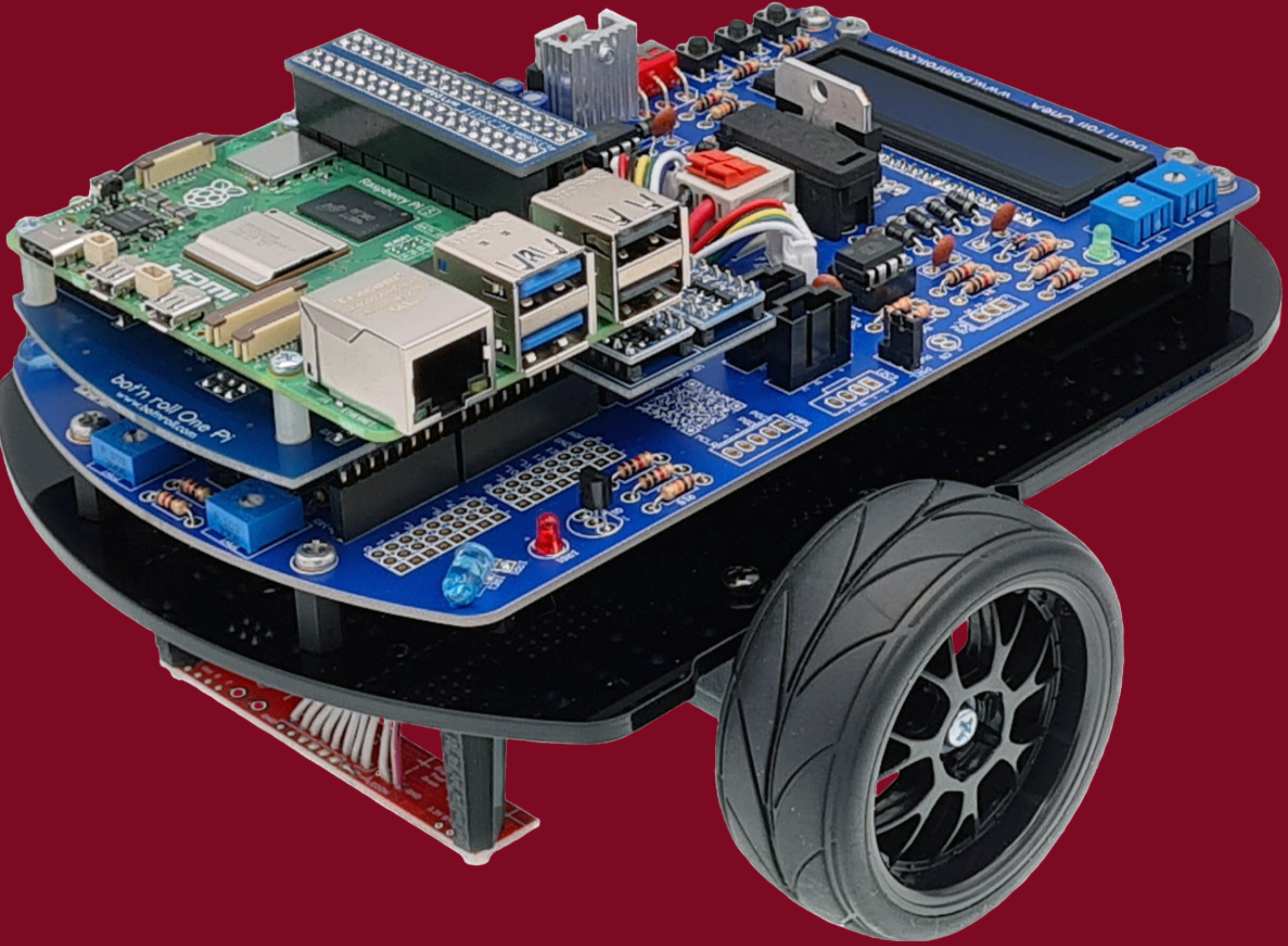
bot'n roll ONE A$^+$

build your own robot



add-on Raspberry Pi®
user manual

Conteúdo

Revisão do Documento: 13 de Janeiro de 2025

1. Introduction

The **Bot'n Roll ONE A+**, with the *Add-On Raspberry Pi*, is programmed using the Python language, and the programming environment can be any IDE that supports *Python3*. Instead of the ATmega328 microcontroller, the **Bot'n Roll ONE A+** now receives commands directly from the **Raspberry Pi 5**.

The robot has a second microcontroller, a **PIC18F45K22**, pre-programmed with software developed by *botnroll.com*. In the **Bot'n Roll ONE A+**, it functions as a *slave* device that executes the command orders from the *master*, the **Raspberry Pi 5**.

The two microcontrollers in the **Bot'n Roll ONE A+** communicate with each other through the *SPI (Serial Peripheral Interface) bus*. The microcontrollers exchange information in a coordinated and well-defined manner. For this, a *data transfer protocol* was developed between the *master* and the *slave*. The *master* uses a command list corresponding to control orders, and each command generates a response from the *slave*. The command list and the way data is transmitted between *master* and *slave* are defined in the "**BnrOneAPlus" library**.

The "**BnrOneAPlus" library** for *Python* allows the user to control the robot in a simple way, as long as the library's commands are used correctly. These commands are listed and explained in this manual.

The **PIC18F45K22** can be programmed in *C language* using the **MPLABX IDE** development environment and the **XC8** compiler from *Microchip*, or other compatible software. However, this should only be done by advanced users, as programming the **PIC18F45K22** to include a new feature also requires updating the "**BnrOneAPlus" library** so that *Python* can use the new functionality.

Contact *botnroll.com* if you would like to see a new feature implemented in your **Bot'n Roll ONE A+**!

1. Programming the Bot'n Roll ONE A+ with the AddOn Raspberry Pi

To program the **Bot'n Roll ONE A+**, you need to have your computer and robot set up with all the necessary tools, namely:

- ANNEX A.1: Add-On Raspberry Pi Installation on Bot'n Roll One A+
- Annex B: Installation of the Linux Image Provided by Bot'n Roll One A+
- Annex C: CONNECTING TO THE RASPBERRY PI ON THE BOT'N ROLL ONE A+

The **Python language** is a necessary tool for programming the **Bot'n Roll ONE A+**. If you're not yet very comfortable with Python, the **library examples** are a great guide to get you started in the world of programming. Also, check out the **RoboParty presentations** on Python programming, and of course, there are thousands of web pages explaining the Python language.

2. Library *BnrOneAPlus* for Python

A **library** is a set of "pre-built" code that you can insert and use in your program. To use the “**BnrOneAPlus”** library, simply include it in your Python code:

from onepi.one import BnrOneAPlus

And create an instance of the **BnrOneAPlus** class:

one = BnrOneAPlus(0, 0)

From this point on, you have access to all the functions in the library, which are preceded by the instance you defined, meaning: *one.library_function()*

A **library** is usually created for data or hardware manipulation and can contain one or multiple `.py` files.

The “**BnrOneAPlus”** library was specifically designed for handling the hardware associated with the **PIC18F45K22**, allowing the **Raspberry Pi** to interact with it via the **SPI communication bus**. The **Raspberry Pi** has access to all hardware and functionalities defined within the library and the **PIC18F45K22** software.

The “**BnrOneAPlus”** library and the **PIC18F45K22 software** were developed to work together, meaning any modification in one requires adjustments in the other.

The Python library is available online at:

- **https://github.com/botnroll/bnronepi**

Its respective documentation and description of the available functions can be found at:

- **https://botnroll.github.io/bnronepi.**

3. Python Programming Language

Python was developed in the early 1990s by Guido van Rossum at the Centrum Wiskunde & Informatica (CWI) in the Netherlands. It was created with the goal of being a simple, readable, and powerful language, allowing programmers to focus on problem-solving rather than complex implementation details. Since its release, Python has been continuously improved and updated, making it extremely versatile and widely used in various fields such as web development, automation, data science, artificial intelligence, and scientific computing. The first official version, **Python 1.0**, was released in 1991. In 2000, **Python 2.0** was introduced, bringing several important improvements but also some incompatibilities. In 2008, **Python 3.0** was released, a redesigned version that fixed many limitations of the original language, though it was not compatible with previous versions. Since then, Python has continued to evolve, with regular updates that make the language even more powerful and suited to modern needs. All these updates and revisions aim to make Python a **versatile tool for developing applications on personal computers**, where processing power and memory are not major limitations. Thus, to successfully program your device, you only need to understand some **basic rules of Python** and how to use the **specific modules available for the hardware**. All Python programs follow a **logical structure** based on **functions and blocks of code**, which can be defined as needed. In **interactive or continuous programs**, such as those used to control devices, it is common to define a function to initialize variables, configure devices, or set the system’s initial state. This function can be named, for example, **setup()**, and it will only run once at the beginning of the program. Here, you can configure **input and output pins, initialize communications like SPI, Serial, or I2C**, and set other necessary parameters. After initialization, the program enters a **main loop**, which can be implemented as:

```
while True:
    # Main program logic
```

This loop runs indefinitely, **continuously repeating** the instructions inside it. This is where you define the **core behavior** of your program, updating sensors, sending commands, and implementing the **logic or "intelligence" of your robot!** Python programming itself is not explained in detail in this manual. Instead, we refer you to the **examples in the “BnrOneAPlus” library** and general **Python documentation**. All the code is properly commented, and you will need to **experiment and test** to understand how it works. However, here are some **tips** we have learned over time at **botnroll.com**:

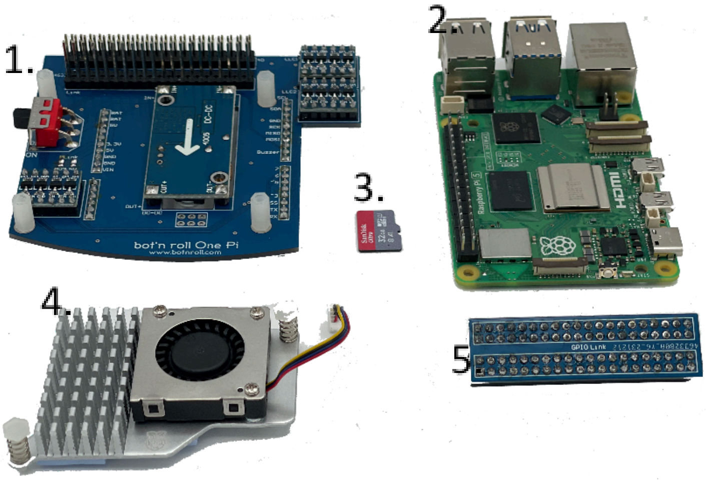
- **Create new programs** starting from the basic examples. Try combining **3 or 4 functionalities** of the robot in the same program using the basic examples as a foundation!
- **A program rarely works on the first try!** Don't get discouraged—analyze the problem and fix it!
- **Insert code gradually** and test it frequently to ensure everything works as expected.
- **You will spend more time testing a program than writing it!** A well-functioning program requires thorough debugging.
- **Use debugging tools** such as the **LED, the serial monitor, or the LCD** to print variable values and check if the program reaches specific parts of the code.
- **Programming is like learning a new sport**—at first, it’s tough because you lack experience, don’t know all the rules, and might feel lost. But with **practice, learning, and improvement**, you’ll soon be part of the **main team!**

The **Bot'n Roll ONE A+** allows interaction with a wide range of hardware. In addition to all the components you can add to your **Bot’n Roll ONE A+**, there are also numerous **shields** that can be attached to your **Raspberry Pi**, as well as a vast number of **Python libraries** available. By combining these resources, the possibilities for expansion are practically **limitless**—your imagination is the only limit!

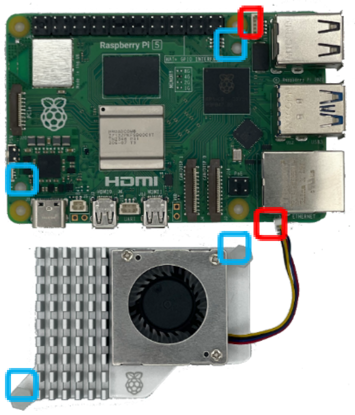
ANNEX A.1: Add-On Raspberry Pi Installation on Bot’n Roll One A+

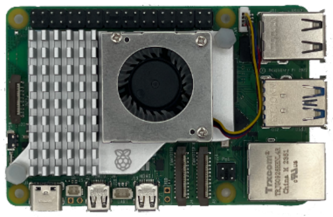
In the **Add-On Raspberry Pi** for the **Bot’n Roll ONE A+**, you will find the following components.

1. Shield Bot’n Roll One Pi
2. Rasberry Pi 5
3. MicroSD card 64GB
4. Fan and heatsink for Raspberry Pi 5
5. GPIOLink for Shield

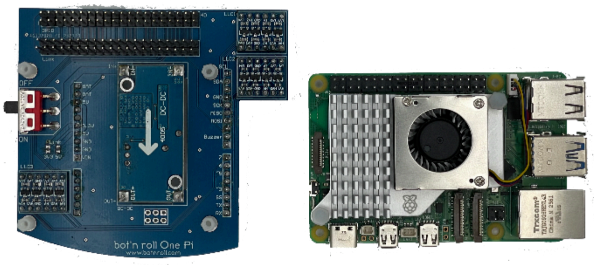


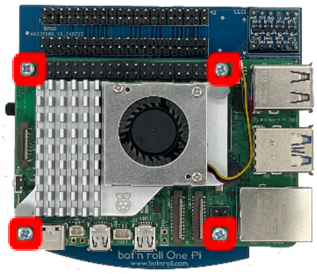
Install the **fan** on your **Raspberry Pi 5** by aligning both **spring clips** with the designated holes on the Raspberry Pi (blue boxes). Apply pressure to the clips until they securely pass through the board. **Do not forget to remove the protective plastic** from the heatsink. Then, connect the **fan connector** to the designated port, as indicated in the image (red box).



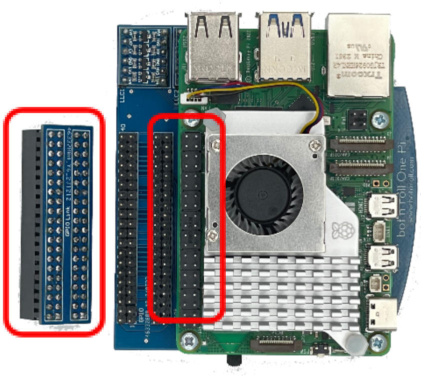


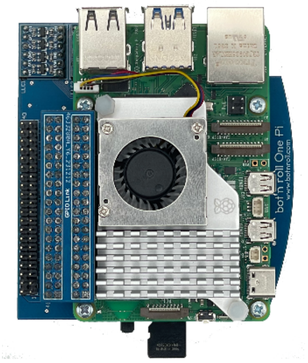
Install the **Raspberry Pi** onto the **Shield** by aligning the **four mounting holes** of the Raspberry Pi with the **spacers** on the Shield. Then, secure it by tightening the **four M2.5x5 screws**.



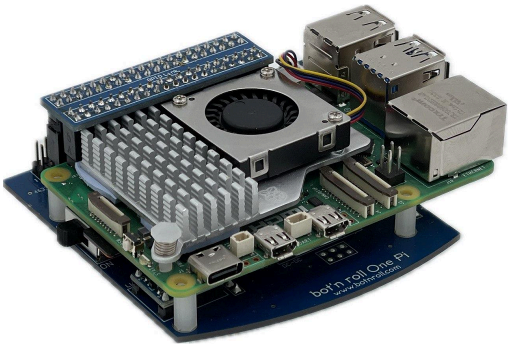


After properly mounting the **Raspberry Pi** onto the **Shield**, you should also install the **GPIOLink**. Once the image is written to the **microSD card**, you can insert it into the **Raspberry Pi** as well.

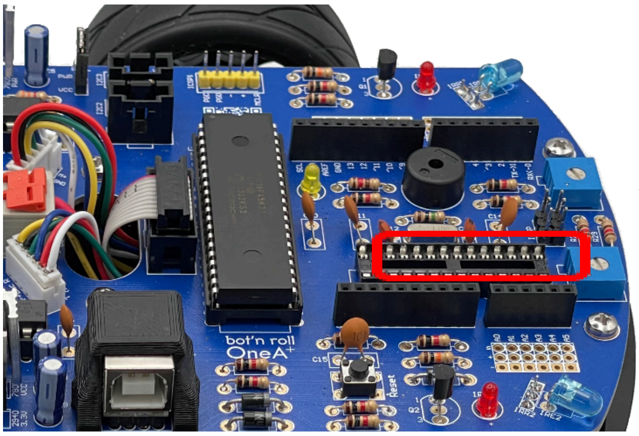




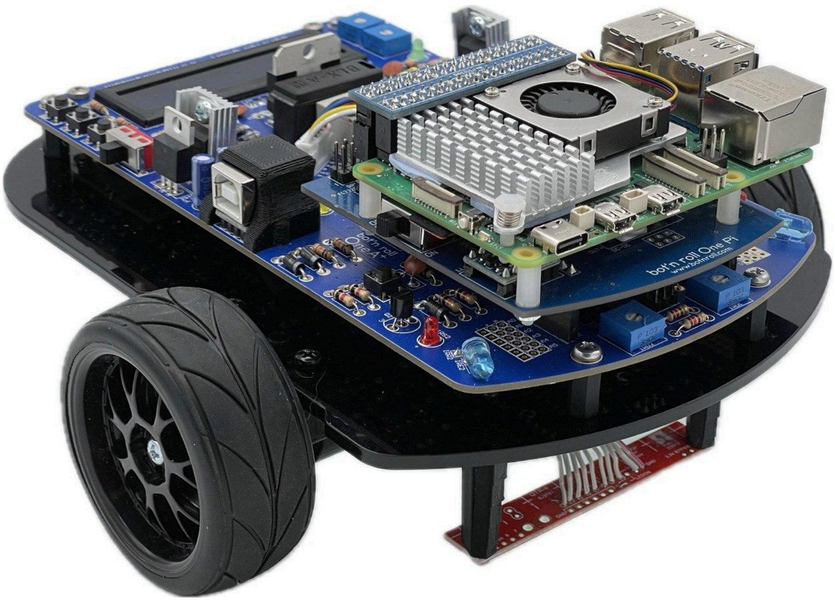
At the end of the **Shield** assembly, you should have something like the **Figure**.



The **Add-On Raspberry Pi** functions as an alternative to the **Arduino**, and as such, the first step is to remove the **ATMega328P**. Carefully remove the **ATMega328P** so that the **Bot'n Roll ONE A+** looks like it does in the **Figure**.



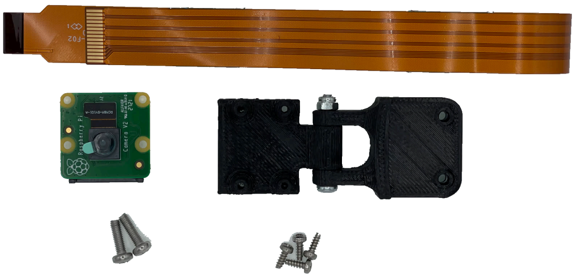
Next, place the **Add-On Raspberry Pi shield** onto the **Bot'n Roll ONE A+,** as shown in the image. The shield's pins should align with the corresponding **pinout** on the **Bot'n Roll**. Ensure that all the pins on the bottom of the shield are properly inserted into their respective slots.



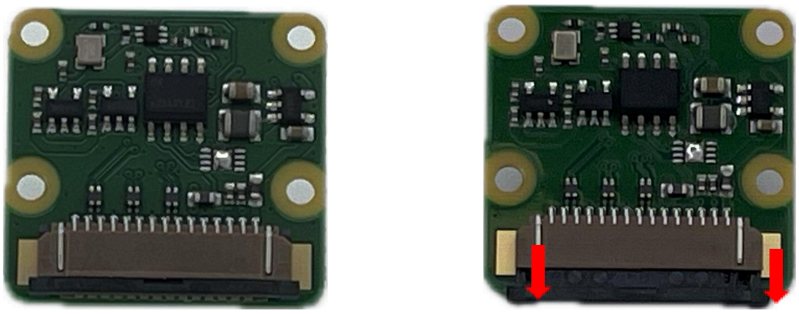
Annex A.2: Assembly of AddOn Vision for Raspberry Pi no Bot'n Roll One A +.

The **Vision Add-On** for the **Raspberry Pi** on the **Bot'n Roll ONE A+** includes the following components:

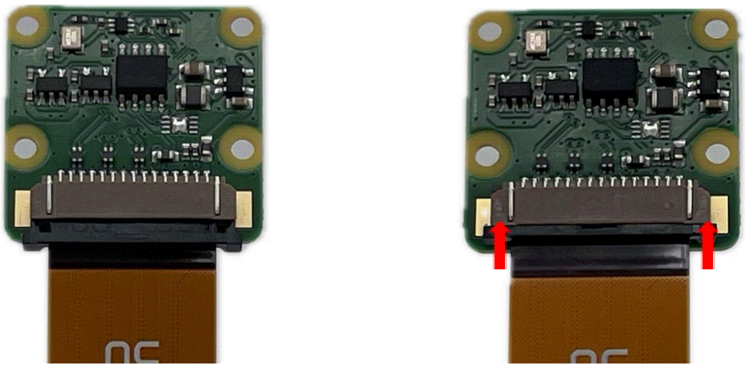
- A Camera for Raspberry Pi
- A Cable to connect the Camera to the Raspberry Pi 5
- A Mount to attach it to the Bot'n Roll ONE A+
- 4x M2.5 screws
- 2x M3 screws



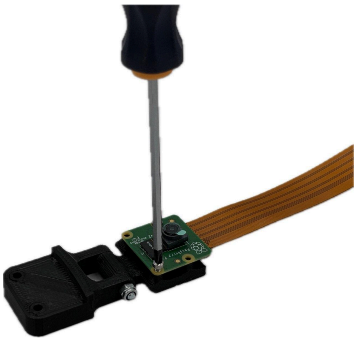
The assembly of the **Vision Add-On** starts with the **Camera** itself. The cable included in the Add-On is used to connect the camera to the **Raspberry Pi**. To connect the camera, you must **open the black latch**, as shown in the image.

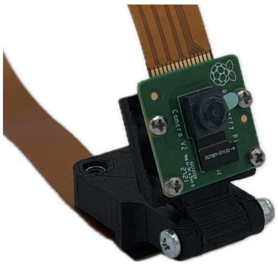


Insert the cable with the **same orientation** as shown in the following image and **close the camera's black latch** again.

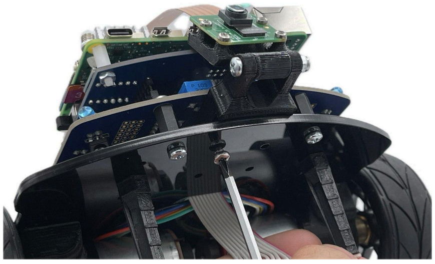


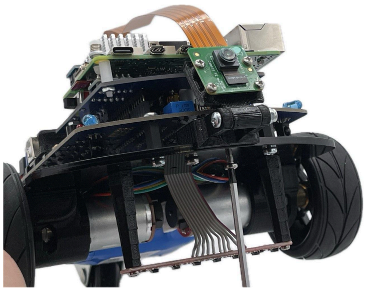
Using the **4x M2.5 screws** provided, secure the **camera to its respective mount**. After this process, the **Vision Add-On will be ready to be installed** on the Bot'n Roll ONE A+.



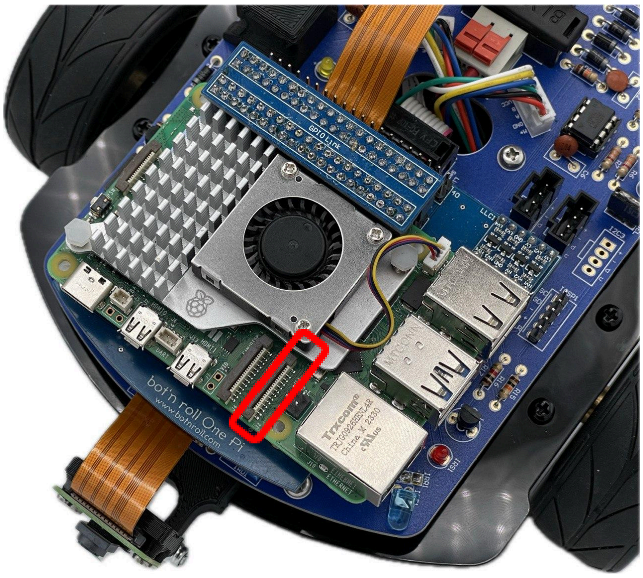


The assembly is done using the **two M3 screws**, fitting the **mount between the PCB and the acrylic plate**, as shown in the image, and tightening the screws with a **Phillips screwdriver**.

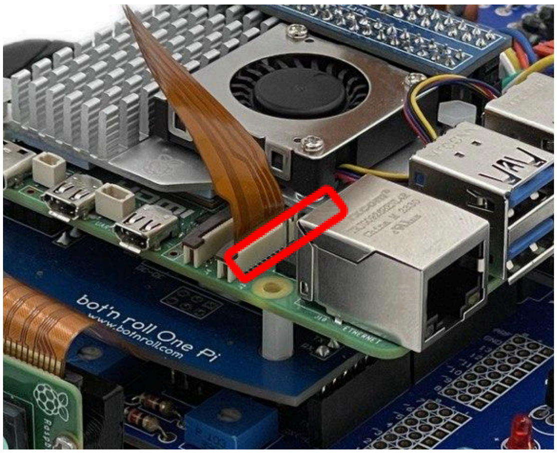




After securing the camera, you must also connect the cable to the **designated port on the Raspberry Pi 5**. The Raspberry Pi 5 supports **two cameras/screens**, and its connector is the same as the one on the camera. Therefore, you need to **lift the same locking tabs** to ensure the correct connection of the camera.

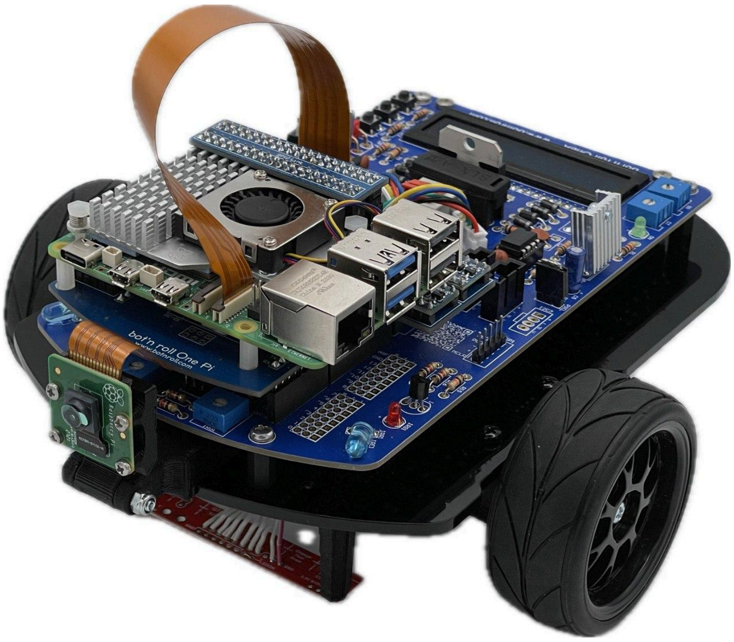


Lift the locking tabs and insert the **camera cable** with the **orientation shown in the image**. The **golden pins** on the cable should be **facing the USB ports** of the Raspberry Pi. Then, push the **black locking tab back down** to secure the cable.



In the end, your **Bot'n Roll ONE A+** will look like the **figure shown**, ready to be **programmed** and to **explore the world of Python** and all its functionalities.

To start using the **Raspberry Pi**, you must **configure the Linux image**. The steps to do so can be found in **Annex B: Installation of the Linux Image Provided by Bot'n Roll One A+**.



ANNEX B: INSTALLATION OF THE LINUX IMAGE PROVIDED BY BOT'N ROLL ONE A+

The **Raspberry Pi**, being a **microcomputer**, requires an **operating system**. The most commonly used one is **Raspbian**. In this annex, you will learn **how to install the operating system on the microSD card**, allowing you to **power on and interact with your Raspberry Pi**.

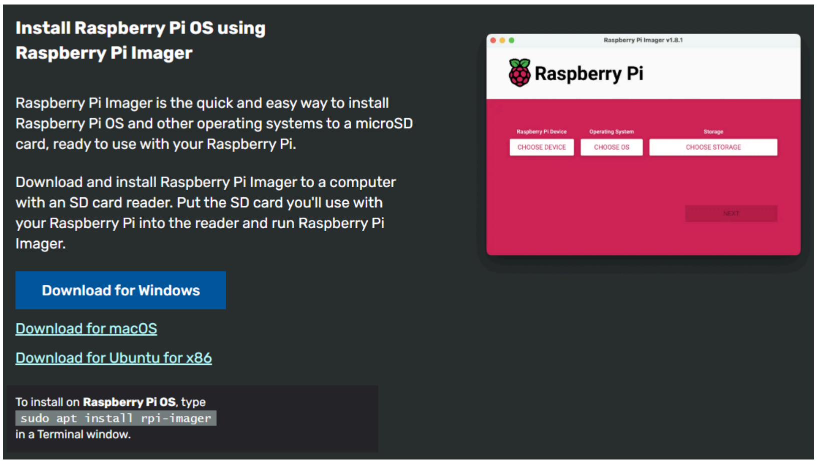
Required Materials:

- A computer with internet access
- A microSD card (provided in the Raspberry Pi AddOn for Bot'n Roll One A+)
- A microSD card reader
- The Raspberry Pi Imager software installed on your computer (see section B.1)

B.1 RASPBERRY PI IMAGER SOFTWARE INSTALLATION

Visit the Raspberry Pi software page: https://www.raspberrypi.com/software/

In the section "Install Raspberry Pi OS using Raspberry Pi Imager", click "Download for Windows". If you are using Linux or MacOS, download the version for your specific operating system. Install Raspberry Pi Imager. Run the downloaded file and follow the on-screen installation instructions.



Install Raspberry Pi OS using Raspberry Pi Imager

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

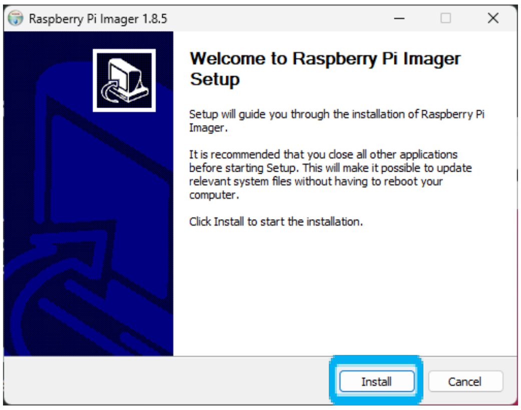
Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

To install on **Raspberry Pi OS**, type

```
sudo apt install rpi-imager
```

in a Terminal window.

Once the download is complete, open the file and proceed with the installation. Follow the on-screen instructions to complete the process.

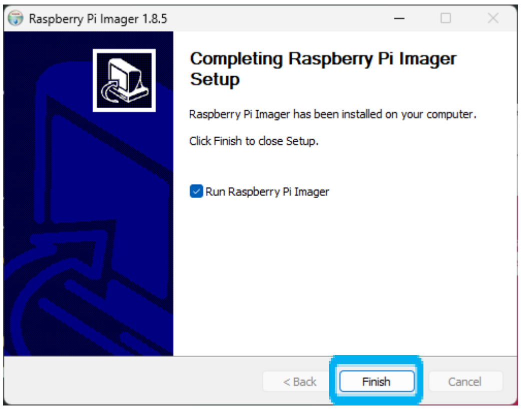


Welcome to Raspberry Pi Imager Setup

Setup will guide you through the installation of Raspberry Pi Imager.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Install to start the installation.



Completing Raspberry Pi Imager Setup

Raspberry Pi Imager has been installed on your computer.

Click Finish to close Setup.

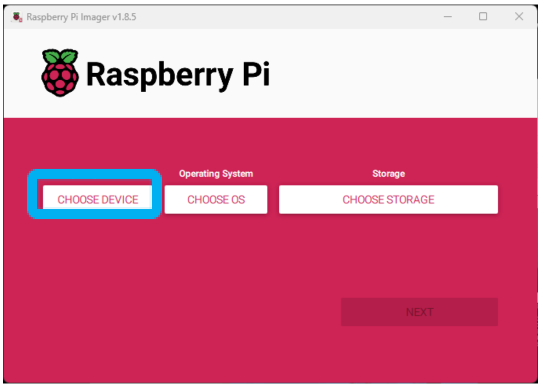
☑ Run Raspberry Pi Imager

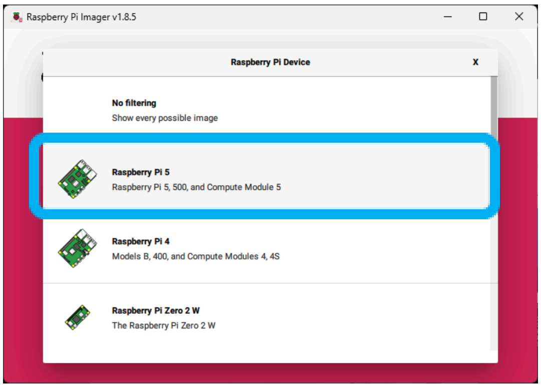
B.2 Raspberry Pi Imager Installation on the MicroSD Card

On the support page for the Bot’n Roll One A+, you will find a link to download a Raspbian image that already includes the following:

- Raspbian Operating System
- Python3 properly installed
- BnrOneAPlus library for Python already installed
- SPI communication already enabled
- Camera support already set up
- Example code for the Bot’n Roll One A+ on the desktop
- Python programming environments already installed (Thonny or VSCode)

1º Open the Raspberry Pi Imager software**,** click on **“Choose Device”** and then select Raspberry Pi.





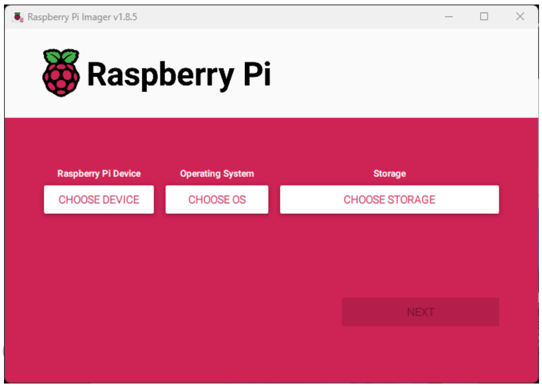
No filtering
Show every possible image

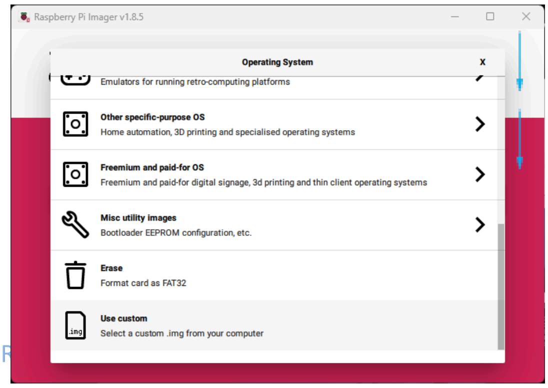
Raspberry Pi 5
Raspberry Pi 5, 500, and Compute Module 5

Raspberry Pi 4
Models B, 400, and Compute Modules 4, 4S

Raspberry Pi Zero 2 W
The Raspberry Pi Zero 2 W

2º Next, click on "Choose OS," then scroll down to the last option and click on "Use Custom."





Emulators for running retro-computing platforms

Other specific-purpose OS
Home automation, 3D printing and specialised operating systems

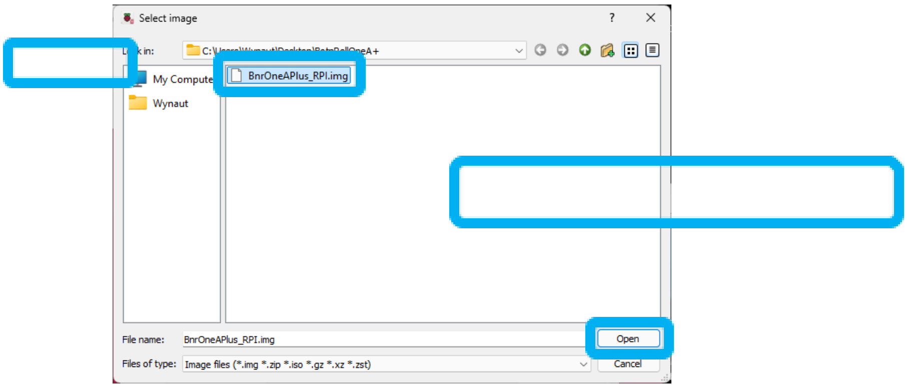
Freemium and paid-for OS
Freemium and paid-for digital signage, 3d printing and thin client operating systems

Misc utility images
Bootloader EEPROM configuration, etc.

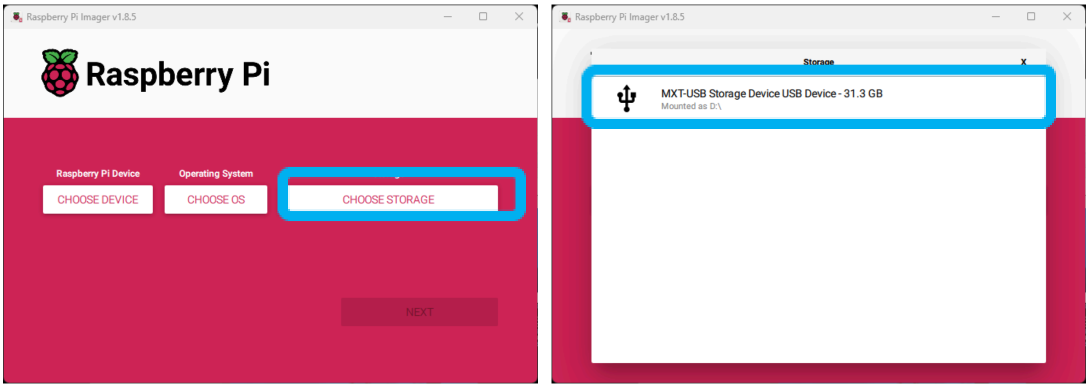
Erase
Format card as FAT32

Use custom
Select a custom .img from your computer

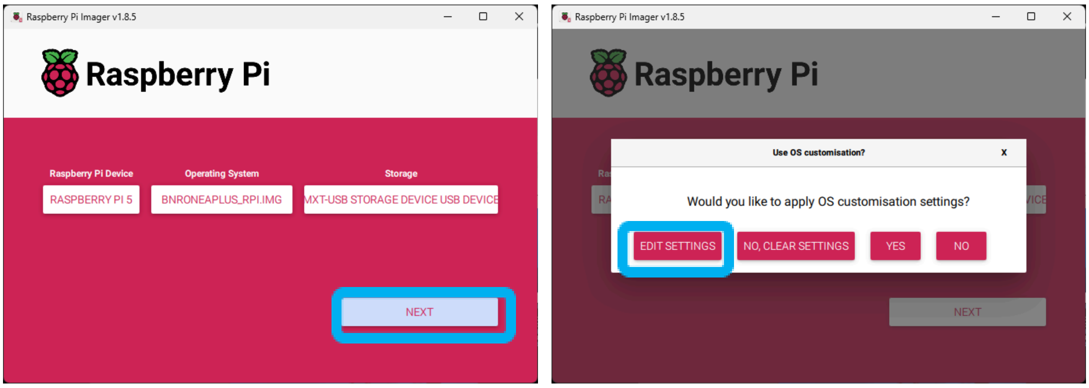
3º A window will appear to select the Linux image. Choose the **“BnrOneAPlus_RPI.img”** file that you previously downloaded and click **"Open"**.


Select image
?
Look in: C:\Users\Wynaut\Desktop\BotnRollOneA+
My Compute
Wynaut
BnrOneAPlus_RPI.img
File name: BnrOneAPlus_RPI.img
Files of type: Image files (*.img *.zip *.iso *.gz *.xz *.zst)
Open
Cancel

4º With the device and image already selected, insert the provided microSD card into your computer and click **"Choose Storage"**. Then, select the microSD card from the list of available devices.



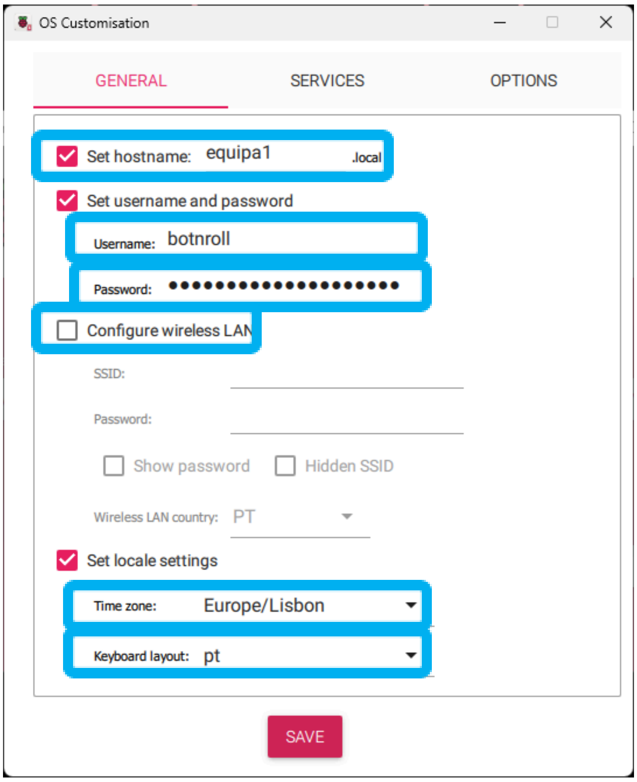
5º Next, click **"Next"**. A menu will appear, and you should click **"Edit Settings"**. Here, you will define the robot's name, username, and password, as well as configure your Wi-Fi network settings.



During the event, you will not be able to connect to the available Wi-Fi network by following this tutorial. It is permitted to use a wireless connection during the event. An explanation of how to connect your Raspberry Pi with a network cable during RoboParty is provided in ANNEX C.

6º In the menu that appears, you should define the following parameters.

- **Set Hostname** -> Robot name: This will be the name used later to connect to the Raspberry Pi.
- **Set username and password:**
  - **Username** -> This will be the login name for accessing the operating system.
  - **Password** -> user password
- **Configure Wireless LAN** ->
  - During the RoboParty, this setting cannot be configured and should be disabled.
- **Set Locale Settings:**
  - **Time Zone** -> You should select the region where you are located.
  - **Keyboard Layout** -> Select the keyboard layout you will use (e.g., PT for Portuguese).



☑ Set hostname: equipa1 .local

☑ Set username and password

Username: botnroll

Password: •••••••••••••••••••

☐ Configure wireless LAN

SSID:

Password:

☐ Show password ☐ Hidden SSID

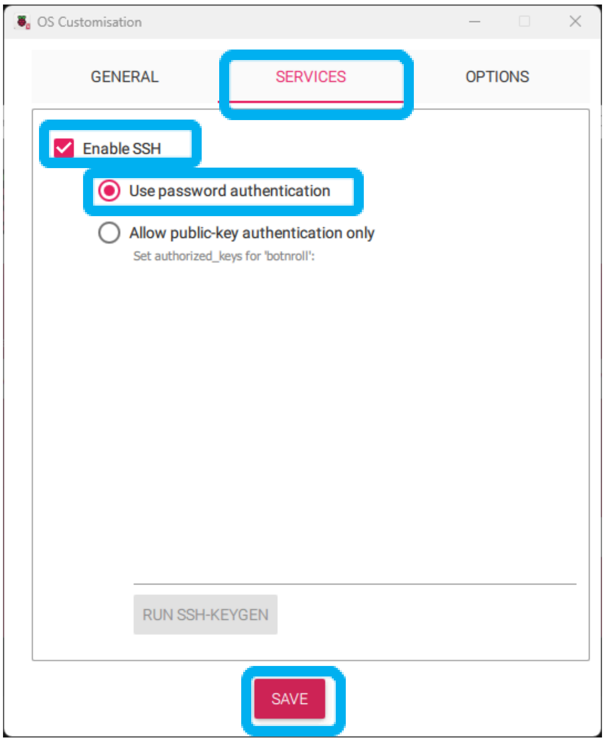
Wireless LAN country: PT

☑ Set locale settings

Time zone: Europe/Lisbon

Keyboard layout: pt

7º In the same window, under the **"Services"** tab, you must enable **SSH** by clicking on **"Enable SSH".** You should also select the option **"Use password authentication".** Finally, click on **"Save"**.



OS Customisation

GENERAL | SERVICES | OPTIONS

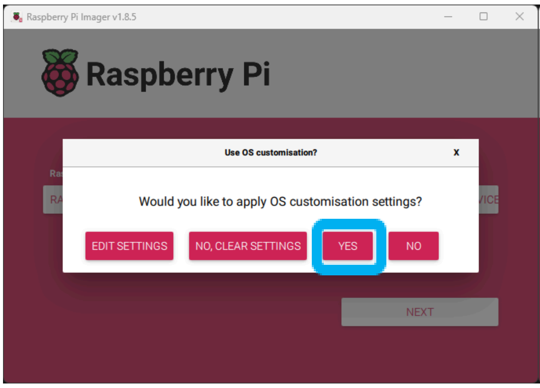
☑ Enable SSH

● Use password authentication

○ Allow public-key authentication only

Set authorized_keys for 'botnroll':

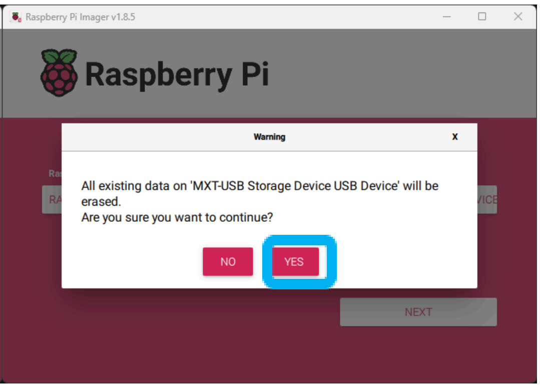
8º After completing the image configuration, click **"Yes"** and then click **"Yes"** again.



Raspberry Pi Imager v1.8.5

Use OS customisation?

Would you like to apply OS customisation settings?

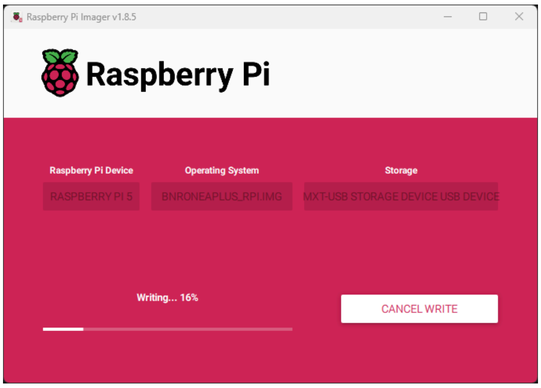


Raspberry Pi Imager v1.8.5

Warning

All existing data on 'MXT-USB Storage Device USB Device' will be erased.
Are you sure you want to continue?

The software will write the image to the microSD card and then verify it.



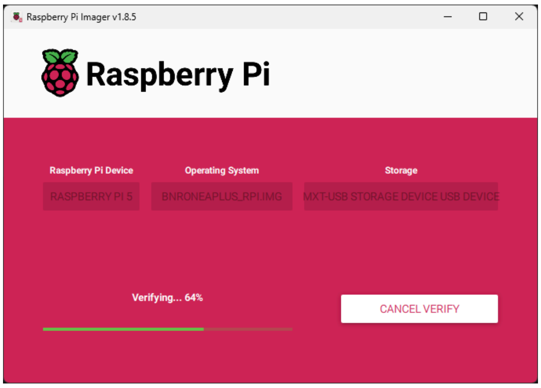
Raspberry Pi Imager v1.8.5

Raspberry Pi Device: RASPBERRY PI 5

Operating System: BNRONEAPLUS_RPI.IMG

Storage: MXT-USB STORAGE DEVICE USB DEVICE

Writing... 16%



Raspberry Pi Imager v1.8.5

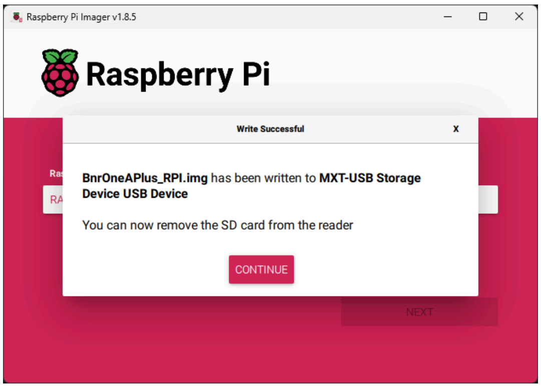
Raspberry Pi Device: RASPBERRY PI 5

Operating System: BNRONEAPLUS_RPI.IMG

Storage: MXT-USB STORAGE DEVICE USB DEVICE

Verifying... 64%

9º At the end of the verification, you can close the program, remove the microSD card from your computer, and insert it into the Raspberry Pi 5.



BnrOneAPlus_RPI.img has been written to **MXT-USB Storage Device USB Device**

You can now remove the SD card from the reader

After inserting the microSD card into your Raspberry Pi, your Bot'n Roll ONE A+ is ready to be programmed in Python. Check **Annex C: CONNECTING TO THE RASPBERRY PI ON THE BOT'N ROLL ONE A+** for further instructions.

Annex C: Connecting To The Raspberry Pi On The Bot'n Roll ONE A+

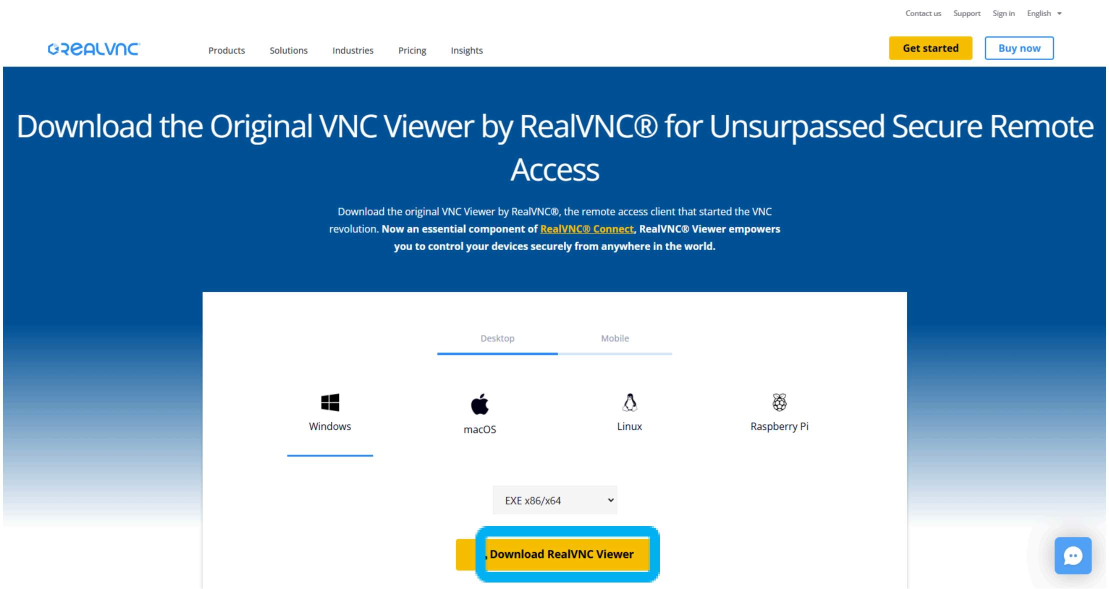
The **Raspberry Pi 5** is a microcomputer, and there are multiple ways to program it, each with its own advantages and disadvantages. The first option is to connect a **keyboard and monitor** to your Raspberry Pi and use it like a regular computer. However, this setup can become impractical when testing the robot. This **Annex C** will introduce you to some alternative methods.

Included in your kit, you will find an **Ethernet cable**. You can use this cable to interact with and program your Raspberry Pi. This will be the primary connection method to access your **Raspberry Pi 5** and program your **Bot'n Roll ONE A+**.

The **Linux image** pre-installed on your Raspberry Pi already has the necessary network configurations set up. If you install an image that was **not provided by botnroll.com**, refer to Annex D, which explains all the necessary steps to configure your **Raspberry Pi** correctly.

C.1 VNC Viewer Installation

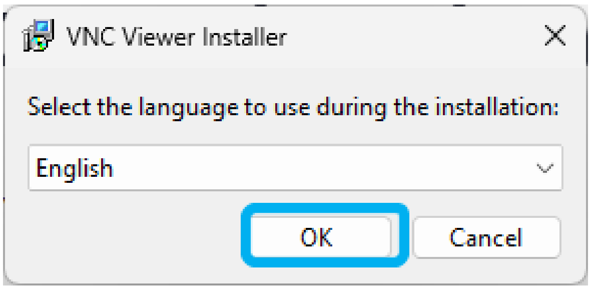
1º Open the link "https://www.realvnc.com/en/connect/download/viewer/" and download the version specified for your personal computer.



Download the Original VNC Viewer by RealVNC® for Unsurpassed Secure Remote Access

Download the original VNC Viewer by RealVNC®, the remote access client that started the VNC revolution. Now an essential component of RealVNC® Connect, RealVNC® Viewer empowers you to control your devices securely from anywhere in the world.

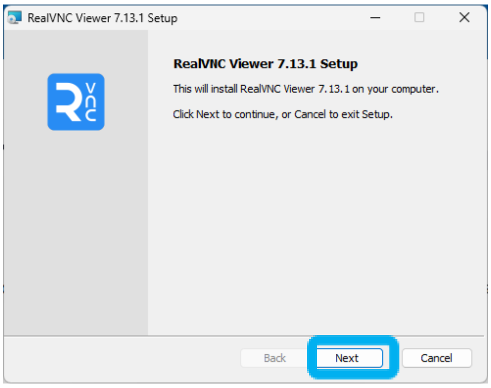
2º Open the executable you downloaded and proceed with the installation. Choose your preferred language and click "**OK**".



Select the language to use during the installation:

English

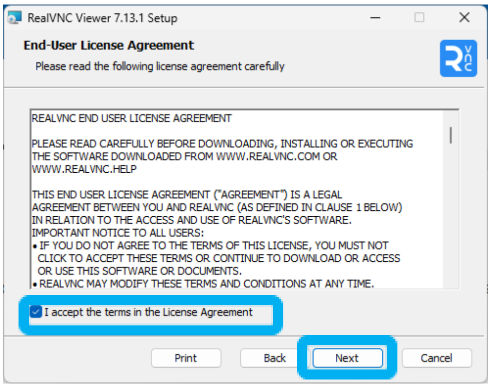
3º Next, click on "**Next**", then select "**I accept the terms in the License Agreement**", and click "**Next**" again.


RealVNC Viewer 7.13.1 Setup

RealVNC Viewer 7.13.1 Setup

This will install RealVNC Viewer 7.13.1 on your computer.

Click Next to continue, or Cancel to exit Setup.


RealVNC Viewer 7.13.1 Setup

End-User License Agreement

Please read the following license agreement carefully

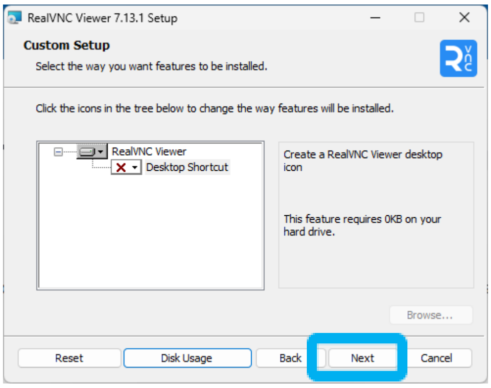
REALVNC END USER LICENSE AGREEMENT

PLEASE READ CAREFULLY BEFORE DOWNLOADING, INSTALLING OR EXECUTING THE SOFTWARE DOWNLOADED FROM WWW.REALVNC.COM OR WWW.REALVNC.HELP

THIS END USER LICENSE AGREEMENT ("AGREEMENT") IS A LEGAL AGREEMENT BETWEEN YOU AND REALVNC (AS DEFINED IN CLAUSE 1 BELOW) IN RELATION TO THE ACCESS AND USE OF REALVNC'S SOFTWARE.
IMPORTANT NOTICE TO ALL USERS:
- IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, YOU MUST NOT CLICK TO ACCEPT THESE TERMS OR CONTINUE TO DOWNLOAD OR ACCESS OR USE THIS SOFTWARE OR DOCUMENTS.
- REALVNC MAY MODIFY THESE TERMS AND CONDITIONS AT ANY TIME.

☑ I accept the terms in the License Agreement

4º Click "**Next**" again, then click "**Install**." You will be prompted for administrator access to proceed with the software installation.


RealVNC Viewer 7.13.1 Setup

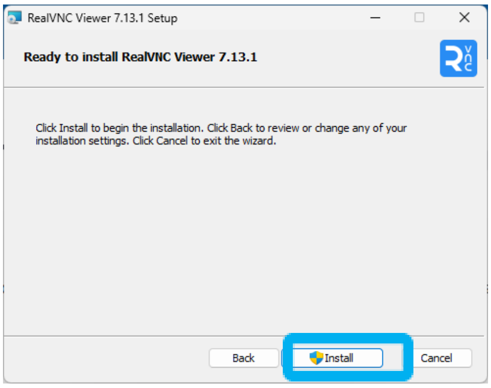
Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

Create a RealVNC Viewer desktop icon

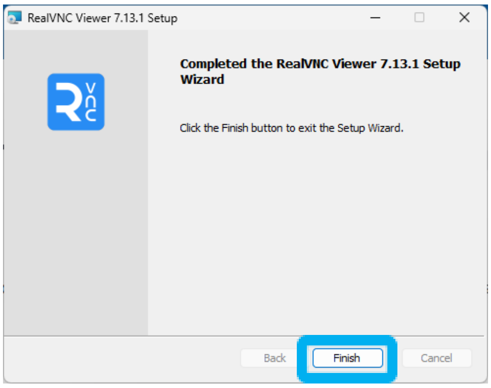
This feature requires 0KB on your hard drive.


RealVNC Viewer 7.13.1 Setup

Ready to install RealVNC Viewer 7.13.1

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

5º At the end of the progress bar, the installation of VNC Viewer will be complete, and you can click "**Finish**."

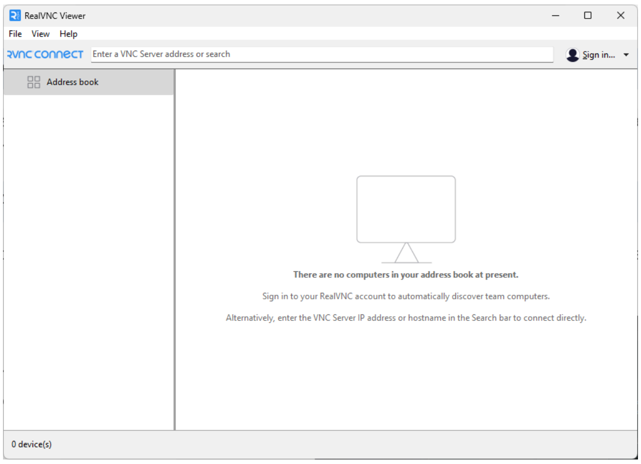

RealVNC Viewer 7.13.1 Setup

Completed the RealVNC Viewer 7.13.1 Setup Wizard

Click the Finish button to exit the Setup Wizard.

C.2 Raspberry Pi Connection with VNC Viewer

After having VNC Viewer properly installed, you can proceed with connecting to your Raspberry Pi. Open the VNC Viewer software, and you will find the following menu.



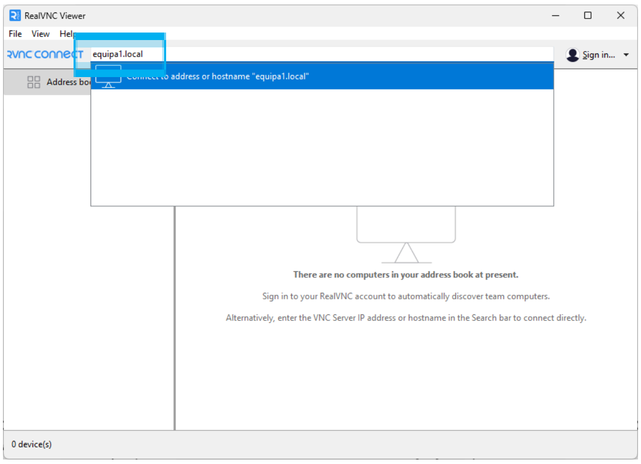
There are no computers in your address book at present.

Sign in to your RealVNC account to automatically discover team computers.

Alternatively, enter the VNC Server IP address or hostname in the Search bar to connect directly.

0 device(s)

Enter the HOSTNAME that you defined during the installation of the Linux image on your Raspberry Pi, followed by **“.local”**. For example, if your hostname was **“equipa1”**, you should type **“equipa1.local”** in the search bar and press enter.



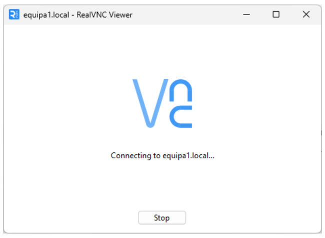
There are no computers in your address book at present.

Sign in to your RealVNC account to automatically discover team computers.

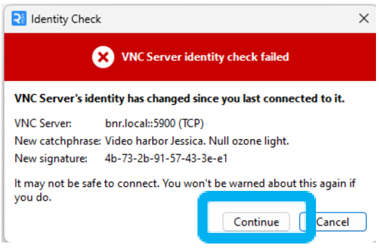
Alternatively, enter the VNC Server IP address or hostname in the Search bar to connect directly.

0 device(s)

When you press enter, a new window will appear to attempt the connection, and another window where you must enter the previously defined **username** and **password**, then click **“OK”**. If a window appears before the login one, confirming the port, click **“continue”**.



Connecting to equipa1.local...

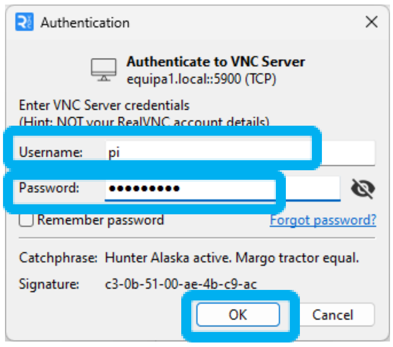


VNC Server identity check failed

VNC Server's identity has changed since you last connected to it.

VNC Server: bnr.local::5900 (TCP)
New catchphrase: Video harbor Jessica. Null ozone light.
New signature: 4b-73-2b-91-57-43-3e-e1

It may not be safe to connect. You won't be warned about this again if you do.



Authenticate to VNC Server
equipa1.local::5900 (TCP)

Enter VNC Server credentials
(Hint: NOT your RealVNC account details)

Username: pi
Password: •••••••••
□ Remember password

Catchphrase: Hunter Alaska active. Margo tractor equal.
Signature: c3-0b-51-00-ae-4b-c9-ac

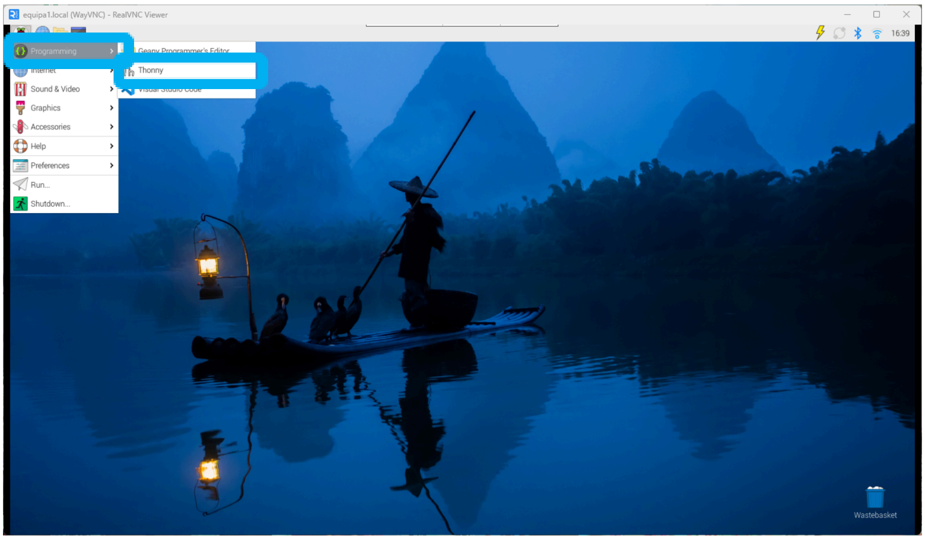
By clicking **"OK"**, the Raspberry Pi desktop environment will appear. It contains a folder called **"OnePi_Examples"** which includes examples for the BnrOneAPlus library, ready to interact with the Bot'n Roll ONE A+.

C.3 Python Programming Environment

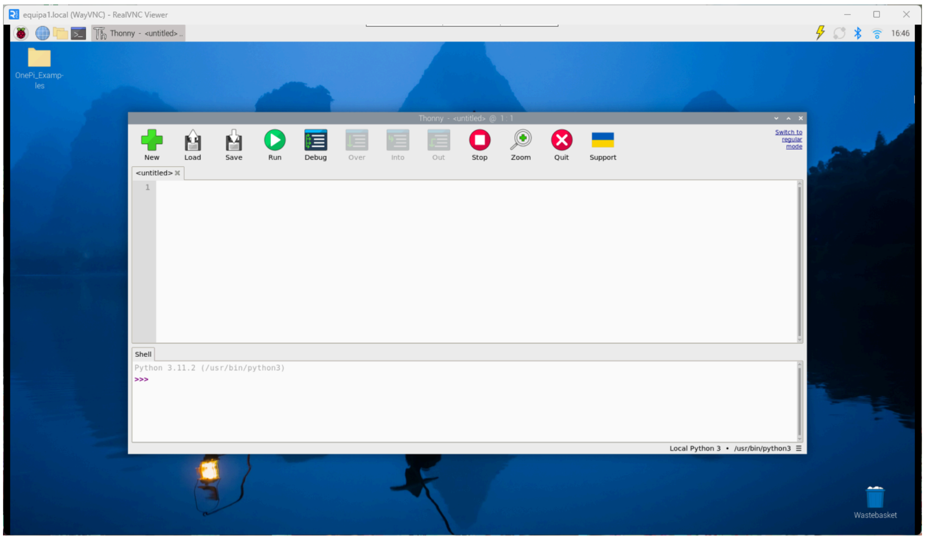
Just like programming the Bot'n Roll ONE A+ in Arduino, programming in Python also requires an interface where the codes that will interact with the robot will be written. Here, two solutions are presented: a basic one using the Thonny IDE, and an intermediate one using Visual Studio Code. Choose one of the options and see how to start programming in Python.

THONNY:

Thonny is a Python programming IDE ideal for beginners. To open it, go to the applications menu, select the "**Programming**" option, and then the "**Thonny**" option.



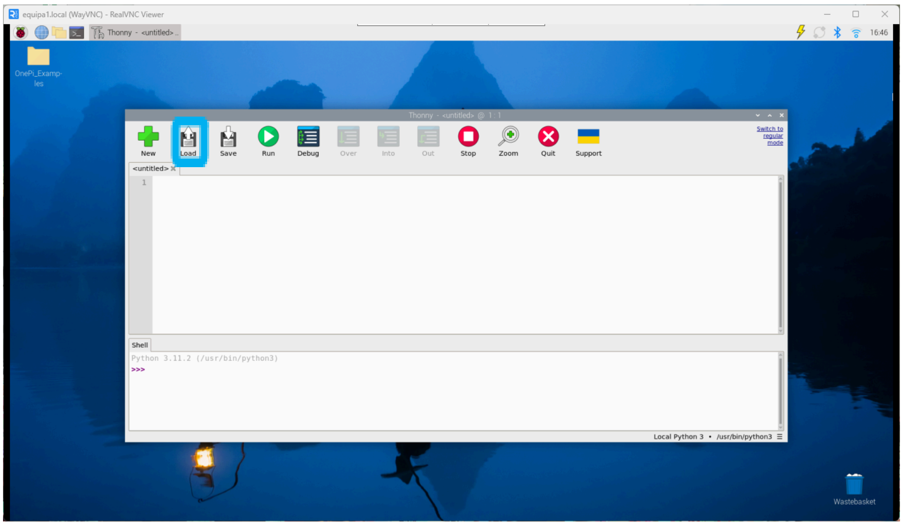
The Thonny software will open, and you will see the following window. Here is a list of the different buttons on the interface and their functionality.



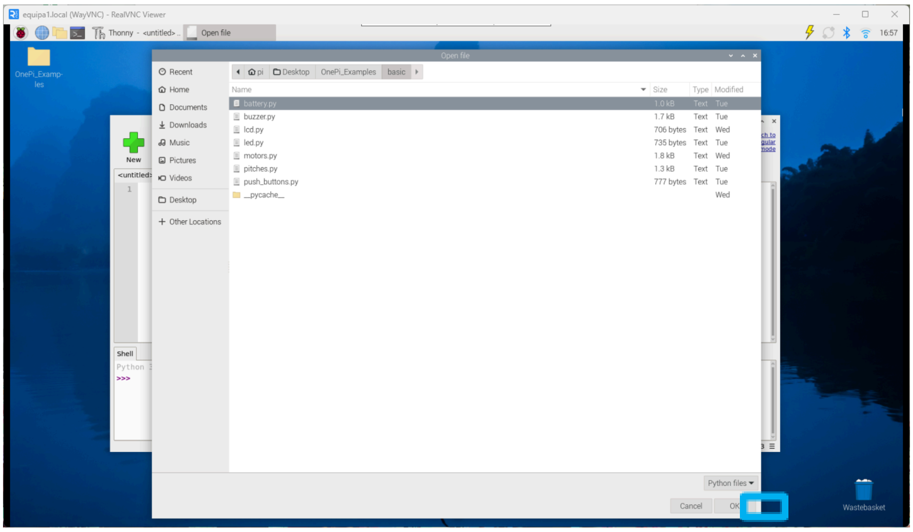
```
Python 3.11.2 (/usr/bin/python3)
>>>
```

- New - Create a new document
- Load - Open a programm
- Save - Save the current programm
- Run - Run the current programm
- Stop - Stop the programm

To get started with the "BnrOneAPlus" library examples for Python, click the "**Load**" button, navigate to the "**OnePi_Examples**" folder located on the desktop, and choose an example.



```
Python 3.11.2 (/usr/bin/python3)
>>>
```

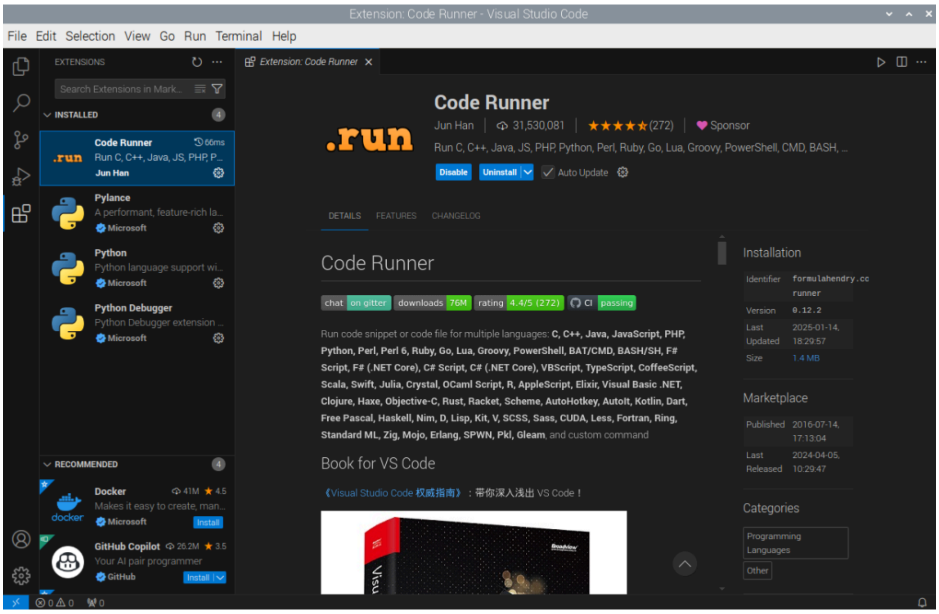


Name	Size	Type	Modified
battery.py	1.0 kB	Text	Tue
buzzer.py	1.7 kB	Text	Tue
lcd.py	706 bytes	Text	Wed
led.py	735 bytes	Text	Tue
motors.py	1.8 kB	Text	Wed
pitches.py	1.3 kB	Text	Tue
push_buttons.py	777 bytes	Text	Tue
__pycache__			Wed

After opening an example of your choice, you only need to click the "**Run**" button.

Visual Studio Code:

For using Python code in Visual Studio Code, it is recommended to use the "**Code Runner**" extension. You can check if it is installed by going to the Extensions menu and searching for its name. If it is not installed, click the "**Install**" button, and afterward, you should encounter an environment like this.



Run C, C++, Java, JS, PHP, Python, Perl, Ruby, Go, Lua, Groovy, PowerShell, CMD, BASH, ...

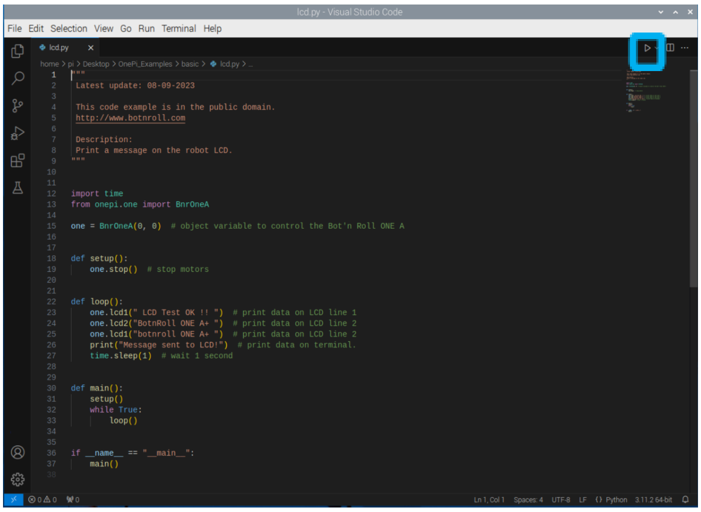
Code Runner

Run code snippet or code file for multiple languages: **C, C++, Java, JavaScript, PHP, Python, Perl, Perl 6, Ruby, Go, Lua, Groovy, PowerShell, BAT/CMD, BASH/SH, F# Script, F# (.NET Core), C# Script, C# (.NET Core), VBScript, TypeScript, CoffeeScript, Scala, Swift, Julia, Crystal, OCaml Script, R, AppleScript, Elixir, Visual Basic .NET, Clojure, Haxe, Objective-C, Rust, Racket, Scheme, AutoHotkey, AutoIt, Kotlin, Dart, Free Pascal, Haskell, Nim, D, Lisp, Kit, V, SCSS, Sass, CUDA, Less, Fortran, Ring, Standard ML, Zig, Mojo, Erlang, SPWN, Pkl, Gleam**, and custom command

Book for VS Code

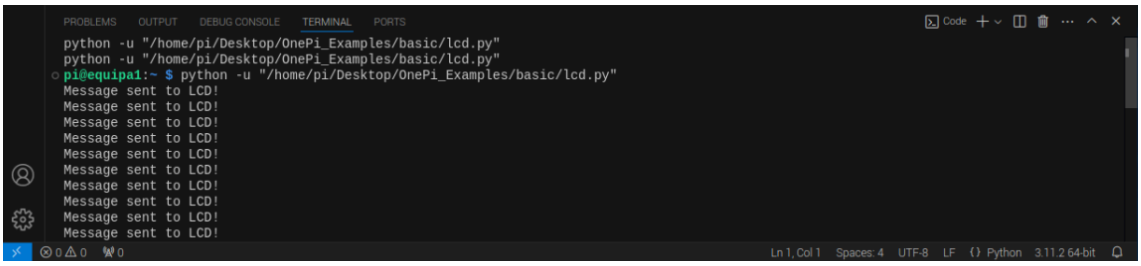
《Visual Studio Code 权威指南》：带你深入浅出 VS Code！

With the extension installed, when you open a Python code, you can execute it by clicking the ▶ "Run" button.


lcd.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
lcd.py
home > pi > Desktop > OnePi_Examples > basic > lcd.py > ...
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.11.2 64-bit

```python
"""
Latest update: 08-09-2023

This code example is in the public domain.
http://www.botnroll.com

Description:
Print a message on the robot LCD.
"""


import time
from onepi.one import BnrOneA

one = BnrOneA(0, 0)  # object variable to control the Bot'n Roll ONE A


def setup():
    one.stop()  # stop motors


def loop():
    one.lcd1(" LCD Test OK !! ")  # print data on LCD line 1
    one.lcd2("BotnRoll ONE A+ ")  # print data on LCD line 2
    one.lcd1("botnroll ONE A+ ")  # print data on LCD line 2
    print("Message sent to LCD!")  # print data on terminal.
    time.sleep(1)  # wait 1 second


def main():
    setup()
    while True:
        loop()


if __name__ == "__main__":
    main()
```

When running the code, a terminal will appear below where you can interact with the code.



```
python -u "/home/pi/Desktop/OnePi_Examples/basic/lcd.py"
python -u "/home/pi/Desktop/OnePi_Examples/basic/lcd.py"
pi@equipa1:~ $ python -u "/home/pi/Desktop/OnePi_Examples/basic/lcd.py"
Message sent to LCD!
Message sent to LCD!
Message sent to LCD!
Message sent to LCD!
Message sent to LCD!
Message sent to LCD!
Message sent to LCD!
Message sent to LCD!
Message sent to LCD!
Message sent to LCD!
```

Anexo D: Library Installation and Networks Configuration

In case you are using a pre-installed Linux image that was not provided by www.botnroll.com, you must follow the instructions in this annex to use the Raspberry Pi Add-On with the Bot’n Roll ONE A+. Support for this situation will be limited to what is presented in this document, and no additional external assistance will be provided during the RoboParty. It is recommended to use the previously provided image, which can be installed by following ANNEX B, as this tutorial is intended only for advanced users.

D.1 Raspberry Pi Configuration of SPI and I2C

If you already have a Raspberry Pi setup with SPI and I2C enabled, skip to step D.2.

1. Download Raspberry Pi Imager https://www.raspberrypi.com/software/
2. Using the Imager, install Raspberry Pi OS (32-bit) on a micro SD card (64GB).
3. Insert the micro SD card into the Raspberry Pi's socket.
4. Start the Raspberry Pi
5. Follow the on-screen instructions to set up the desktop environment.

To activate the SPI and I2C follow these steps:

1. Open a terminal (CTRL+ALT+T) and run "sudo raspi-config".
2. Use the down arrow to select "Interface Options".
3. Down arrow until SPI.
4. Select "Yes" when it asks you to enable SPI.
5. Repeat the process to enable I2C.
6. Reboot the Raspberry Pi

D.2 Python Library Installation Manually

You can choose to install the “onepi” library using one of the two options listed below.

Install using pip (option 1):

```
pip3 install onepi --break-system-packages
```

Install using the github (option 2):

Run the following lines of code to install via GitHub.

```
mkdir ~/src
cd ~/src
git clone https://github.com/botnroll/bnronepi # or download and extract zip file
cd bnronepi-main
python setup.py sdist
pip install
```

Troubleshooting

If you encounter an error when installing Matplotlib, you may need to install the OpenBLAS library. You can do this via the command line:

```
sudo apt-get install libopenblas-dev
```