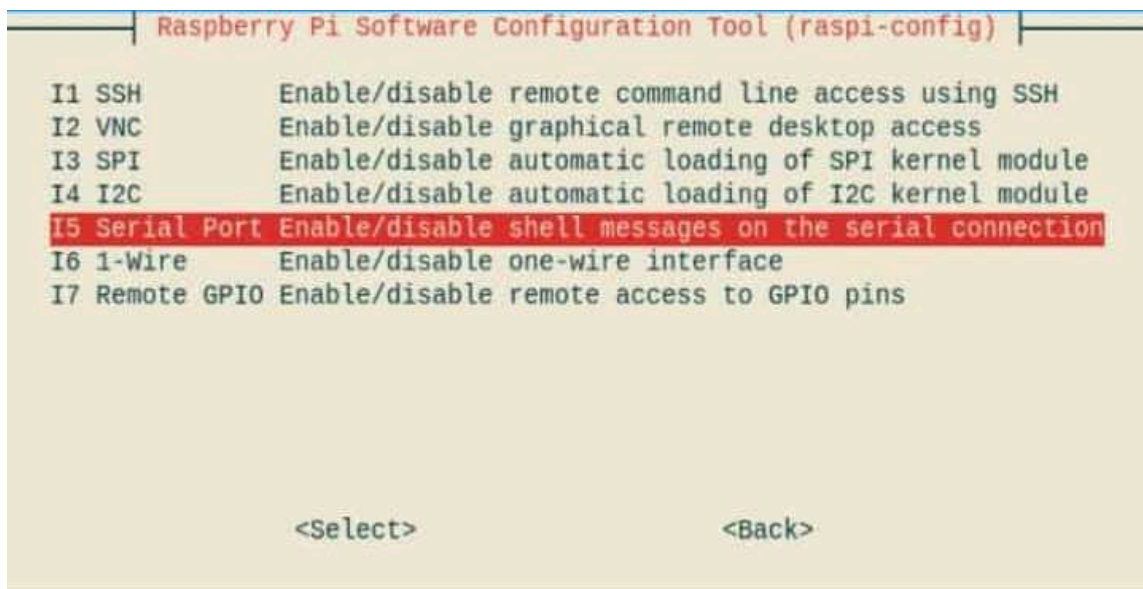
RASPBERRY PI	FINGERPRINT SENSOR
VCC	RED
GND	GREEN
TX	BLACK
RX	YELLOW

Step 1: Enable Serial Hardware (UART)

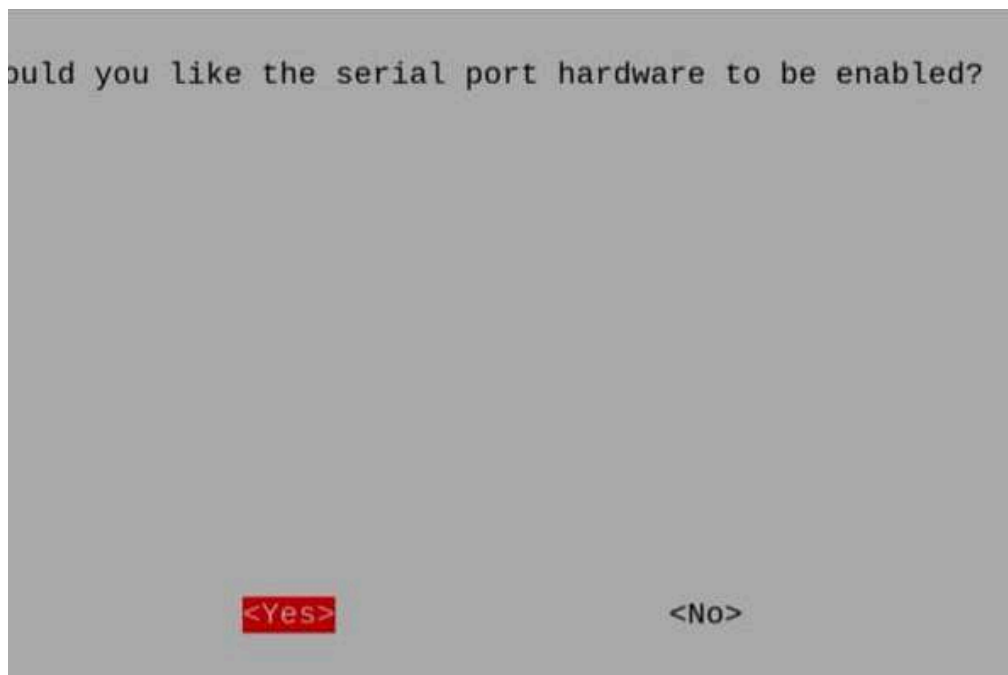
- Click the shortcut for the terminal on the left side of the taskbar and type the shell command:
sudo raspi-config
- This will immediately launch the Raspberry Pi Software Configuration Tool** – showing the following window inside your terminal:



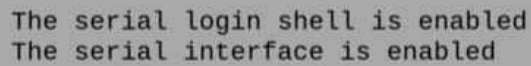
- Select **Interface Options > Serial Port** to enable UART. Unlike the Raspberry Pi Configuration application, both Serial Port and Serial Console are located within a single “Interface Options” group.



- Would you like a login shell to be accessible over serial? Select **No**.
- Would you like the serial port hardware to be enabled? Select **Yes**.



- Finally, the terminal will display what you had chosen as a way to present the interface changes.

A screenshot of a terminal window with a grey background. It displays two lines of text in a monospaced font: "The serial login shell is enabled" and "The serial interface is enabled". At the bottom center, there is a red button with the text "<Ok>" in white.

```
The serial login shell is enabled
The serial interface is enabled
```

- Finish and **Reboot** your Pi.

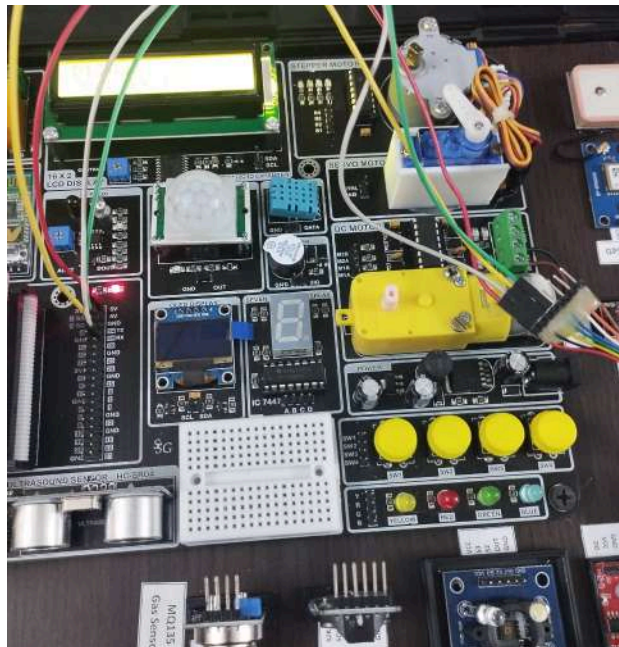
Step 2: Set Up and Execution

Do follow the same terminal setup steps as you did earlier for the Relay to activate your virtual environment, then install the specific biometric library.

- Activate Environment: `source env/bin/activate`
- Install Required Libraries:

```
pip install adafruit-circuitpython-fingerprint pyserial gpiozero lgpio
```

- Create the script
- Run the script



Shell x

```
>>> %Run fingerprint.py
```

```
1. Enroll Finger | 2. Start Recognition | 3. Exit
```

```
Select: 1
```

```
Enter ID # (1-127) to save: 1
```

```
Step 1: Place finger...
```

```
Remove finger...
```

```
Place same finger again...
```

```
? SUCCESS! Saved to ID #1
```

```
1. Enroll Finger | 2. Start Recognition | 3. Exit
```

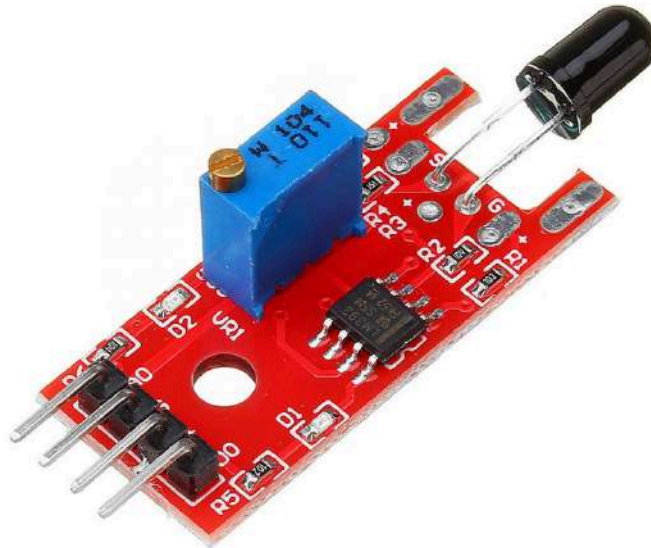
```
Select: 2
```

```
--- Monitoring Mode Active ---
```

```
? MATCH: ID #1
```

22. FLAME SENSOR

A flame sensor (or detector) is a device that detects the presence of a fire or flame by sensing its unique light (UV/IR) or electrical properties, often used in furnaces, boilers, and fire safety systems for fast detection and to trigger alarms or shut down fuel lines.



- **Electrical Detection (Flame Rod):** In some systems, a metal rod (flame rod) detects the flame's electrical conductivity, which changes when the flame is present.
- **Signal Processing :** A control unit process the signal from the sensor to confirm the flame and make decisions.

Connections of Flame SENSOR with RASPBERRY PI

RASPBERRY PI	FLAME SENSOR
VCC	VCC
GND	GND
GPIO 17	D0
GPIO 16	BUZZER



```
Shell X
?? FLAME DETECTED!
? No flame.
?? FLAME DETECTED!
? No flame.
?? FLAME DETECTED!
? No flame.
?? FLAME DETECTED!
? No flame.
?? FLAME DETECTED!
? No flame.
```

23. GPS SENSOR

A GPS module is a compact electronic device that receives signals from Global Positioning System satellites to calculate and provide precise location, velocity, and time data for various applications, acting as a miniature navigation system that determines latitude, longitude, and altitude by processing satellite transmissions.



These modules are essential for smartphones, drones, vehicles, and IoT devices, integrating a receiver, antenna, processor, and output interface to translate complex satellite data into usable location information.

Connections of GPS SENSOR with RASPBERRY PI

RASPBERRY PI	GPS
5V	VCC
GND	GND
RX	TX
TX	RX

Step 1: Enable Serial Hardware (UART)

The GPS module communicates via Serial. You must enable the hardware port to allow the Pi to read the coordinates.

1. Open the terminal: **sudo raspi-config**
2. Navigate to Interface Options -> **Serial Port**.
3. **Login shell over serial?** Select **No**.
4. **Serial port hardware enabled?** Select **Yes**.
5. **Reboot** your Raspberry Pi 5.

Step 2: Set Up and Execution

Do follow the same terminal setup steps as you did earlier for the Buzzer to activate your virtual environment, then install the NMEA parsing library.

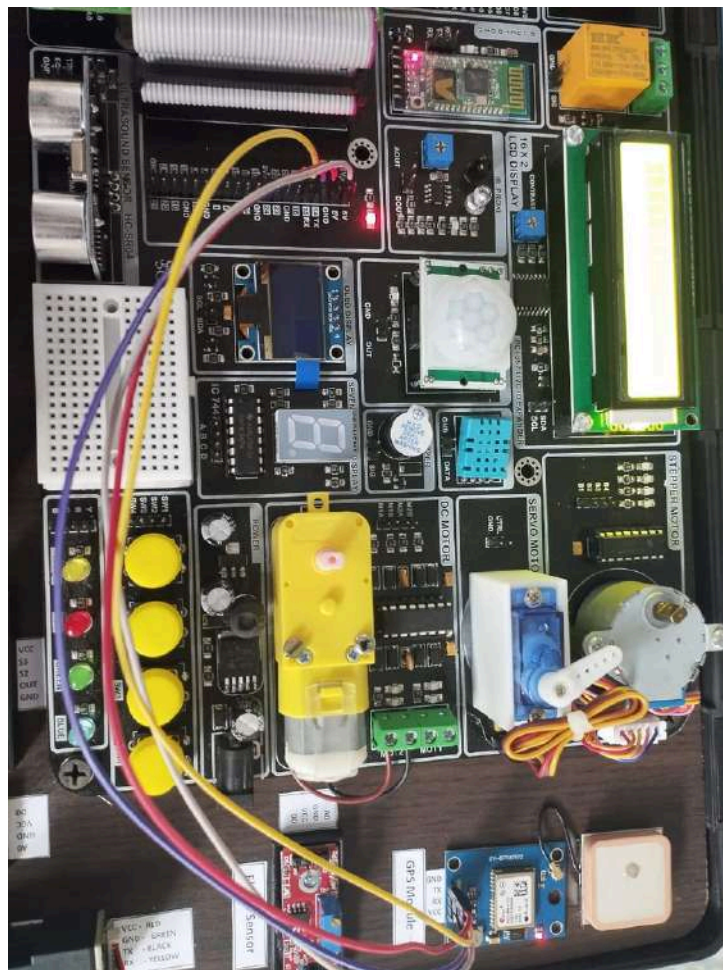
- **Activate Environment:**

`source env/bin/activate`

- **Install Required Library:**

`pip install pynmea2 pyserial`

- **Create the script**
- **Run the script**



Shell x

```
>>> %Run GPS.py
```

```
▣ Connection established on /dev/ttyAMA0  
Waiting for valid coordinates...
```

```
=====
```

```
▣ Latitude : 21.103743
```

```
▣ Longitude : 79.047758
```

```
▣ Click Link: https://www.google.com/maps/search/?api=1&query=21.103743%2C79.047758
```

```
=====
```


Microphone



Step 1: Connect the Microphone

1. Plug your USB microphone into one of the USB ports on the Raspberry Pi.
2. Wait about 5 seconds for the system to recognize the new hardware.

Step 2: Create a Folder for Your Recordings

Just like with the camera, you need a place to keep your voice files.

1. Right-click on your **Desktop**.
2. Select **New Folder**.
3. Name the folder **recordings** (all lowercase).

Step 3: Prepare the Recording Script

Create a new Python file (for example: mic_test.py) and paste the code you have. This script is smart—it will:

- Find your microphone automatically.

- Record your voice for 5 seconds.
- Save the file with the exact date and time so you don't lose it.
- **Play the sound back to you** so you can hear if it worked.

Step 4: Run the Voice Test

1. Open your **Terminal**.
2. Run your script by typing:
3. Bash
4. python3 mic_test.py
5. **Important:** As soon as you see the message "**Speak into the microphone now...**", speak clearly into the mic.
6. After 5 seconds, the recording will stop, and the Raspberry Pi will play your voice back through your speakers or headphones.

Step 5: Verify Your Files

1. Open the **recordings** folder on your Desktop.
2. You should see a file named something like [voice_20240520_1200.wav](#).
3. You can double-click this file anytime to listen to it again.