TensorFlow Image Analysis Tutorial

Ryan Wood

Updated from Rutuja Shivarkar's previous tutorial

July 11, 2022

Developing an occupancy counting algorithm from thermal images using Python

If you have any questions, feel free to email me at rpw6fx@virginia.edu

Table of Contents

List of Figures

Objective

The objective of this research project is to develop and train an algorithm that is counts the number of individuals in a room using a thermal camera to make energy and cost-efficient decisions regarding the heating and air conditioning in the room. An algorithm developed by the computer with machine learning training was tested using validation images through a Python script. The final part of the project is to test the system with real-time image capturing and detection.

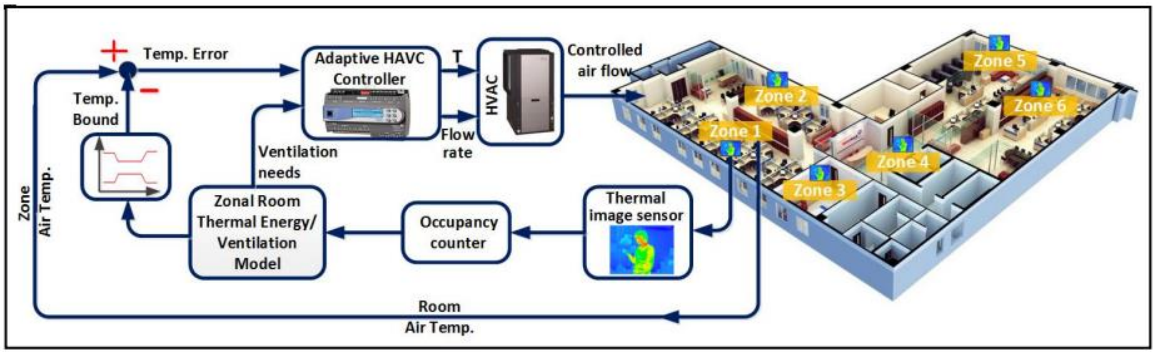


Figure 1. The proposed system architecture to detect the presence of people

List of Materials

CanaKit Raspberry Pi 4 Starter Kit

- Raspberry Pi 4 Model B (4 GB RAM)
- SanDisk 32 GB MicroSD Card
- Raspberry Pi Premium Case
- CanaKit USB-C Power Supply
- CanaKit Fan
- Micro HDMI Cable
- USB Card Reader
- Set of 3 Aluminum Heat Sinks
- CanaKit Quick-Start Guide
- CanaKit GPIO Reference Card

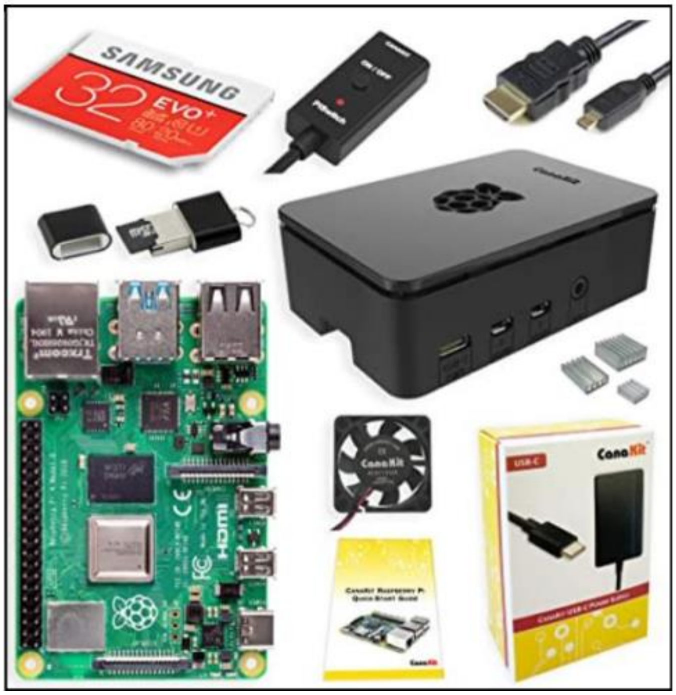


Figure 2. CanaKit Raspberry Pi 4 Starter Kit

An exclusive Starter Kit for Raspberry Pi from CanaKit includes the Raspberry Pi 4. The Raspberry Pi 4 offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation boards while retaining backward compatibility and similar power consumption. The Raspberry Pi 4 provides desktop performance comparable to entry-level x86 PC systems. The Raspberry Pi 4 comes in three onboard RAM options for even further performance.

Due to the higher power requirements, the Raspberry Pi 4 requires at least a 3.0A USB-C power supply. The CanaKit 3.5A USB-C power supply (UL Listed) that is included in this kit is specially designed and tested for the Raspberry Pi 4. This power supply incorporates a noise filter.

The standard HDMI port that was part of previous generation Raspberry Pi generation boards is replaced on the Raspberry Pi 4 by two Micro HDMI ports to provide dual monitor support. The CanaKit Starter Kit includes a high-quality 4K60P Micro HDMI to HDMI cable (6 feet) to provide the best output display resolution support.

The kit includes the popular CanaKit Raspberry Pi case adapted to be fully compatible with the Pi 4 and comes in a black high-gloss finish, as well as a set of 3 Aluminum Heat Sinks and a CanaKit low-noise fan to help keep the Pi 4 running cool.

HDMI Cable

- Raspberry Pi has an HDMI output port that is compatible with the HDMI port of most modern computer monitors. Many computer monitors may also have DVI or VGA ports. Raspberry Pi 4 has two micro-HDMI ports, allowing us to connect two separate monitors. We need either a micro-HDMI to HDMI cable or a standard HDMI to HDMI cable plus a micro-HDMI to HDMI adapter, to connect Raspberry Pi 4 to a screen.

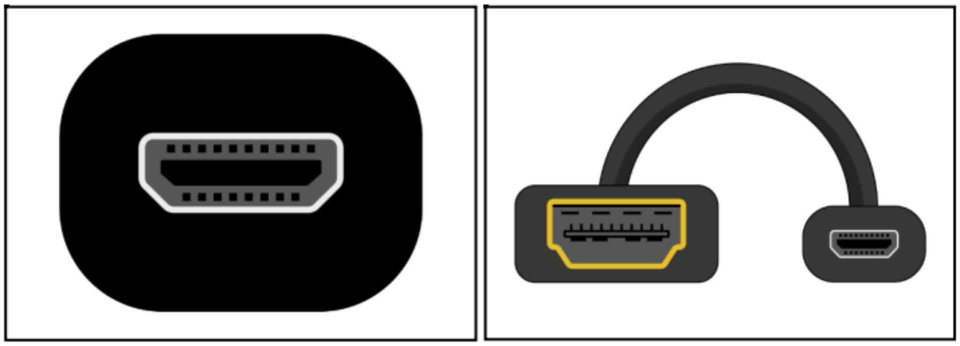


Figure 3. Micro-HDMI to HDMI Adapter

VGA Cable

- Some screens only have a VGA port. To connect the Raspberry Pi to such a screen, we can use an HDMI to VGA adapter.

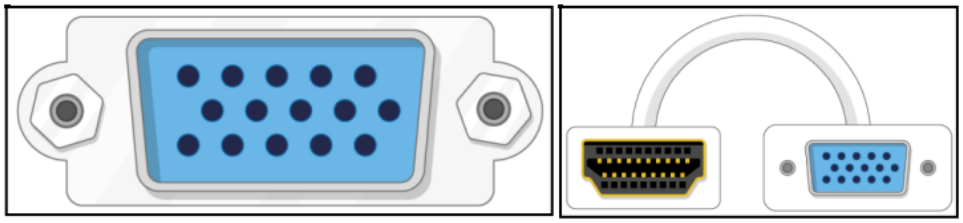


Figure 4. HDMI to VGA Adapter

FLIR Lepton 2.5 (Thermal Image Sensor 80 x 60)

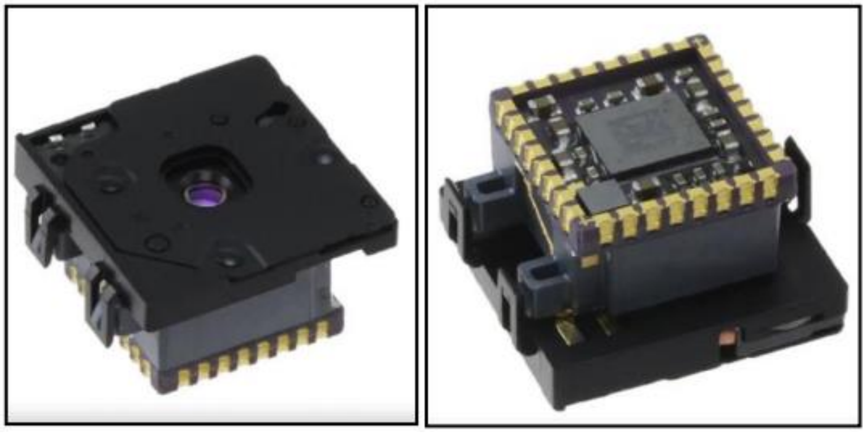
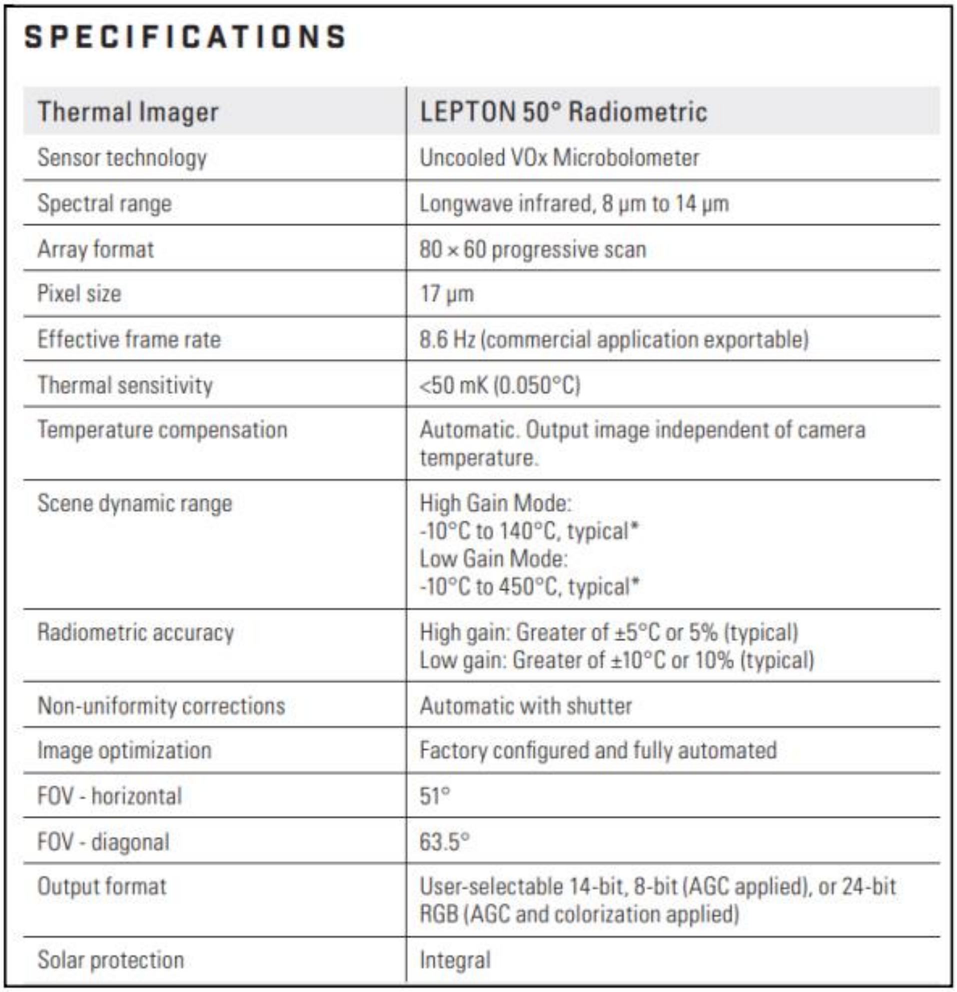


Figure 5. FLIR Lepton 2.5 (Thermal Image Sensor 80 x 60)

The FLIR Lepton® is a revolutionary radiometric-capable LWIR camera solution that is smaller than a dime, to inside a smartphone, and is one-tenth the cost of traditional IR cameras. With a focal plane array of 80 x 60 active pixels, Lepton easily integrates into native mobile devices and other electronics as an IR sensor or thermal imager.

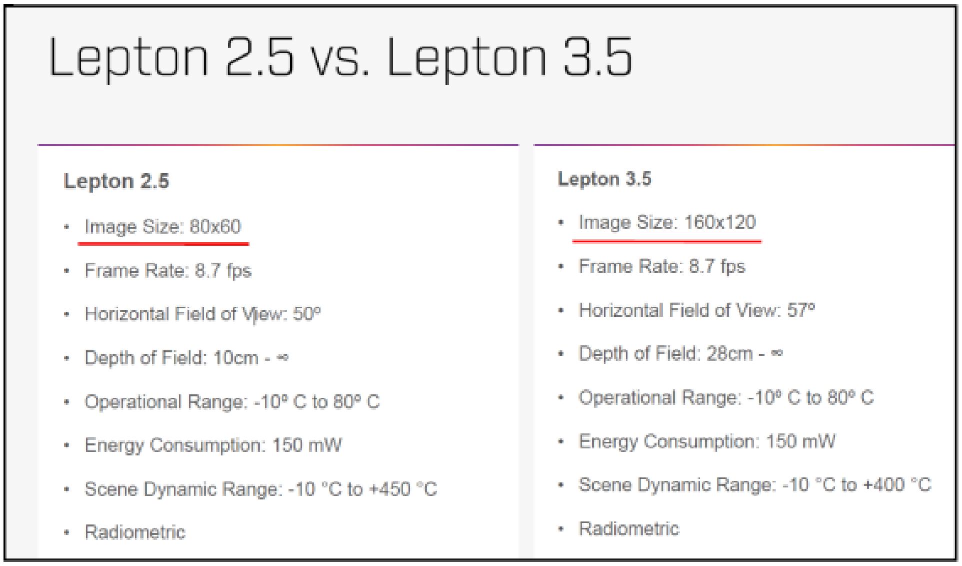
The radiometric Lepton captures accurate, calibrated, and noncontact temperature data in every pixel of each image for even greater utility in commercial applications. Non-radiometric versions are also available. Lepton is the technology behind many groundbreaking thermal imagers from FLIR, from the FLIR ONE thermal imaging accessory for smartphones to the MR176 Imaging Moisture Meter. New devices equipped with thermal imaging have incredible potential in the areas of safety, temperature measurement, and presence detection. And due to high volume manufacturing techniques, FLIR can deliver Lepton at a price point that is an order of magnitude below other thermal camera cores.



SPECIFICATIONS

Thermal Imager	LEPTON 50° Radiometric
Sensor technology	Uncooled VOx Microbolometer
Spectral range	Longwave infrared, 8 µm to 14 µm
Array format	80 × 60 progressive scan
Pixel size	17 µm
Effective frame rate	8.6 Hz (commercial application exportable)
Thermal sensitivity	<50 mK (0.050°C)
Temperature compensation	Automatic. Output image independent of camera temperature.
Scene dynamic range	High Gain Mode: -10°C to 140°C, typical* Low Gain Mode: -10°C to 450°C, typical*
Radiometric accuracy	High gain: Greater of ±5°C or 5% (typical) Low gain: Greater of ±10°C or 10% (typical)
Non-uniformity corrections	Automatic with shutter
Image optimization	Factory configured and fully automated
FOV - horizontal	51°
FOV - diagonal	63.5°
Output format	User-selectable 14-bit, 8-bit (AGC applied), or 24-bit RGB (AGC and colorization applied)
Solar protection	Integral

Figure 6. Specifications of the FLIR Lepton 2.5



Lepton 2.5 vs. Lepton 3.5

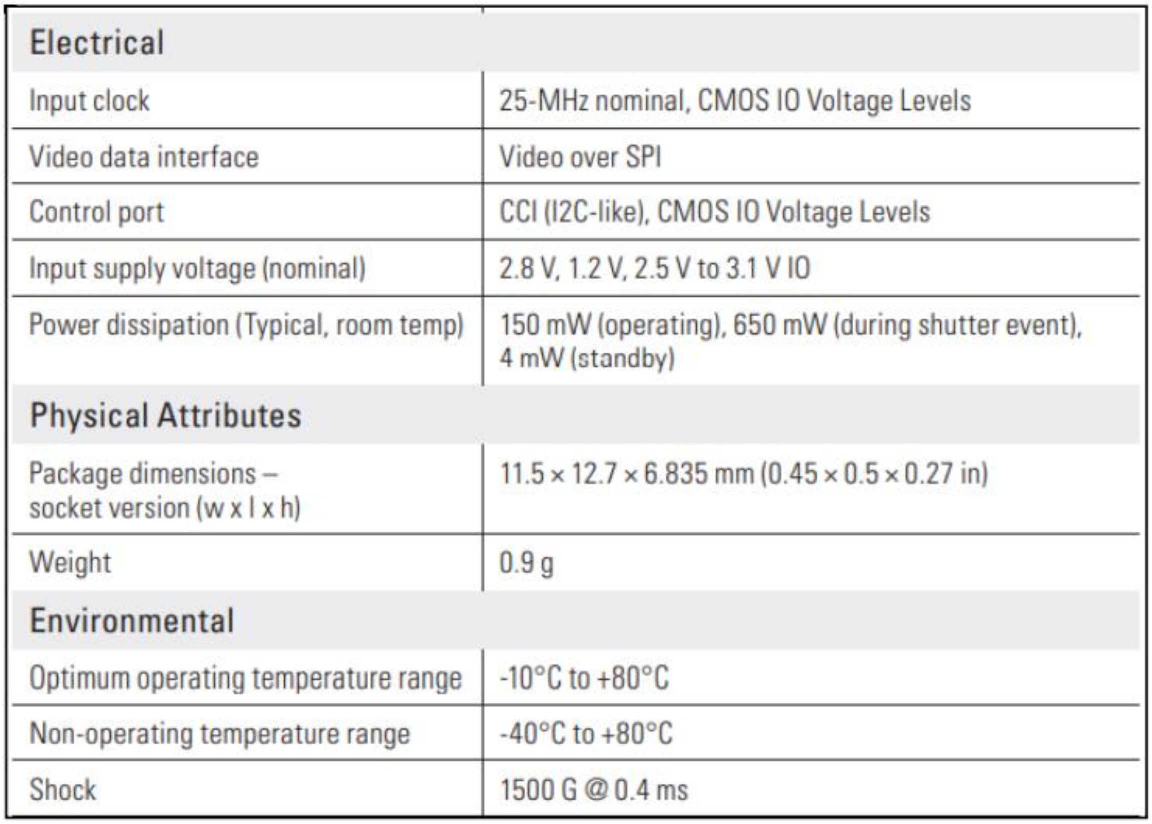
Lepton 2.5

- Image Size: 80x60
- Frame Rate: 8.7 fps
- Horizontal Field of View: 50°
- Depth of Field: 10cm - ∞
- Operational Range: -10° C to 80° C
- Energy Consumption: 150 mW
- Scene Dynamic Range: -10 °C to +450 °C
- Radiometric

Lepton 3.5

- Image Size: 160x120
- Frame Rate: 8.7 fps
- Horizontal Field of View: 57°
- Depth of Field: 28cm - ∞
- Operational Range: -10° C to 80° C
- Energy Consumption: 150 mW
- Scene Dynamic Range: -10 °C to +400 °C
- Radiometric

Figure 7. Comparison between the FLIR Lepton 2.5 and FLIR Lepton 3.5



Electrical	
Input clock	25-MHz nominal, CMOS IO Voltage Levels
Video data interface	Video over SPI
Control port	CCI (I2C-like), CMOS IO Voltage Levels
Input supply voltage (nominal)	2.8 V, 1.2 V, 2.5 V to 3.1 V IO
Power dissipation (Typical, room temp)	150 mW (operating), 650 mW (during shutter event), 4 mW (standby)
Physical Attributes	
Package dimensions – socket version (w x l x h)	11.5 × 12.7 × 6.835 mm (0.45 × 0.5 × 0.27 in)
Weight	0.9 g
Environmental	
Optimum operating temperature range	-10°C to +80°C
Non-operating temperature range	-40°C to +80°C
Shock	1500 G @ 0.4 ms

Figure 8. Electrical Specifications of the FLIR Lepton 2.5

FLIR Lepton® Camera Breakout Board v2.0

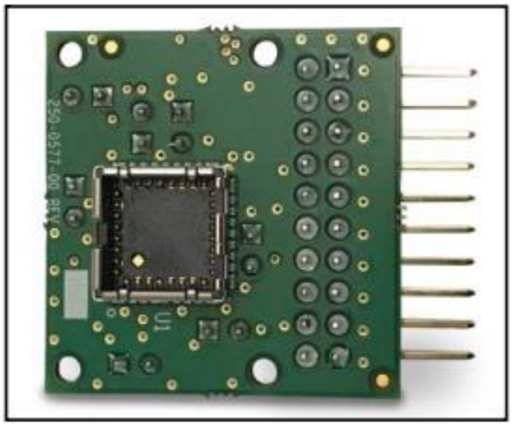
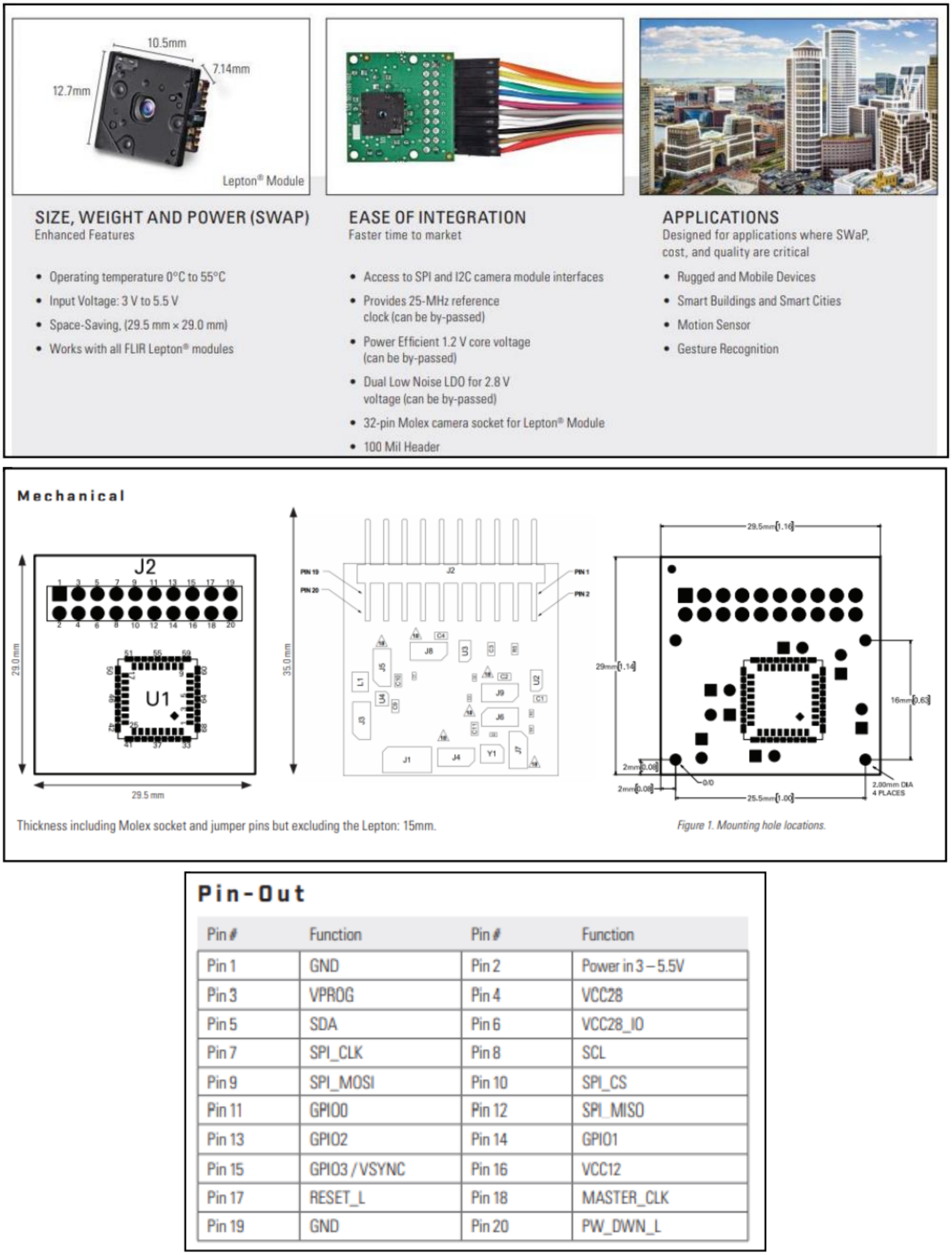


Figure 9. FLIR Lepton Camera Breakout Board v2.0

The FLIR Lepton® Thermal Camera Breakout Board is an easy-to-interface evaluation board to quickly connect all versions of the FLIR Lepton camera module to common platforms like Raspberry Pi* or custom hardware such as mobile development kits. It provides onboard power supplies, generated from 3 – 5.5V, and a master clock. Local power supplies, the master clock, and the powerup sequence components can all be bypassed using a jumper.


10.5mm
7.14mm
12.7mm
Lepton® Module

SIZE, WEIGHT AND POWER (SWAP)
Enhanced Features

- Operating temperature 0°C to 55°C
- Input Voltage: 3 V to 5.5 V
- Space-Saving, (29.5 mm × 29.0 mm)
- Works with all FLIR Lepton® modules

EASE OF INTEGRATION
Faster time to market

- Access to SPI and I2C camera module interfaces
- Provides 25-MHz reference clock (can be by-passed)
- Power Efficient 1.2 V core voltage (can be by-passed)
- Dual Low Noise LDO for 2.8 V voltage (can be by-passed)
- 32-pin Molex camera socket for Lepton® Module
- 100 Mil Header

APPLICATIONS
Designed for applications where SWaP, cost, and quality are critical

- Rugged and Mobile Devices
- Smart Buildings and Smart Cities
- Motion Sensor
- Gesture Recognition

Mechanical

Thickness including Molex socket and jumper pins but excluding the Lepton: 15mm.

Figure 1. Mounting hole locations.

Pin-Out

Pin #	Function	Pin #	Function
Pin 1	GND	Pin 2	Power in 3 – 5.5V
Pin 3	VPROG	Pin 4	VCC28
Pin 5	SDA	Pin 6	VCC28_IO
Pin 7	SPI_CLK	Pin 8	SCL
Pin 9	SPI_MOSI	Pin 10	SPI_CS
Pin 11	GPIO0	Pin 12	SPI_MISO
Pin 13	GPIO2	Pin 14	GPIO1
Pin 15	GPIO3 / VSYNC	Pin 16	VCC12
Pin 17	RESET_L	Pin 18	MASTER_CLK
Pin 19	GND	Pin 20	PW_DWN_L

Figure 10. Specifications and Pin-Out diagram of the FLIR Lepton Camera Breakout Board v2.0

Jumper Wires

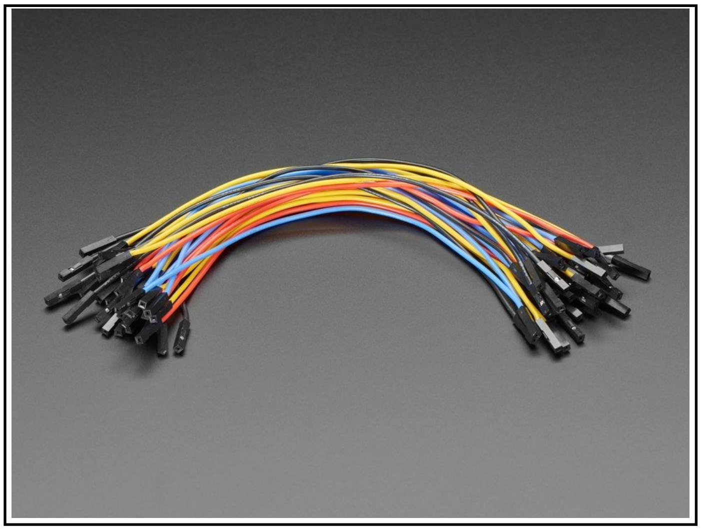


Figure 11. Female - Female Jumper Wires

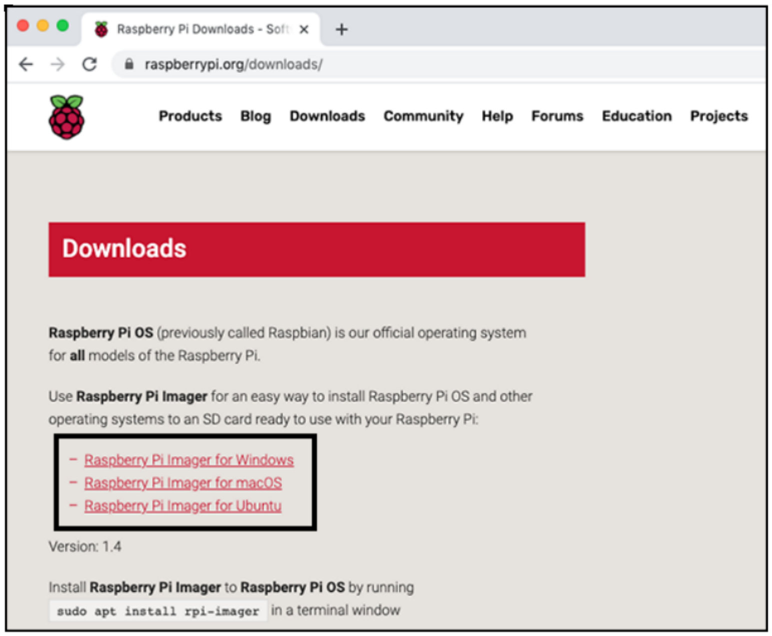
Computer Monitor with an HDMI port

Keyboard and Mouse

Procedure

Setting up the SD Card

- To install Raspberry Pi OS on the SD card, we 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 our SD card.



Downloads

Raspberry Pi OS (previously called Raspbian) is our official operating system for **all** models of the Raspberry Pi.

Use **Raspberry Pi Imager** for an easy way to install Raspberry Pi OS and other operating systems to an SD card ready to use with your Raspberry Pi:

- Raspberry Pi Imager for Windows
- Raspberry Pi Imager for macOS
- Raspberry Pi Imager for Ubuntu

Version: 1.4

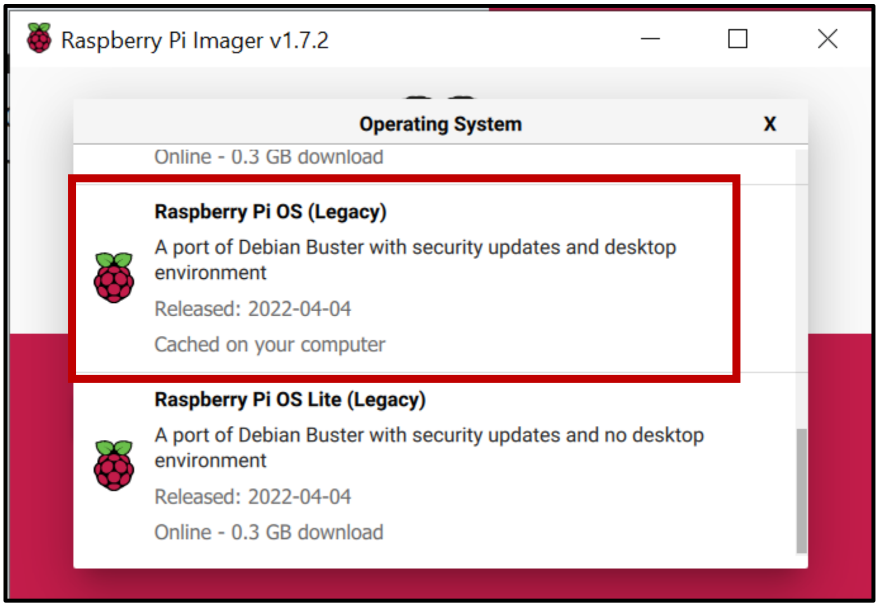
Install **Raspberry Pi Imager** to **Raspberry Pi OS** by running `sudo apt install rpi-imager` in a terminal window

Figure 12. Downloading the Raspberry Pi Imager onto your device

- Go to raspberrypi.com/software/ and click on the link for the Raspberry Pi Imager that matches your operating system
- When the download finishes, click on it to launch the installer

Using the Raspberry Pi Imager

- Anything that is stored on the SD card will be overwritten during formatting. If the SD card currently has any files on it, e.g., from an older version of Raspberry Pi OS, we can back up these files first to prevent permanently losing them.
- Follow the instructions to install and run the Raspberry Pi Imager.
- Insert the SD card into the computer or laptop SD card slot.
- In the Raspberry Pi Imager, select the OS that you want to install and the SD card you would like to install it on.
- Note: You will need to be connected to the internet the first time for the Raspberry Pi Imager to download the OS that you choose. That OS will then be stored for future offline use. Being online for later uses means that the Raspberry Pi imager will always give you the latest version.
- After clicking "Choose OS", follow the prompt "Raspberry Pi OS (Other)", and find the Legacy OS that installs Debian Buster.
  - The current version (Debian Bullseye) does not support some of the features, so an earlier version must be installed to make the camera work.



Online - 0.3 GB download

Raspberry Pi OS (Legacy)
A port of Debian Buster with security updates and desktop environment
Released: 2022-04-04
Cached on your computer

Raspberry Pi OS Lite (Legacy)
A port of Debian Buster with security updates and no desktop environment
Released: 2022-04-04
Online - 0.3 GB download

Figure 13. Choosing the correct operating system

- Next, click on the "Choose Storage" prompt. Find the storage device that the SD card in plugged into and select that.
- Then, click the "WRITE" button and wait for the Raspberry Pi Imager to finish writing
- Once the installation is done, remove the SD card from the computer and SD card reader

Connecting the Raspberry Pi

- Now get everything connected to the Raspberry Pi. It is important to do this in the right order so that all the components are safe.

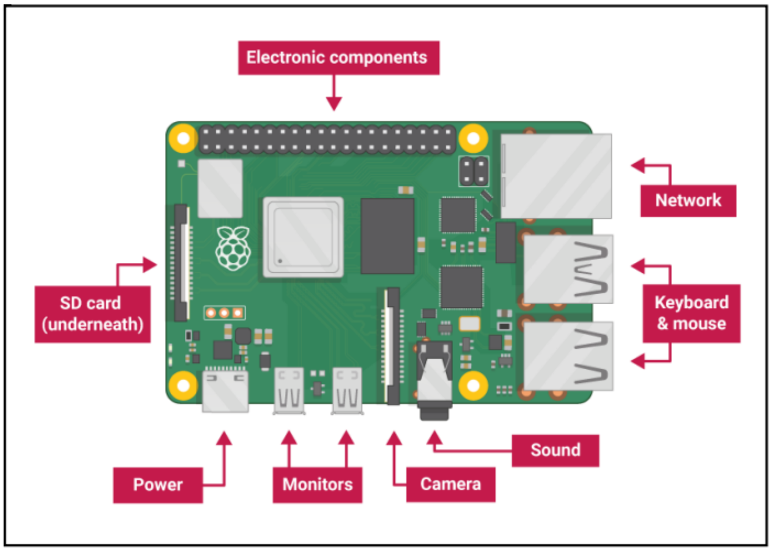


Figure 14. Components of the Raspberry Pi 4

- Insert the SD card we have set up with Raspberry Pi OS into the microSD card slot on the underside of the Raspberry Pi.

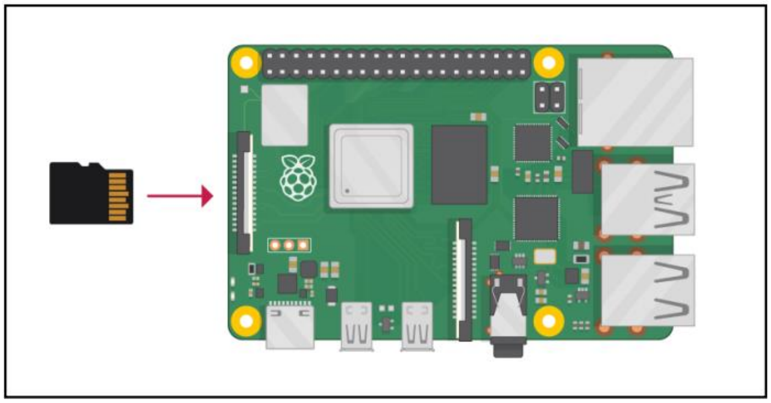


Figure 15. How to insert an SD card into the Raspberry Pi 4

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

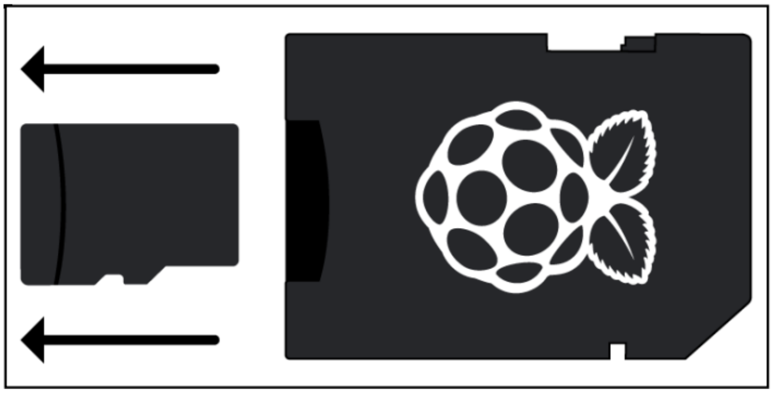


Figure 16. Taking an SD card out of a larger adapter

- Find the USD connector end of the mouse's cable and connect the mouse to a USB port on Raspberry Pi (it does not matter which port we use).

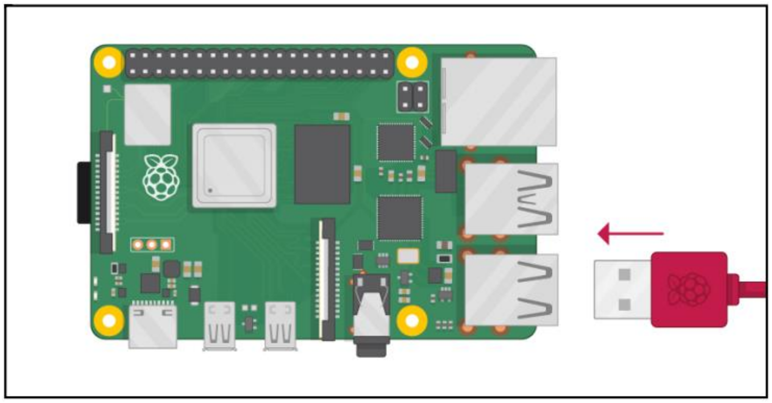


Figure 17. Plugging mouse into Raspberry Pi 4

- Connect the keyboard in the same way.

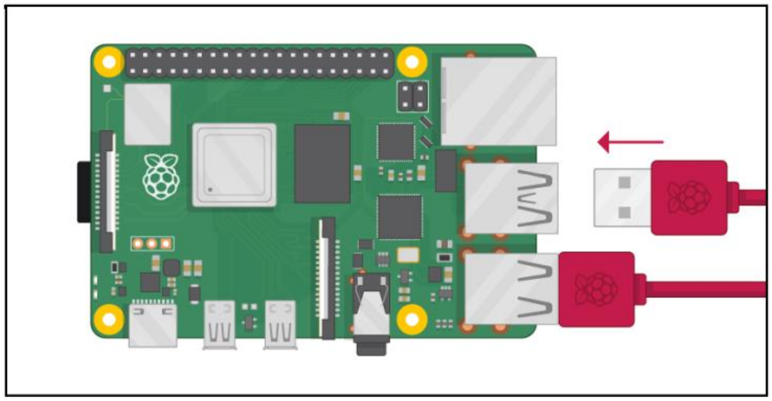


Figure 18. Plugging keyboard into Raspberry Pi 4

- Make sure the screen is plugged into a wall socker and switched ON. Look at the HDMI port(s) on the Raspberry Pi – notice that they have a flat side on top. Use a cable to connect the screen to the Raspberry Pi 4’s HDMI port – use an adapter if necessary. Connect the screen to the first of the Raspberry Pi 4’s HDMI ports, labeled HDMI 0.
    - Note: Make sure the cord is plugged into HDMI 0 rather than HDMI 1.

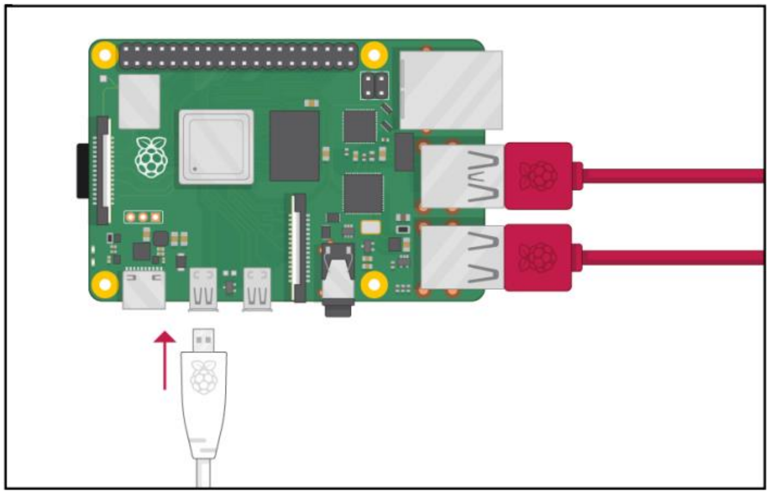


Figure 19. Plugging monitor into Raspberry Pi 4

- Note: Nothing will display on the screen because the Raspberry Pi is not running yet.

Starting up the Raspberry Pi

- The Raspberry Pi does not have a power switch. As soon as we connect it to a power outlet, it will turn ON. Plug the power supply into a socket and connect it to the Raspberry Pi’s power port.

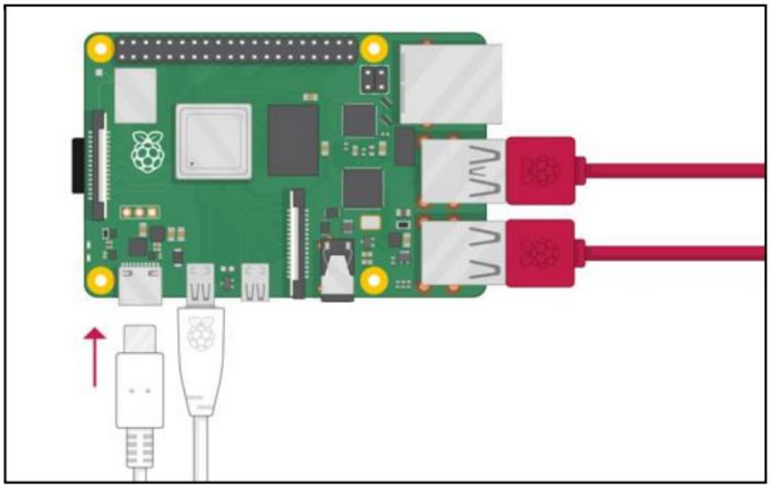


Figure 20. Plugging power supply cord into Raspberry Pi 4

- A red light should light up on the Raspberry Pi, which indicates that the Raspberry Pi is connected to power. As it boots up, there should be raspberries that appear in the top left corner of the screen
- After a few seconds, the Raspberry Pi OS desktop will appear.

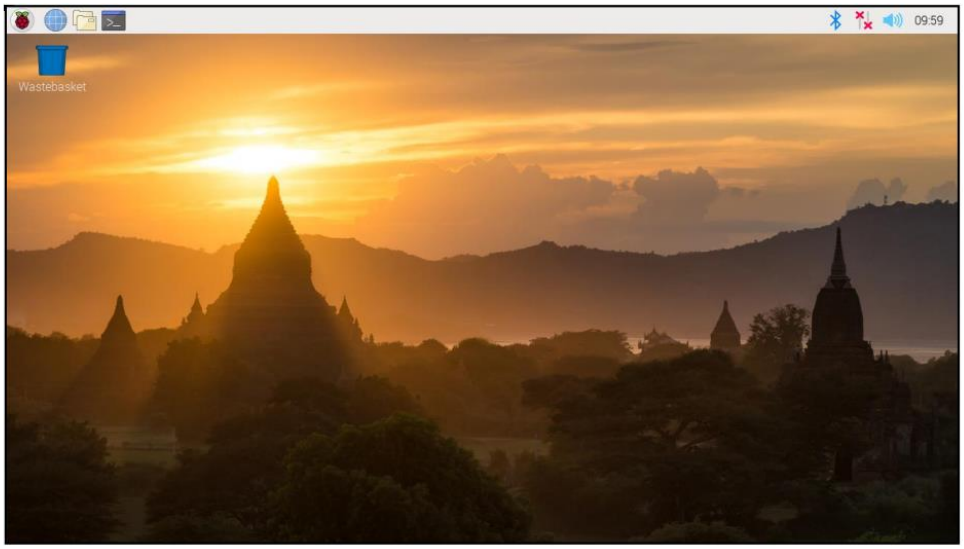
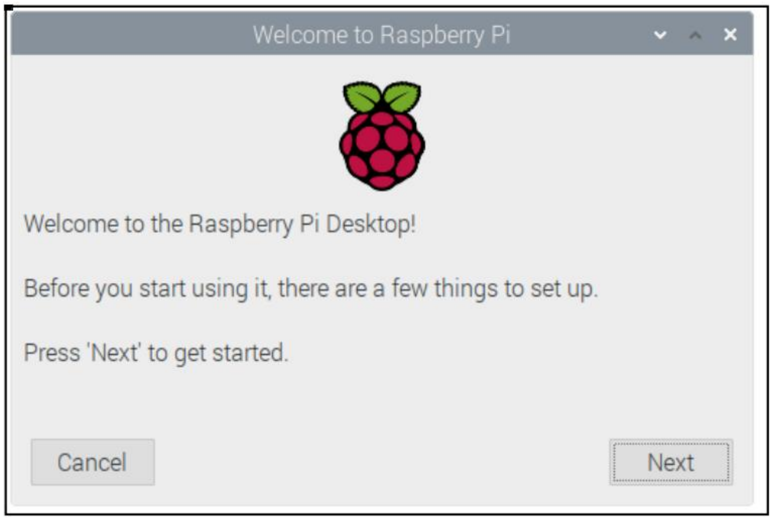


Figure 21. Raspberry Pi Buster Desktop

Finishing the Setup

- When the Raspberry Pi starts for the first time, the Welcome to Raspberry Pi application will appear to help through the initial setup.



Welcome to Raspberry Pi

Welcome to the Raspberry Pi Desktop!

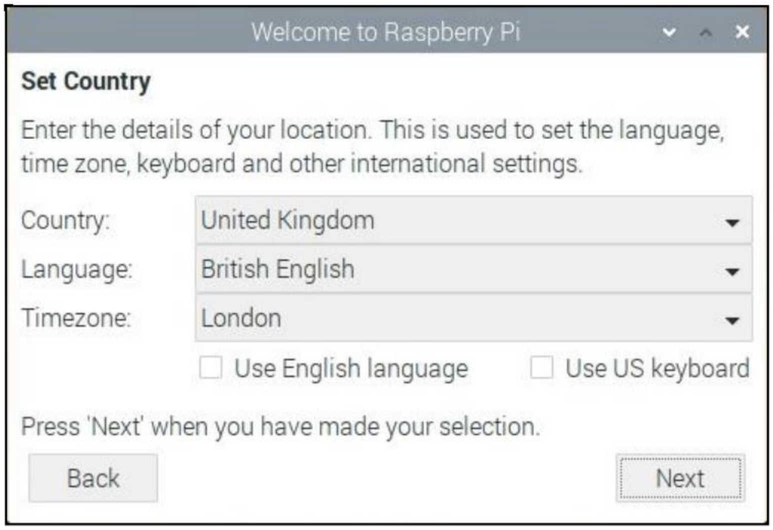
Before you start using it, there are a few things to set up.

Press 'Next' to get started.

Figure 22. Raspberry Pi 4 Startup Screen

- Click on “Next” to start the setup.

- Then, set the country, language, and time zone before clicking on “Next” again.



Welcome to Raspberry Pi

Set Country

Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.

Country: United Kingdom

Language: British English

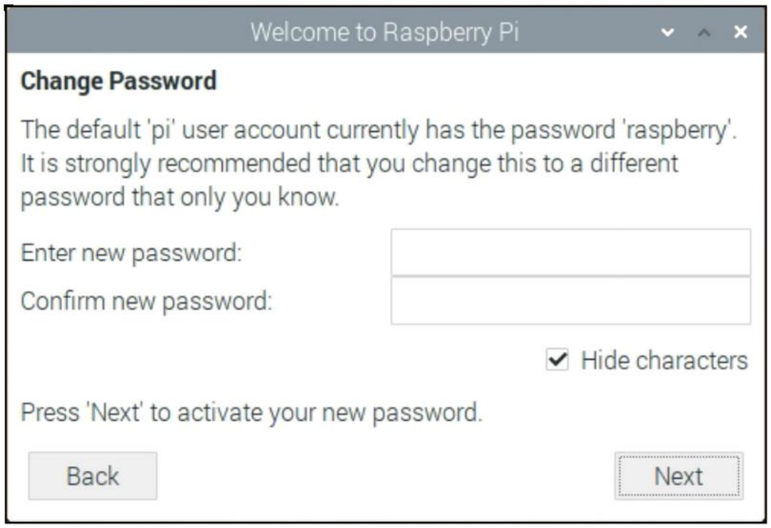
Timezone: London

□ Use English language □ Use US keyboard

Press 'Next' when you have made your selection.

Figure 23. Location Setup

- Enter a new password for the Raspberry Pi and click on “Next”.



Welcome to Raspberry Pi

Change Password

The default 'pi' user account currently has the password 'raspberry'. It is strongly recommended that you change this to a different password that only you know.

Enter new password:

Confirm new password:

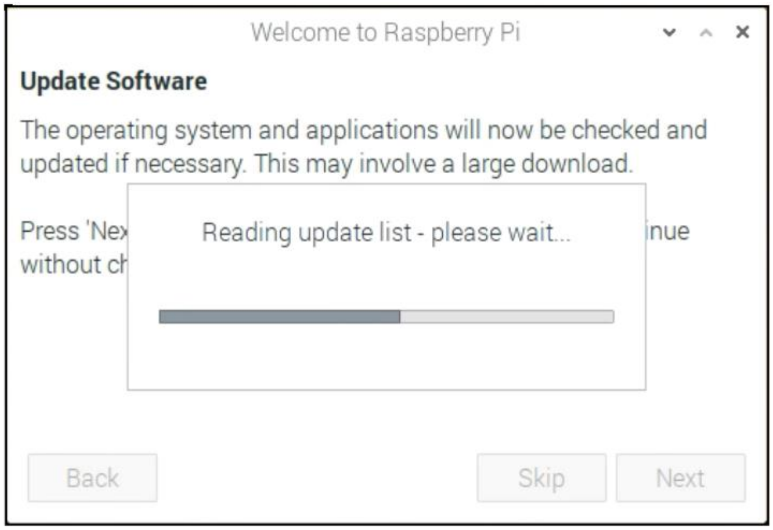
☑ Hide characters

Press 'Next' to activate your new password.

Figure 24. Password Setup

- Skip the Wi-Fi setup for the time being, as there is a special way to connect to the UVA Eduroam Wi-Fi network.

- Allow the wizard check for updates to the Raspberry Pi OS and install them (it could take a while).



Welcome to Raspberry Pi

Update Software

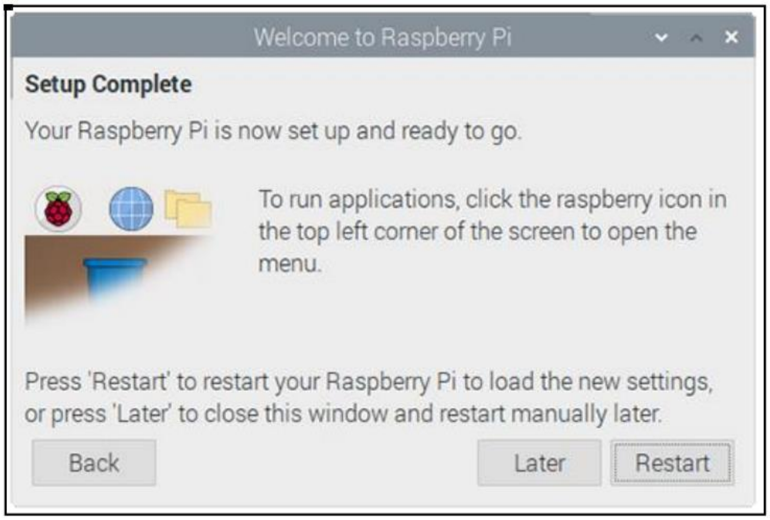
The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Nex inue without ch

Reading update list - please wait...

Figure 25. Checking for Updates Screen

- Click on "Restart" to finish the setup.
  - Note: The Pi will need to reboot if it needs to complete an update.



Welcome to Raspberry Pi

Setup Complete

Your Raspberry Pi is now set up and ready to go.

To run applications, click the raspberry icon in the top left corner of the screen to open the menu.

Press 'Restart' to restart your Raspberry Pi to load the new settings, or press 'Later' to close this window and restart manually later.

Figure 26. End of Setup Screen

Configuring the Raspberry Pi

- Most of the Raspberry Pi’s settings, such as the password, can be controlled through the “Raspberry Pi Configuration” application found in “Preferences” on the menu.
- For all the following screenshots, mimic your settings to match the screenshots (with the exception of the camera, as that needs to be adjusted)

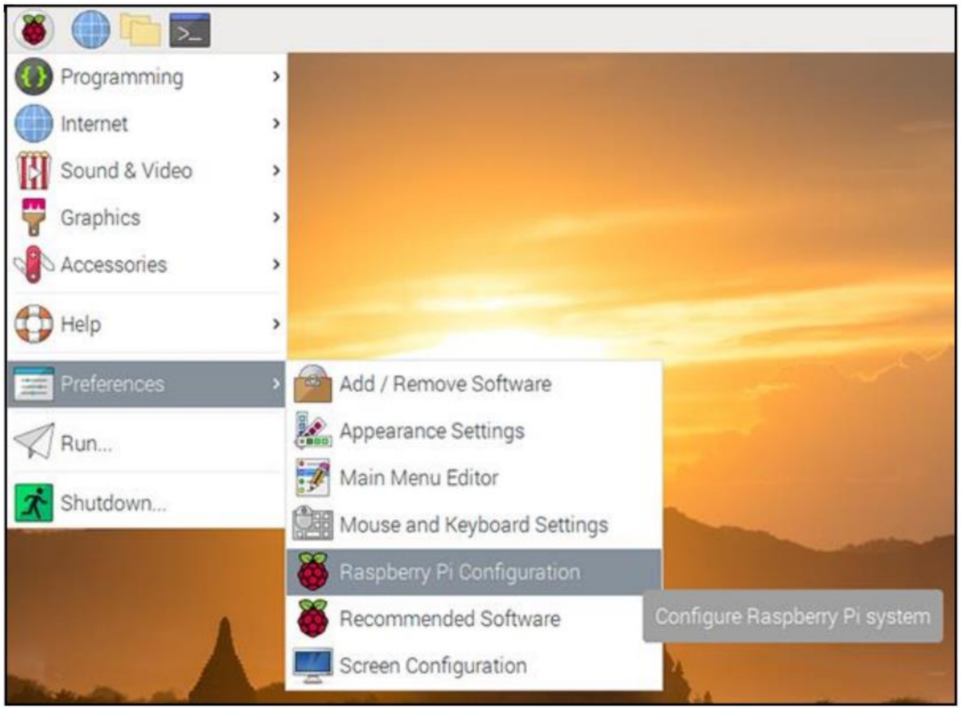
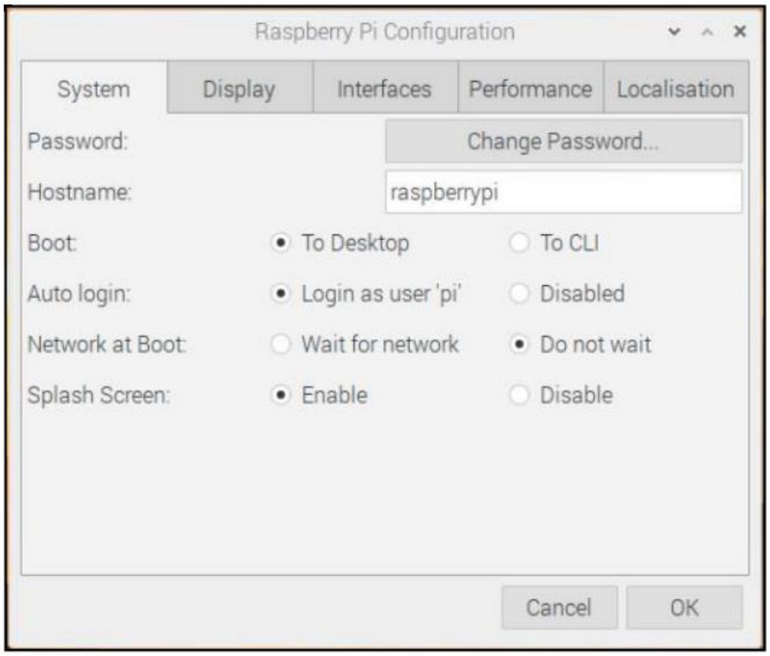


Figure 27. Accessing Raspberry Pi 4 Configuration Settings

- In the “System” tab, the basic system settings of the Raspberry Pi can be changed.



Password:

Hostname: raspberrypi

Boot: ● To Desktop ○ To CLI

Auto login: ● Login as user 'pi' ○ Disabled

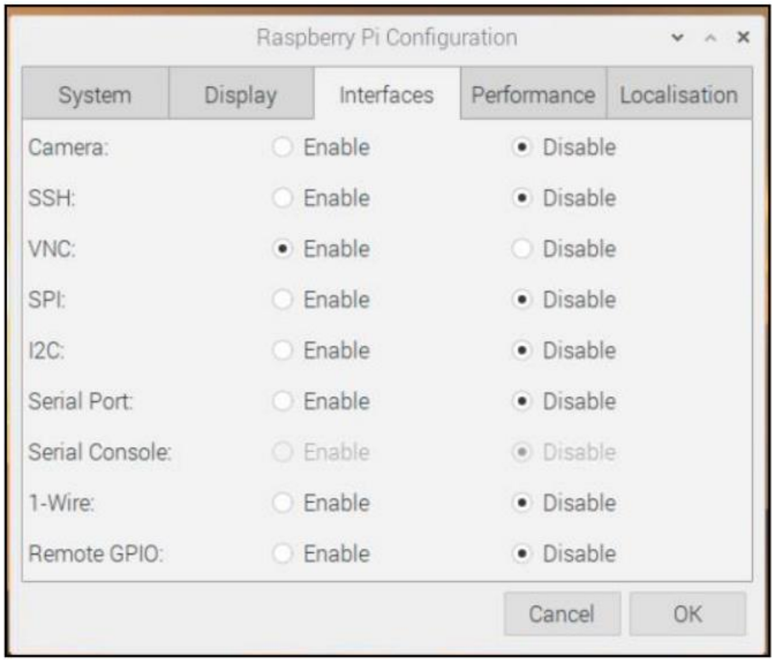
Network at Boot: ○ Wait for network ● Do not wait

Splash Screen: ● Enable ○ Disable

Figure 28. System Configuration

- Password: set the password of the pi user (it is a good idea to change the password from the factory default ‘raspberry’).
- Boot: select to show the Desktop or CLI (command-line interface) when the Raspberry Pi starts.
- Auto Login: enabling this option will make the Raspberry Pi automatically log in whenever it starts.
- Network at Boot: selecting this option will cause the Raspberry Pi to wait until a network connection is available before starting.
- Splash Screen: choose whether to show the splash (startup) screen when the Raspberry Pi boots.

- In the “Interfaces” tab, devices and components can be linked to the Raspberry Pi using a lot of different types of connections. The Interfaces tab is where these different connections can be turned ON or OFF so that the Raspberry Pi recognizes that it has something linked to it via a particular type of connection.

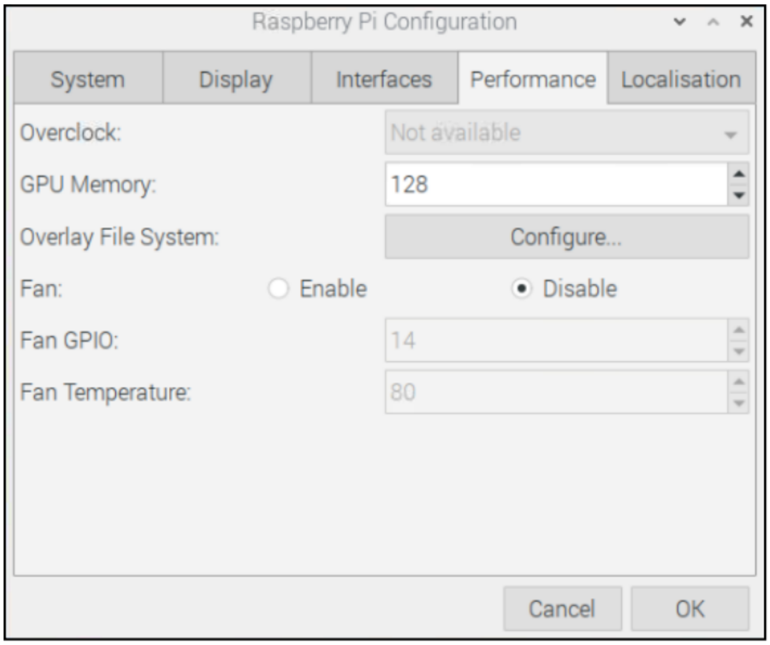


Camera:	○ Enable	● Disable
SSH:	○ Enable	● Disable
VNC:	● Enable	○ Disable
SPI:	○ Enable	● Disable
I2C:	○ Enable	● Disable
Serial Port:	○ Enable	● Disable
Serial Console:	○ Enable	● Disable
1-Wire:	○ Enable	● Disable
Remote GPIO:	○ Enable	● Disable

Figure 29. Interfaces Configuration

- Camera: enable the Raspberry Pi Camera Module
  - Enable this function: occasionally a warning comes up that prevents the camera from working correctly if this is not enabled
- SSH: allow remote access to the Raspberry Pi from another computer using SSH
- VNC: allow remote access to the Raspberry Pi Desktop from another computer using VNC
- SPI: enable the SPI GPIO pins
- I2C: enable the I2C GPIO pins
- Serial: enable the Serial (Rx, Tx) GPIO pins
- 1-Wire: enable the 1-Wire GPIO pin
- Remote GPIO: allow access to the Raspberry Pi’s GPIO pins from another computer

- In the “Performance” tab, the performance settings of the Raspberry Pi can be changed if necessary for a particular project. Changing the settings, however, can result in the Pi performing erratically or not working.



Overclock: Not available

GPU Memory: 128

Overlay File System: Configure...

Fan: ○ Enable ● Disable

Fan GPIO: 14

Fan Temperature: 80

Figure 30. Performance Configuration

- ○ Overclock: change the CPU speed and voltage to increase performance GPU
- ○ Memory: change the allocation of memory given to the GPU

- In the “Localization” tab, the Raspberry Pi’s settings can be changed to be specific to a country or location

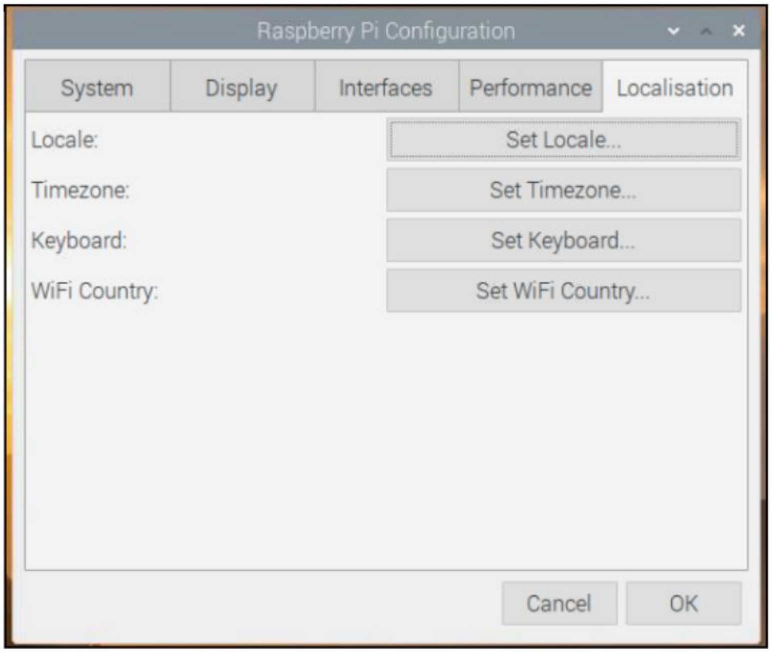
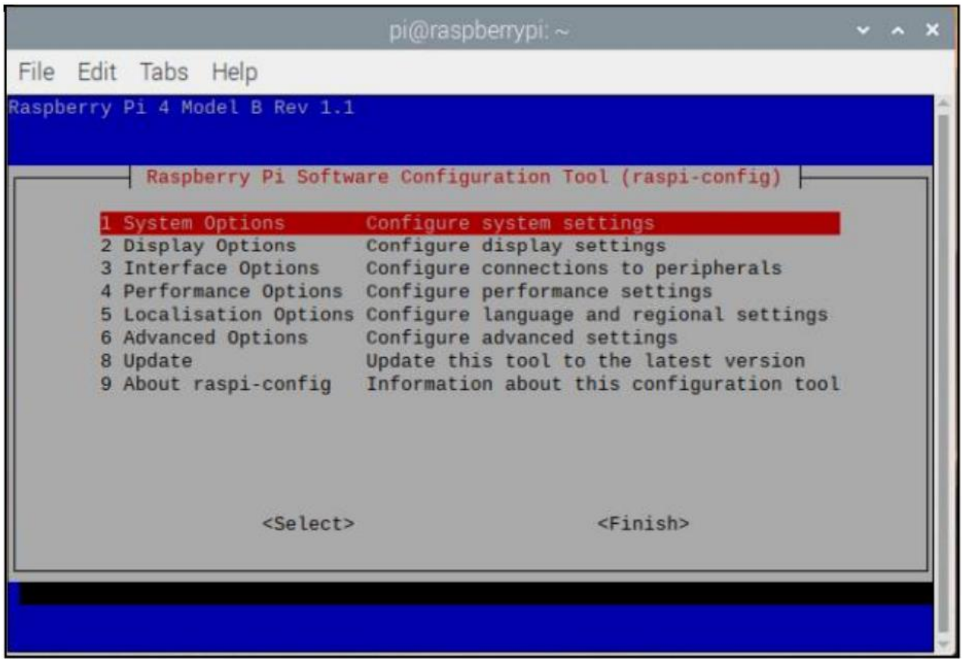


Figure 31. Localization Configuration

  - Locale: set the language, country, and character set used by our Raspberry Pi
  - Time zone: set the time zone
  - Keyboard: changes the keyboard layout Wi-Fi
  - Country: set the Wi-Fi country code

Alternative Configuration

- Open a terminal and type in “sudo raspi-config”. After running the command, a blue screen with options inside a grey box should appear.

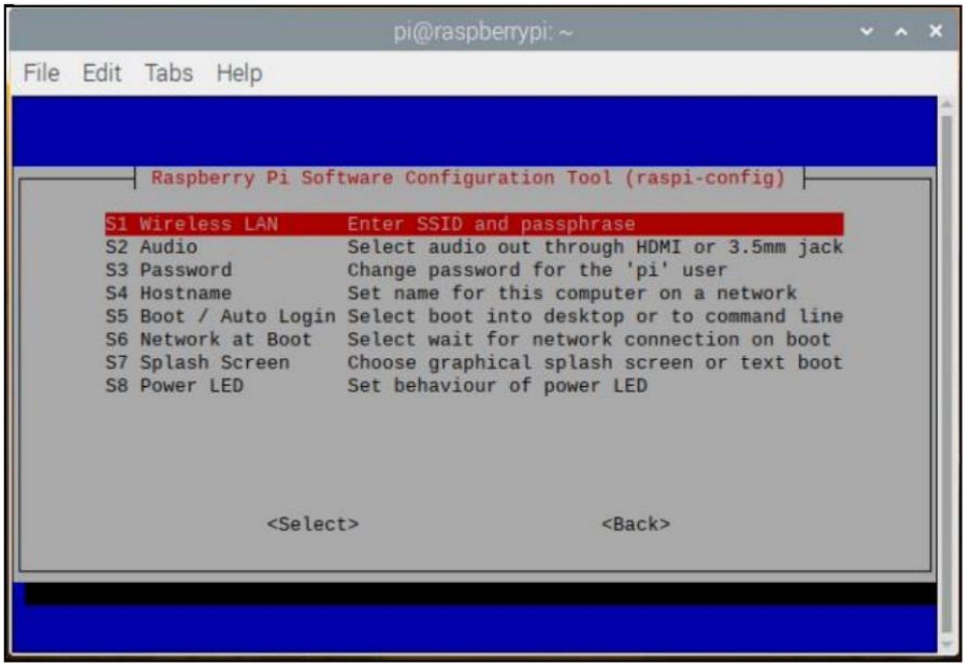


```
Raspberry Pi 4 Model B Rev 1.1

┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

        1 System Options       Configure system settings
        2 Display Options      Configure display settings
        3 Interface Options    Configure connections to peripherals
        4 Performance Options  Configure performance settings
        5 Localisation Options Configure language and regional settings
        6 Advanced Options     Configure advanced settings
        8 Update               Update this tool to the latest version
        9 About raspi-config   Information about this configuration tool



                  <Select>                     <Finish>
```

Figure 32. raspi-config menu

- Moving around the menu: Use the up and down arrow keys to move the highlighted selection between the options available. Pressing the tab key will jump out of the Options menu and take you to the buttons. Pressing left will take you back to the options.
- What raspi-config does: raspi-config aims to provide the functionality to make the most common configuration changes. This may result in automated edits to /boot/config.txt and various standard Linux configuration files. Some options require a reboot to take effect. If we changed any of those, raspi-config will ask if we wish to reboot now when we select the button.

- System Options: The system options submenu allows configuration changes to be made during various parts of the boot, login, and networking process, along with some other system-level changes.

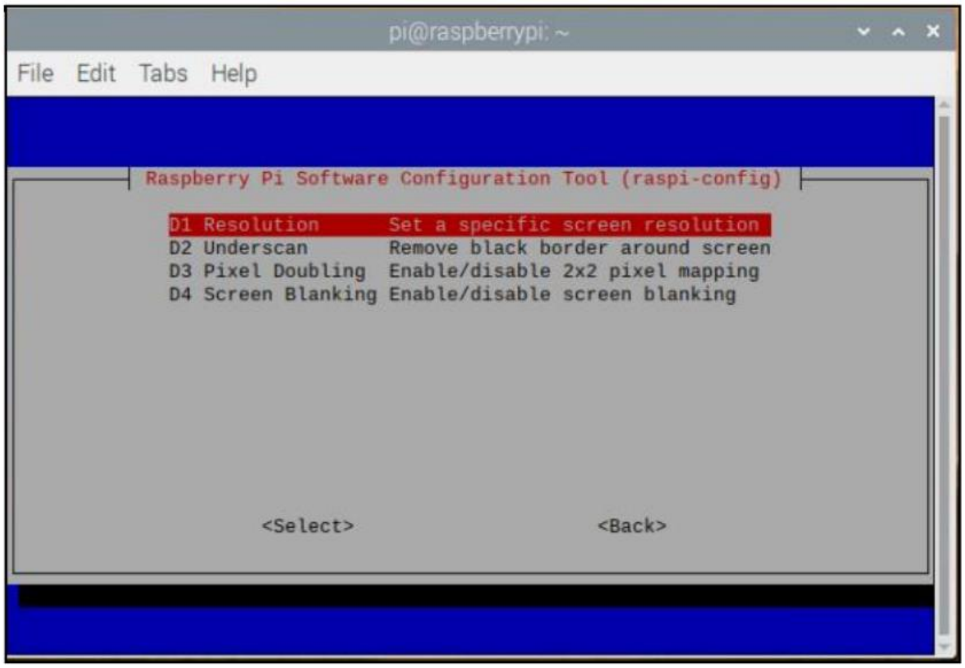


```
Raspberry Pi Software Configuration Tool (raspi-config)

S1 Wireless LAN        Enter SSID and passphrase
S2 Audio               Select audio out through HDMI or 3.5mm jack
S3 Password            Change password for the 'pi' user
S4 Hostname            Set name for this computer on a network
S5 Boot / Auto Login   Select boot into desktop or to command line
S6 Network at Boot     Select wait for network connection on boot
S7 Splash Screen       Choose graphical splash screen or text boot
S8 Power LED           Set behaviour of power LED

           <Select>                        <Back>
```

Figure 33. raspi-config System Options

- Wireless LAN: Allows setting of the wireless LAN SSID and passphrase.
- Audio: Specify the audio output destination.
- Password: The default user on Raspberry Pi OS is pi with the password raspberry. It can be changed here.
- Hostname: Set the visible name for this Pi on a network.
- Boot / Autologin: From this submenu, select whether to boot to console or desktop and whether there needs to be a log in or not. If automatic login is selected, the pi user will be logged in.
- Network at Boot: Use this option to wait for a network connection before letting boot proceed.
- Splash Screen: Enable or disable the splash screen displayed at boot time.
- Power LED: If the model of Pi permits it, the behavior of the power LED can be changed using this option.

- Display Options

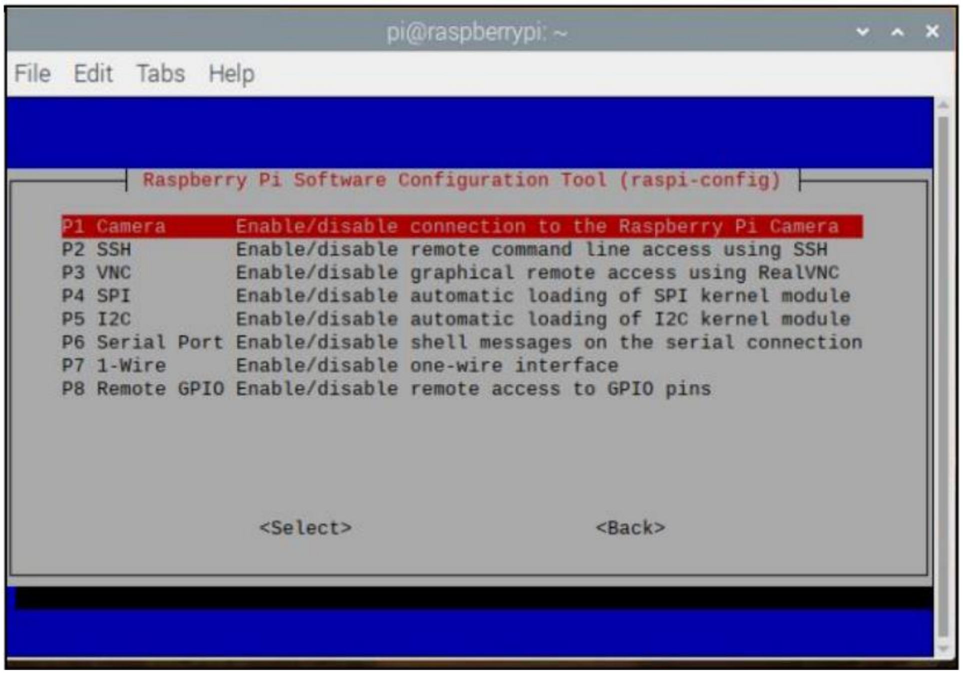


```
                Raspberry Pi Software Configuration Tool (raspi-config)

                 D1 Resolution       Set a specific screen resolution
                 D2 Underscan        Remove black border around screen
                 D3 Pixel Doubling   Enable/disable 2x2 pixel mapping
                 D4 Screen Blanking  Enable/disable screen blanking

                        <Select>                     <Back>
```

Figure 34. raspi-config Display Options

- Resolution: Define the default HDMI/DVI video resolution to use when the system boots without a monitor being connected. This can influence RealVNC if the VNC option is enabled.
- Underscan: Old TV sets had a significant variation in the size of the picture they produced; some had cabinets that overlapped the screen. TV pictures were therefore given a black border so that none of the pictures was lost; this is called overscan. Modern TVs and monitors do not need the border, and the signal does not allow for it. If the initial text shown on the screen disappears off the edge, we need to enable overscan to bring the border back. Any changes will take effect after a reboot. We can have greater control over the settings by editing config.txt. On some displays, particularly monitors, disabling overscan will make the picture fill the whole screen and correct the resolution. For other displays, it may be necessary to leave overscan enabled and adjust its values.
- Pixel Doubling: Enable/disable 2x2 pixel mapping.
- Screen Blanking: Enable or disable screen blanking.

- Interfacing Options: In this submenu, there are the following options to enable/disable: Camera, SSH, VNC, SPI, I2C, Serial, 1-wire, and Remote GPIO.

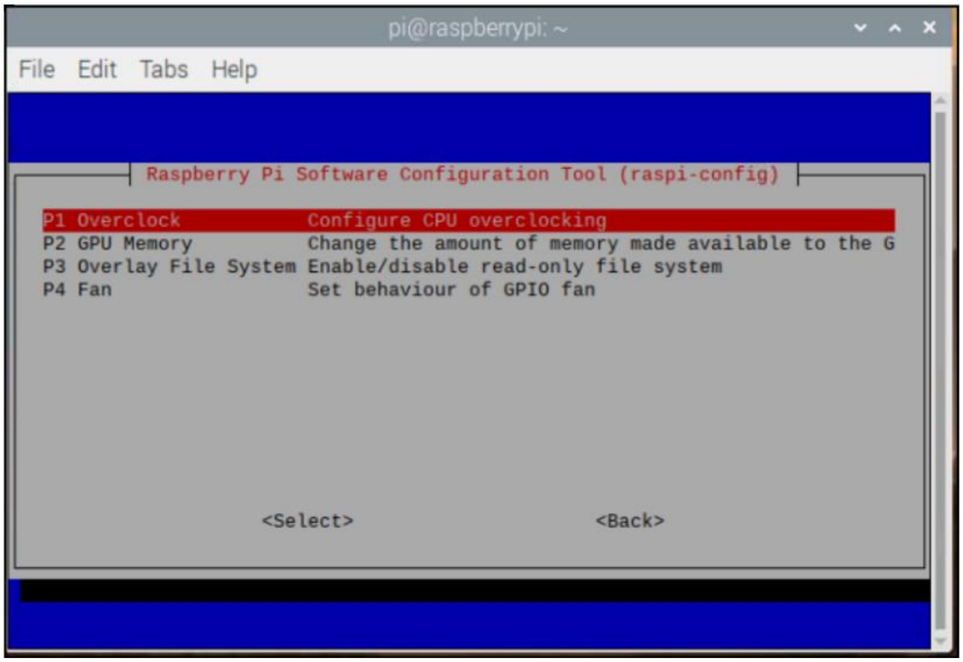


```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/disable connection to the Raspberry Pi Camera
P2 SSH         Enable/disable remote command line access using SSH
P3 VNC         Enable/disable graphical remote access using RealVNC
P4 SPI         Enable/disable automatic loading of SPI kernel module
P5 I2C         Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell messages on the serial connection
P7 1-Wire      Enable/disable one-wire interface
P8 Remote GPIO Enable/disable remote access to GPIO pins


                    <Select>                    <Back>
```

Figure 35. raspi-config Interfacing Options

- Camera: Enable/disable the CSI camera interface.
- SSH: Enable/disable remote command-line access to your Pi using SSH. SSH allows to remotely access the command line of the Raspberry Pi from another computer. SSH is disabled by default.
- VNC: Enable/disable the RealVNC virtual network computing server.
- SPI: Enable/disable SPI interfaces and automatic loading of the SPI kernel module, needed for products such as PiFace.
- I2C: Enable/disable I2C interfaces and automatic loading of the I2C kernel module.
- Serial: Enable/disable shell and kernel messages on the serial connection.
- 1-wire: Enable/disable the Dallas 1-wire interface. This is usually used for DS18B20 temperature sensors.
- Remote GPIO: Enable or disable remote access to the GPIO pins.

- Performance Options

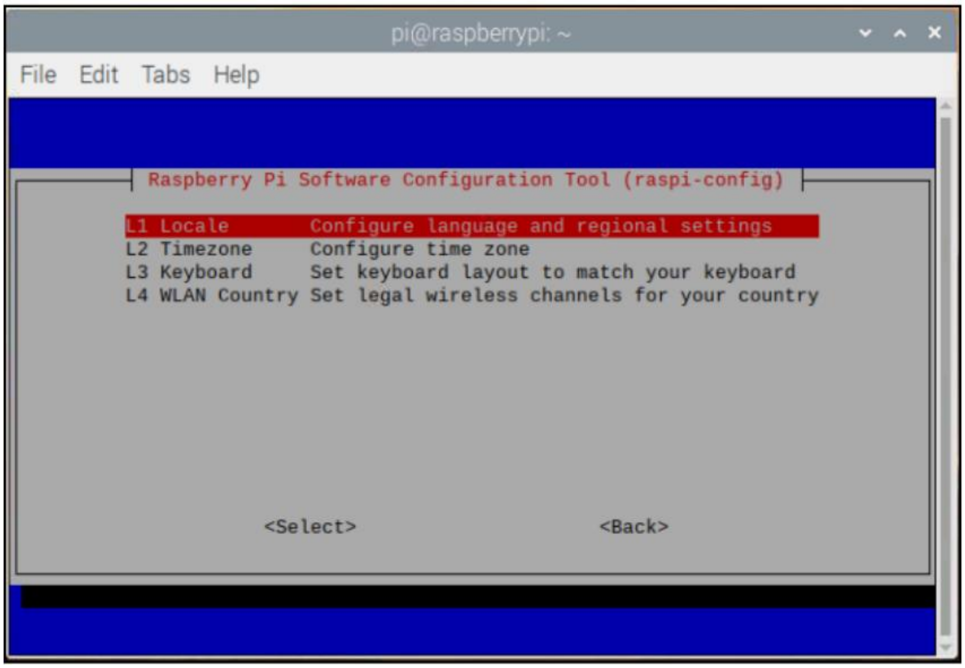


```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Overclock           Configure CPU overclocking
P2 GPU Memory          Change the amount of memory made available to the G
P3 Overlay File System Enable/disable read-only file system
P4 Fan                 Set behaviour of GPIO fan




             <Select>                    <Back>
```

Figure 36. raspi-config Performance Options

- Overclock On some models, it is possible to overclock the Raspberry Pi's CPU using this tool. The overclocking we can achieve will vary; overclocking too high may result in instability. Selecting this option shows the following warning: 29 Be aware that overclocking may reduce the lifetime of your Raspberry Pi. If overclocking at a certain level causes system instability, try a more modest overclock. Hold down the Shift key during boot to temporarily disable overclocking.
- GPU Memory: Change the amount of memory made available to the GPU.
- Overlay File System: Enable or disable a read-only filesystem.
- Fan: Set the behavior of a GPIO connected fan.

- Localization Options: The localization submenu allows users to choose between keyboard layout, time zone, locale, and wireless LAN country code.

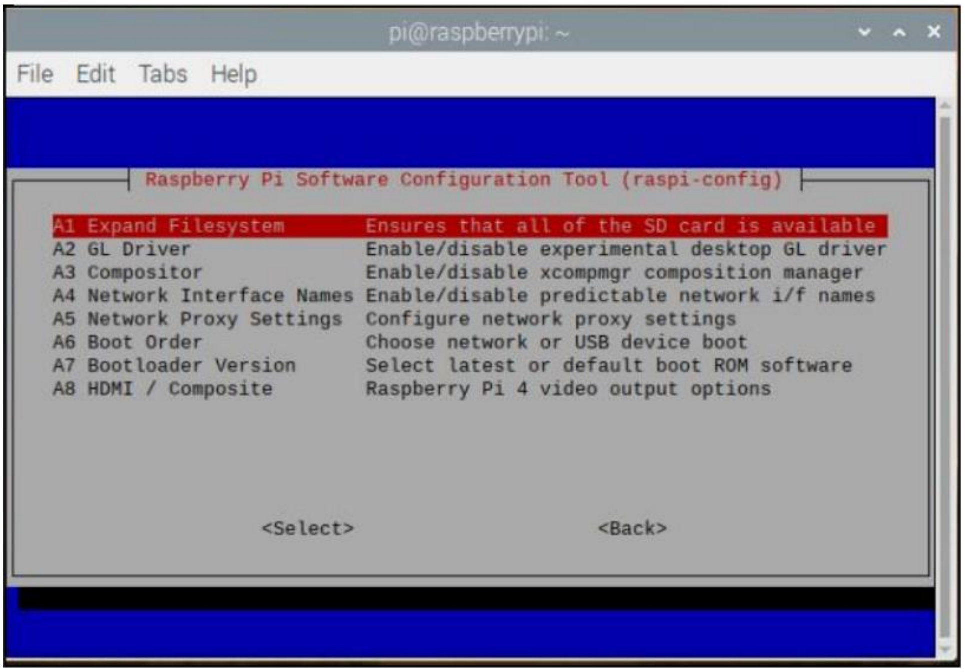


```
Raspberry Pi Software Configuration Tool (raspi-config)

L1 Locale        Configure language and regional settings
L2 Timezone      Configure time zone
L3 Keyboard      Set keyboard layout to match your keyboard
L4 WLAN Country  Set legal wireless channels for your country

          <Select>                    <Back>
```

Figure 37. raspi-config Localization Options

- ○ Locale: Select a locale, for example en_GB.UTF-8 UTF-8.
- ○ Time Zone: Select your local time zone, starting with the region, e.g., USA, then selecting a city, e.g., Detroit. Type a letter to skip down the list to that point in the alphabet.
- ○ Keyboard: This option opens another menu that allows you to select your keyboard layout. It will take a long time to display while it reads all the keyboard types. Changes usually take effect immediately but may require a reboot.
- ○ WLAN Country: This option sets the country code for your wireless network.

- Advanced Options



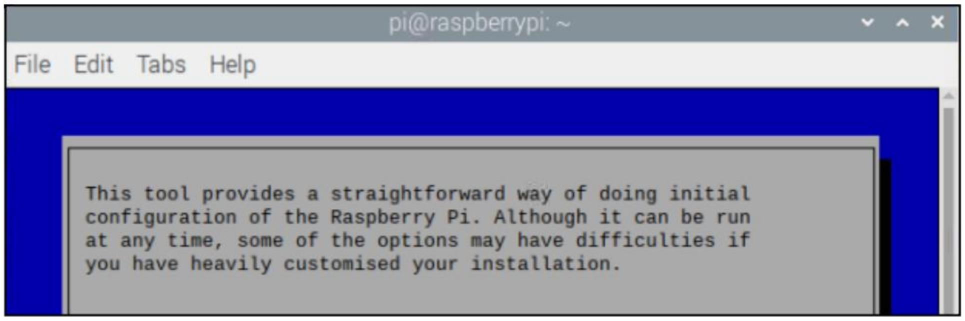
```
Raspberry Pi Software Configuration Tool (raspi-config)

A1 Expand Filesystem          Ensures that all of the SD card is available
A2 GL Driver                  Enable/disable experimental desktop GL driver
A3 Compositor                 Enable/disable xcompmgr composition manager
A4 Network Interface Names    Enable/disable predictable network i/f names
A5 Network Proxy Settings     Configure network proxy settings
A6 Boot Order                 Choose network or USB device boot
A7 Bootloader Version         Select latest or default boot ROM software
A8 HDMI / Composite           Raspberry Pi 4 video output options
```

Figure 38. raspi-config Advanced Options

- Expand Filesystem: Consider the Raspberry Pi OS was installed using NOOBS, then the filesystem will have been expanded automatically. There may be a rare occasion where this is not the case, e.g., if a smaller SD card is copied onto a larger one. In this case, this option should be used to expand the installation to fill the whole SD card, giving more space to use for files. The Raspberry Pi will need to be rebooted to make this available. Note that there is no confirmation: selecting the option begins the partition expansion immediately.
- GL Driver: Enable/disable the experimental GL desktop graphics drivers.
- GL (Full KMS): Enable/disable the experimental OpenGL Full KMS (kernel mode setting) desktop graphics driver.
- GL (Fake KMS): Enable/disable the experimental OpenGL Fake KMS desktop graphics driver.
- Legacy: Enable/disable the original legacy non-GL VideoCore desktop graphics driver.
- Compositor: Enable/Display the xcompmgr composition manager.
- Network Interface Names: Enable or disable predictable network interface names.
- Network Proxy Settings: Configure the network's proxy settings.
- Boot Order: On the Raspberry Pi4, it can be specified whether to boot from USB or network if the SD card is not inserted.

- Bootloader Version: On the Raspberry Pi4, the system can be told to use the very latest boot ROM software, or default to the factory default if the latest version causes problems.

- Update: Update this tool to the latest version.
- About raspi-config: Selecting this option shows the following text:



```
This tool provides a straightforward way of doing initial
configuration of the Raspberry Pi. Although it can be run
at any time, some of the options may have difficulties if
you have heavily customised your installation.
```

Figure 39. raspi-config description

Connecting the Raspberry Pi 4 to UVA’s Eduroam Wi-Fi

Follow the instructions at the following link to connect the Raspberry Pi 4 to UVA’s Eduroam Wi-Fi: https://scholarslab.lib.virginia.edu/blog/raspberry-pi-uva-eduroam/

Note: Be sure to follow all the steps, however, wherever the instructions use “Ammon Edwin Shepherd 21.p12”, be sure to replace that with the name of your certificate. Additionally, in the wpa_supplicant.conf file, be sure to use your UVA email and the password you set to make sure the connection can be successfully completed.

GPIO

A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board. A 40-pin GPIO header is found on all current Raspberry Pi boards (unpopulated on Pi Zero and Pi Zero W). Before the Pi 1 Model B+ (2014), boards comprised a shorter 26-pin header.

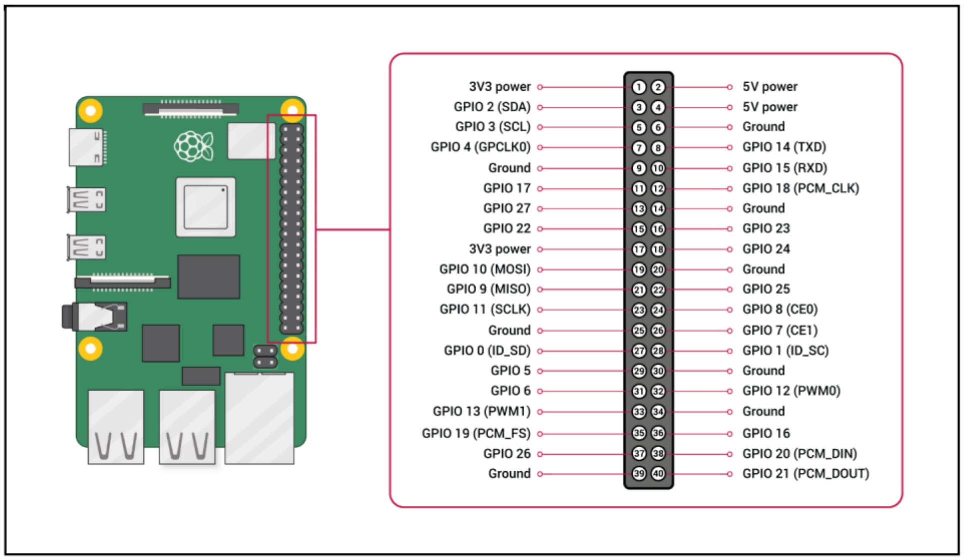


Figure 40. GPIO Pinout

Any of the GPIO pins can be designated (in software) as an input or output pin and used for a wide range of purposes.

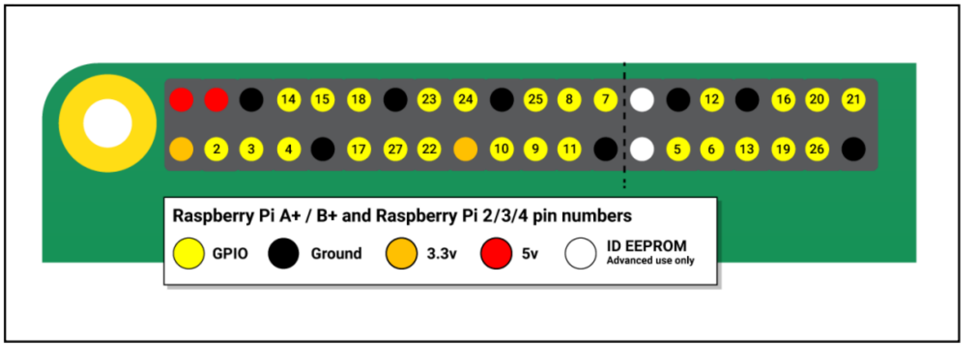


Figure 41. Color-Coded GPIO Pinout

Note: the numbering of the GPIO pins is not in numerical order; GPIO pins 0 and 1 are present on the board (physical pins 27 and 28) but are reserved for advanced use (see below).

Types of Pins

- Voltages:
  - Two 5V pins and two 3V3 pins are present on the board, as well as several ground pins (0V), which are unconfigurable. The remaining pins are all general-purpose 3V3 pins, meaning outputs are set to 3V3 and inputs are 3V3-tolerant.
- Outputs:
  - A GPIO pin designated as an output pin can be set to high (3V3) or low (0V).
- Inputs:
  - A GPIO pin designated as an input pin can be read as high (3V3) or low (0V). This is made easier with the use of internal pull-up or pull-down resistors. Pins GPIO2 and GPIO3 have fixed pull-up resistors, but for other pins, this can be configured in software.
- PWM (pulse-width modulation):
  - Software PWM available on all pins Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19
- SPI:
  - SPI0: MOSI (GPIO10), MISO (GPIO9), SCLK (GPIO11), CE0 (GPIO8), CE1 (GPIO7)
  - SPI1: MOSI (GPIO20), MISO (GPIO19), SCLK (GPIO21), CE0 (GPIO18), CE1 (GPIO17), CE2 (GPIO16)
- I2C:
  - Data: (GPIO2), Clock (GPIO3)
  - EEPROM Data: (GPIO0), EEPROM Clock (GPIO1)
- Serial:
  - TX (GPIO14); RX (GPIO15)

Connecting the Thermal Camera to the Raspberry Pi 4

- Connect the FLIR Lepton 2.5 with the FLIR Lepton® Camera Breakout Board v2.0 as shown in the figure below.

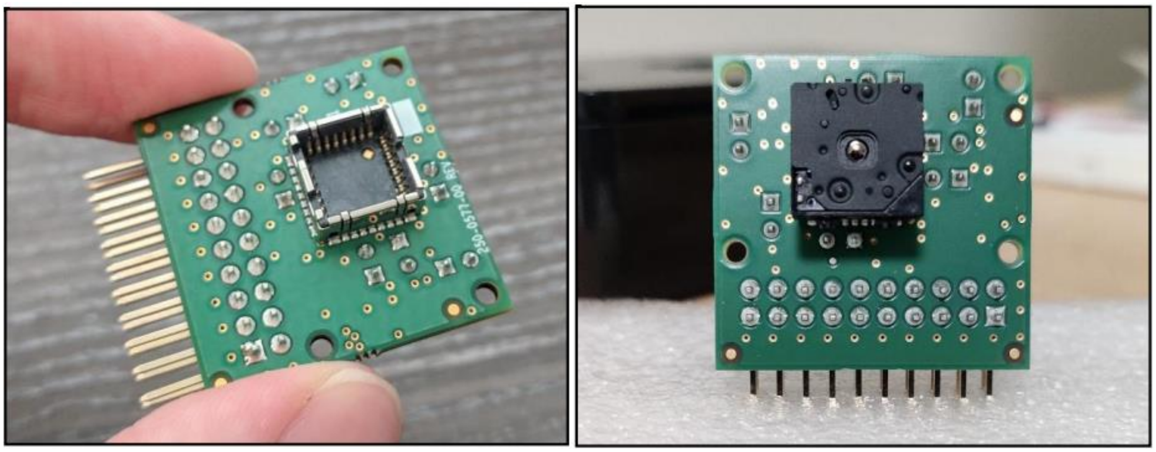
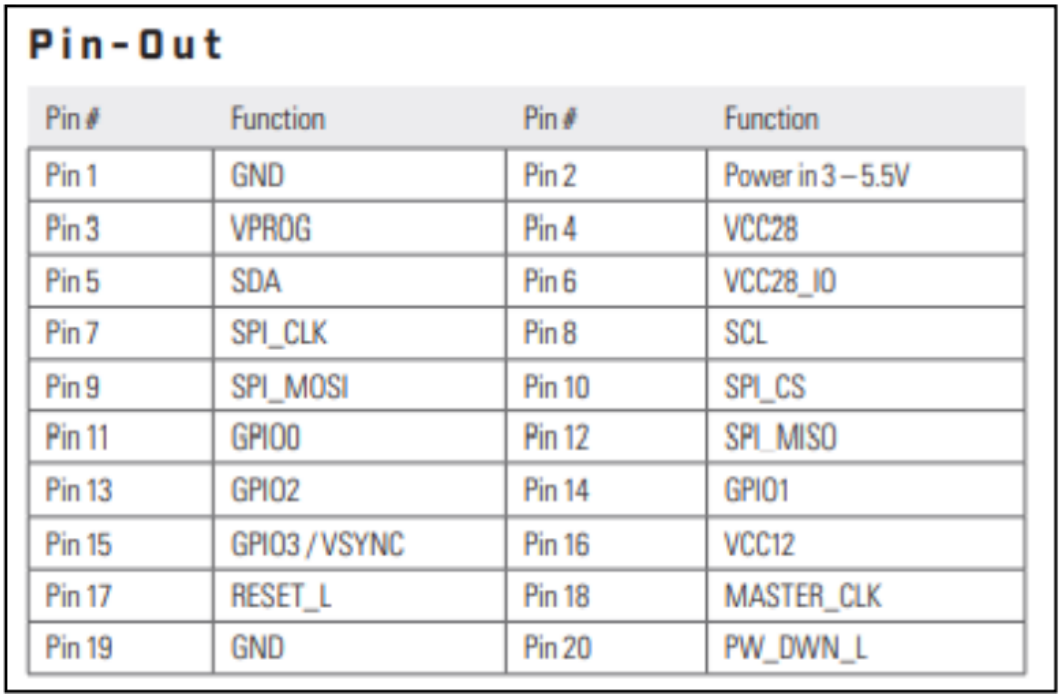


Figure 42. How to Connect the FLIR Lepton 2.5 to the Breakout Board

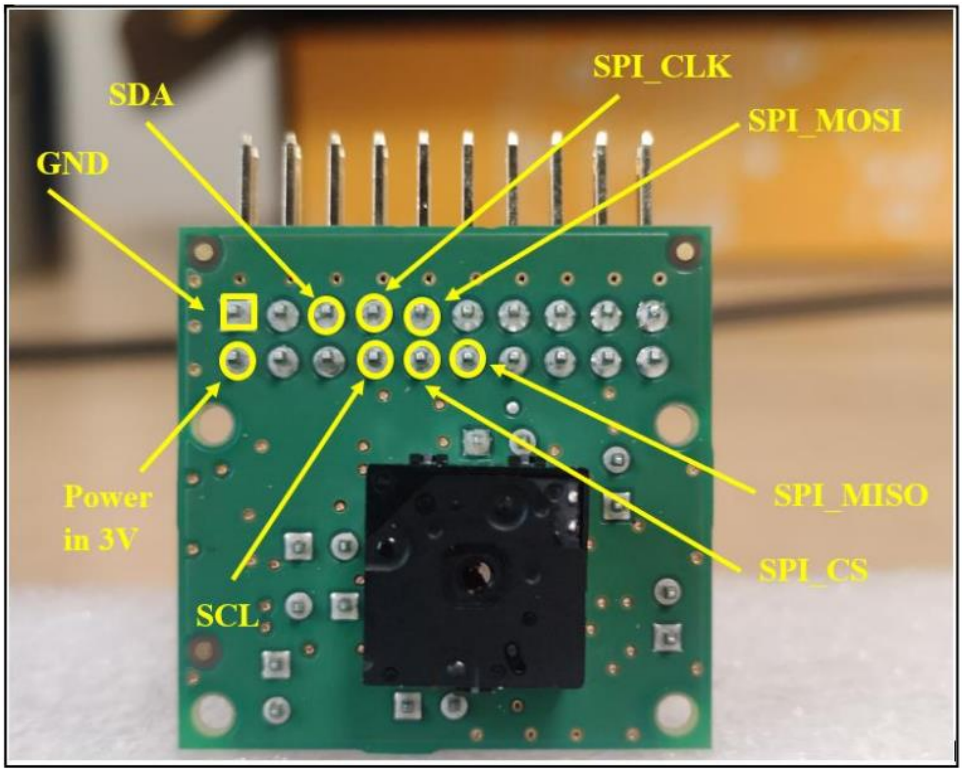
- Connect the Camera Breakout Board with the Raspberry Pi through the GPIO pins.



Pin-Out

Pin #	Function	Pin #	Function
Pin 1	GND	Pin 2	Power in 3 – 5.5V
Pin 3	VPROG	Pin 4	VCC28
Pin 5	SDA	Pin 6	VCC28_IO
Pin 7	SPI_CLK	Pin 8	SCL
Pin 9	SPI_MOSI	Pin 10	SPI_CS
Pin 11	GPIO0	Pin 12	SPI_MISO
Pin 13	GPIO2	Pin 14	GPIO1
Pin 15	GPIO3 / VSYNC	Pin 16	VCC12
Pin 17	RESET_L	Pin 18	MASTER_CLK
Pin 19	GND	Pin 20	PW_DWN_L

Camera Breakout Board Pin Number	Camera Breakout Board Pin Function	GPIO Pin
Pin 1	GND	Ground
Pin 2	Power in 3 – 5.5V	3V3 Power
Pin 5	SDA	GPIO 2
Pin 7	SPI_CLK	GPIO 11
Pin 8	SCL	GPIO 3
Pin 9	SPI_MOSI	GPIO 10
Pin 10	SPI_CS	GPIO 8 (CE0)
Pin 12	SPI_MISO	GPIO 9



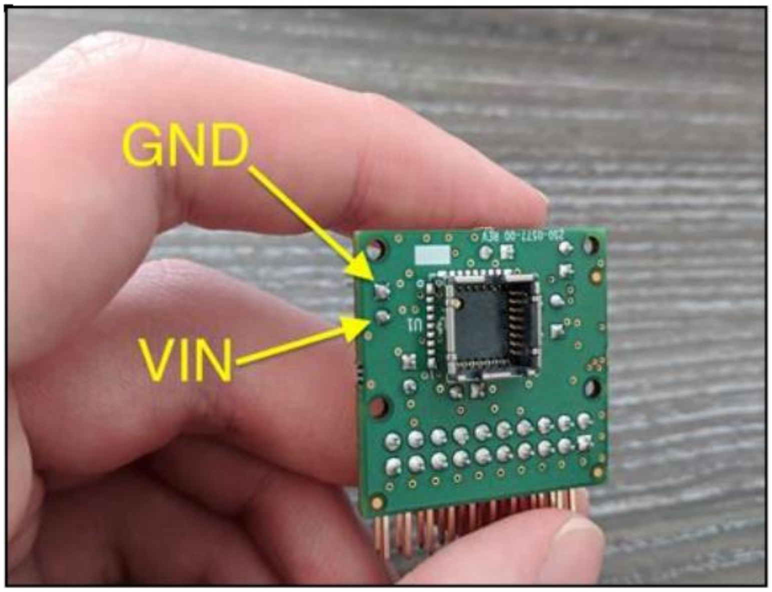


Figure 43. Details on How to Connect the Breakout Board to the Raspberry Pi 4

Here are my recommended connections:

- Connect “GND” on the Breakout Board to pin 6 on the Pi
- Connect “Power in 3V” on the Breakout Board to pin 1 on the Pi
- Connect “SDA” on the Breakout Board to pin 3 on the Pi
- Connect “SPI_CLK” on the Breakout Board to pin 23 on the Pi
- Connect “SCL” on the Breakout Board to pin 5 on the Pi
- Connect “SPI_MOSI” on the Breakout Board to pin 19 on the Pi
- Connect “SPI_CS” on the Breakout Board to pin 24 on the Pi
- Connect “SPI_MISO” on the Breakout Board to pin 21 on the Pi

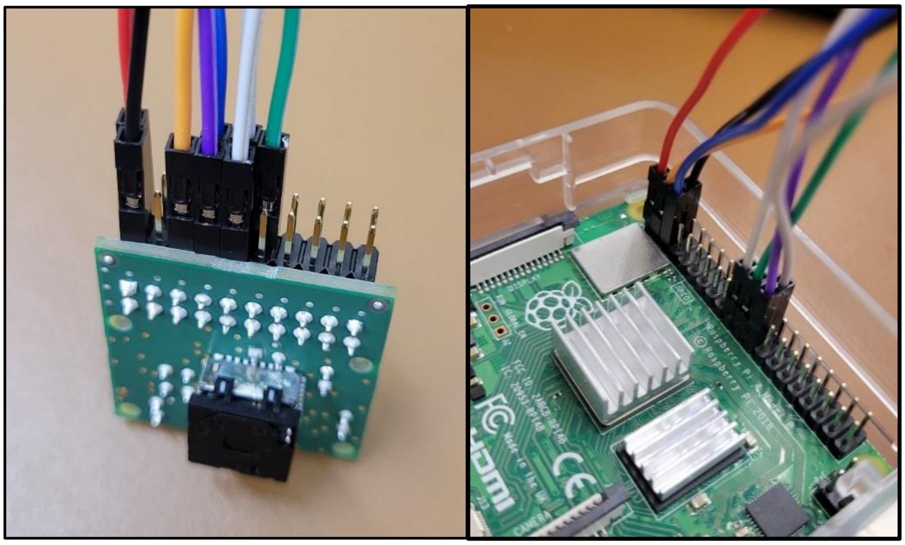
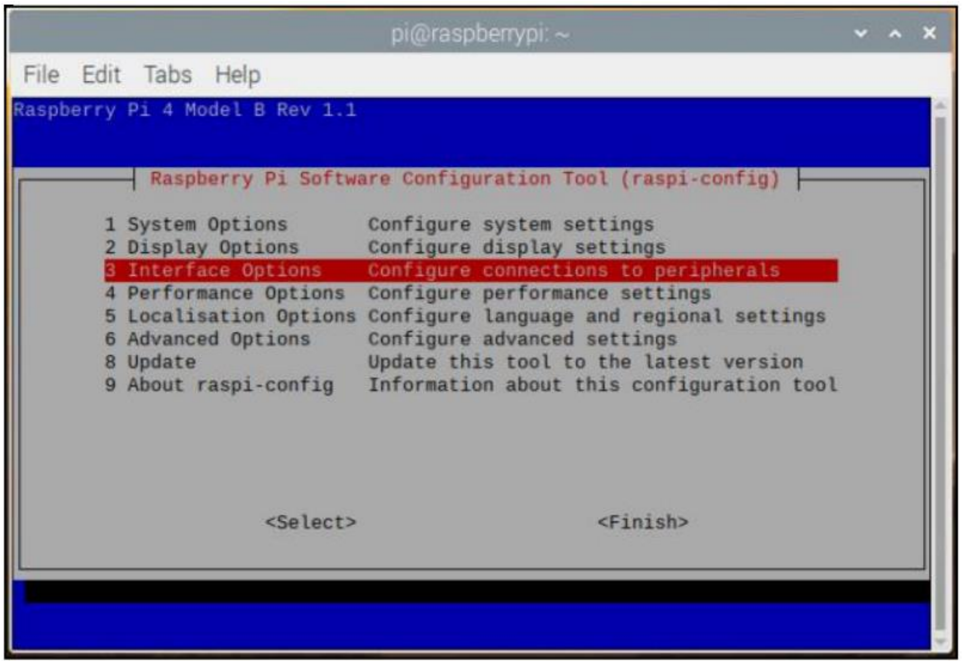


Figure 44. Final Connections Between Raspberry Pi 4 and Breakout Board

Software

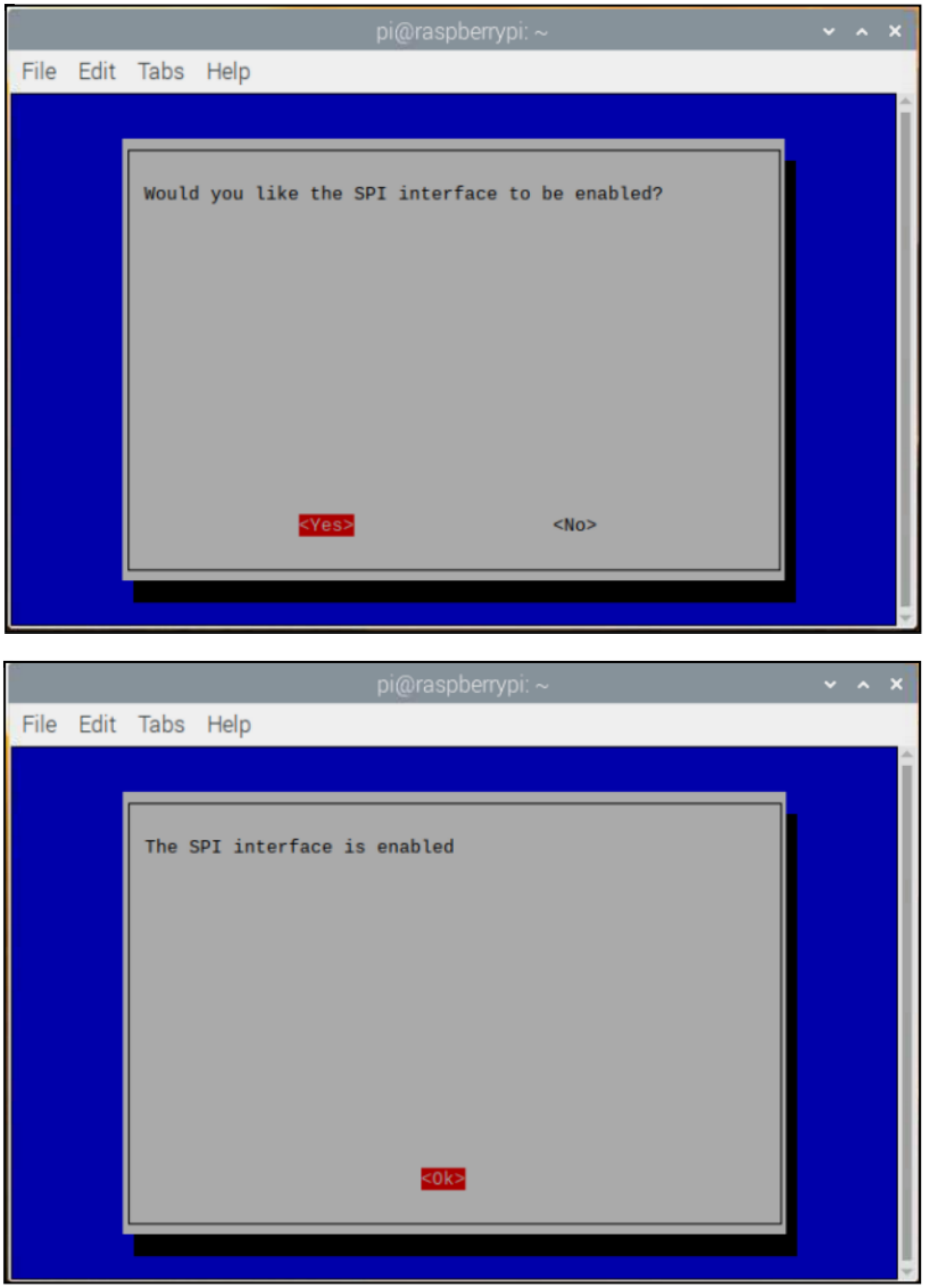
As mentioned earlier, the Raspbian OS must be installed on the Raspberry Pi. After booting it up, open the terminal. The first task is enabling the Pi's SPI and I2C interfaces. To do this, run the command "sudo raspi-config" in the terminal.



```
Raspberry Pi 4 Model B Rev 1.1

          Raspberry Pi Software Configuration Tool (raspi-config)

        1 System Options       Configure system settings
        2 Display Options      Configure display settings
        3 Interface Options    Configure connections to peripherals
        4 Performance Options  Configure performance settings
        5 Localisation Options Configure language and regional settings
        6 Advanced Options     Configure advanced settings
        8 Update               Update this tool to the latest version
        9 About raspi-config   Information about this configuration tool

                     <Select>                     <Finish>
```

Figure 45. raspi-config menu

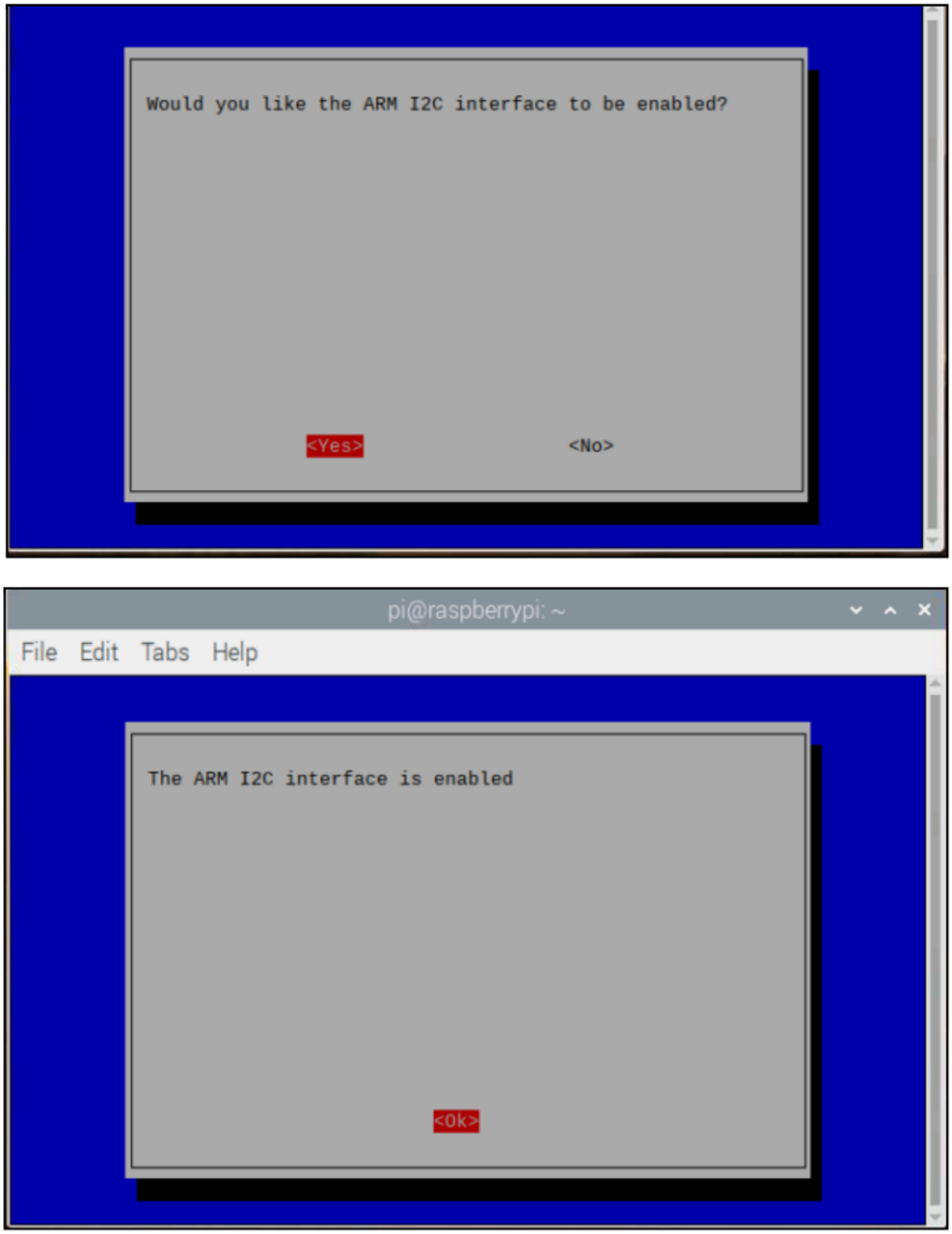
Select “Interface Options”, then click “SPI”. Enable the SPI interface. Follow the same process for the I2C. Once finished, the system will ask to reboot. Reboot the system so that the changes will be applied.



```
Would you like the SPI interface to be enabled?
```

```
The SPI interface is enabled
```

Figure 46. Enabling the SPI Interface

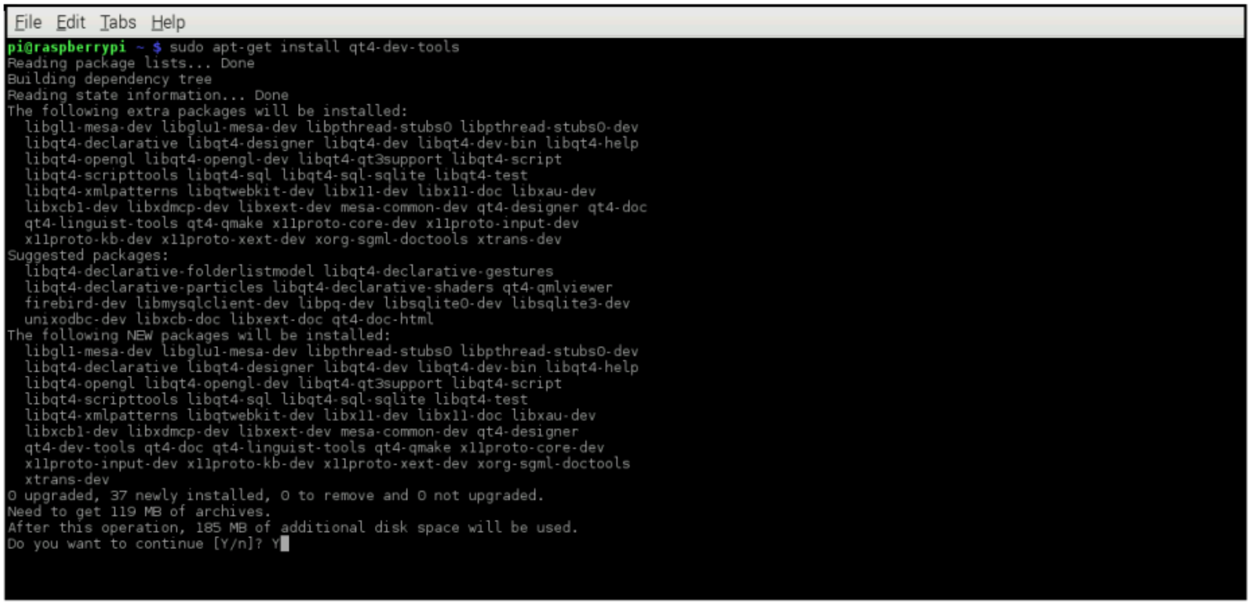


Would you like the ARM I2C interface to be enabled?

The ARM I2C interface is enabled

Figure 47. Enabling the I2C Interface

After enabling the SPI and I2C interfaces, the first step is to install the QT dev tools module. After opening a terminal, run the command “sudo apt-get install qt4-dev-tools”.



```
pi@raspberrypi ~ $ sudo apt-get install qt4-dev-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libgl1-mesa-dev libglu1-mesa-dev libpthread-stubs0 libpthread-stubs0-dev
  libqt4-declarative libqt4-designer libqt4-dev libqt4-dev-bin libqt4-help
  libqt4-opengl libqt4-opengl-dev libqt4-qt3support libqt4-script
  libqt4-scripttools libqt4-sql libqt4-sql-sqlite libqt4-test
  libqt4-xmlpatterns libqtwebkit-dev libx11-dev libx11-doc libxau-dev
  libxcb1-dev libxdmcp-dev libxext-dev mesa-common-dev qt4-designer qt4-doc
  qt4-linguist-tools qt4-qmake x11proto-core-dev x11proto-input-dev
  x11proto-kb-dev x11proto-xext-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  libqt4-declarative-folderlistmodel libqt4-declarative-gestures
  libqt4-declarative-particles libqt4-declarative-shaders qt4-qmlviewer
  firebird-dev libmysqlclient-dev libpq-dev libsqlite0-dev libsqlite3-dev
  unixodbc-dev libxcb-doc libxext-doc qt4-doc-html
The following NEW packages will be installed:
  libgl1-mesa-dev libglu1-mesa-dev libpthread-stubs0 libpthread-stubs0-dev
  libqt4-declarative libqt4-designer libqt4-dev libqt4-dev-bin libqt4-help
  libqt4-opengl libqt4-opengl-dev libqt4-qt3support libqt4-script
  libqt4-scripttools libqt4-sql libqt4-sql-sqlite libqt4-test
  libqt4-xmlpatterns libqtwebkit-dev libx11-dev libx11-doc libxau-dev
  libxcb1-dev libxdmcp-dev libxext-dev mesa-common-dev qt4-designer
  qt4-dev-tools qt4-doc qt4-linguist-tools qt4-qmake x11proto-core-dev
  x11proto-input-dev x11proto-kb-dev x11proto-xext-dev xorg-sgml-doctools
  xtrans-dev
0 upgraded, 37 newly installed, 0 to remove and 0 not upgraded.
Need to get 119 MB of archives.
After this operation, 185 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Figure 48. Installing qt4-dev-tools in terminal

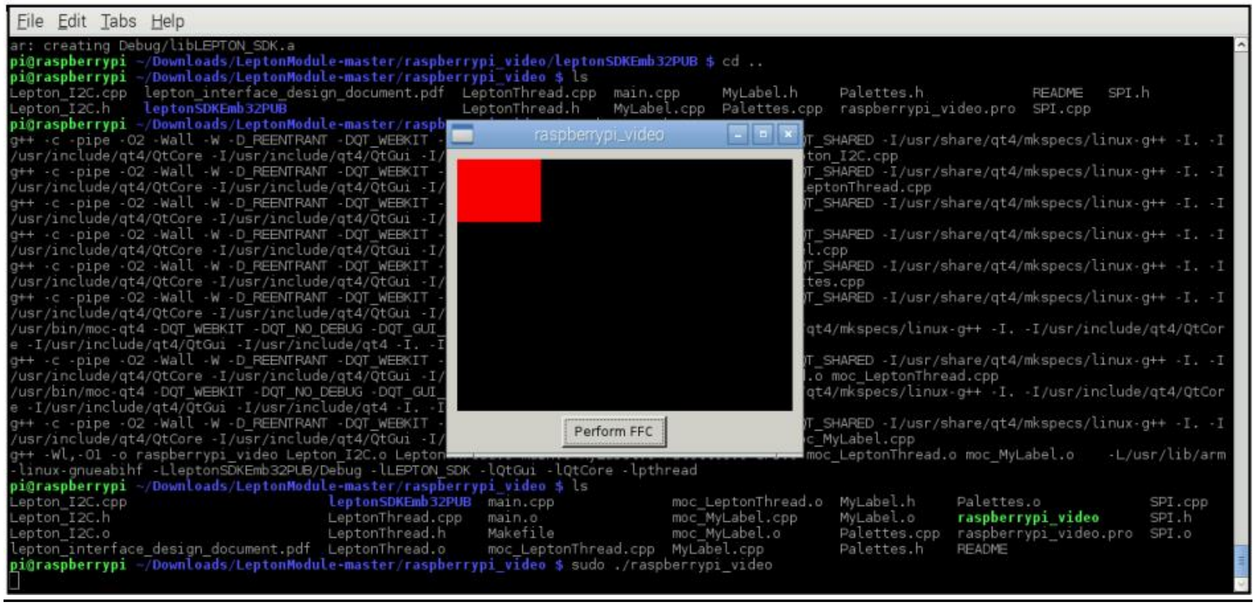
Setting Up

1. Go to https://github.com/groupgets/LeptonModule on the Pi and download the code as a zip file
2. After downloading, move into the right directory in the terminal by running the command “cd Downloads”
3. Unzip the file using the command “unzip LeptonModule-master.zip”
4. Change the working directory using “cd LeptonModule-master/software/raspberrypi_libs/leptonSDKEmd32PUB”
5. Run the make command by typing “make” and then hitting the Enter key
6. Once that process finishes, run the command “cd ..” twice to back up into the “software” directory

raspberrypi_video

1. Run the command “cd raspberrypi_video” to move into the video directory.
2. Run the command “qmake && make” to compile the code.
3. Next, run the command “sudo ./raspberrypi_video”. After this command is executed, the video should pop up in a new window
4. Troubleshooting Tip: Although it shouldn’t happen with the recommended connections, you may get an error like the one shown below: a red square in a black window. If this is the case, carefully remove the Lepton module from the breakout board. That is right, pull it from the socket, while it is powered. Then (again, very carefully) pop it back into

place. Images should start pouring in. Remember, if you have issues getting the camera to work with pin 26 on the Pi, you may need to adjust the CS pin to pin 24 (next to the CLK pin) like this tutorial from FLIR.



```
ar: creating Debug/libLEPTON_SDK.a
pi@raspberrypi ~/Downloads/LeptonModule-master/raspberrypi_video/leptonSDKEmb32PUB $ cd ..
pi@raspberrypi ~/Downloads/LeptonModule-master/raspberrypi_video $ ls
Lepton_I2C.cpp  lepton_interface_design_document.pdf  LeptonThread.cpp  main.cpp     MyLabel.h     Palettes.h             README   SPI.h
Lepton_I2C.h    leptonSDKEmb32PUB                     LeptonThread.h    MyLabel.cpp  Palettes.cpp  raspberrypi_video.pro  SPI.cpp
pi@raspberrypi ~/Downloads/LeptonModule-master/raspb
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -DQT_WEBKIT -                       T_SHARED -I/usr/share/qt4/mkspecs/linux-g++ -I. -I
/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/                       ton_I2C.cpp
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -DQT_WEBKIT -                       T_SHARED -I/usr/share/qt4/mkspecs/linux-g++ -I. -I
/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/                       eptonThread.cpp
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -DQT_WEBKIT -                       T_SHARED -I/usr/share/qt4/mkspecs/linux-g++ -I. -I
/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -DQT_WEBKIT -                       T_SHARED -I/usr/share/qt4/mkspecs/linux-g++ -I. -I
/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/                       l.cpp
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -DQT_WEBKIT -                       T_SHARED -I/usr/share/qt4/mkspecs/linux-g++ -I. -I
/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/                       tes.cpp
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -DQT_WEBKIT -                       T_SHARED -I/usr/share/qt4/mkspecs/linux-g++ -I. -I
/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/
/usr/bin/moc-qt4 -DQT_WEBKIT -DQT_NO_DEBUG -DQT_GUI_                       qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCor
e -I/usr/include/qt4/QtGui -I/usr/include/qt4 -I. -I
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -DQT_WEBKIT -                       T_SHARED -I/usr/share/qt4/mkspecs/linux-g++ -I. -I
/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/                       .o moc_LeptonThread.cpp
/usr/bin/moc-qt4 -DQT_WEBKIT -DQT_NO_DEBUG -DQT_GUI_                       qt4/mkspecs/linux-g++ -I. -I/usr/include/qt4/QtCor
e -I/usr/include/qt4/QtGui -I/usr/include/qt4 -I. -I
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -DQT_WEBKIT -                       T_SHARED -I/usr/share/qt4/mkspecs/linux-g++ -I. -I
/usr/include/qt4/QtCore -I/usr/include/qt4/QtGui -I/                       c_MyLabel.cpp
g++ -Wl,-O1 -o raspberrypi_video Lepton_I2C.o Lepton                       moc_LeptonThread.o moc_MyLabel.o    -L/usr/lib/arm
-linux-gnueabihf -LleptonSDKEmb32PUB/Debug -lLEPTON_SDK -lQtGui -lQtCore -lpthread
pi@raspberrypi ~/Downloads/LeptonModule-master/raspberrypi_video $ ls
Lepton_I2C.cpp                        leptonSDKEmb32PUB  main.cpp               moc_LeptonThread.o  MyLabel.h     Palettes.o             SPI.cpp
Lepton_I2C.h                          LeptonThread.cpp   main.o                 moc_MyLabel.cpp     MyLabel.o     raspberrypi_video      SPI.h
Lepton_I2C.o                          LeptonThread.h     Makefile               moc_MyLabel.o       Palettes.cpp  raspberrypi_video.pro  SPI.o
lepton_interface_design_document.pdf  LeptonThread.o     moc_LeptonThread.cpp   MyLabel.cpp         Palettes.h    README
pi@raspberrypi ~/Downloads/LeptonModule-master/raspberrypi_video $ sudo ./raspberrypi_video
```

Figure 49. raspberrypi_video error

Heat signatures should be seen while standing in front of the camera:

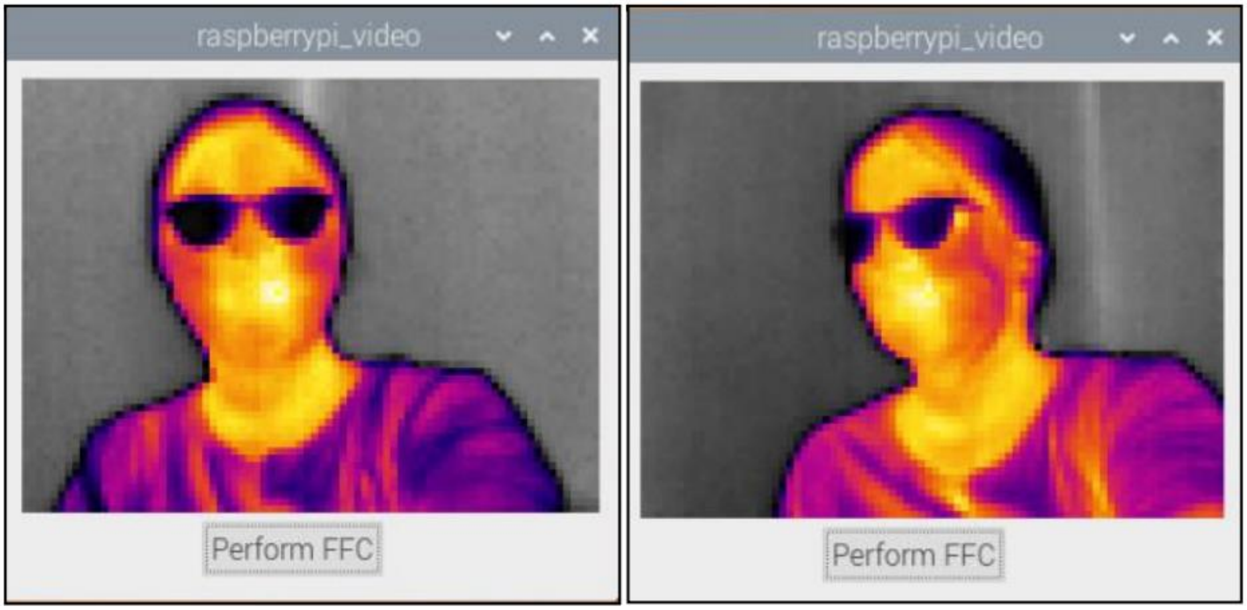


Figure 50. Example Video Captures

raspberrypi_capture

1. The next task is to capture these thermal images for further image processing. As we have already enabled the SPI and I2C interfaces there is no need for enabling them again. Follow below-given steps below to capture the thermal images in .pgm format.
2. Move into the raspberrypi_capture directory by running the command “cd ..” followed by “cd raspberrypi_capture”.
3. Run the command “sudo vi /etc/modules”
    a. The file already contains “i2c-dev”. Copy the code “spi-bcm2708” and right click above the i2c-dev line, choosing to paste the spi line in. Then copy a “#” sign to your clipboard. Paste it in before “i2c-dev” and “spi-bcm2708” to comment both lines out
    b. Then, type “:w” and press the Enter key to write those changes to the file. The file can now be closed
4. Next, a line of code must be changed in order to capture the images. Go to the File Explorer on the Raspberry Pi and navigate to the raspberrypi_capture directory. Double click on the “raspberrypi_capture.c” file to open it
    a. Once the file opens, scroll down to line 46. Change the line so it reads “static const char *device = “/dev/spidev0.0”; “ instead of having a “0.1” at the end
    b. Save the file, then it can be closed
5. Next, in the terminal (while in the raspberrypi_capture directory), run the command “gcc raspberrypi_capture.c”. This command compiles the code
6. Finally, run the command “sudo ./a.out” to capture an image
    a. Once this command is executed, there should be an image in the raspberrypi_capture folder called “IMG_0000.pgm” which is a grayscale picture of whatever the camera was pointed at while the “sudo ./a.out” command was executed.
    b. Any further images taken will just increase the number (0001, 0002, …)

Training the Dataset

- Before beginning training the dataset, it would be incredibly useful to complete the first two tutorial classes from the DeepLearning.AI TensorFlow Developer Professional Certificate course on Coursera.
    - For reference, it took me (CS experience of CS 1110, 2110, and 2150) about 20-25 hours to complete the first two courses
    - These two classes will help you to understand how to build a proficient model and train it to properly fit a dataset by adjusting certain parameters in Python code
    - The first course is "Introduction to TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning"
        - https://www.coursera.org/learn/introduction-tensorflow?specialization=tensorflow-in-practice
    - The second course is "Convolutional Neural Networks in TensorFlow"
        - https://www.coursera.org/learn/convolutional-neural-networks-tensorflow?specialization=tensorflow-in-practice
    - The last two courses in the specialization are not as useful, as course three deals with text, and course four with time-spaced data, both of which are not related to the image analysis done here
- After finishing those classes, you should have a good understanding of how to develop a model and adjust the parameters to better fit the data
- To code a model, I used the Anaconda Spyder IDE, but any Python IDE should work if you can install/import TensorFlow into it
    - Pretty sure you can just run "pip install tensorflow" in the command line terminal in Spyder to install
- Here is a GitHub link to the code and file setup I have so my model can be run correctly. If you download it as a zip, unzip it, and the Python file should run correctly
    - https://github.com/rpw6fx/TensorFlow-Research

Results

- After spending time tweaking the model to fit the data as best as possible, my final model had accuracy and loss graphs like the ones below
- After training with the model shown in the GitHub python code 25 times, the averages for training and validation accuracy, as well as training and validation loss were:
    - Training Accuracy: 98.442%
    - Validation Accuracy: 97.087%
        - Started even with training accuracy (mid 98s), but as I trained more, it kept dropping slightly to end around 96% after training 25 times
    - Training Loss: 0.0547
    - Validation Loss: 0.1427
        - Same as validation accuracy, started close to training loss (0.07ish), but as training went on, kept increasing to around 0.23 by the end of training for 25 times

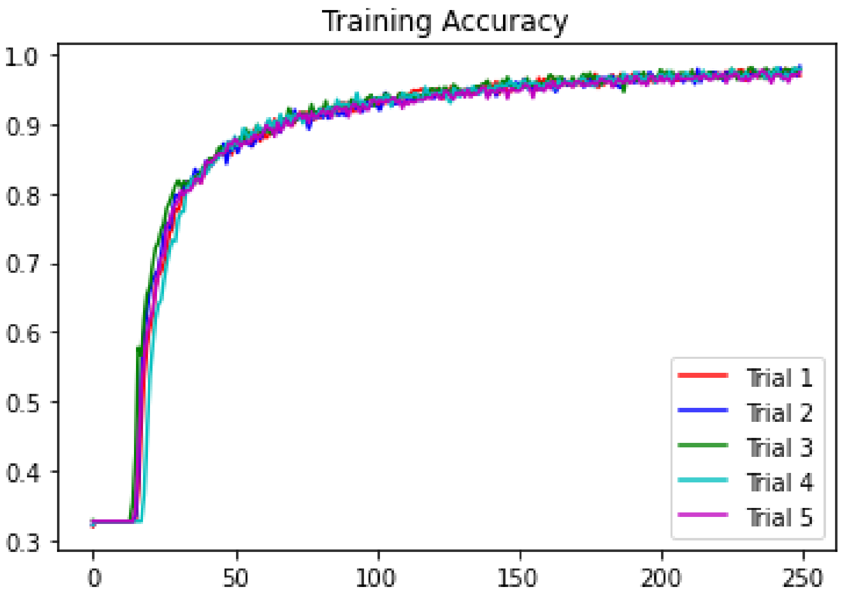


Figure 51. Training Accuracy Plot

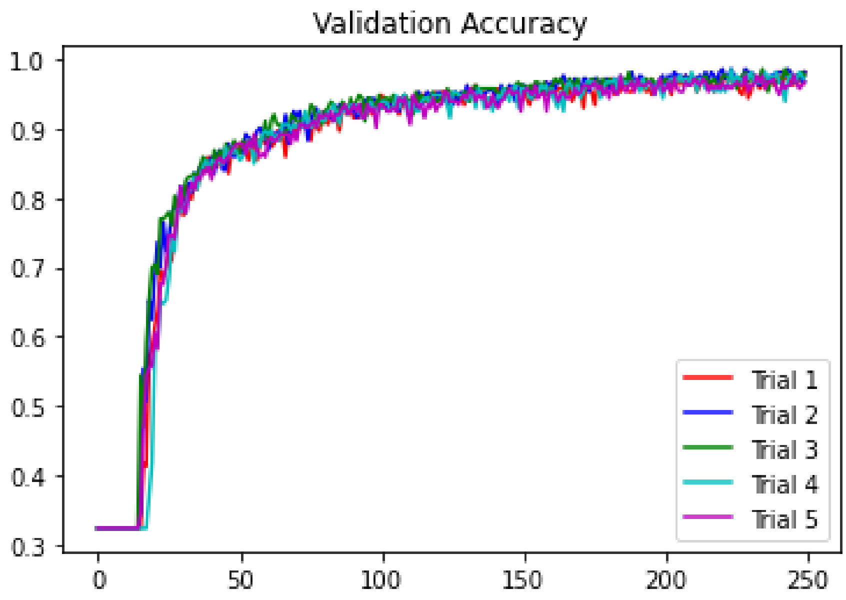


Figure 52. Validation Accuracy Plot

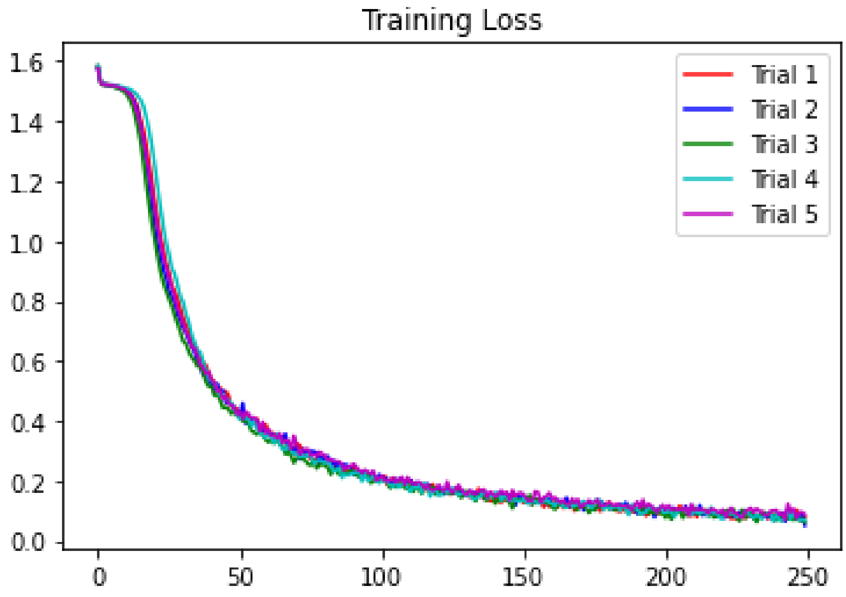


Figure 53. Training Loss Plot

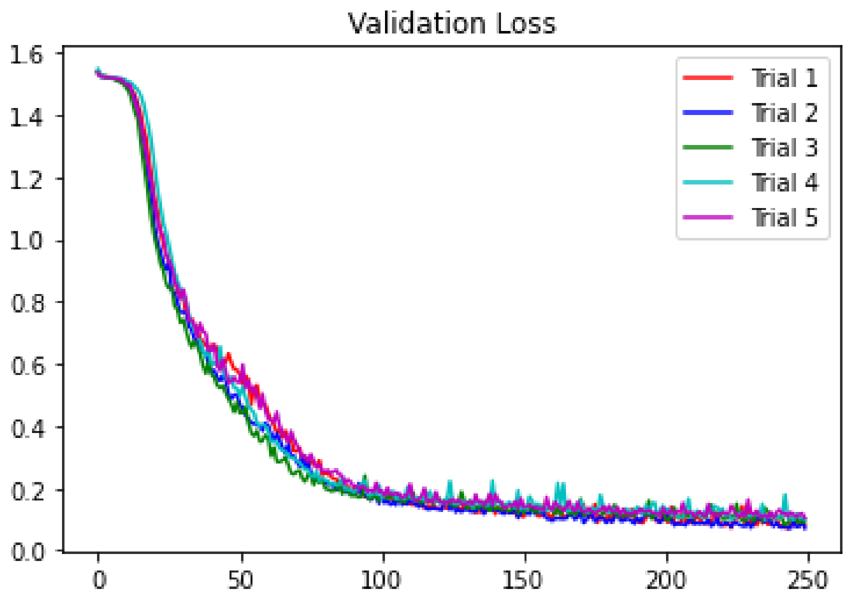


Figure 54. Validation Loss Plot

Conclusion and Future Scope

Throughout the course of this project, I was able to set up the Raspberry Pi 4 and make it compatible with the FLIR Lepton camera module to allow the saving of images to the file explorer. By using the Coursera TensorFlow courses, I was able to learn how to develop, train, and validate a model that precisely fit the data to around 98.5% training and validation accuracy.

In the future, the dataset should be trained on a wider variety of images, as many people in the frame had relatively similar poses and distances from the camera, contributing to overfitting the model to the data. Additionally, the model should be trained on more images. While around 1000 images give us a good start, adding thousands more images (and ones that mimic the proposed situation) with variety should help with overfitting and validate that the model is correct. This project focused solely on still images. In future analysis, it will be incredibly useful to develop some way of calculating the number of people in the room with live video, so changes can be made to the HVAC system directly when they are needed. Finally, the camera model and Raspberry Pi OS are both out of date. If possible, it will be important to find a way to update the camera to the most updated model (FLIR Lepton 3.5) and the Raspberry Pi 4 Operating System to the newest release (currently Debian Bullseye) while still being able to complete the project, so the research is completely up to date with industry standards.

References/Installation Links

1. TensorFlow Tutorials: https://www.coursera.org/professional-certificates/tensorflow-in-practice
2. Raspberry Pi OS: https://www.raspberrypi.com/software/
3. Connecting Raspberry Pi to UVA Wi-Fi: https://scholarslab.lib.virginia.edu/blog/raspberry-pi-uva-eduroam/
4. Lepton Module GitHub: https://github.com/groupgets/LeptonModule