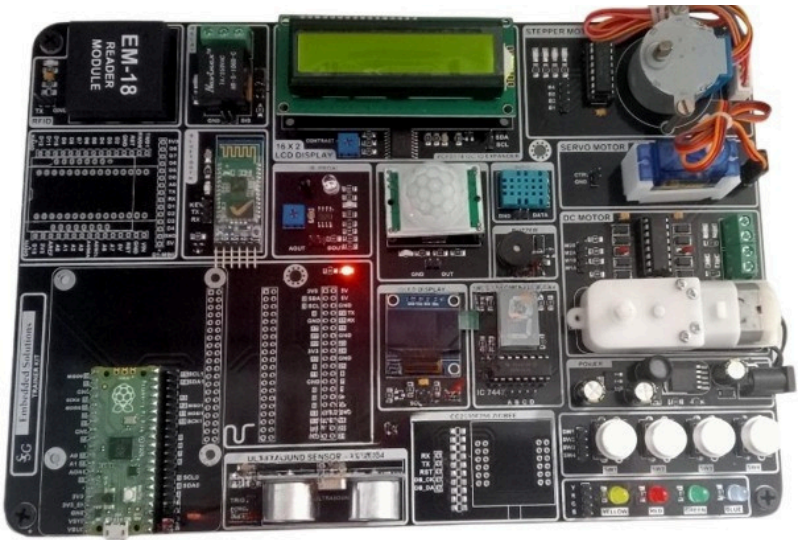
SSG RASPBERRY PICO DEVELOPMENT BOARD

Raspberry Pi Pico is a development board that uses Raspberry Pi microcontrollers (**RP2040** or **RP2350**). A Raspberry Pi Pico is programmed using **MicroPython**, **C**, or **C++** and, unlike other Raspberry Pi devices, doesn't run Linux or support removable storage. Instead, Raspberry Pi Pico is programmed by flashing binaries to the on-board flash memory.



A Raspberry Pi device is a compact computer that is commonly used in two main ways: either as a **desktop computer** or as a **headless (remote) computer**.

Raspberry Pi devices are available in several series:

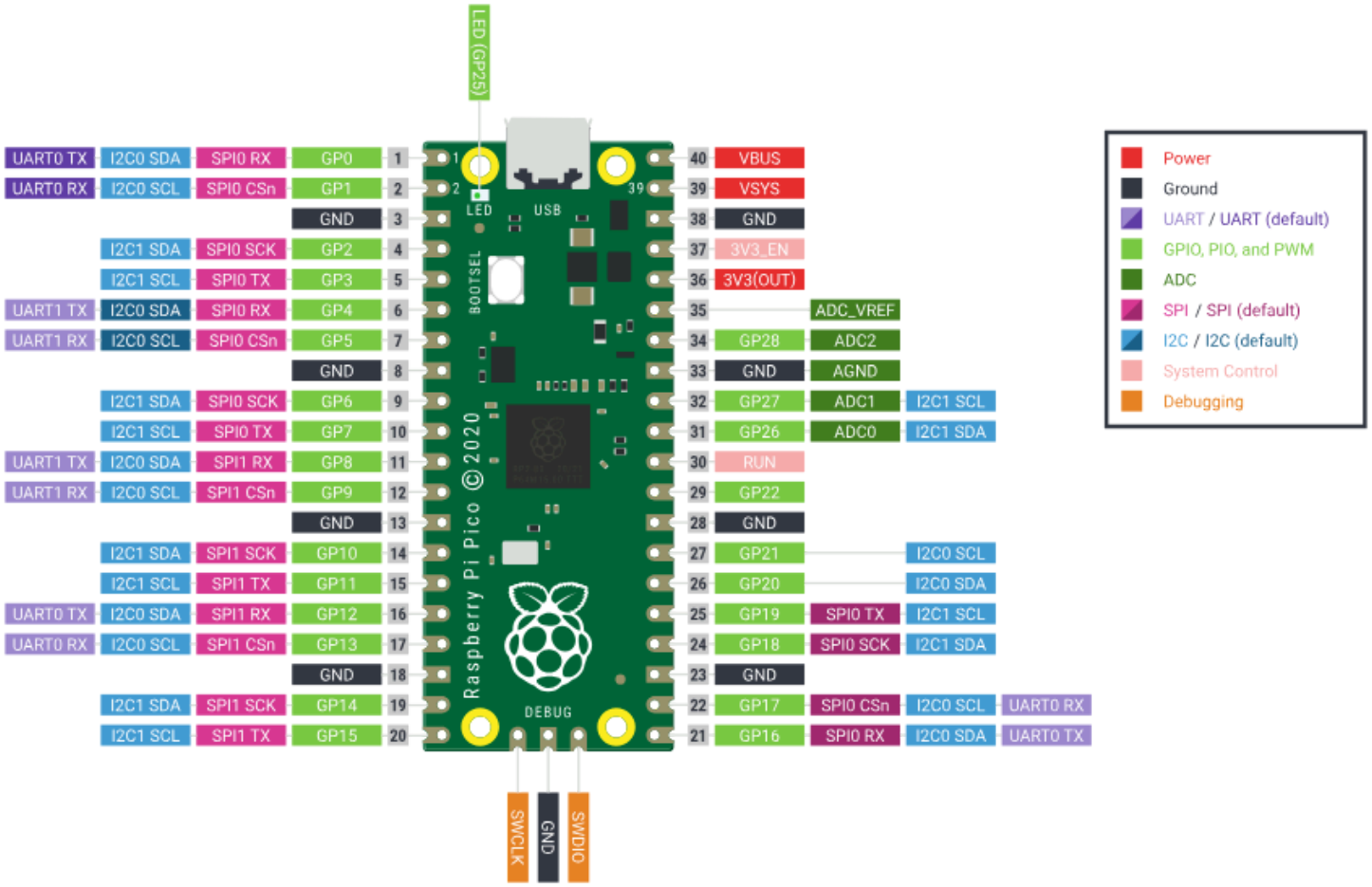
- **Raspberry Pi single-board computers (SBC).** Classic or 'flagship' Raspberry Pi devices that you can connect to a monitor, keyboard, and mouse like a desktop computer.
- **Raspberry Pi Zero.** A smaller version of flagship Raspberry Pi devices with fewer USB ports and lower power consumption.
- **Raspberry Pi keyboard computers.** Devices that have the computer built into a keyboard case, designed to be plugged into a monitor and mouse.
- **Raspberry Pi compute modules.** Small system-on-module versions of Raspberry Pi computers designed for industrial and embedded applications.
- **Raspberry Pi Pico.** A microcontroller board that uses Raspberry Pi microcontrollers (RP2040 or RP2350).

Key features

Raspberry Pi **Pico W** and **Pico WH** offer the same key features as Raspberry Pi **Pico** and **Pico H**, but with added wireless functionality. Specifically, Raspberry Pi Pico W and Pico WH offer the following key features:

- **RP2040 microcontroller chip**, including:
  - Dual-core M0+ processor.
  - Flexible clock running up to 133 MHz.
  - 264 kB of SRAM.
  - USB 1.1 controller and PHY with device and host support.
  - Low-power sleep and dormant modes.
  - Accurate clock and timer.
  - Accelerated floating-point libraries.
  - Temperature sensor.
  - Eight Programmable I/O (PIO) state machines for custom peripheral support.
    - Flexible, user-programmable high-speed I/O.
    - Can emulate interfaces such as SD card and VGA.
- **Board features**:
  - Drag-and-drop programming using mass storage over USB.
  - 2 MB of on-board flash memory.
  - 26 multi-function GPIO pins. For more information, see Wireless board layout.
  - Debug connector. For more information, see Debug connector (SWD).
- **Peripheral interfaces**:
  - Two SPI
  - Two I2C
  - Two UART
  - Three 12-bit ADC
  - 16 PWM channels
- **Wireless connectivity**:
  - Wi-Fi (802.11n):
    - Single-band (2.4 GHz).
    - WPA3, a modern security protocol for wireless connectivity.
    - Soft access point for broadcasting a Wi-Fi network, supporting up to four clients.
  - Bluetooth 5.2:
    - Support for Bluetooth Low Energy (BLE) Central and Peripheral roles.
    - Support for Bluetooth Classic.

Raspberry Pi Pico Pin Configuration



The majority of the microcontroller pins are brought to the user I/O pins on the left and right edges of the board. Four microcontroller I/O pins are used for internal functions: driving an LED, on-board Switched Mode Power Supply (SMPS) power control, and sensing system voltages. Aside from the ground (GND) pins, the 40 pins on the left and right side of a non-wireless Pico board include:

- 26 GPIO pins with overlapping functions (SPI, I2C, UART, and so on) that can be configured for different project needs.
- Power pins located in the top-right of the board, including:
  - 3V3(OUT) at pin 36 (3.3 V regulated output).
  - VSYS at pin 39 (2 V to 5 V system input).
  - VBUS at pin 40 (5 V input from USB).
- Special-purpose pins:

- 3V3_EN at pin 37. Pulling this pin to ground turns off the Raspberry Pi Pico.
- ADC_VREF at pin 35. Provides the reference voltage for ADC converters.
- AGND at pin 33 (also a GND pin). An analogue ground used to provide noise-free ground for sensors.
- RUN at pin 30. The microcontroller reset pin.

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.



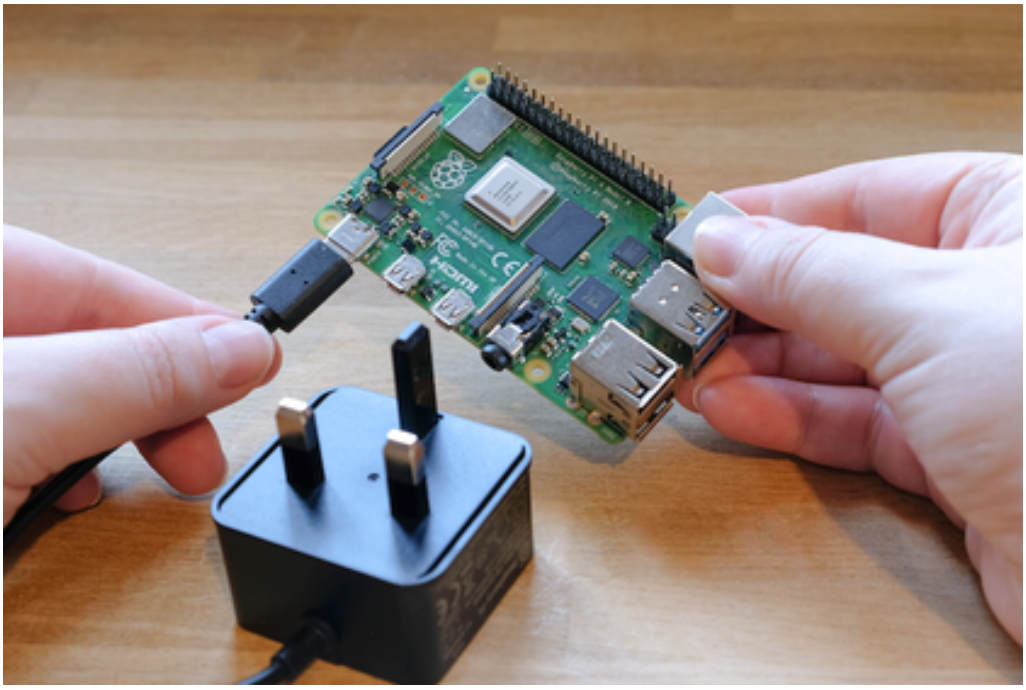
3 Mouse

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



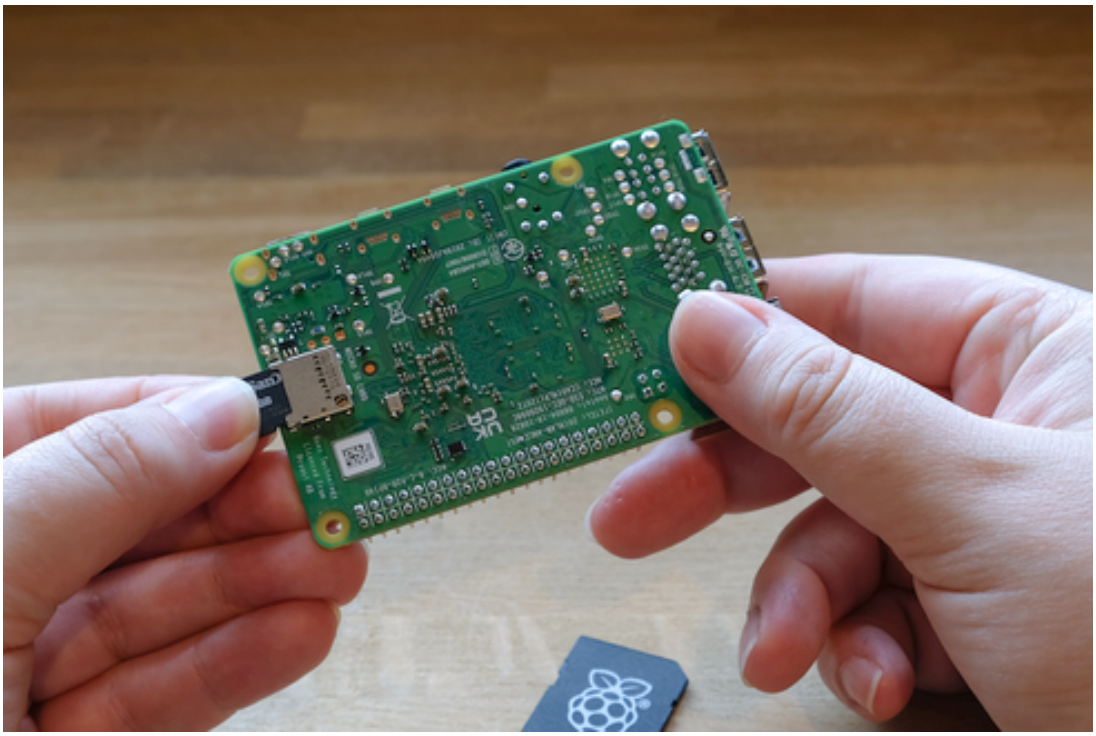
4 Power Supply

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.

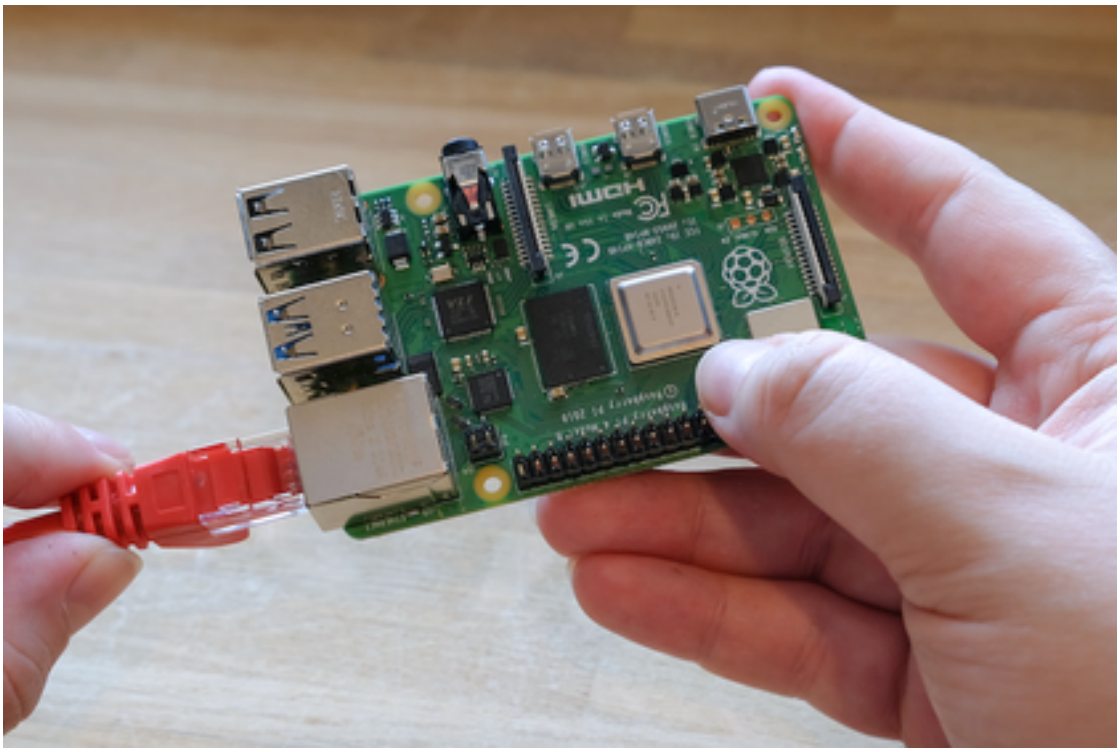
All Raspberry Pi models support audio over USB. All Raspberry Pi models equipped with Bluetooth support Bluetooth audio.

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

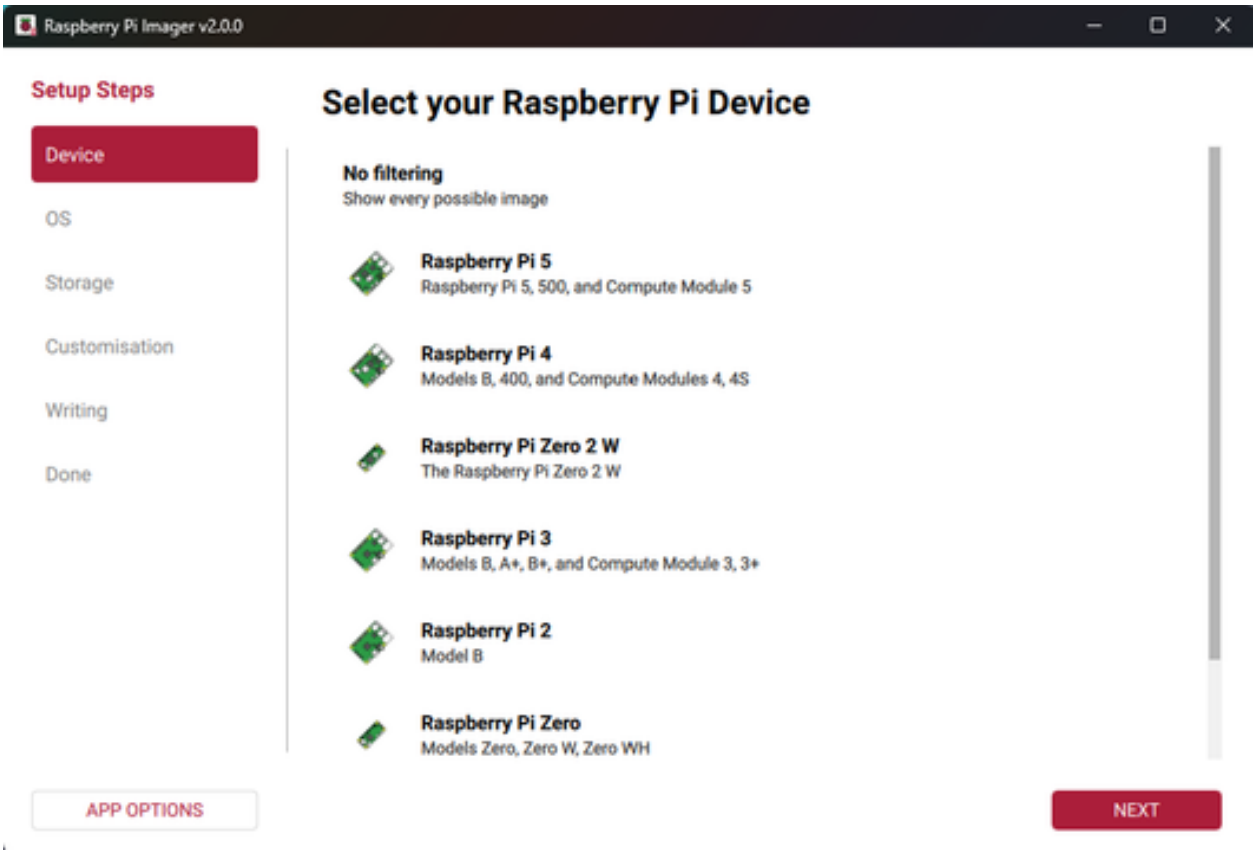
Download for Linux(x86_64)

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

Step 2. Configure the fundamentals

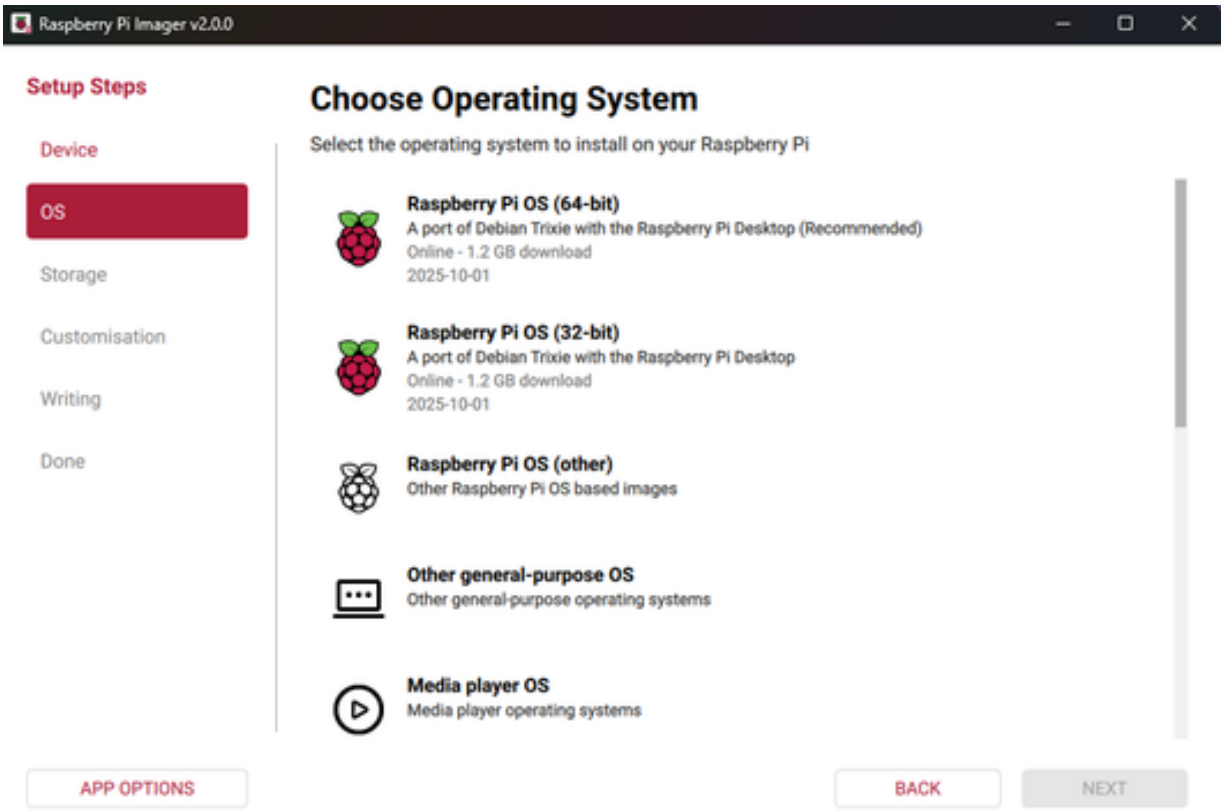
Tell Raspberry Pi Imager what combination of hardware and operating system you want to use:

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



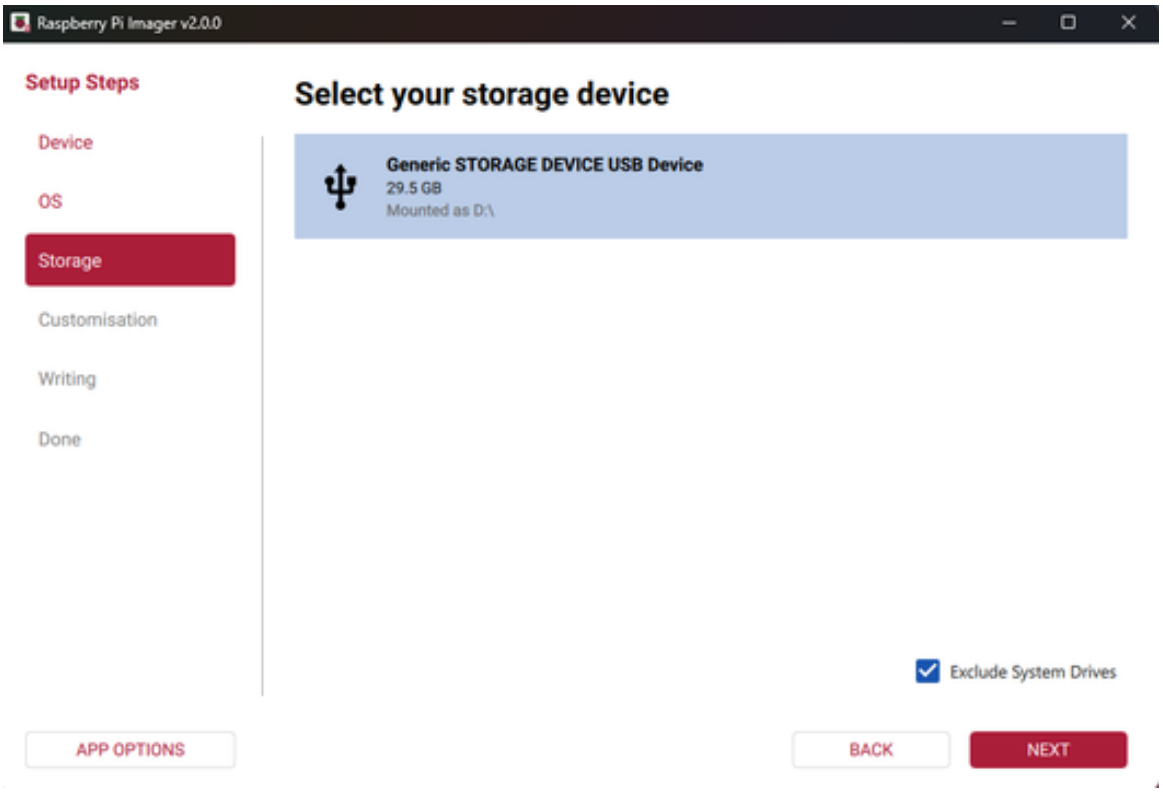
2. In the OS tab, choose from the available operating systems. Select Next.

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.

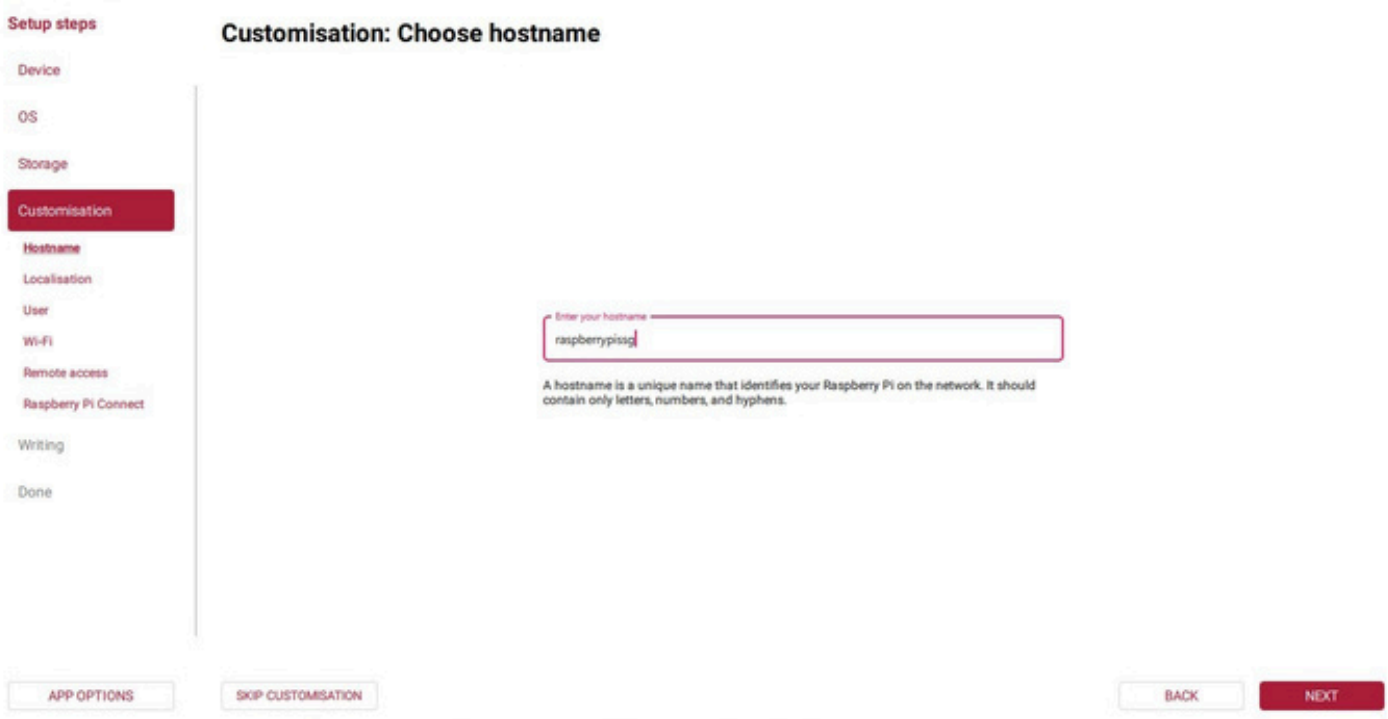


Warning

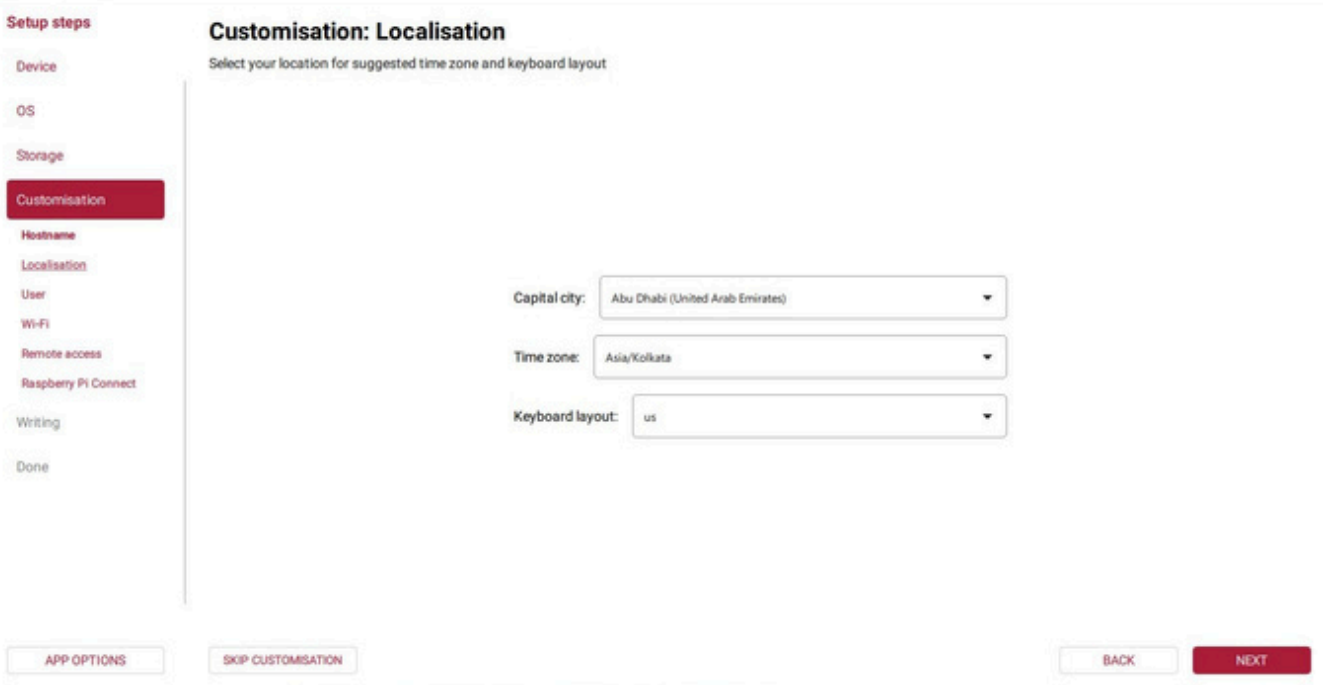
If you have more than one storage device connected to your computer, be sure to choose the correct device! You can often identify storage devices by size. If you're unsure, disconnect other devices until you've identified the device you want to image. Imager warns you if you attempt to overwrite a system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

Step 3. Customise your operating system

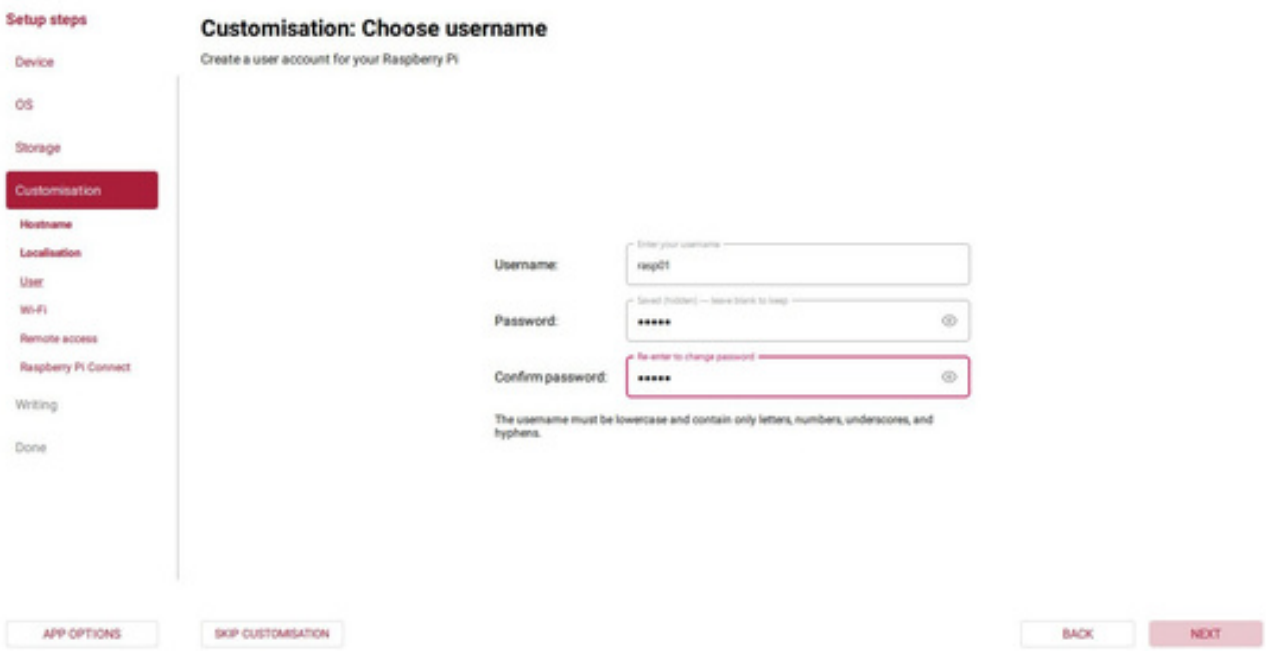
In the Customisation > Hostname tab, enter a host name for your Raspberry Pi that includes only letters, numbers, and hyphens. Select Next.



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.



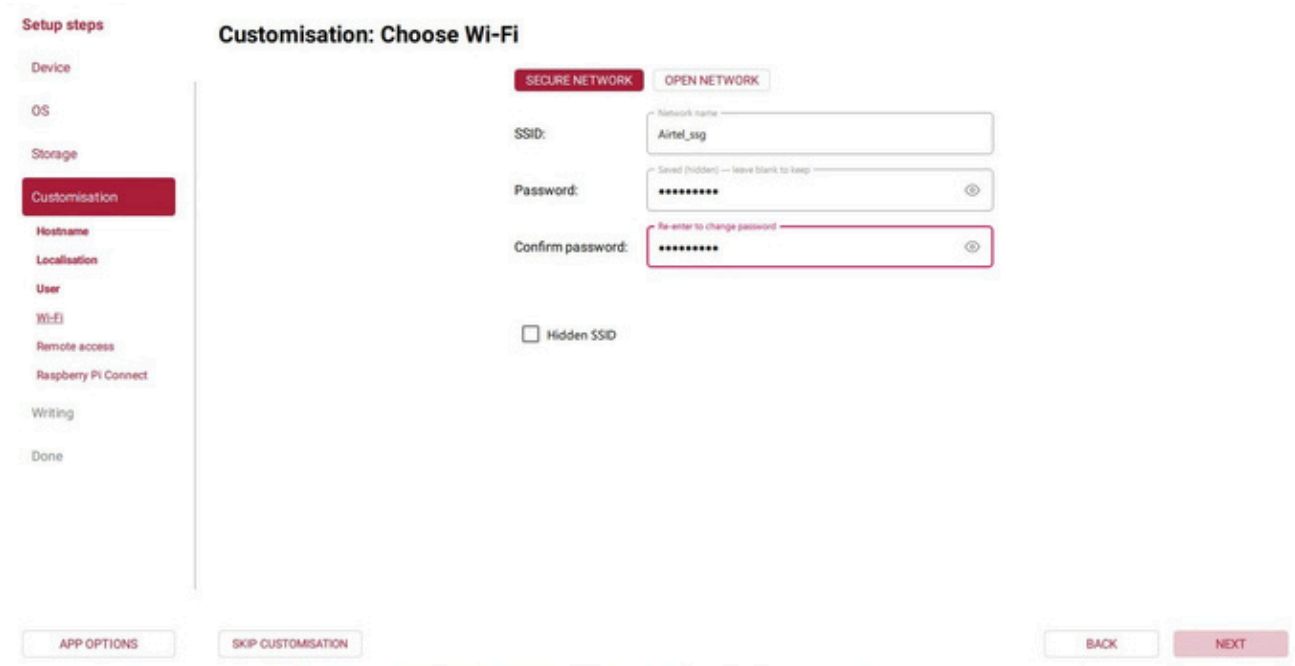
In the Customisation > User tab, enter a username and password for the admin user of your Raspberry Pi. Select Next.

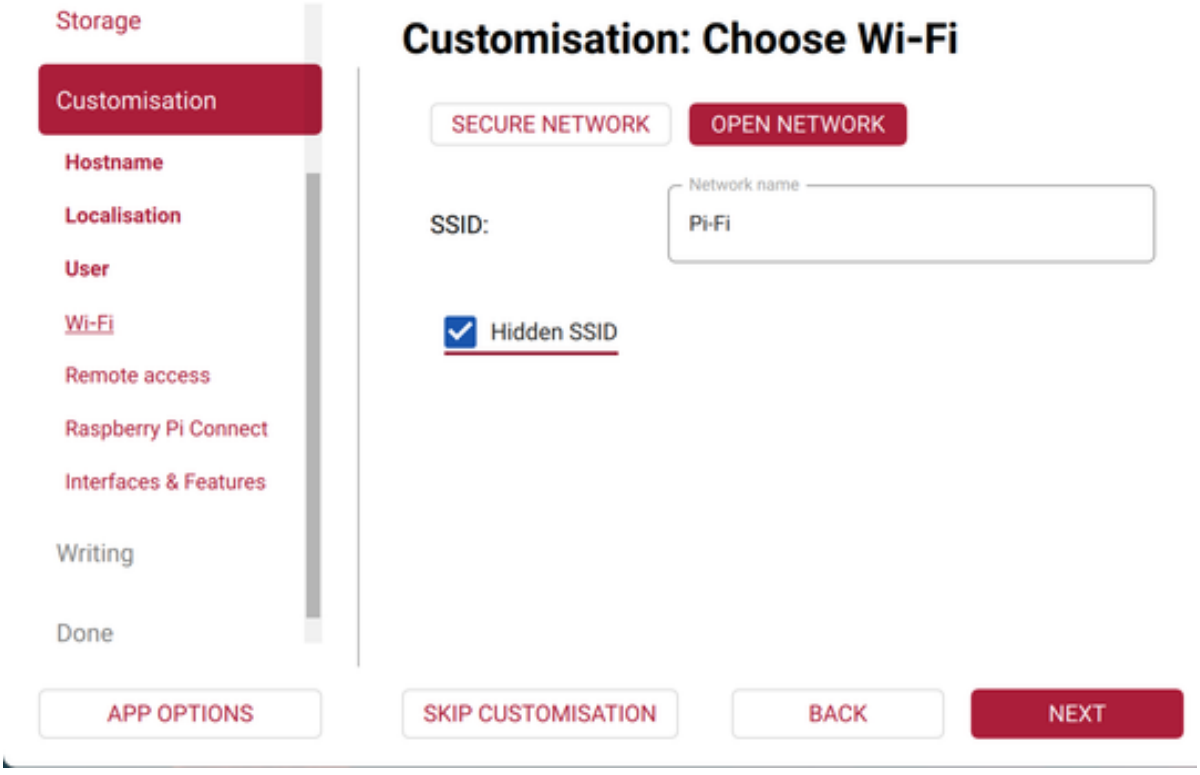


In the Customisation > Wi-Fi tab, on first use, Imager pre-fills the SSID (name) and password of the Wi-Fi network you're currently connected to.

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.





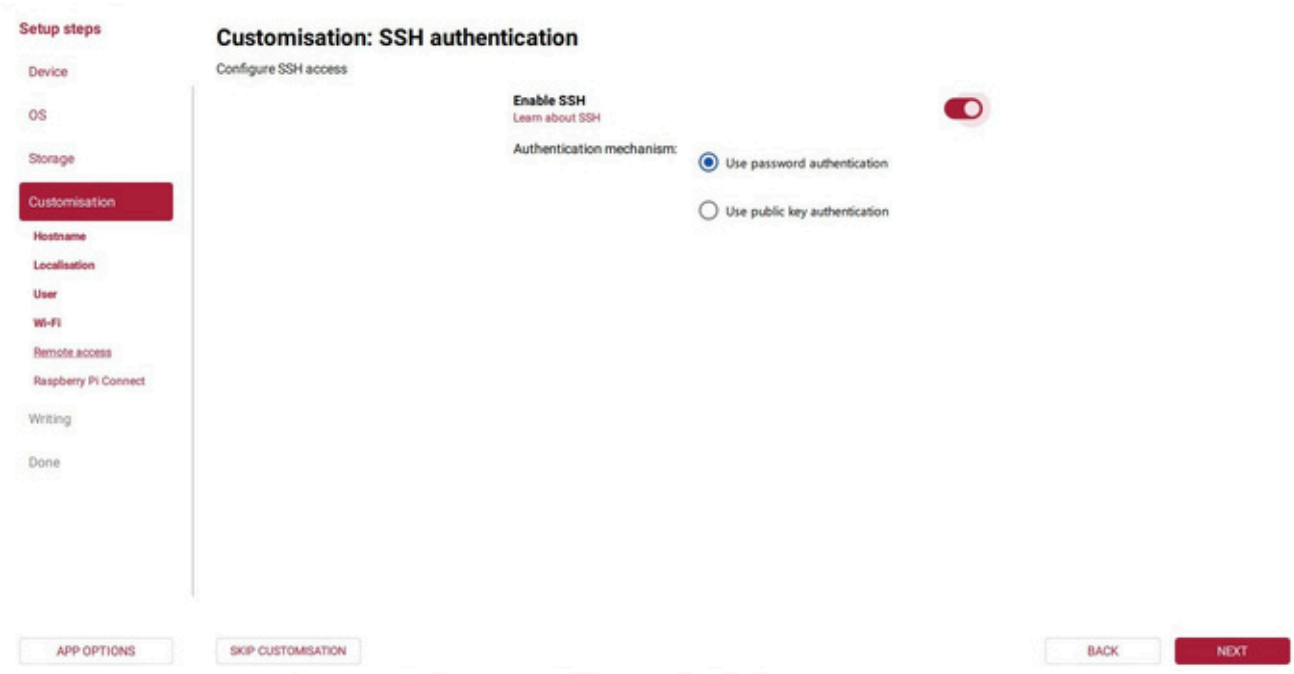
In the Customisation > Remote Access tab, configure your SSH settings.

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

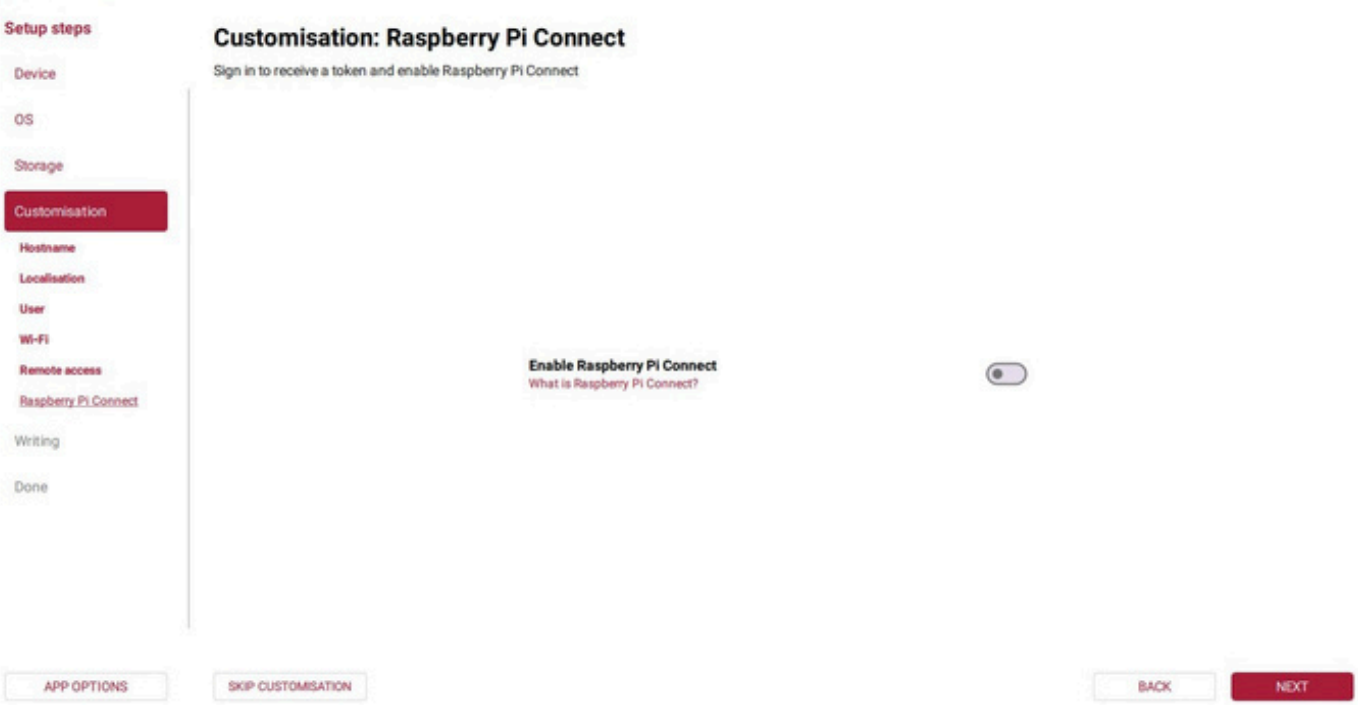
If you plan to run a headless Raspberry Pi, either enable this option or configure Raspberry Pi Connect for your device.

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)



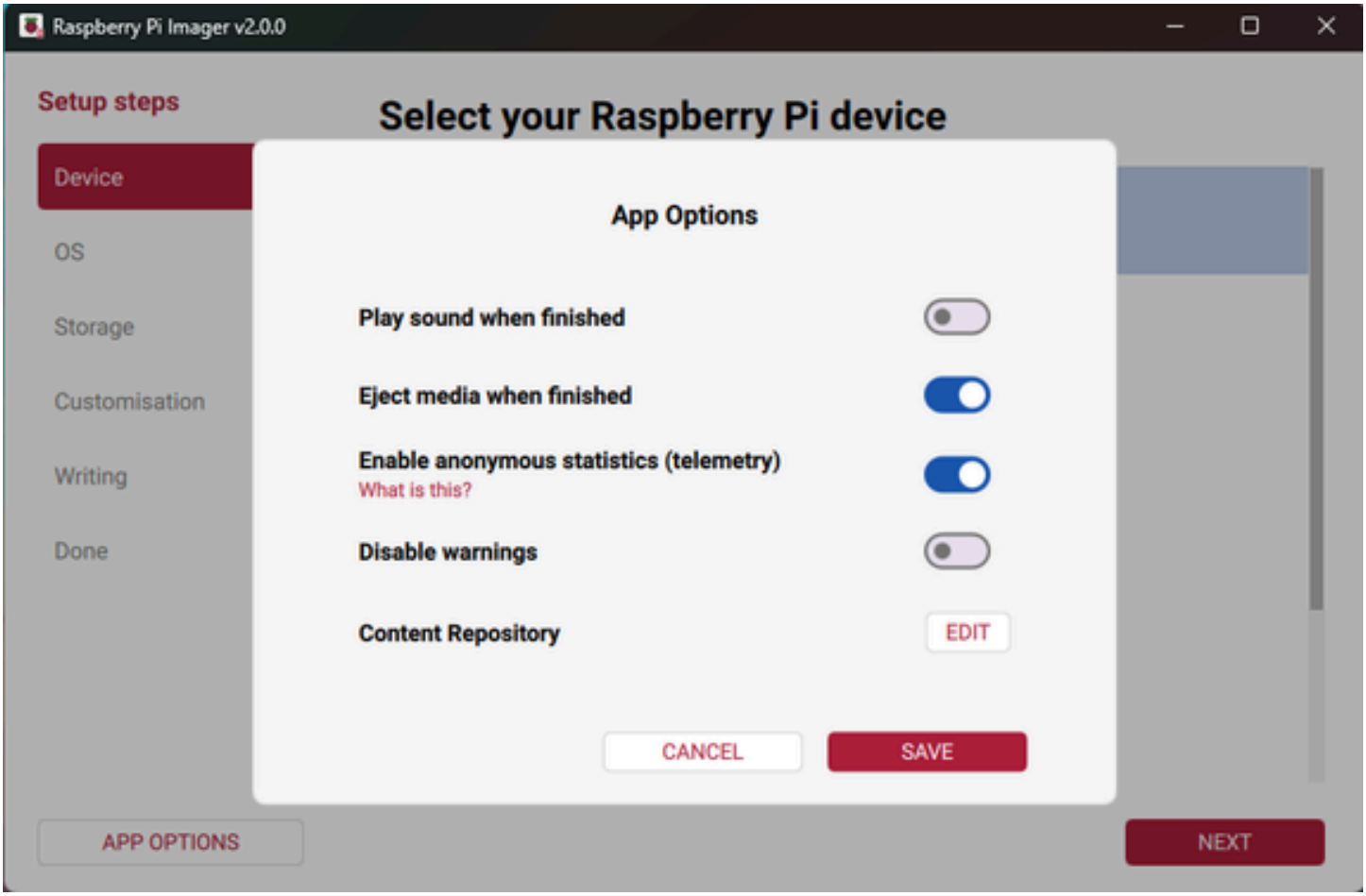
Step 4. Set the Imager options

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.

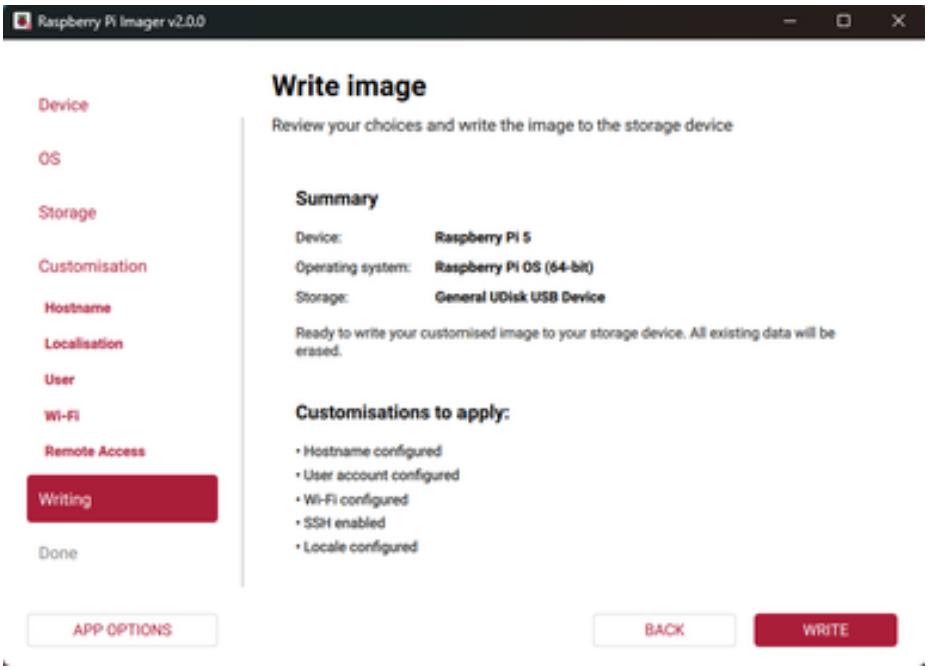
(Advanced users) The App Options dialog box also lets you set a custom repository to retrieve the operating system and devices list from. For more information, see Imager repo URL.



Step 5. Write the image to the storage device

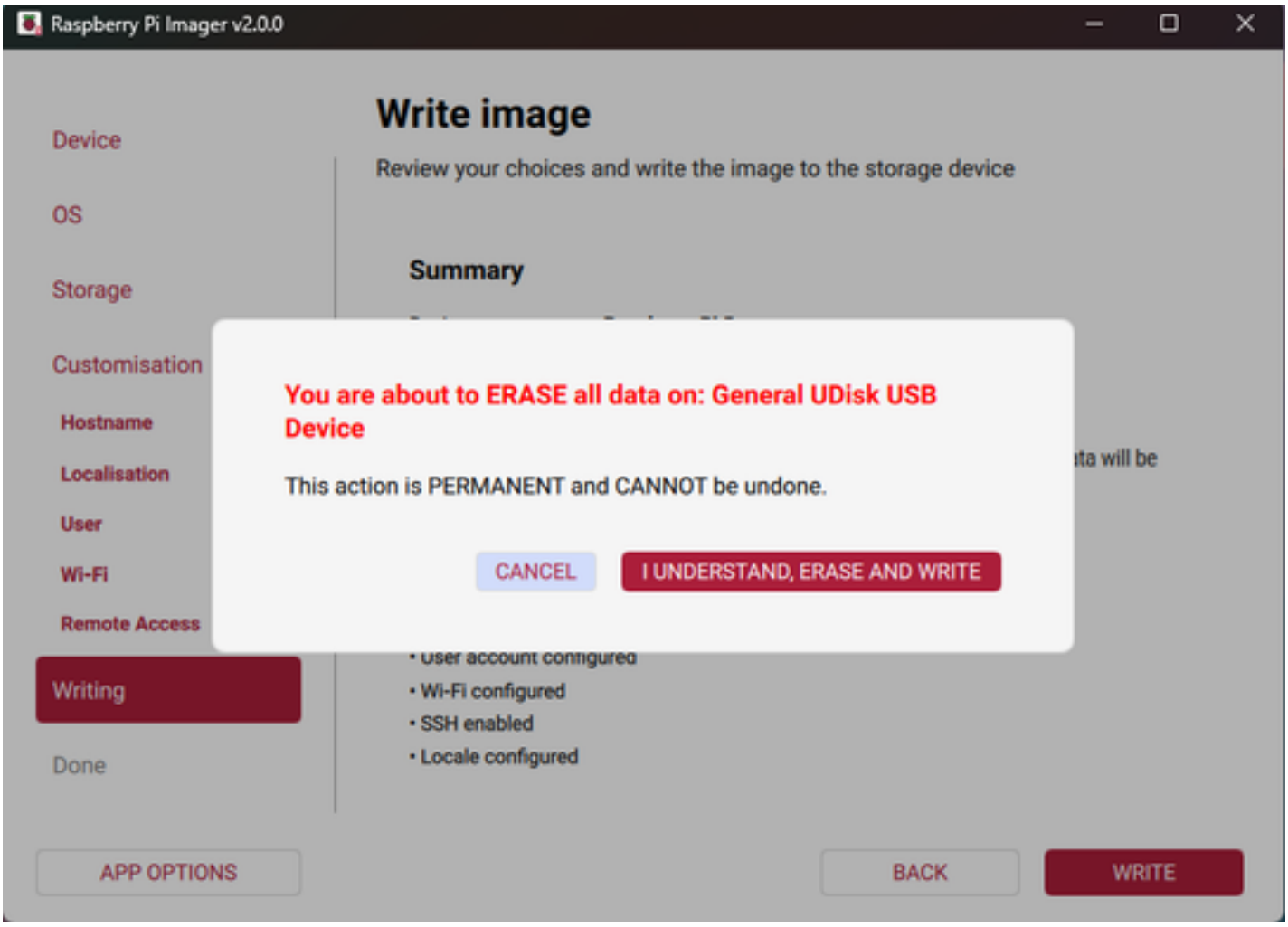
After you've gone through the tabs in Imager, it shows a summary page where you can review your choices.

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

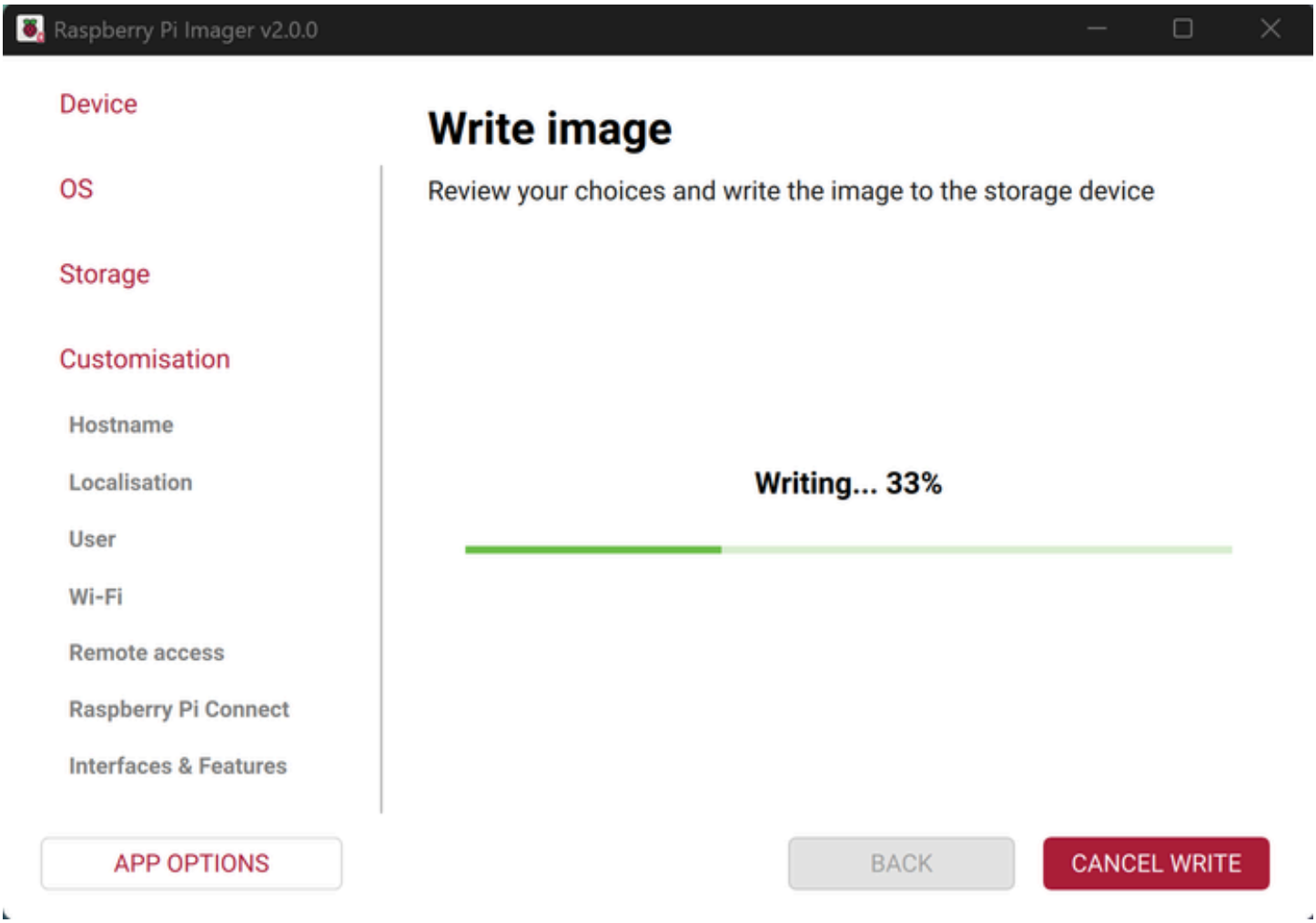


- Imager shows a prompt warning you that you're about to overwrite all data on your storage media. To continue, select I understand, erase and write.

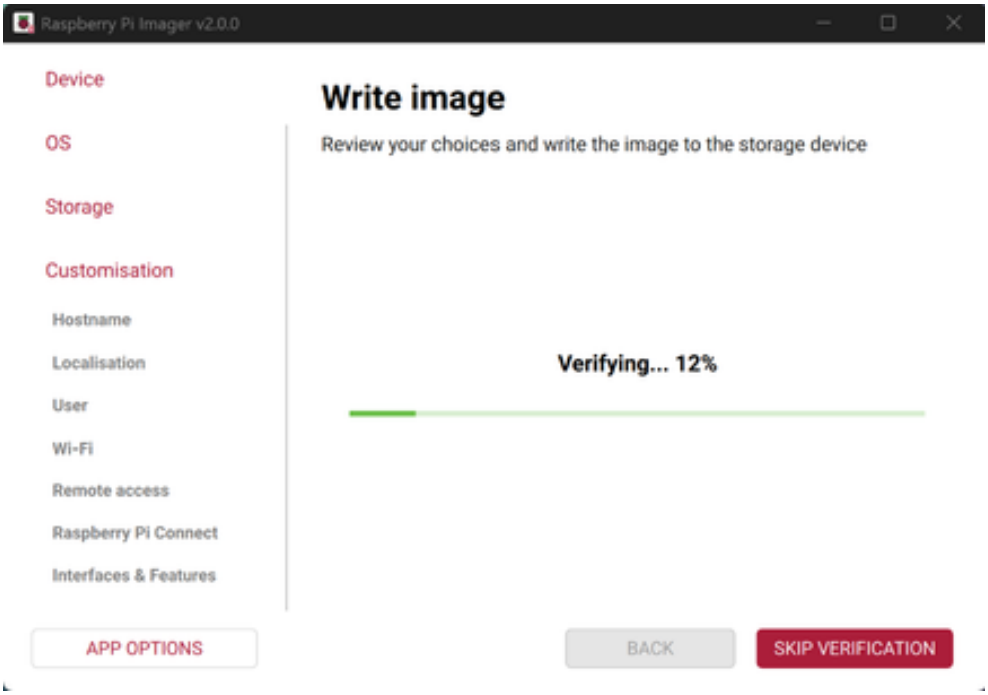
If you see an admin prompt asking for permissions to read and write to your storage medium, grant Imager the permissions to proceed.



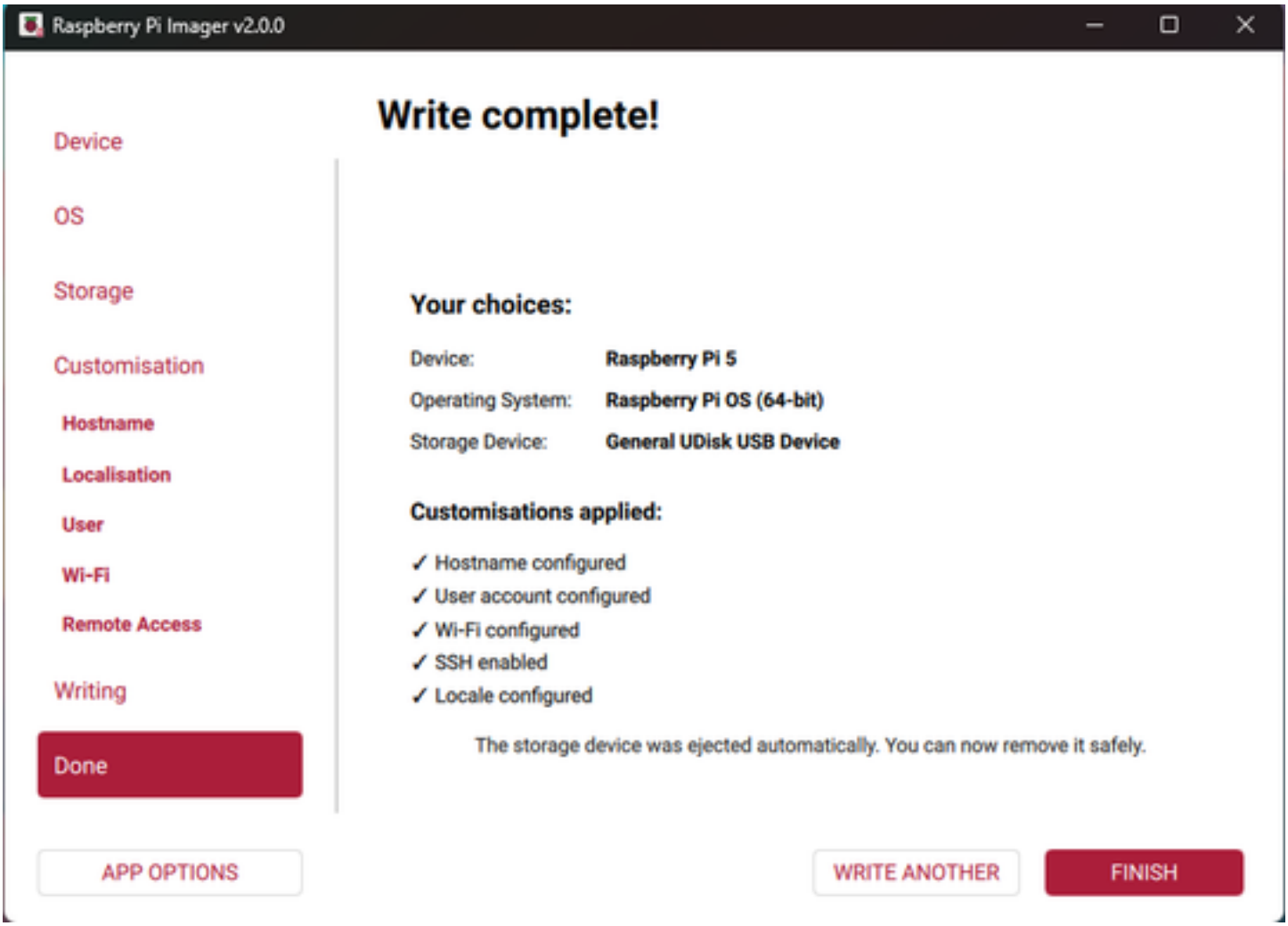
Imager begins writing your image to the storage media. This can take a few minutes; grab a cup of coffee while you wait.



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.

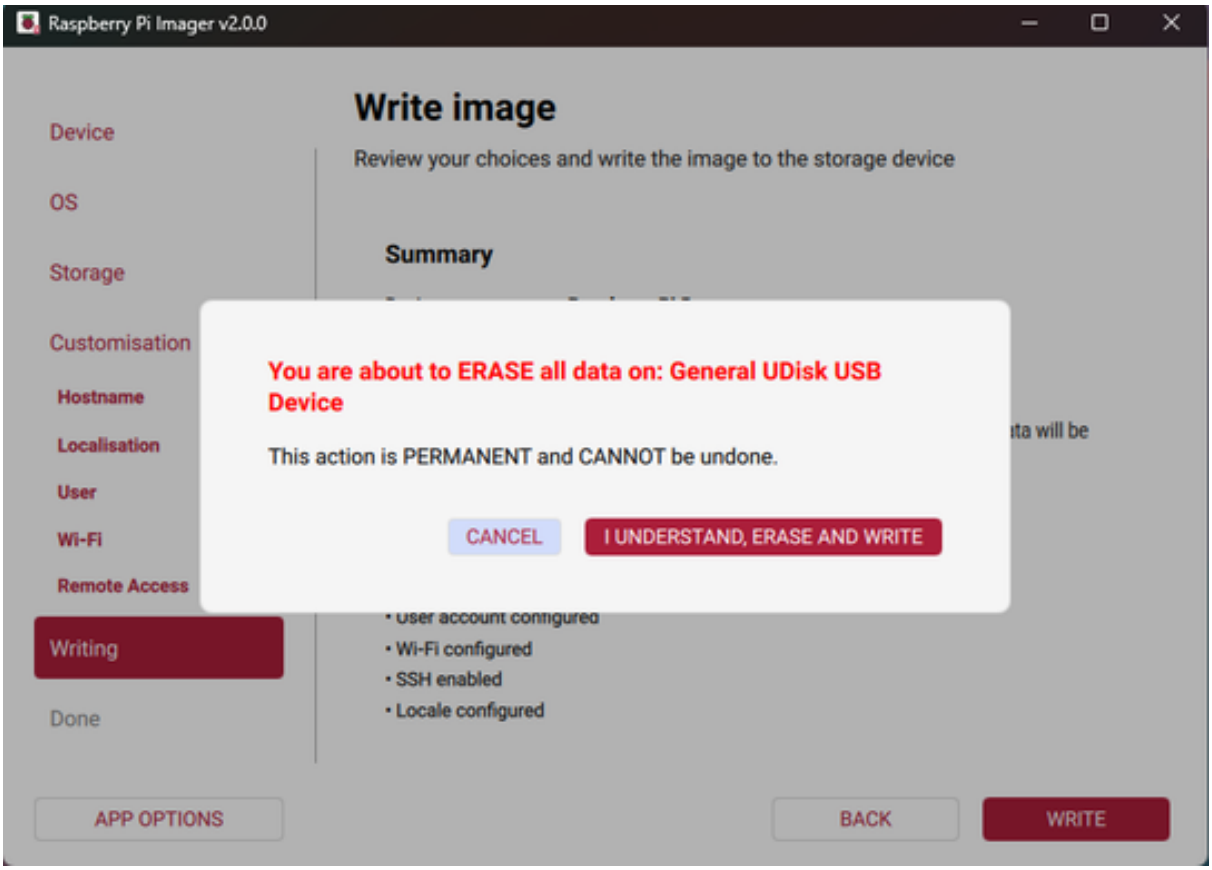


After the image is written and verified, Imager shows the Done tab. Select Finish to exit Imager.



SSG EMBEDDED SOLUTIONS

info@ssges.co.in

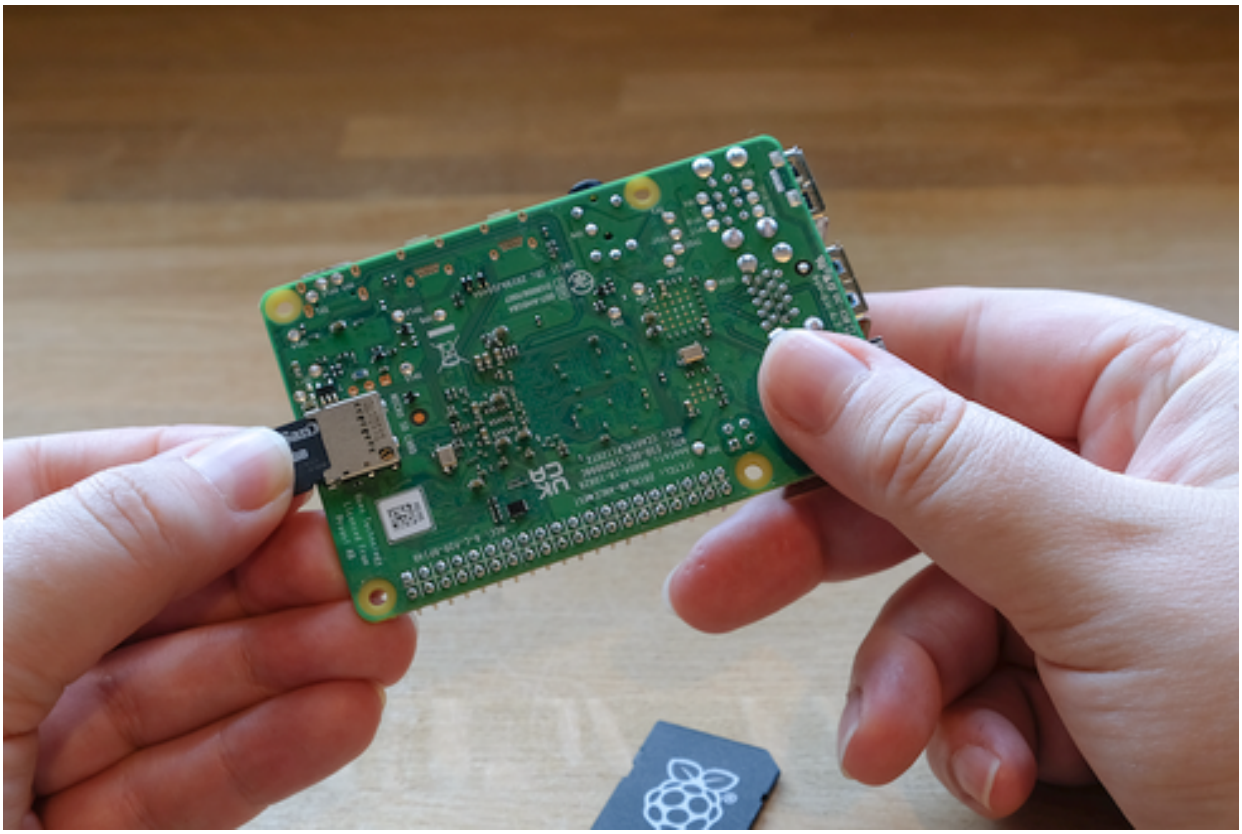


After the image is written and verified, Imager shows the Done tab. Select Finish to exit Imager.

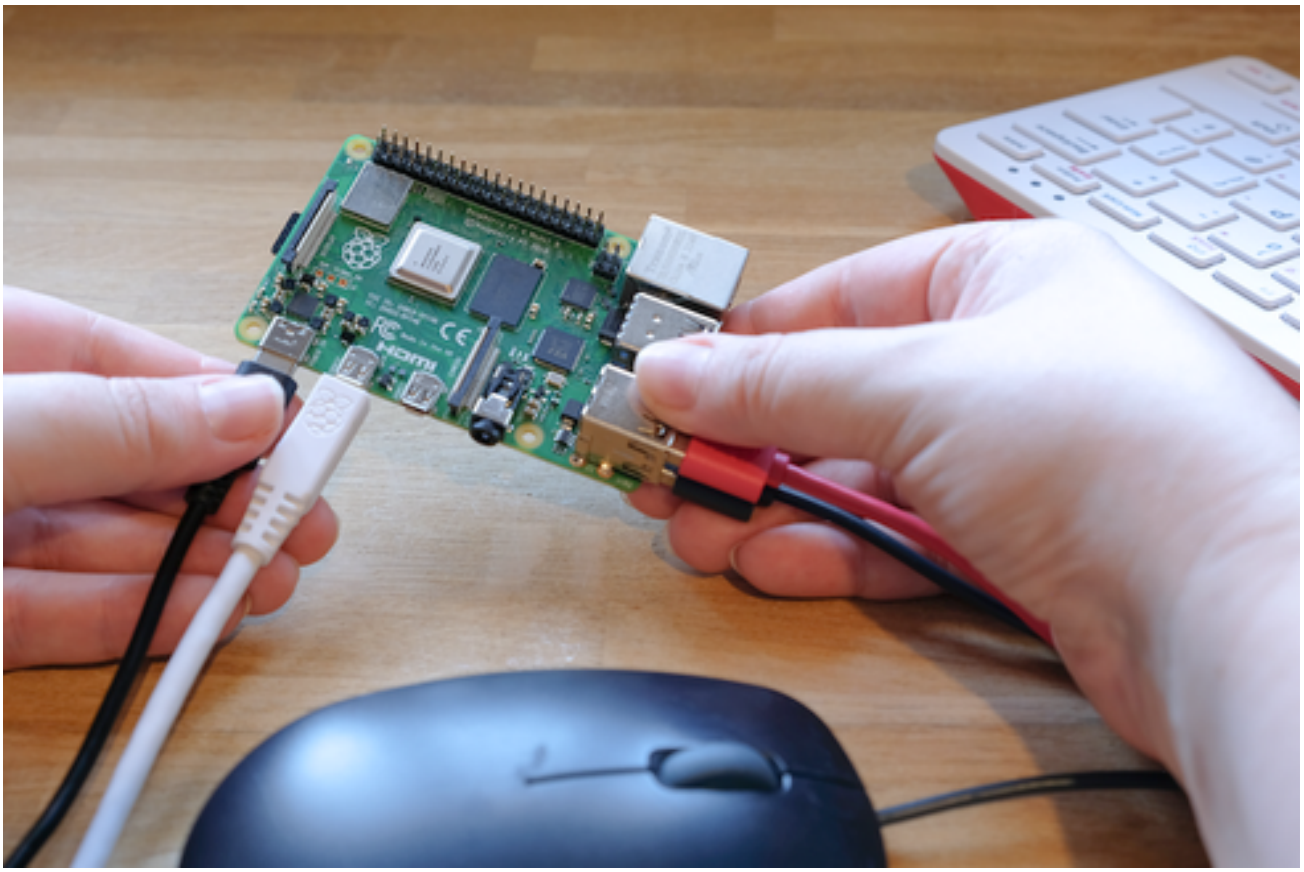
Set up your Raspberry Pi

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.

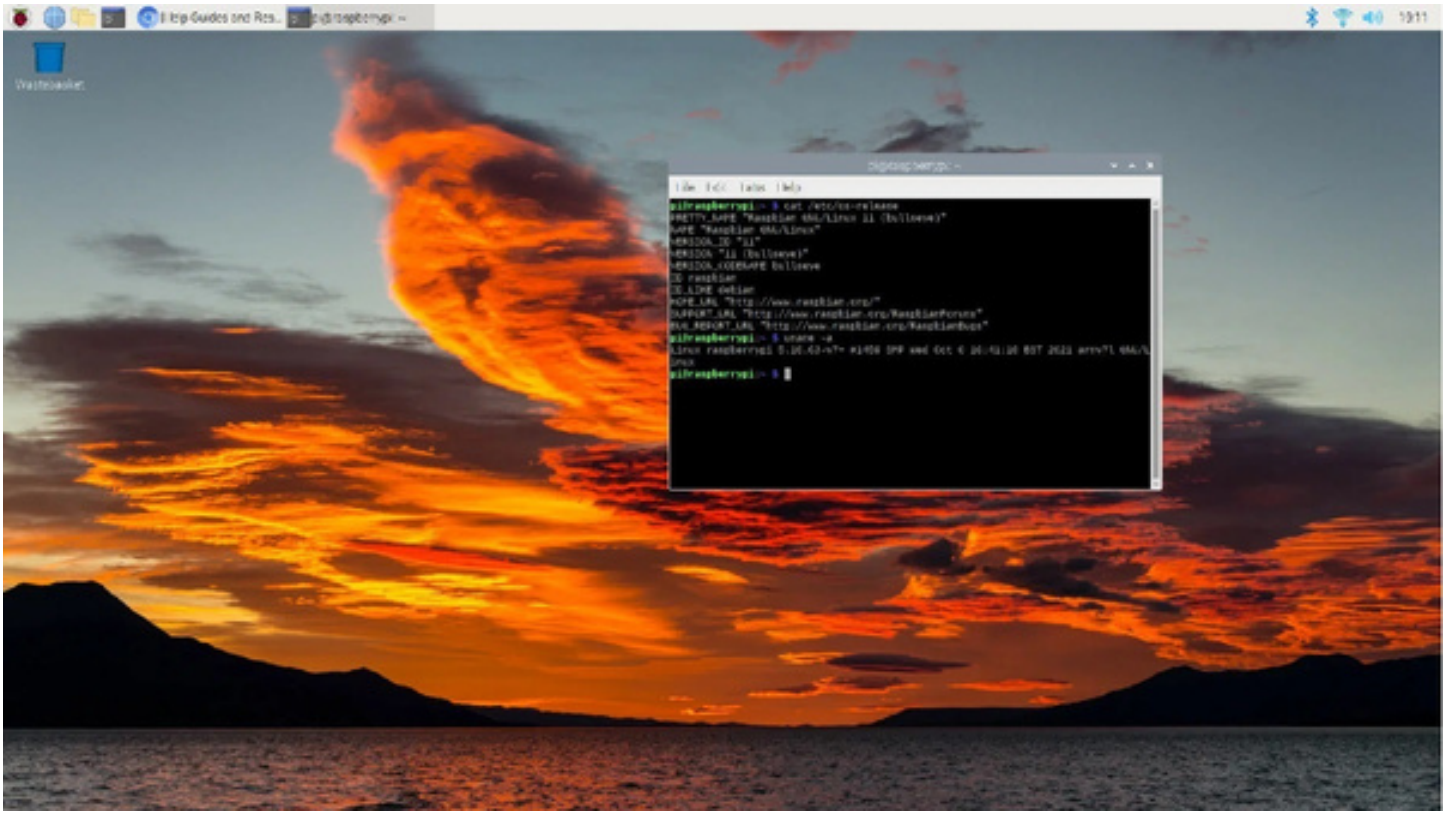


Then, plug in any other peripherals, such as your mouse, keyboard, and monitor.



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

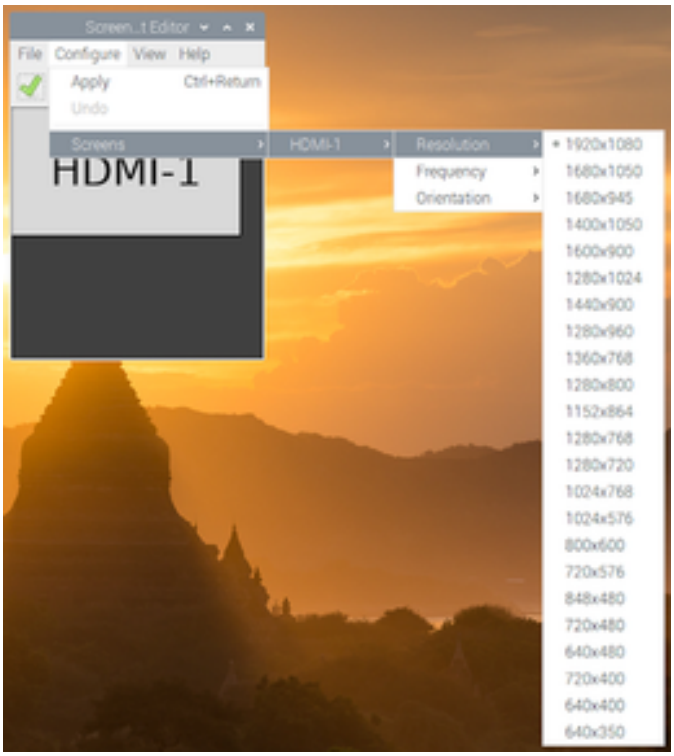


Changing Your Screen Resolution on Raspberry Pi

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.

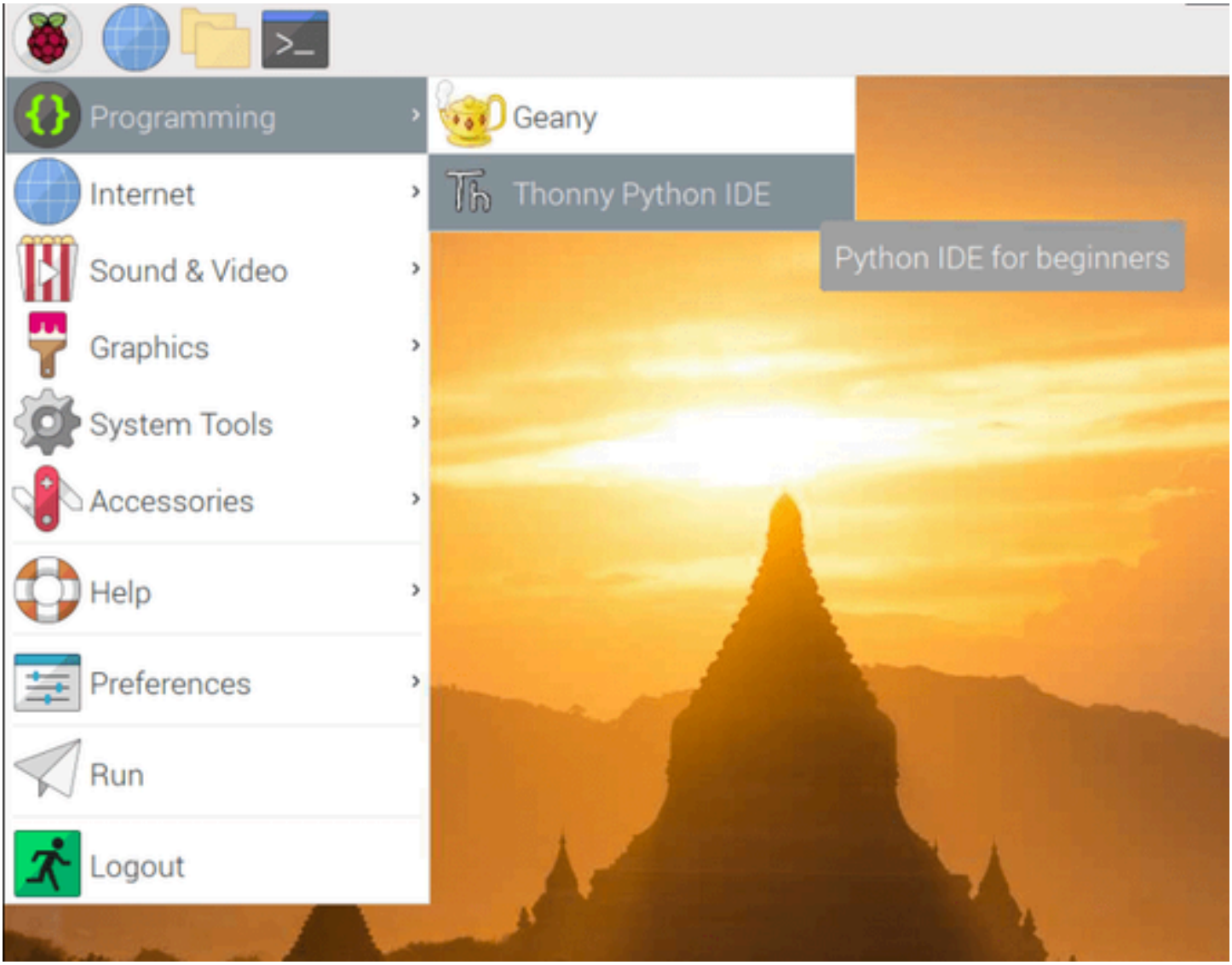


1. Select yes and reboot.

info@ssges.co.in

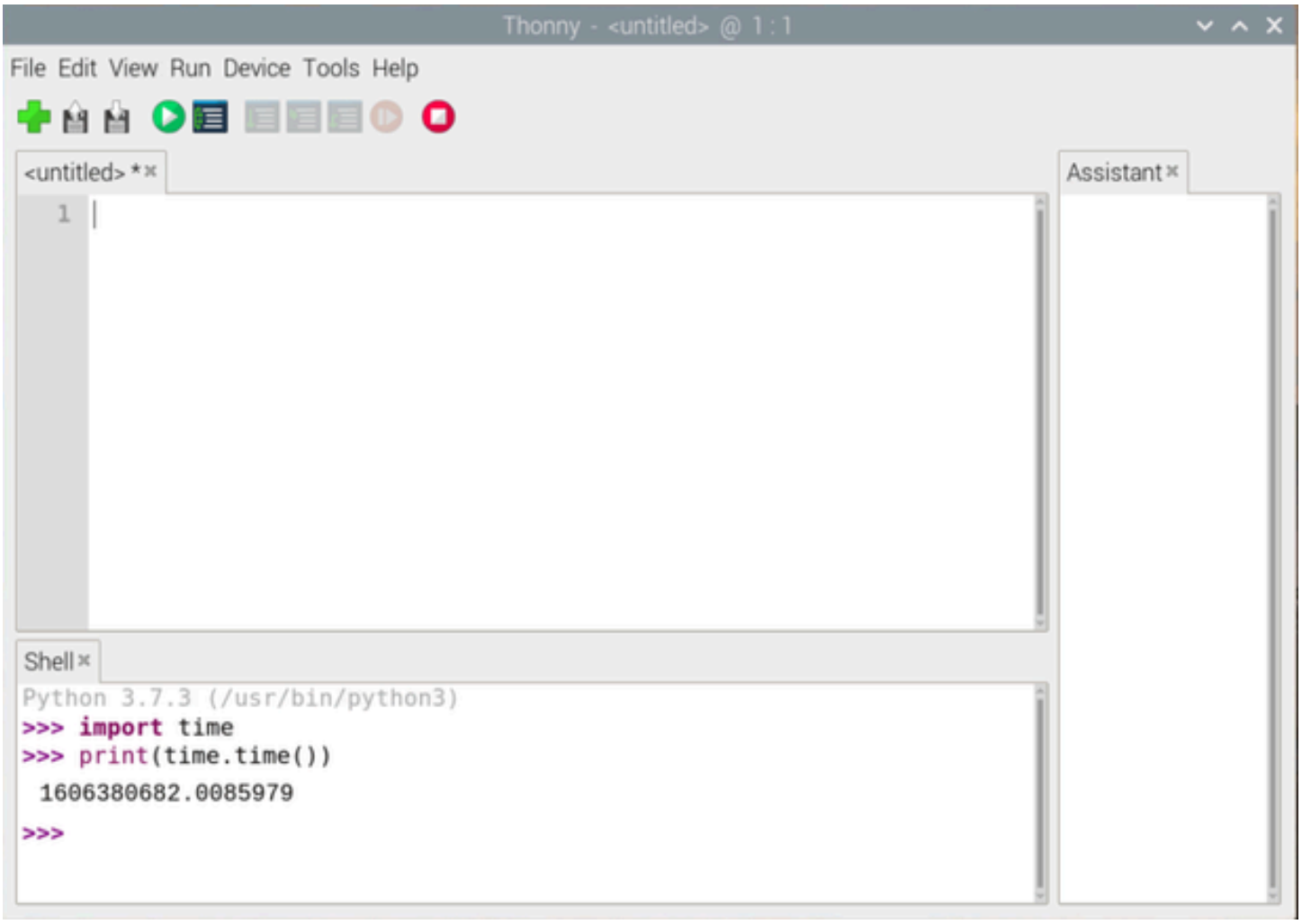
How to start Thonny IDE on Raspberry Pi

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.

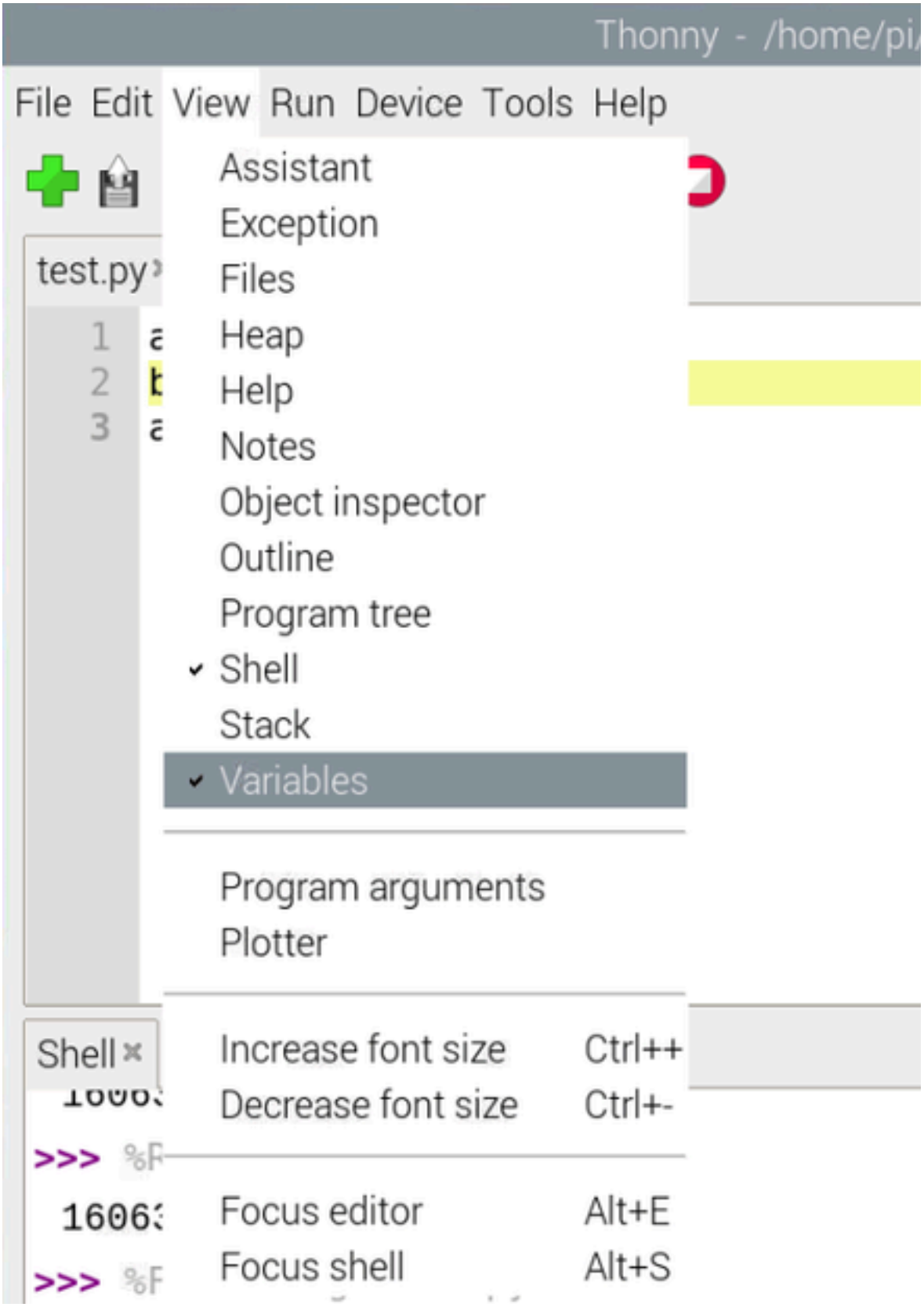


SSG EMBEDDED SOLUTIONS

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

On board Interfaces Available

- 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)


SSG
Embedded solutions

•