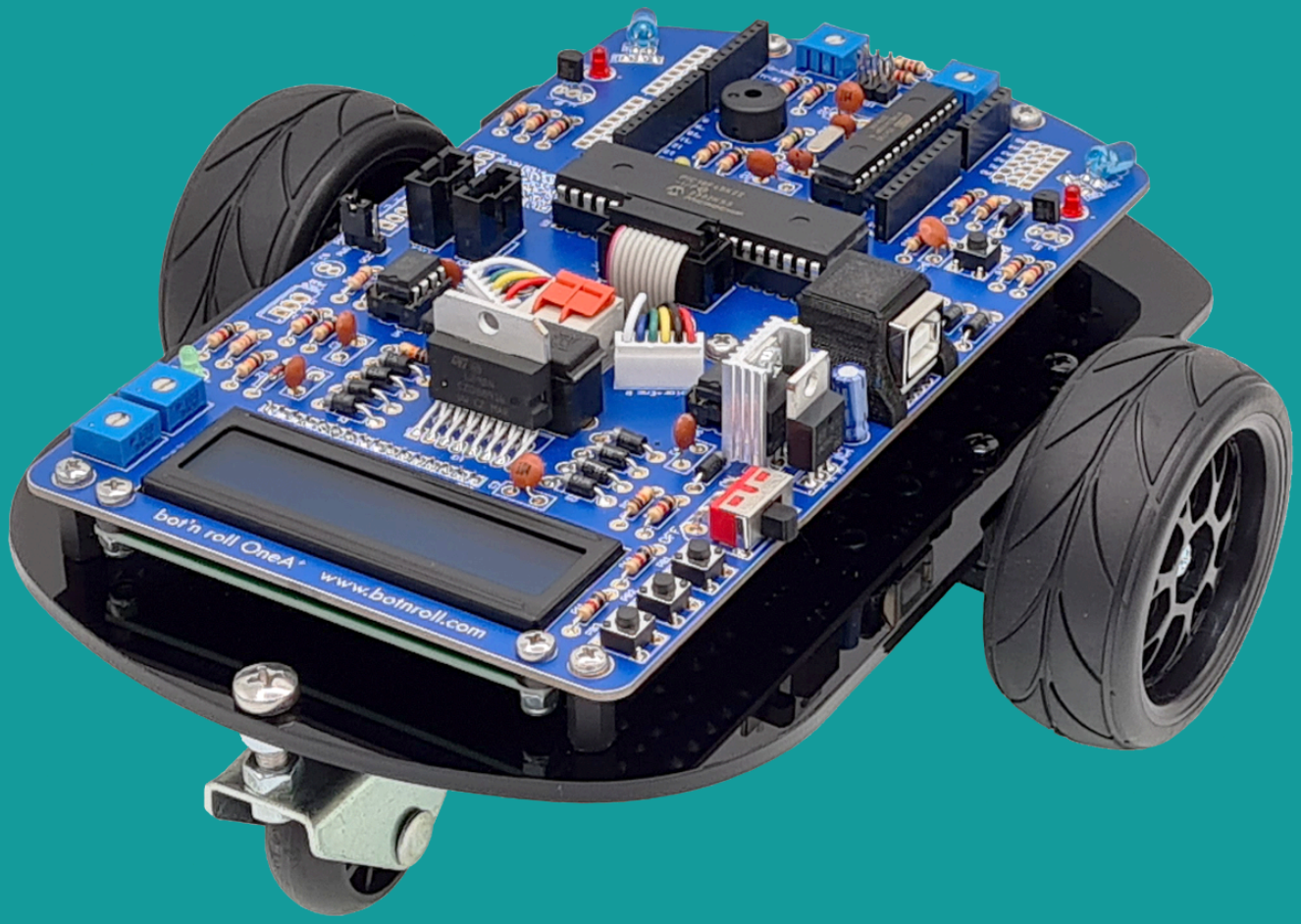
bot'n roll ONE A+

build your own robot



Arduino® software user manual

CONTENT

Revisão do Documento: 2 de Março de 2025

1. Introduction

The Bot'n Roll ONE A+ is programmed using the C language with the Arduino IDE programming environment. The ATmega328 microcontroller in the robot has the Arduino Uno bootloader, so the robot is programmed as if it were an Arduino Uno.

The robot has a second microcontroller, a PIC18F45K22, which comes 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 ATmega328 master.

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

The "BnrOneAPlus" Arduino library allows the user to control the robot easily—just by correctly using the library's commands in the Arduino IDE. These commands are listed and explained in this manual.

Although both microcontrollers can be programmed in C by the user, only the ATmega328 with the Arduino bootloader is programmed on a daily basis using the "BnrOneAPlus" library.

The PIC18F45K22 can be programmed in C using the MPLABX IDE programming environment and the Microchip XC8 compiler 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 the Arduino can use the new functionality.

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

1. Programming the Bot'n Roll ONE A+

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

- VCP driver installed, the USB port driver for the Bot'n Roll ONE A+ (Annex A);
- Arduino IDE installed (Annex B1);
- BnrOneAPlus library installed in the Arduino IDE (Annex B2).

For detailed information on installing the items mentioned above, refer to Annexes A and B at the end of this manual.

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

2. BnrOneAPlus Library for Arduino

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

#include<BnrOneAPlus.h>

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

BnrOneAPlus one;

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

A library is usually created for data or hardware manipulation and always consists of at least two files. However, in the case of Arduino, there is an additional file with the **".txt"** extension:

- A file with the **".h"** extension (header), which contains a list of all the functions, commands, and definitions of the library.
- A file with the **".cpp"** extension (C++ source), which contains the implementation of all the functions declared in the header file.
- A **keywords.txt** file, which allows the Arduino IDE to recognise the library functions and display them in a different colour from the rest of the code.

The **"BnrOneAPlus"** library was created to control the hardware associated with the **PIC18F45K22** and allows the Arduino to interact with it via the SPI communication bus. The Arduino has access to all the hardware and functionalities defined in the library and in the **PIC18F45K22** software. The **"BnrOneAPlus"** library and the **PIC18F45K22** software were designed to work together, meaning that any change to one requires an adjustment to the other.

The library is available online at:

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

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

- **https://botnroll.github.io/BnrOneAPlus/class_bnr_one_a_plus.html.**

3. ARDUINO IDE

The Arduino development environment includes a text editor for writing code, a message area, a text console, a toolbar with the most important functions, and several menus. It connects to the Arduino hardware of the Bot'n Roll ONE A+ to upload code and communicate with the robot.

An Arduino program is called a "sketch", written in the text editor and saved on your computer with the ".ino" extension.

The message area displays information about saving and exporting programs, as well as any errors.

The console provides detailed text messages about errors and other information.

In the bottom right corner of the window, you can see information about the board being programmed and the serial port in use.

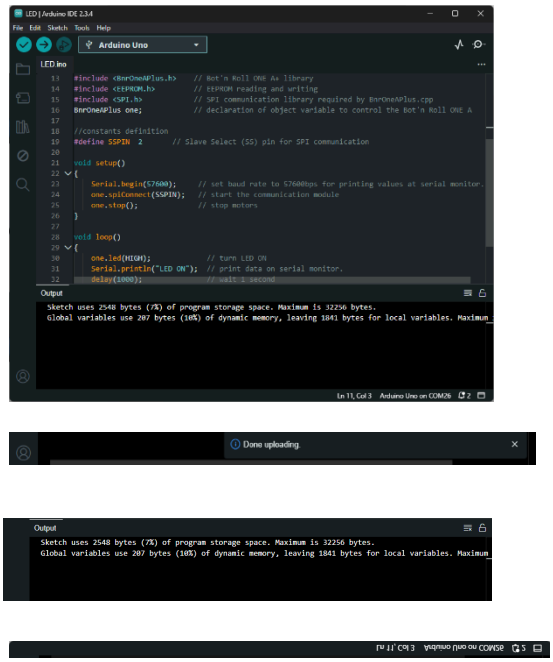


Fig. 1: Módulos do Arduino IDE

The buttons on the toolbar and their function:

Verify: Check for errors in the code

Upload: Compile the code and upload it to the Arduino.

New: Create a new sketch.

Open: Open a sketch saved on the computer.

Save: Save the sketch.

Serial Monitor: Open the serial monitor.

The serial monitor allows you to view data sent from the Arduino to the computer and also send data from the computer to the Arduino. It is very useful in programming, as you can print text and variable values here, making it easier to debug (find errors) in your program. When you open the serial monitor, your program on the Arduino restarts.

4. C Language

The C language was developed in 1972 by Dennis Ritchie at Bell Labs in New Jersey. It was created to be a powerful and fast language for use in the Unix operating system that was being developed at the time. Over the years, it has been improved and updated, proving to be robust and reliable, and eventually being adopted by other operating systems such as Windows, macOS, and Linux.

C has been evolving continuously since the first version, known as "K&R C". In 1989, the first official standard specification was published by the American National Standards Institute (ANSI C). In 1990, the International Organization for Standardization (ISO) introduced ISO C. In 1999, the C99 standard was released, followed by periodic revisions aimed at making the language more powerful and better suited to modern needs.

All **Arduino programs** have two **mandatory** routines, or functions:

- The **setup()** routine, which is executed **only once** at the start of the program. Here, you should place all the necessary code to initialize variables, configure **input and output pins**, set up **SPI, Serial, and I2C communication**, and perform any other required configurations.

- After the setup, your program enters the **loop()** routine, where it remains indefinitely. The term **loop** means cycle, and in this case, it is an **infinite loop**, meaning that when the program reaches the end of the cycle, it starts over again! This is where you write your program and create the **intelligence** for your robot!

This manual does not explain **C++ programming** itself. Instead, we recommend using the **"BnrOneAPlus" library examples** and general **Arduino examples**. All the code is well-commented, and you'll need to experiment and test it to understand how it works.

Here are some tips:

- **Create new programs** based on **basic examples**. Try combining **3 or 4 functionalities** of the robot in the same program!
- **Programs rarely work perfectly the first time!** Don't get discouraged—find the issue and fix it!
- **Add code gradually** and **test frequently** to check if everything works as expected.
- **Testing takes longer than writing the program!** A good program requires thorough testing!
- **Use debugging tools** like the **LED, serial monitor, or LCD** to print variable values and check if the program reaches certain parts of the code.

Programming is like learning a new sport!

At first, it feels **difficult and frustrating** because you don't have the necessary skills, don't know the rules, and feel a bit lost. But with practice and training, you **learn, improve, and get better** in every way. **With effort, you'll eventually become a pro!**

The **Bot'n Roll ONE A+** supports **a wide range of hardware**. There are additional components, commonly known as **Arduino shields**, that allow you to do **almost anything** you can imagine—and they are compatible with the Bot'n Roll ONE A+! **All shields come with libraries** to help you integrate them, and **your imagination is the only limit!**

Anexo A: VCP Driver Installation for USB converter (RS232)

The driver allows your computer's operating system to communicate with the Bot'n Roll ONE A+.

To install the driver, visit the Bot'n Roll ONE A+ support page (http://botnroll.com/one-a-plus/) and download it by clicking on "VCP Driver - Windows" or "VCP Driver - Mac OS X", according to your operating system. Once the download is complete, extract the file with the ".zip" extension and run the application.

Whenever you connect the robot to the computer using the USB cable, a virtual COM port (VCP) is created, through which communication between the Bot'n Roll ONE A+ and the PC takes place. The application for programming the robot uses this port to communicate with the Bot'n Roll ONE A+ and transfer programs to the robot.

The USB-Serial converter used in the Bot'n Roll ONE A+ is a PoUSB12 from PoLabs, which uses the CP2102 Bridge device from Silicon Labs.

Anexo B: Programming IDE 2.0

The software used for programming the robot is Arduino IDE 2.0. This application is necessary for editing programs in C++ language. It is also used to transfer your programs to the Bot'n Roll ONE A+.

B.1 Arduino IDE Instalation

To install the Arduino IDE, visit the Bot'n Roll ONE A+ support page at (http://botnroll.com/one-a-plus/). In the "Software | Drivers" section, click on "Arduino IDE" to install it according to your operating system.

Once the download is complete, extract the "**.zip**" file and place the extracted folder in a directory of your choice on your computer.

This folder contains several subfolders and files, including the **"arduino.exe"** application, the executable that launches the Arduino IDE.

B.2 BnrOneAPLus library installation on Arduino IDE 2

Libraries are your working tools in programming. The "BnrOneAPlus" library, developed by botnroll.com for the Arduino IDE, contains all the necessary commands for controlling the robot. This library must be installed in the Arduino IDE.

To install the library in **Arduino™ IDE 2**, click on the **"Sketch" tab → "Include Library" → "Manage Libraries..."**, search for "BnrOneAPlus" and install the library by clicking the "Install" button.

To install the library in **Arduino™ IDE 1**, click on the **"Sketch" tab → "Include Library" → "Add .ZIP Library..."**, select the BnrOneAPlus.zip file, and the library will be installed automatically. Close and reopen the Arduino™ IDE to ensure the library is functional!

The procedures for installing the VCP driver and the "BnrOneAPlus" library are also described in the Assembly Manual.

B.3 Configuring the communication with the robot

Before performing this step, make sure you have installed the VCP driver correctly (see ANNEX A). Connect the Bot'n Roll ONE A+ to the computer using the supplied USB cable. At this point, a COM port will be automatically assigned for communication with the robot.

Open the Arduino IDE, and in the top bar of the window, find a menu where you can choose the Board and the Port. Select the board "**Arduino Uno**". The Bot'n Roll ONE A+ will be programmed as if it were an Arduino Uno.

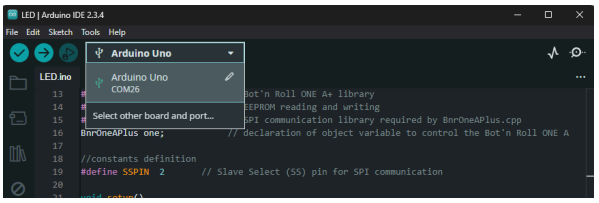


Fig. Selecionar a placa a programar

If it does not appear instantly, you can click on "**Select other board and port...**" and the following menu will appear. Choose the board "**Arduino Uno**" and the corresponding COM port to program the Bot'n Roll ONE A+.

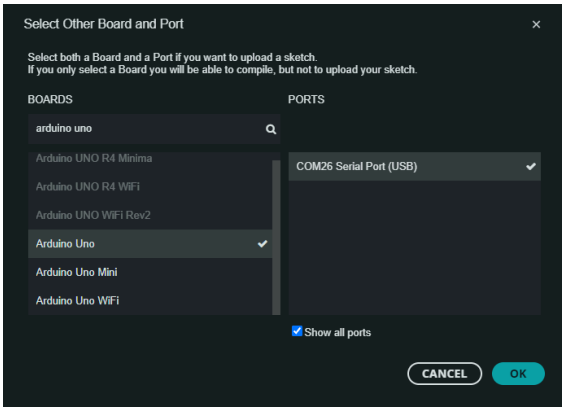


Fig. Selecionar a Porta Série

If no COM port is available, it is most likely that you have not correctly installed the VCP driver for the USB-to-Serial converter.

Open the Windows Device Manager and look for the item labelled "**Ports (COM & LPT)**". Expanding this item, you will see all the assigned COM ports.

"**Silicon Labs CP210x USB to UART Bridge**" is the designation that identifies the connection port to the Bot'n Roll ONE A+. (In the example from the figure, COM21 was assigned.)

If the item "**Silicon Labs CP210x USB to UART Bridge**" does not appear, you will need to correctly install the VCP driver.

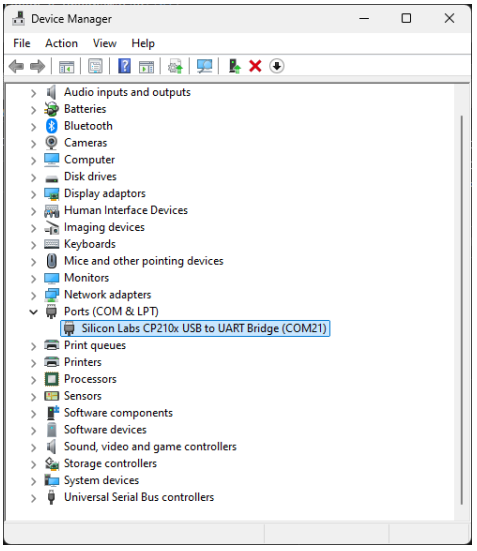


Fig. Portas COM no Gestor de Dispositivos

B.4 Loading a Program to the Bot'n Roll ONE A+

In the Arduino programming environment, you will find several example programs that you can load onto the robot.

Click on **"File -> Examples -> 01.Basics -> Blink"** and a new window will appear with the code for this example.

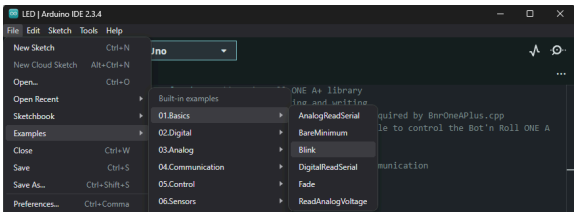


Fig. Carregar um programa de exemplo

Click on **"File -> Upload"** or press the arrow symbol pointing to the right to upload the program to the robot. Once the upload is complete, you should see the yellow L LED blinking every second!

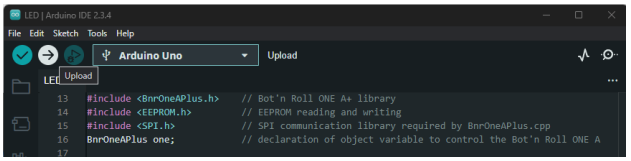


Fig. Enviar o programa para o robô

Clicking on **"File -> Examples -> BnrOneAPlus -> ..."** you will find all the example programs provided by botnroll.com specifically for the Bot'n Roll ONE A+.

In "**File -> Examples -> BnrOneAPlus -> Basic ->...**", you will find the basic programs designed to test all the robot's hardware. You should study and understand these small programs well!

In **"File -> Examples -> BnrOneAPlus -> Advanced -> ..."**, you will find more advanced programs that you should only study after understanding the simpler ones.

In **"File -> Examples -> BnrOneAPlus -> Extra -> ..."**, you will find programs related to the extra components that expand your Bot'n Roll ONE A+.

In **"File -> Examples -> BnrOneAPlus -> Fun Challenge -> ..."**, you will find programs related to the Fun Challenge.

In **"File -> Examples -> BnrOneAPlus -> Calibrate -> ..."**, you will find programs related to calibrating your Bot'n Roll ONE A+.

In **"File -> Examples -> BnrOneAPlus -> CrazyRace -> ..."**, you will find programs related to the CrazyRace challenge.

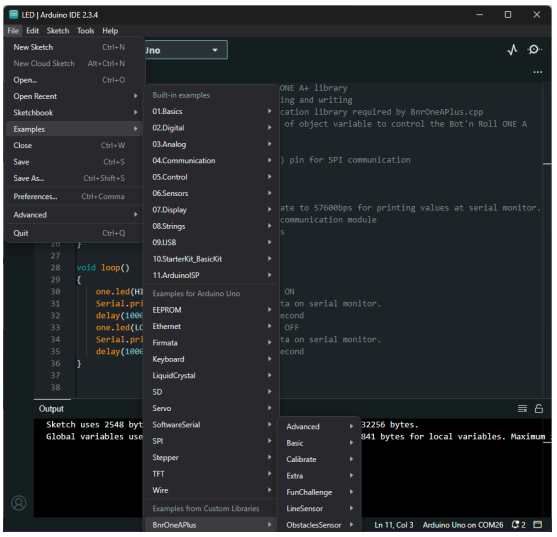


Fig. 7: Programas da biblioteca BnrOneAPlus

In **"File -> Examples -> BnrOneAPlus -> Obstacles Sensor -> ..."**, you will find programs related to the obstacle sensors.

In **"File -> Examples -> BnrOneAPlus -> Line Sensor -> ..."**, you will find programs related to the Line Following Sensor.