AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: Sucessos Criativos, Lda

Title of WP: WP3 – AI-SAVES@ProjectHub

Activity Reference: 3.2 – Project template for AI-SAVES@ProjectHub

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

1. Project Title: 4

2. Project Category: 4

3. Project Description: 4

4. Keywords: 4

5. Target Audience: 4

6. Project Objectives: 4

7. Project Timeline: 5

8. Resources Provided: 5

9. Tools and Technologies Used: 5

10. Step-by-Step Implementation Plan: 5

Step 1: INTRODUCTION TO RASPBERRY PI 5

Step 2: ASSEMBLING THE RASPBERRY PI CASE 12

Step 3: INSTALLATION OF OPERATING SYSTEM ON SD CARD 14

Step 4: First Boot 18

Step 5: Navigating the desktop 19

Step 6: Basic Configuration 20

Step 7: Changing Configurations 22

Step 8: Setting Up TigerVNC for Remote Access on Raspberry Pi 24

Step 9: Updating the System 27

Step 10: Setting a Virtual Environment 29

11. Evaluation Criteria: 34

12. Languages Available: 34

13. Contributors: 34

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

17. Feedback and Continuous Improvement: 35

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

1. Project Title:

Getting Started with Raspberry Pi 5: Setup, Configuration, and Hello World in Python

2. Project Category:

Microelectronics Fundamentals

3. Project Description:

.This project introduces participants to the essential first steps in using the Raspberry Pi 5. It guides them through installing the operating system, configuring system settings, enabling key interfaces, and connecting remotely via VNC. Participants will also set up a Python development environment and write their first script — a simple "Hello World" program — using the Geany IDE. This project ensures that the Raspberry Pi is fully prepared for more advanced electronics or AI applications.

4. Keywords:

Raspberry Pi, Python, AI-VET Kit, Geany

---

5. Target Audience:

Both Educators and Students

6. Project Objectives:

- Learn to install and configure Raspberry Pi OS.
- Enable critical hardware interfaces (SPI, I2C, 1-Wire, VNC).
- Set up remote access to the Raspberry Pi via VNC.
- Create and manage a Python virtual environment.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Configure Geany as a development IDE.
- Write and run a basic “Hello World” Python program.

---

7. Project Timeline:

The estimated duration of this project is approximately 2 hours, assuming the participant has little to no prior experience.

8. Resources Provided:

- AI-VET Kit (Raspberry Pi 5, Case, SD Card)

MATERIAL	
Raspberry Pi 5	
27W USB Type-C Power Supply for Raspberry Pi 5	
64GB MICROSDHC CARD CKINGSTON SDCS2/64GB (IM13)	
Case for Pi 5 With Temperature-Controlled Blower Fan (IM7)	

9. Tools and Technologies Used:

- Python programming language
- Geany IDE -
- VNC Viewer (optional for remote access) https://tigervnc.org

---

10. Step-by-Step Implementation Plan:

Step 1: INTRODUCTION TO RASPBERRY PI

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

The Raspberry Pi is a device that functions as a fully operational computer in a compact, affordable package. It can be used for activities like browsing the web, playing games, learning to write programs, or creating circuits and physical devices. Raspberry Pi is categorized as a single-board computer, which means it is a complete computer built on a single printed circuit board. Although small, with dimensions similar to a credit card, the Raspberry Pi can handle tasks typically performed by larger computers, such as web browsing, gaming, and controlling other devices.

The Raspberry Pi family was created to promote hands-on computer education globally. The non-profit Raspberry Pi Foundation, founded by its creators, originally produced only a few thousand units for testing in 2012, which sold out immediately. Since then, over fifty million boards have been shipped worldwide and have been integrated into homes, classrooms, offices, data centers, factories, and even autonomous boats and satellites.

Unlike a traditional computer, which hides its components inside a case, a standard Raspberry Pi has its parts, ports, and features visible. This open design makes it a useful tool for learning about the different parts of a computer and understanding how to connect various peripherals required to get started.

The next figure shows a Raspberry Pi 5 from above:

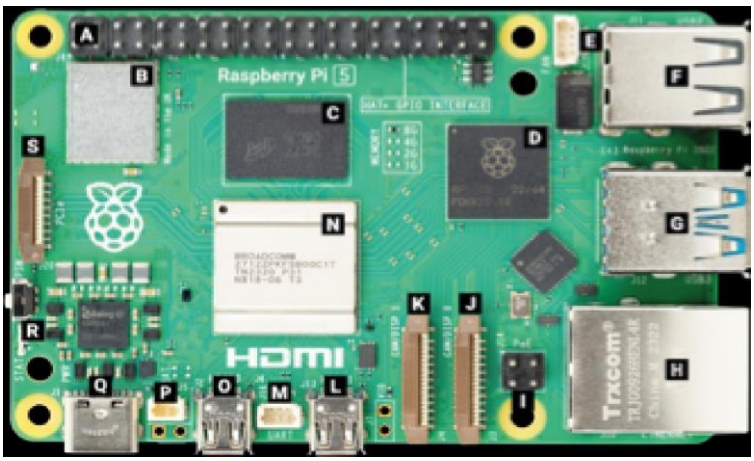


Figure 2- Upper view of the raspberry PI5

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

A	GPIO header	K	CSI/DSI camera/display port 1
B	Wireless	L	Micro HDMI 1
C	RAM	M	Connector for UART serial port
D	RP1 I/O controller chip	N	System-on-chip
E	Connector for fan	O	Micro HDMI 0
F	USB 2.0	P	Real-time clock battery header
G	USB 3.0	Q	USB Type-C power in
H	Ethernet port	R	Power button
I	Power-over-Ethernet (PoE) pins	S	Connector for PCI Express (PCIe)
J	CSI/DSI camera/display port 0		

There is one more port on the Raspberry Pi, but it's located on the underside of the board. Here, a microSD card connector can be found. This connector is used for the Raspberry Pi's storage, where a microSD card holds files, installed software, and the operating system that runs the device.



Figure 3- Raspberry PI SD card connector

The following image displays the pinout diagram for the Raspberry Pi 5 GPIO header, which consists of 40 pins:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

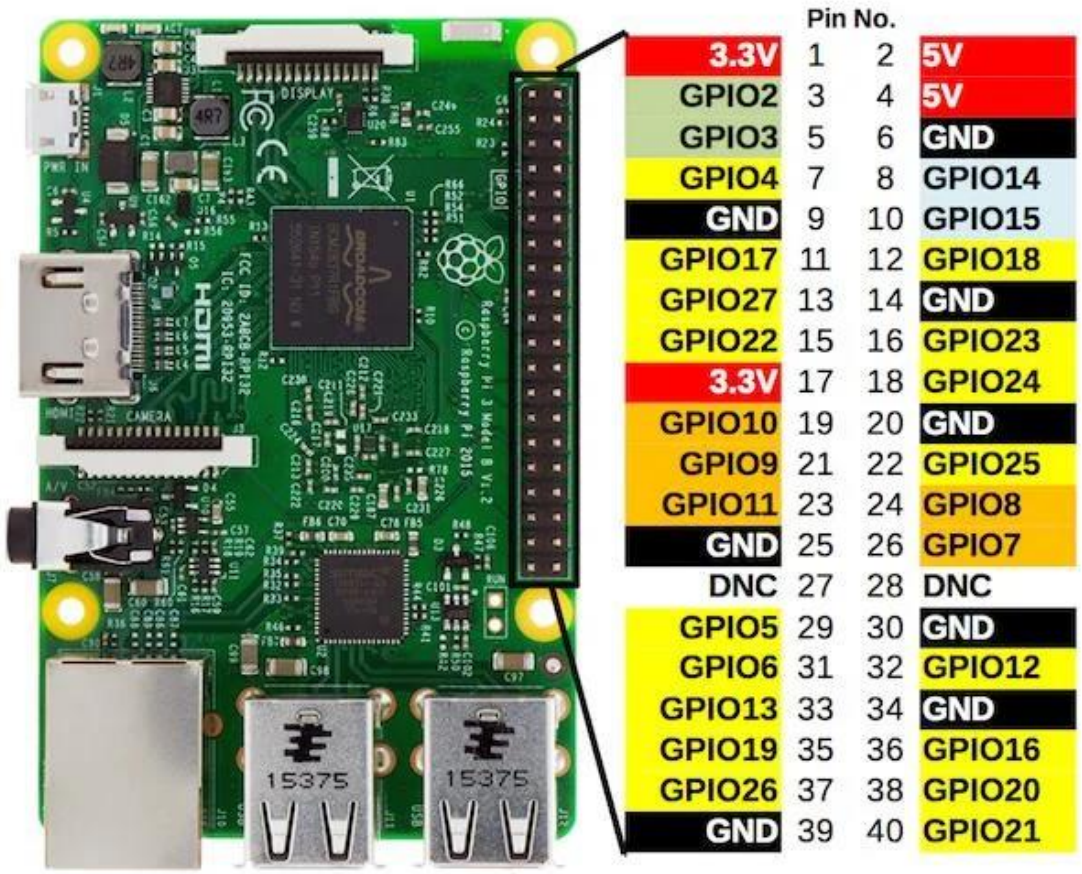


Figure 4- Pinout diagram for the Raspberry Pi 5 GPIO header

This header allows the connection to various peripherals and hardware components, such as sensors, LEDs, and communication modules. The pins are used for power, ground, general-purpose input/output (GPIO), and communication protocols (e.g., I2C, SPI, UART). Below is a detailed explanation of the pinout based on the image:

Power Pins

These pins supply power to external devices:

- Pin 1 : 3.3V Power – Provides a 3.3V power source.

- Pin 2 : 5V Power – Provides a 5V power source.

- Pin 4 : 5V Power – Another 5V power source.

- Pin 17 : 3.3V Power – Additional 3.3V power source.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Ground Pins (GND)

Ground pins are used as a return path for current.

- Pin 6, 9, 14, 20, 25, 30, 34, 39 : Ground pins. These are connected to the system's ground and provide a return path for electric current from devices connected to the GPIO.

GPIO Pins

General Purpose Input/Output (GPIO) pins are programmable pins that can be used as either inputs or outputs for a wide range of devices. GPIO pins are central to many Raspberry Pi projects and are often used to control LEDs, read sensor data, or send signals to other devices.

- Pin 3 (GPIO 2) : I2C1 SDA – Used for I2C communication, this pin is the Serial Data Line (SDA) for I2C communication.
- Pin 5 (GPIO 3) : I2C1 SCL – This is the Serial Clock Line (SCL) for I2C communication.
- Pin 7 (GPIO 4) : GPCLK0 – A general-purpose clock pin, can output a clock signal for use by external devices.
- Pin 11 (GPIO 17) : GPIO pin used for various input/output functions.
- Pin 13 (GPIO 27) : Another GPIO pin, often used for general purposes.
- Pin 15 (GPIO 22) : Another GPIO pin, commonly used for various projects.
- Pin 19 (GPIO 10) : SPI0 MOSI – Master Out Slave In (MOSI) pin for SPI communication.
- Pin 21 (GPIO 9) : SPI0 MISO – Master In Slave Out (MISO) pin for SPI communication.
- Pin 23 (GPIO 11) : SPI0 SCLK – The clock line for SPI communication.
- Pin 29 (GPIO 5) : GPIO pin used for general purposes.
- Pin 31 (GPIO 6) : GPIO pin used for general purposes.
- Pin 33 (GPIO 13) : PWM1 – A Pulse Width Modulation (PWM) pin that can be used for controlling the brightness of LEDs, motors, etc.
- Pin 35 (GPIO 19) : PCM FS – Pulse Code Modulation Frame Sync, used for audio interfaces.
- Pin 37 (GPIO 26) : GPIO pin used for general purposes.
- Pin 12 (GPIO 12) : PWM0 – Another PWM pin, used for controlling things like servos or LEDs.
- Pin 16 (GPIO 23) : GPIO pin used for general purposes.
- Pin 18 (GPIO 24) : GPIO pin used for general purposes.
- Pin 22 (GPIO 25) : GPIO pin used for general purposes.
- Pin 24 (GPIO 8) : SPI0 CE0 – Chip Enable for SPI communication.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- Pin 26 (GPIO 7) : SPI0 CE1 – Another Chip Enable pin for SPI communication.
- Pin 28 (GPIO 1) : EEPROM SCL – Clock line for the EEPROM.
- Pin 32 (GPIO 12) : PWM0 – Another PWM pin.
- Pin 36 (GPIO 16) : GPIO pin used for general purposes.
- Pin 38 (GPIO 20) : PCM DIN – Data input for Pulse Code Modulation, typically used for audio.
- Pin 40 (GPIO 21) : PCM DOUT – Data output for Pulse Code Modulation.

Communication Pins

I2C Communication :

- Pin 3 (GPIO 2) : I2C1 SDA – Serial Data Line for I2C communication.
- Pin 5 (GPIO 3) : I2C1 SCL – Serial Clock Line for I2C communication.

SPI Communication :

- Pin 19 (GPIO 10) : SPI0 MOSI – Master Out Slave In for SPI.
- Pin 21 (GPIO 9) : SPI0 MISO – Master In Slave Out for SPI.
- Pin 23 (GPIO 11) : SPI0 SCLK – Clock line for SPI communication.
- Pin 24 (GPIO 8) : SPI0 CE0 – Chip Enable 0 for SPI.
- Pin 26 (GPIO 7) : SPI0 CE1 – Chip Enable 1 for SPI.

UART Communication :

- Pin 8 (GPIO 14) : UART TX – Transmits data over UART.
- Pin 10 (GPIO 15) : UART RX – Receives data over UART.

PCM (Pulse Code Modulation) :

- Pin 35 (GPIO 19) : PCM FS – Frame Sync, used for audio interfaces.
- Pin 38 (GPIO 20) : PCM DIN – Data input for audio.
- Pin 40 (GPIO 21) : PCM DOUT – Data output for audio.

EEPROM Pins

- Pin 27 (GPIO 0) : EEPROM SDA – Serial Data Line for EEPROM communication.
- Pin 28 (GPIO 1) : EEPROM SCL – Serial Clock Line for EEPROM communication.

PWM Pins

- Pin 33 (GPIO 13) : PWM1 – Can be used for Pulse Width Modulation, which is useful for controlling devices like motors or LEDs.
- Pin 32 (GPIO 12) : PWM0 – Another PWM pin used for controlling devices.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

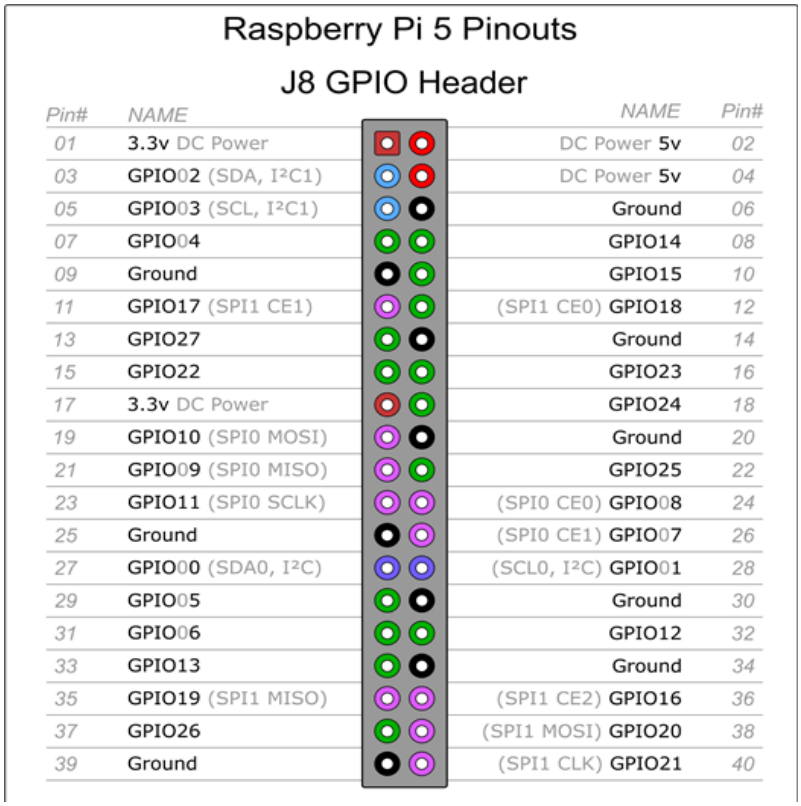


Figure 5 - Pinout diagram for the Raspberry Pi 5 GPIO header in detail

GPIO HEADER NUMBERING SYSTEMS

On the Raspberry Pi GPIO header, each pin has two different numbering systems: the physical pin number and the GPIO (General Purpose Input/Output) number. These numbers serve different purposes, and understanding the distinction between them is important to working with the Raspberry Pi.

1. Physical Pin Number (Pin Number)

- The physical pin number refers to the physical location of the pin on the Raspberry Pi's GPIO header.

- The pins are numbered sequentially from 1 to 40, starting from the top-left corner of the GPIO header (as shown in the image) and moving down and across in a zigzag pattern.

2. GPIO Number (BCM Number)

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- The GPIO number refers to the Broadcom SoC (System on Chip) numbering system, also known as the BCM numbering . These numbers represent the actual GPIO pin numbers controlled by the Raspberry Pi's processor.

- These numbers are specific to the functionality of the pins and are not necessarily sequential.

- Each physical pin is mapped to a specific GPIO number. For example, physical Pin 3 is associated with GPIO 2 and Pin 5 is GPIO 3 .

- GPIO pins can be used for a variety of purposes like input/output, PWM, I2C, SPI, UART, etc.

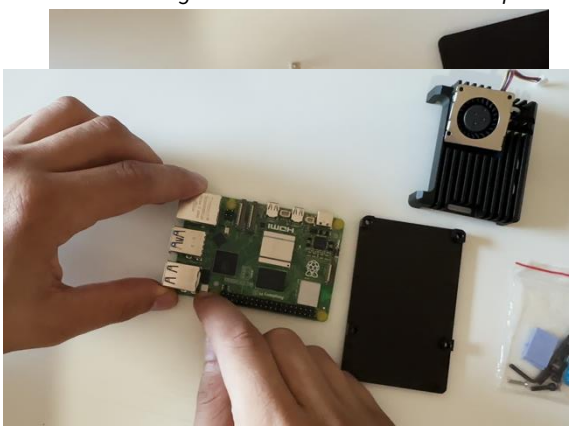
Step 2: ASSEMBLING THE RASPBERRY PI CASE

Begin by unpacking your Raspberry Pi from its box. Raspberry Pi is a robust piece of hardware, but that doesn't mean it's indestructible; try to get into the habit of holding the board by the edges, rather than on its flat sides, and be extra careful around the raised metal pins. If these pins are bent, at best it'll make using add-on boards and other extra hardware difficult and, at worst, might cause a short-circuit that will damage your Raspberry Pi.

Before connecting the fan, you'll need to remove the protection over the fan connection point. This is often a piece of film or a temporary cover that protects the contact points during shipping or assembly.

Figure 6- Removing the protective cover of the fan power connector

Figure 7- Removed cover of the fan power connector



Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

After removing the protective cover, you can proceed to connect the fan to the designated pins on the Raspberry Pi.

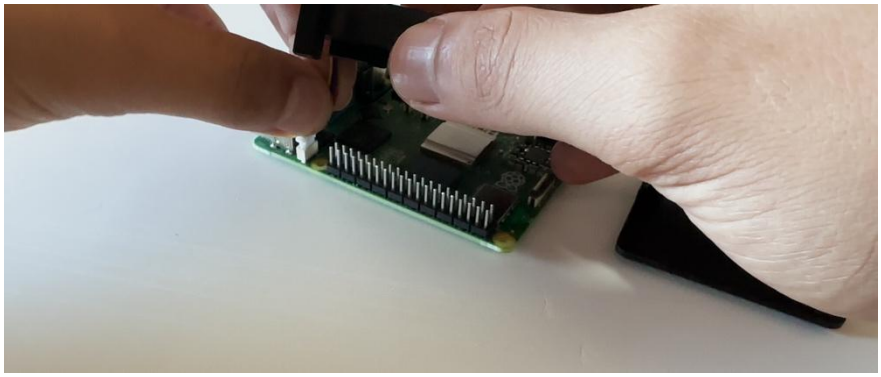


Figure 8- Connecting the fan power

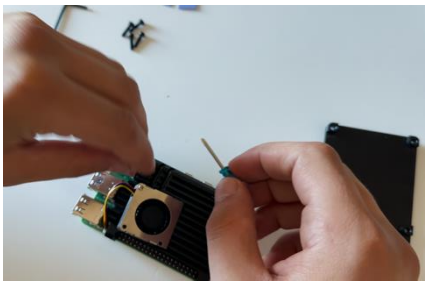
Make sure to handle this process gently to avoid damaging any of the components or connectors.

Align the Raspberry Pi board with the case standoffs. These are the raised parts of the case where the Pi will sit. Attach the board using screws provided. Make sure the USB and HDMI ports align with the case openings.



Figure 9- Fastening the Raspberry Pi board to the case with screws.

Figure 10-Ensuring the proper alignment of the ports in the case.



And finally:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

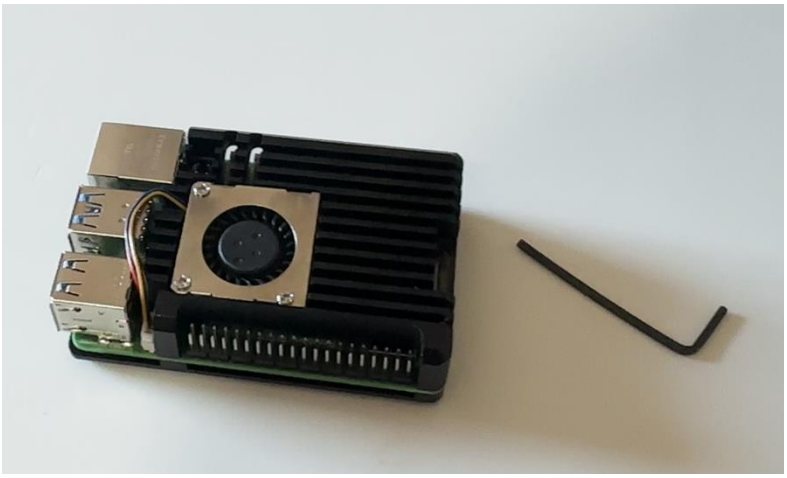


Figure 11- Fully assembled case with the Raspberry Pi securely mounted and the fan in place, ready for use.

Step 3: INSTALLATION OF OPERATING SYSTEM ON SD CARD

Based on Debian, Raspberry Pi OS is the official operating system for Raspberry Pi. The easiest way to install Raspberry Pi OS on a microSD card for your Raspberry Pi is to use the Raspberry Pi Imager tool, downloadable from *https://www.raspberrypi.com/software/ :*

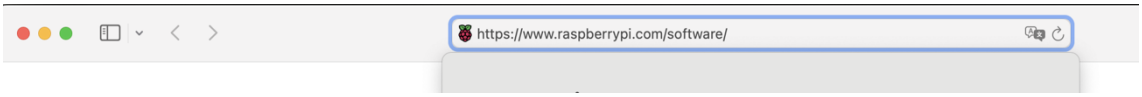
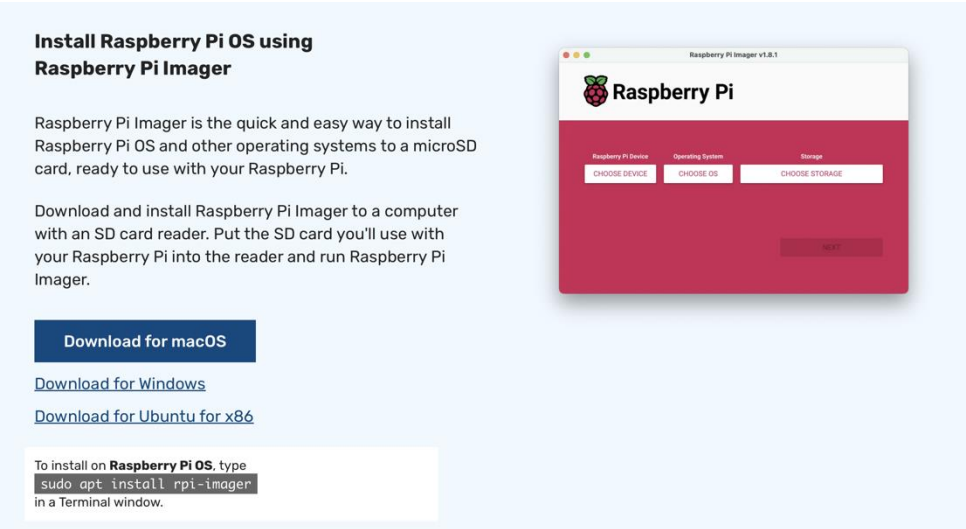


Figure 12- Acces to https://www.raspberrypi.com/software/

The Raspberry Pi Imager application is available for Windows, macOS, and Ubuntu Linux computers, so choose the relevant version for your system



Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Figure 13- Dowload Raspberry PI Imager

You can now attach your microSD card to your computer. You'll need a USB adapter unless your computer has a card reader built in — many laptops do, but not many desktops. Note that the microSD card doesn't need to be preformatted. Launch the Raspberry Pi Imager application

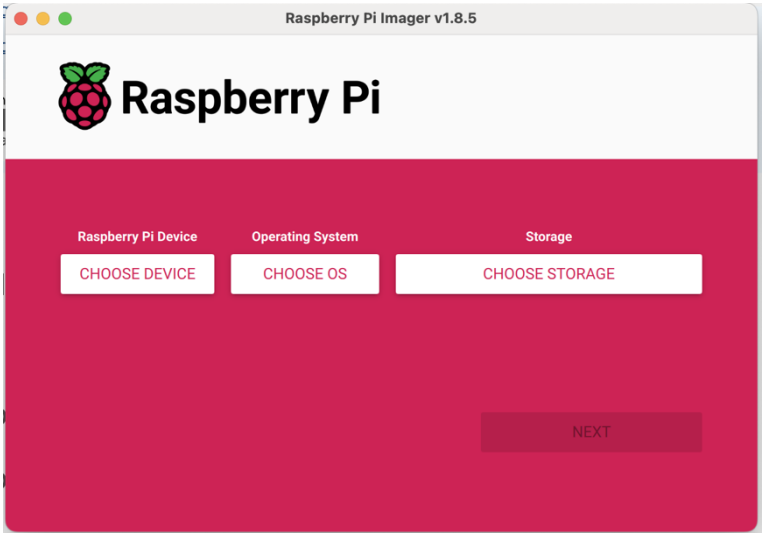


Figure 14- Launch Raspberry PI Imager

Click the Choose Device button to select which model of Raspberry Pi – in this case Raspeberry pi 5:

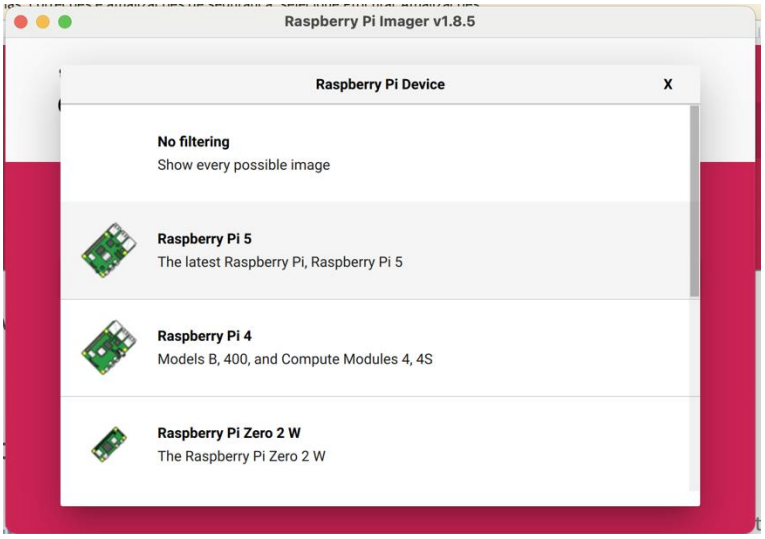


Figure 15- Raspberry PI Imager- Select Device

Next, Click Choose OS to select which operating system you would like to install. In our case will be the first option:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

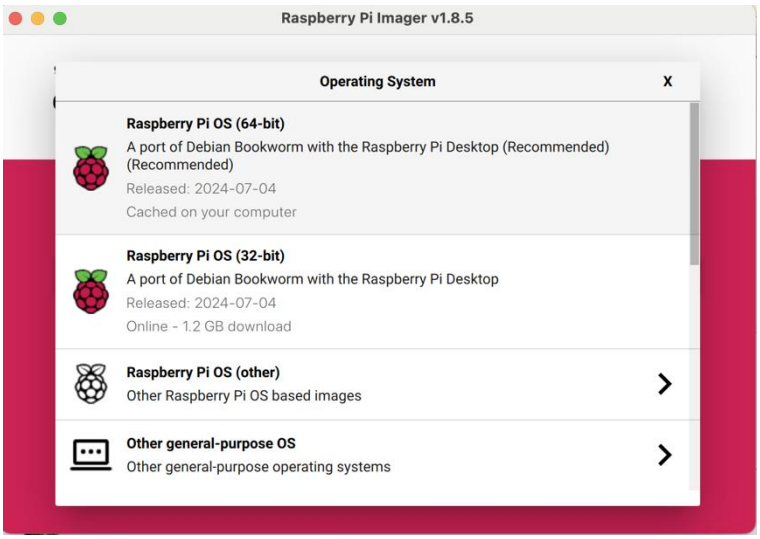


Figure 16- Raspberry PI Imager- Select Operating System

After this step, the following screen shows:

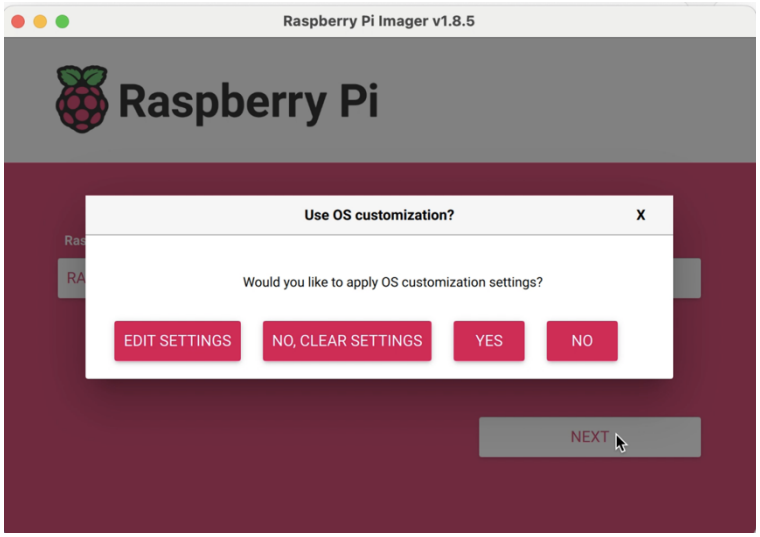


Figure 17- Raspberry PI Imager- Prepare to costumize setting

In this step, it would be better to do some basic customization as described on the next image:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

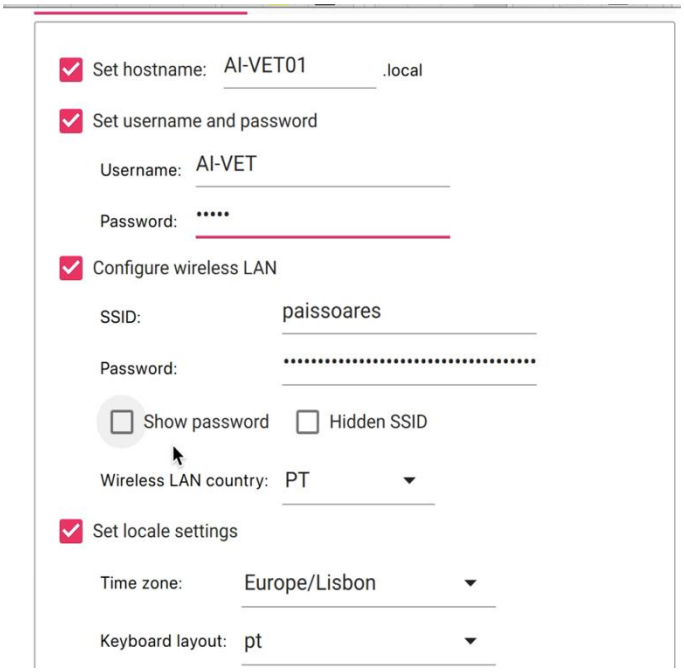


Figure 18- Raspberry PI Imager- Costumizing settings

Set Hostname: A hostname is a label that identifies the device on a local network. In this example, the hostname is set as "AI-VET01", followed by ".local". This means the device will be accessible on the local network using this name.

Set Username and Password: This section allows to set a username and password for accessing the system. The username "AI-VET" is shown, and a password is also configured but hidden for security.

Configure Wireless LAN: This section allows the user to set up wireless network (Wi-Fi) configuration. The SSID (name of the wireless network) is set to "paissoares." The password is entered and hidden by default, but there is an option to reveal it by clicking "Show password."

The Wireless LAN country is set to "PT," which stands for Portugal, ensuring the Wi-Fi operates within the local regulatory standards.

Set Locale Settings: This section handles the time zone and keyboard layout settings. The time zone is set to "Europe/Lisbon," and the keyboard layout is set to "pt" (Portuguese).

After this step, the writing process will start (and will take some time):

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Figure 19- Raspberry PI Imager- Writting

After some minutes the process will be completed:

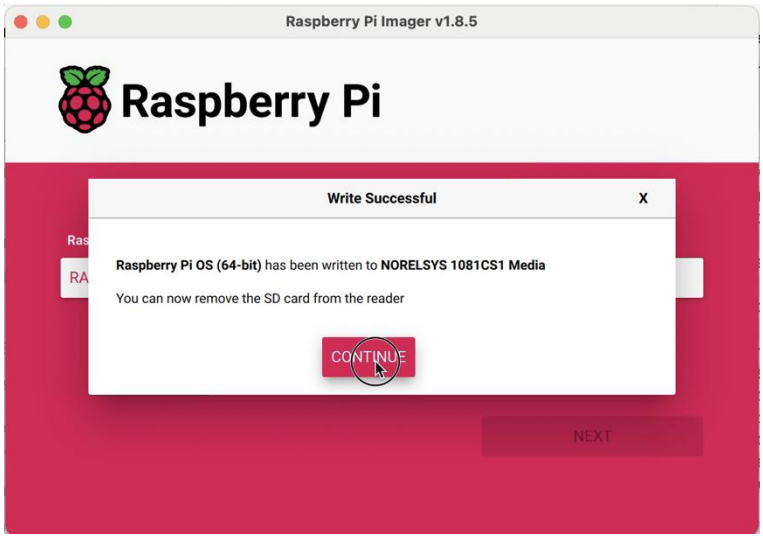
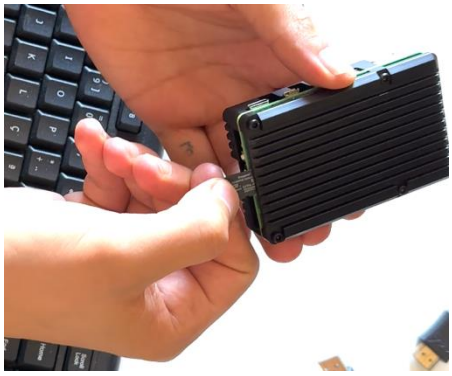


Figure 20- Raspberry PI Imager finishing

Step 4: First Boot

After formatting the card, you can now insert it into the microSD card connector located on the underside of the Raspberry Pi 5.



Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Figure 21- Insert the SD Card

Next, connect the keyboard and mouse to the USB ports, and attach the monitor using the Micro HDMI 0 port



Figure 22- Connecting the keyboard, mouse and the monitor

Lastly, power the Raspberry Pi 5 by connecting it to a power source via the USB-C power port to ensure the system starts up correctly:

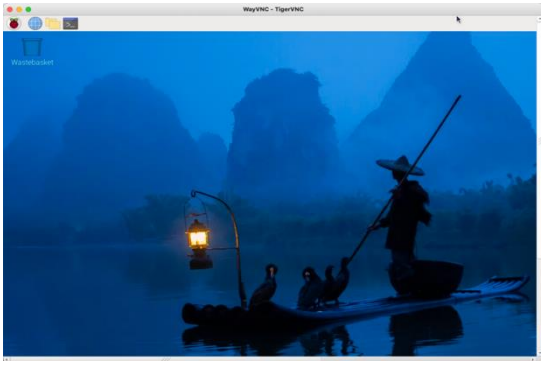


Figure 23- Sarting Raspberry PI

Step 5: Navigating the desktop

The version of Raspberry Pi OS installed is known as 'Raspberry Pi OS with desktop', referring to its main graphical user interface. The following images describe the main components:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

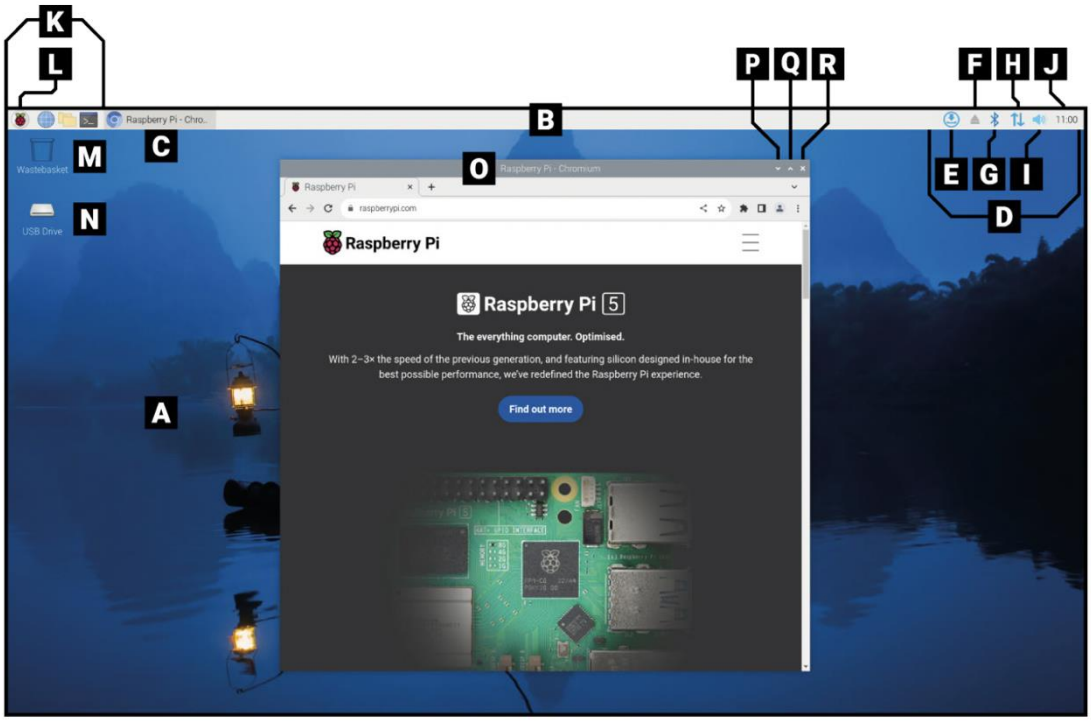


Figure 24- Raspberry PI Desktop

A	Wallpaper	J	Clock
B	Taskbar	K	Launcher
C	Task	L	Menu (or Raspberry Pi icon)
D	System Tray	M	Wastebasket icon
E	Software Update icon	N	Removable drive icon
F	Media eject	O	Window title bar
G	Bluetooth icon	P	Minimise
H	Network icon	Q	Maximise
I	Volume icon	R	Close

Step 6: Basic Configuration

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Click on the Raspberry Pi icon, move your mouse pointer to select the Preferences category, then click on Raspberry Pi Configuration to load it:

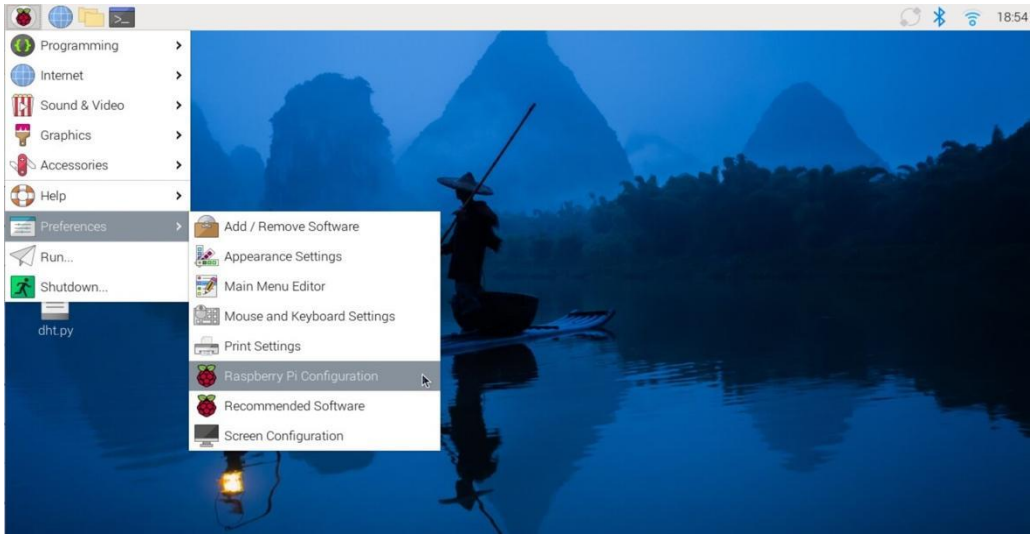


Figure 25- Access to Raspberry Pi Configuration Menu

The tool is split into five tabs:

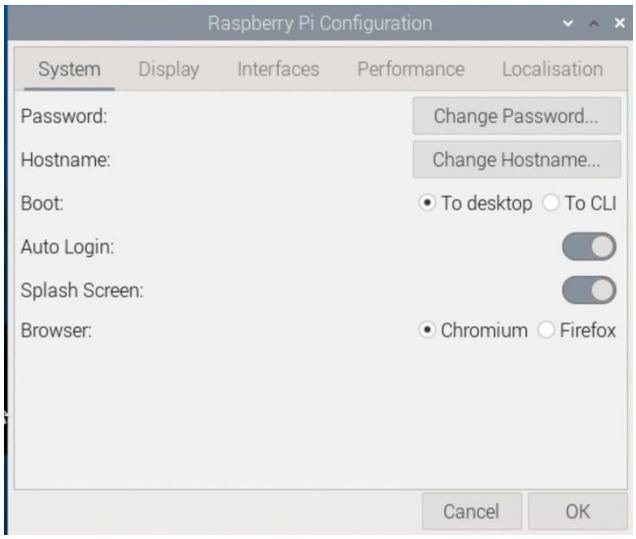


Figure 26- Raspberry Pi Configuration Menu

The first of these is System: this allows you to change the password of your account, set a host name — the name your Raspberry Pi uses on your local wireless or wired network—and alter a range of other settings, including choosing a default web browser. The majority of these shouldn't need changing. Click on the Display tab to

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

bring up the next category. Here you can alter the screen display settings, if needed, to suit your TV or monitor.

The Interface tab offers a range of settings, all of which (except for Serial Console and Serial Port) start off disabled. These settings should only be changed if you're adding new hardware, and even then only if instructed by the hardware's manufacturer.

In the Performance tab can be configured the overlay file system, which is a way to run Raspberry Pi without writing changes to the microSD card.

Finally, click on the Localisation allows to change your locale, which controls things like the language used in Raspberry Pi OS and how numbers are displayed; change the time zone; change the keyboard layout; and set your country for Wi-Fi purposes.

Step 7: Changing Configurations

Change the configurations according with the following image:

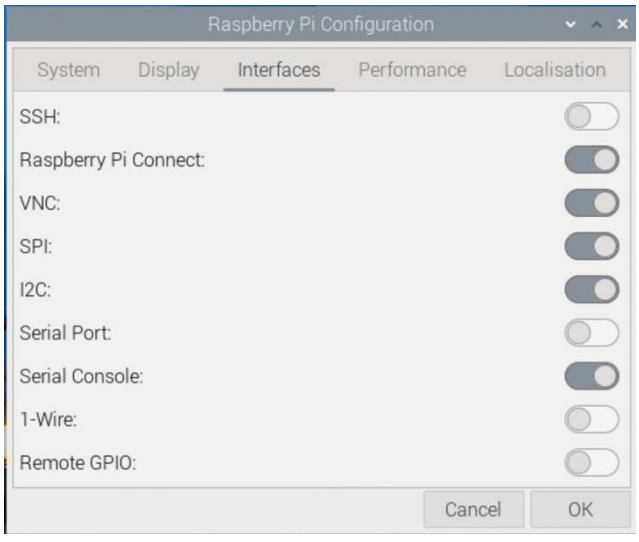


Figure 27- Raspberry Pi Configuration Menu

These settings allow you to customize how your Raspberry Pi interacts with other devices, peripherals and networks. Considering the development of the project cover in this handbook, we will need to enable certain interfaces to connect and communicate with external sensors, actuators, or other systems:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

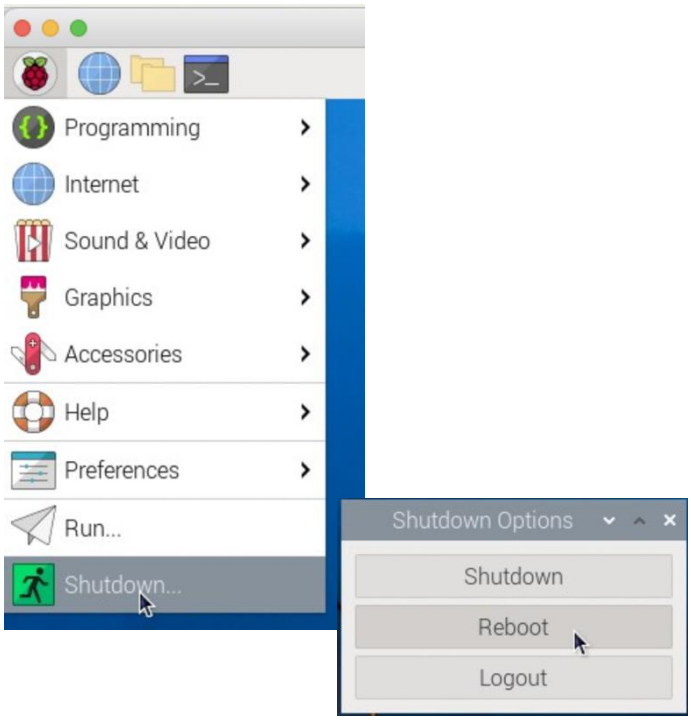
VNC: By enabling VNC (Virtual Network Computing), you can access the Raspberry Pi's desktop environment remotely from another device. This is useful if you want a graphical interface (rather than just command-line access like SSH) without physically connecting a monitor.

SPI: The Serial Peripheral Interface (SPI) is a communication protocol used to connect devices like sensors, displays, or other peripherals. Enabling SPI is important since we will work on projects that involve communication between the Raspberry Pi and these types of devices.

I2C: The Inter-Integrated Circuit (I2C) is another communication protocol, often used to connect low-speed devices like sensors, OLED displays, and other peripherals. You'll need to enable I2C for projects that rely on such components.

1-Wire: 1-Wire is a protocol used for communication with simple devices like temperature sensors. If you're planning to use 1-Wire sensors, enabling this option is necessary.

MAKE SURE YOU REBOOT AFTER THESE COMMANDS!!!



Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Figure 28- Rebboting Raspberry Pi

Step 8: Setting Up TigerVNC for Remote Access on Raspberry Pi

VNC, or Virtual Network Computing, is a powerful tool that allows the remote access and control the graphical desktop of a Raspberry Pi from another device, such as a laptop, tablet, or smartphone. This is particularly useful when working on a headless Raspberry Pi (one without a connected monitor, keyboard, or mouse), as it gives full access to the Pi's graphical user interface (GUI) from anywhere on your local network or even over the internet.

In this training it will be used TigerVNC (https://tigervnc.org). TigerVNC is completely free and open-source, which means there are no costs associated with using it, and the software can be freely modified and improved by the community. This aligns with the spirit of open-source development around the Raspberry Pi and allows users to benefit from continuous updates and improvements.

Follow the following steps in order to configure TigerVNC for Remote Access on Raspberry Pi

1. Go to https://tigervnc.org and download

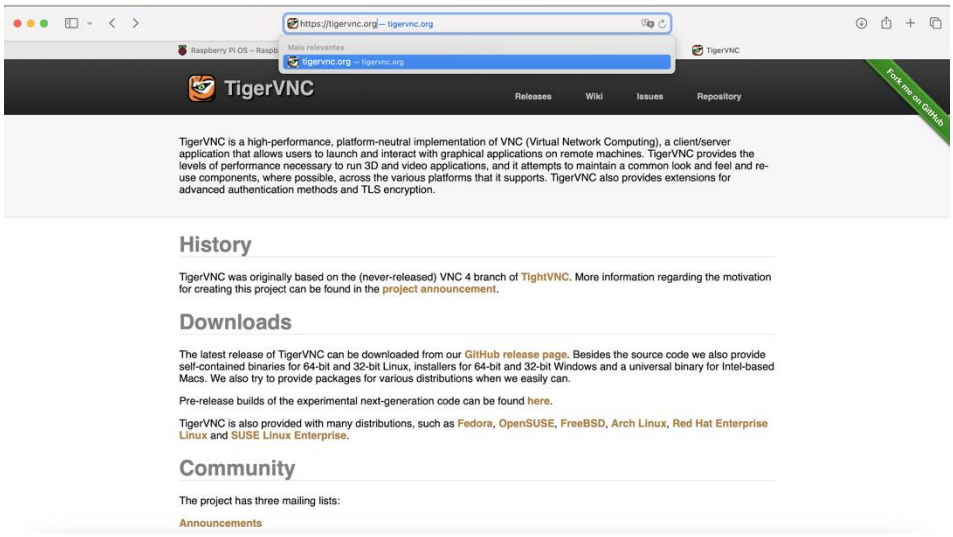


Figure 29- Access to TigerVNC website

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Now that you're on the TigerVNC landing page, click on the "GitHub release page" link to access the downloadable link.

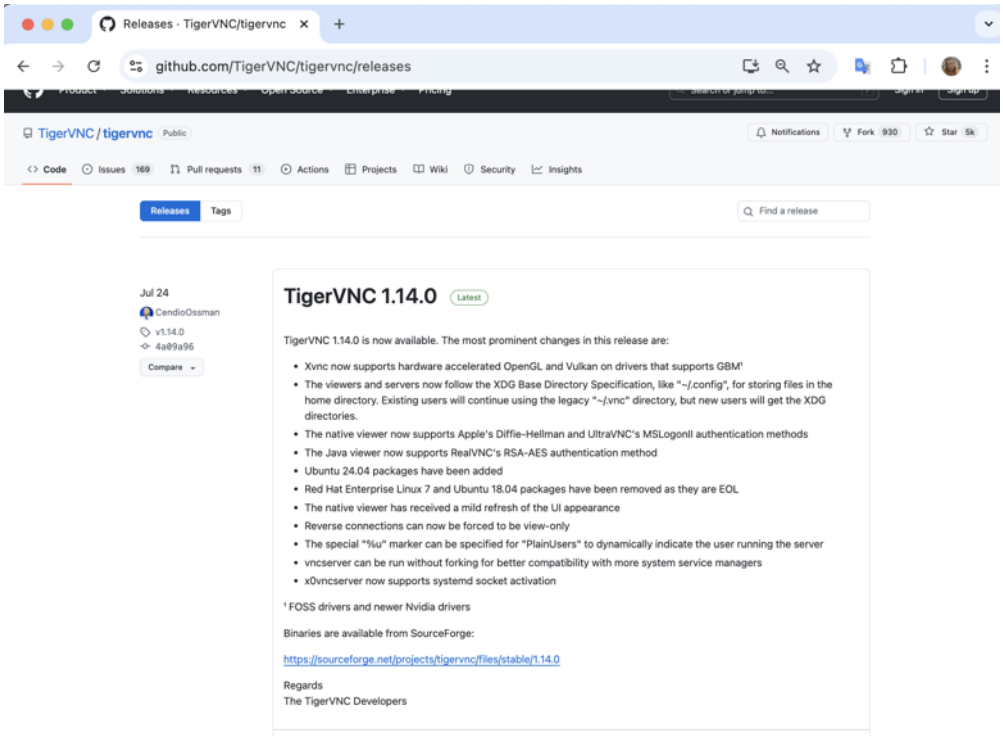


Figure 30- TigerVNC Github area

After opening the GitHub page, look for the "SourceForge" link. Click on it, and you'll be automatically redirected to the page where you can download the app.

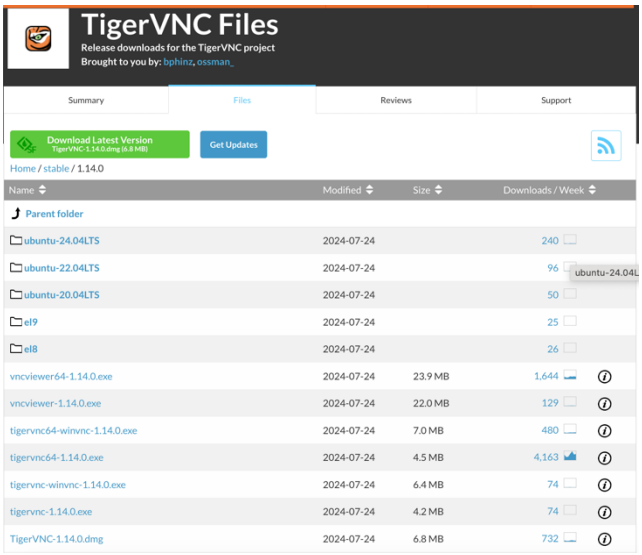


Figure 31- Dowload TigerVNC

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Click on the “Download Latest Version” and you will be ready to go from your computer.

Your device and raspberry PI5 need to be connected on the same network!

On Raspberry PI5 Go to Terminal and execute the command ifconfig:

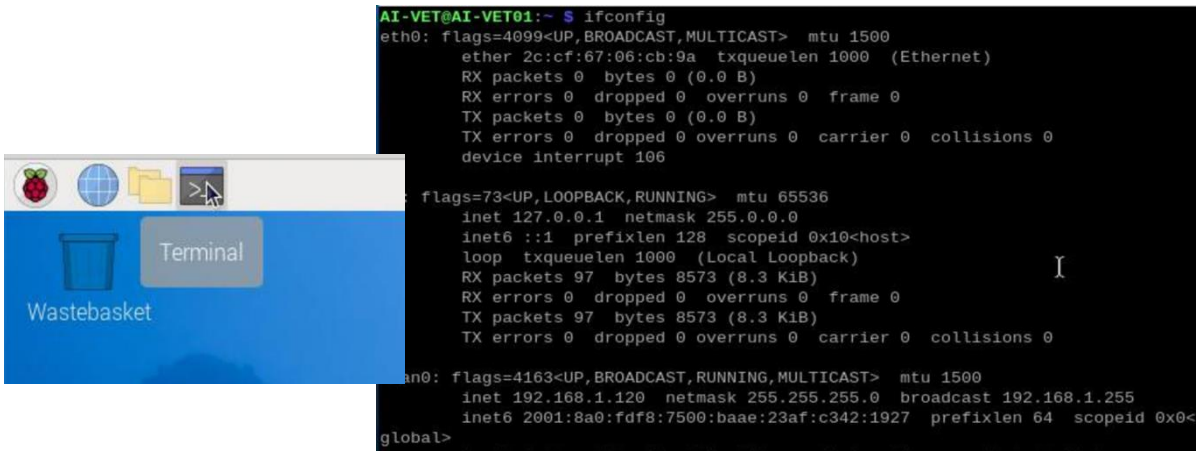


Figure 32- Terminal - ifconfig

As you can see, in this case, the IP address of the Raspberry PI5 is 192.168.1.120. Save your IP address.
Now, go back to your PC and launch TigerVNC. On VNC Server insert the IP Address of your Raspberry PI5:

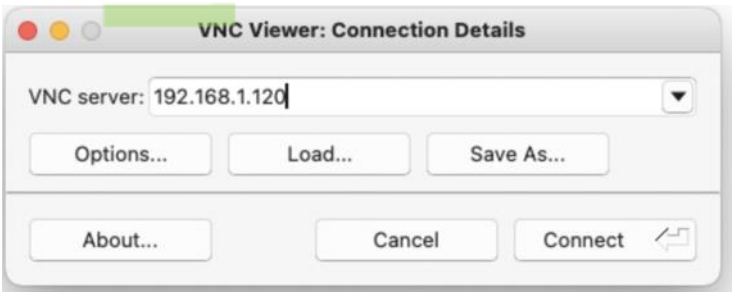


Figure 33- TigerVNC connection details

After click in Connect you will be prompted to log into your Raspberry PI5:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

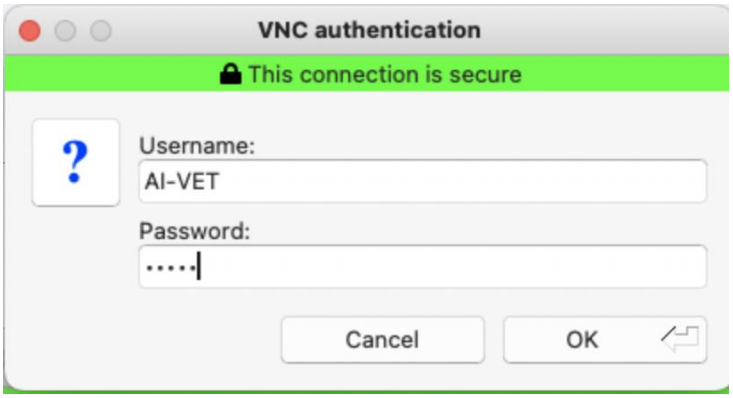


Figure 34- TigerVNC authentication

Now you can control your Raspberry PI5 from your PC!

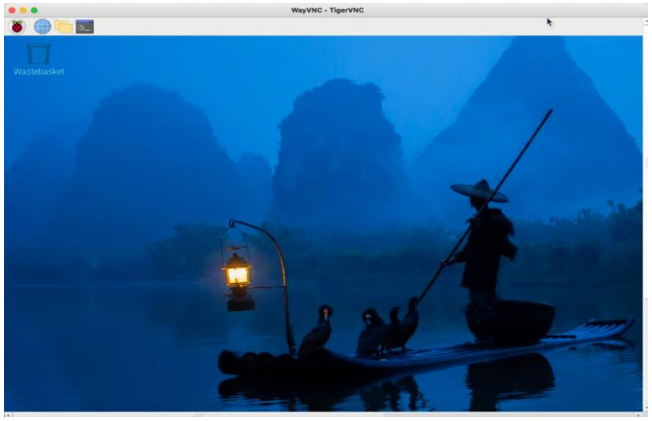
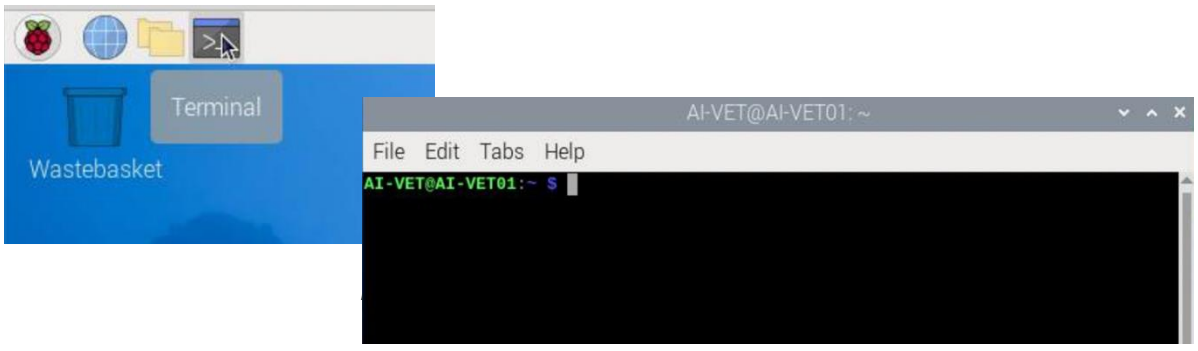


Figure 35- Accessing remotly to Raspberry PI

Step 9: Updating the System

Before starting, we must ensure that your Raspberry Pi and Python versions are up to date.

Open a new terminal window:



After click a new terminal window appear type the following commands:

```
sudo apt update
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

This command refreshes the list of available packages and their versions from all the repositories configured on your system. It doesn't install or upgrade anything, but it updates the local list of available software so the system knows the latest versions and patches that are available for download.

```
sudo apt upgrade
```

This command upgrades the packages on your system to their latest versions based on the updated package list from the apt-get update command. It ensures that all installed packages are brought up to their newest versions, improving performance, fixing bugs, and patching security vulnerabilities.

This process can take a while depending on the number of updates to be installed.



MAKE SURE YOU REBOOT AFTER THESE COMMANDS!!!

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 10: Setting a Virtual Environment

In the Pi 5 Operating System, packages to be installed must be installed in a Python virtual environment using venv. A virtual environment is a container where third-party modules can be installed without affecting the system's Python.

The following command will create an "env" directory for the virtual Python environment:

```
python -m venv env --system-site-packages
```

Now you need to activate the env environment:

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

Your prompt will now display (env), indicating that the system Python is not in use. Instead, the Python version inside the virtual environment is active. Any changes made here will not affect the system Python, and any new modules installed will only apply to the virtual environment:

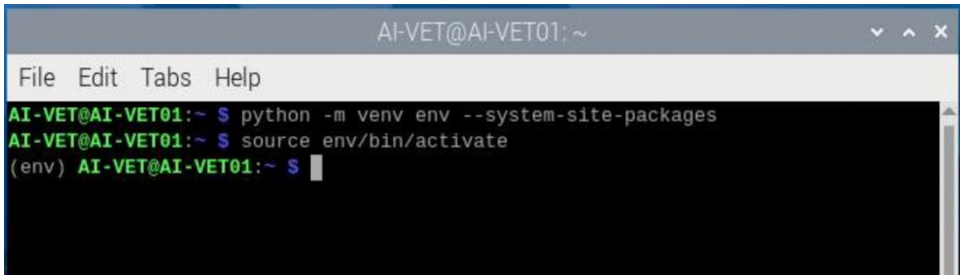


*Figure 37- Terminal prompt with **env** activated*

Step 11: Start Geany IDE

Geany is a lightweight and open-source integrated development environment (IDE) designed for speed and minimal dependencies. It can be used for writing and editing code and supports various programming languages, such as Python, C and C++.

Geany comes pre-installed by default on the Raspberry Pi OS, making it easily accessible for users who want to start coding without the need for additional setup. The following image shows how to start Geany:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

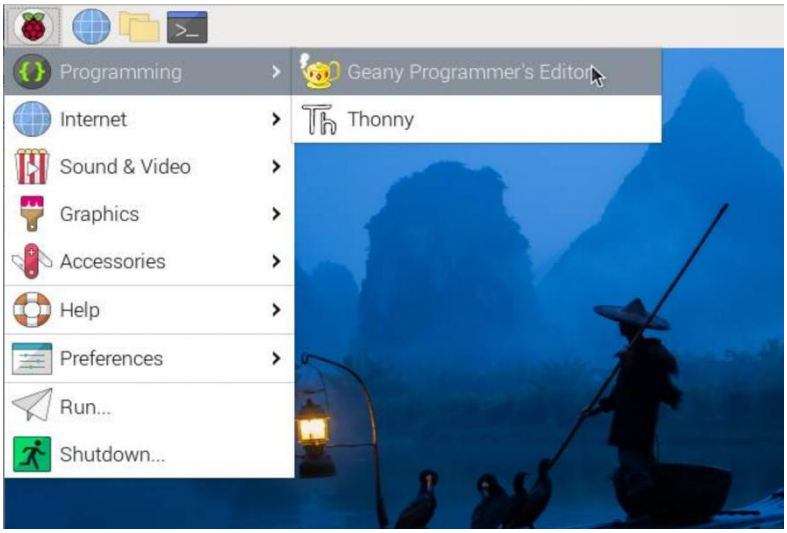


Figure 38- Starting Geany

This is the Geany environment:

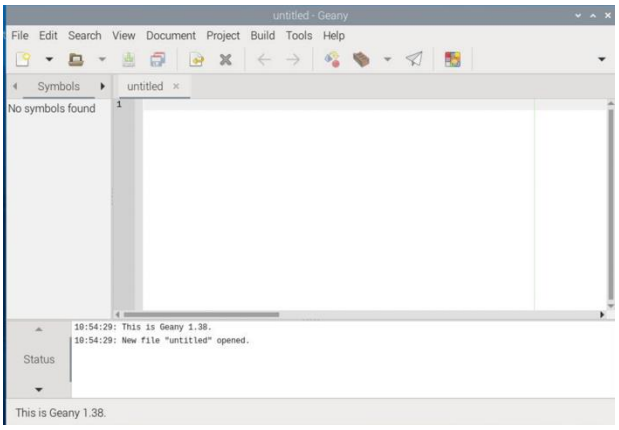


Figure 39- Geany environment

The image shows the Geany (IDE) window open. At the top, the toolbar includes standard options like File, Edit, Search, View, Document, Project, Build, Tools, and Help. Below the toolbar, there are icons for common functions like saving, undo/redo, and file navigation. The main section of the window is a blank, untitled text editing area. At the bottom of the window, the status bar displays messages indicating the version of Geany and confirming that a new file has been opened.

As we created the environment env we need to specify that the Python interpreter within a virtual environment (/env/bin/python) should be used to run Python scripts. This setup is essential when working with a virtual environment because it isolates project dependencies, avoiding potential conflicts with system-installed Python packages.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

First we need to know the path of the Python interpreter in the env virtual environment. To do this, go to the terminal window and write:

```
which python
```

```
(env) AI-VET@AI-VET01:~ $ which python
/home/AI-VET/env/bin/python
(env) AI-VET@AI-VET01:~ $
```

Figure 40- Executing which command

Now, copy the path and go back to Geany:

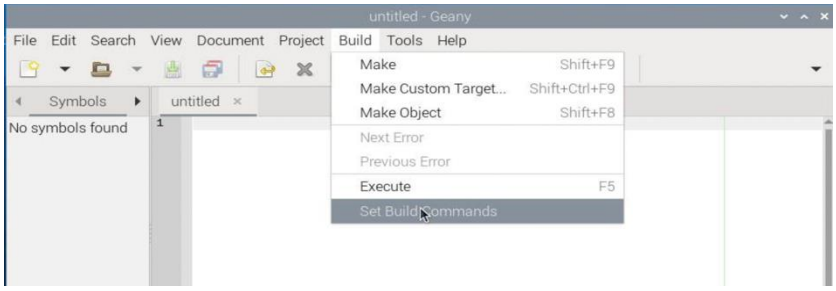


Figure 41- Geany – access to Builtin Commands

And fill the information on the "Execute commands", as shown on the following image:

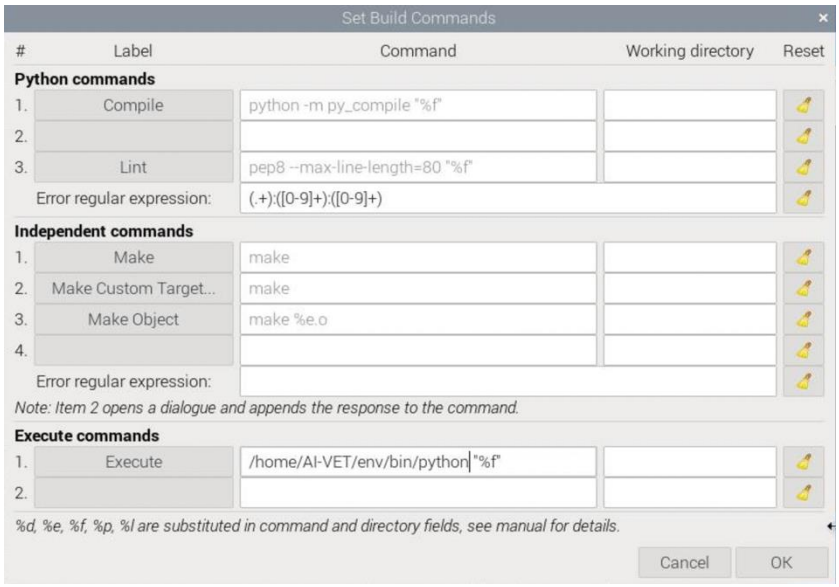


Figure 42- Geany – Set Build Commands

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Note that the %f placeholder refers to the current file in Geany. When you click "Execute" after setting this command, the active Python file will be executed using the specified Python interpreter.



These settings may be lost after a shutdown/reboot. Please always check and reconfigure if necessary.

Finally, we are ready to write the first program in Python!

Write the following Python code:

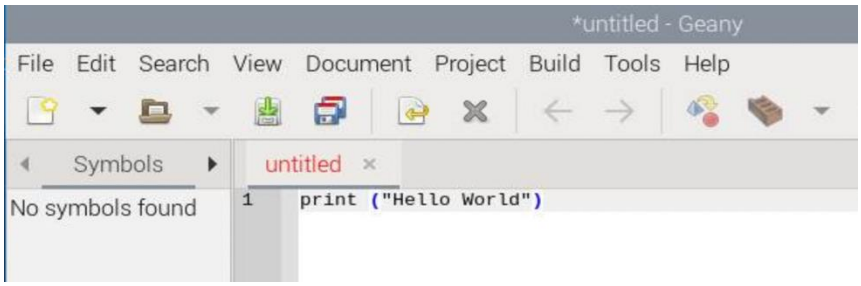


Figure 43- Geany – Hello Word code

Now, make sure you save the file with the extension “py”:

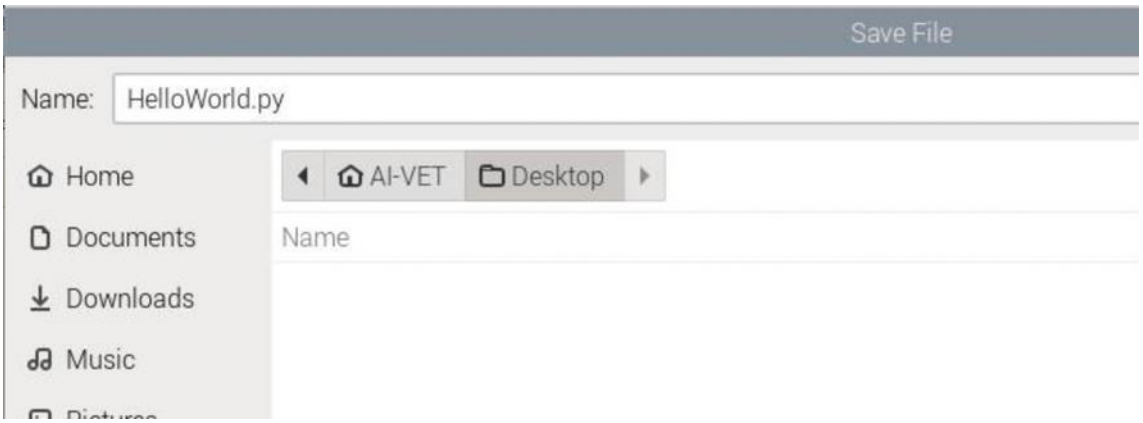


Figure 44- Geany – Saving a file in python format

After this step you can see the python syntax highlight and the program is ready to be executed:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

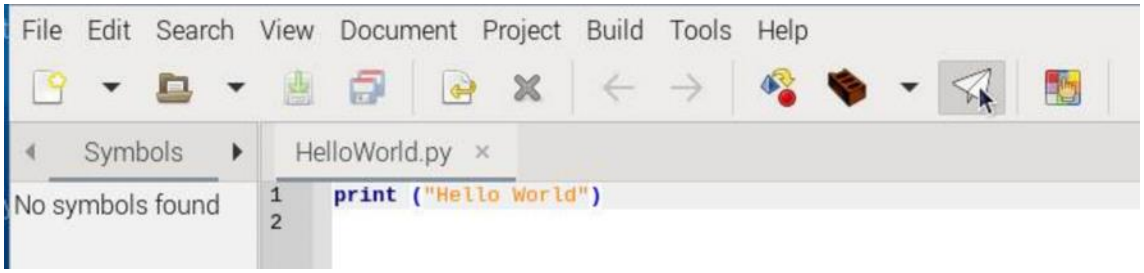


Figure 45- Geany – syntax highlight

And the final result:



Figure 46- Geany – Hello World program

11. Evaluation Criteria:

- **Project Completion:** Raspberry Pi OS is installed, configured, and accessible remotely.
- **System Preparedness:** Required interfaces (SPI, I2C, VNC, etc.) are enabled.
- **Environment Setup:** Virtual environment is active and linked to Geany.
- **Functional Test:** "Hello World" script runs successfully using the configured IDE.
- **Participant Feedback:** Reflection on ease of setup, clarity of instructions, and confidence using the Pi.

---

12. Languages Available:

English, Portuguese (translations in progress)

---

13. Contributors:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Fátima Pais, Luís Soares – Project Designers and Facilitators

14. Feedback and Continuous Improvement:

- **Surveys**: At the end of this milestone, students and educators will complete feedback surveys to assess the effectiveness of the project.
- **Project Reviews**: After project completion, a final review meeting will be held to discuss potential improvements for future AI-based projects.
- **Data Analysis**: Feedback data will be analyzed to identify strengths and areas for improvement, informing future projects.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.