

# Instalar Micropython

## Conceptos previos:

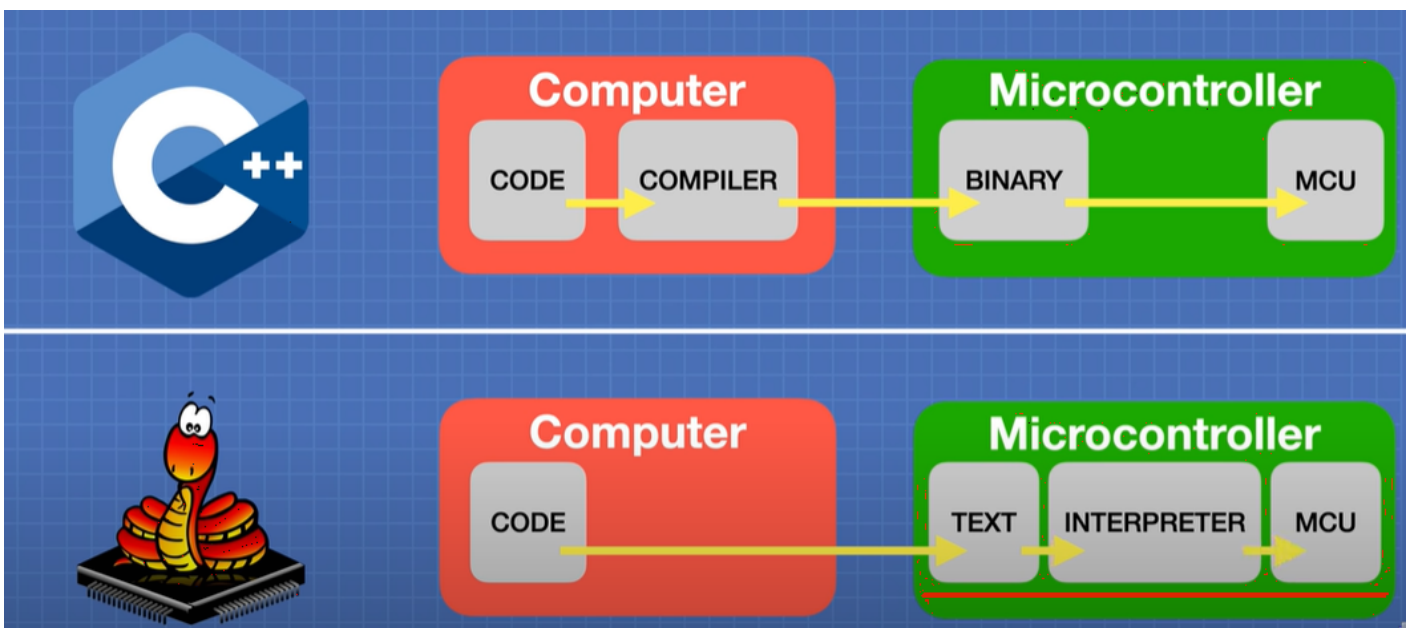
- Los **lenguajes de alto nivel**, es decir el **código**, que es entendible por los humanos (C++, Java, Python...) son textos que se tienen que traducir al lenguaje entendible por el procesador **MCU** (Micro Controler Unit). Este **lenguaje de bajo nivel** que está escrito en **binario** es difícil de entender para los humanos
- El **Compilador** es un programa que **Interpreta** este texto de lenguaje de alto nivel, y lo convierte en lenguaje de bajo nivel
- El Arduino Alvik se puede programar con Arduino IDE como con Micropytno, los dos son de alto nivel

Tanto Micropython como Arduino IDE son lenguajes de tipo CODIGO por lo tanto sólo se aconseja EN SECUNDARIA

Cuando permita lenguaje tipo BLOQUES como Scratch, ya será adecuado para PRIMARIA

## ¿Dónde se compila Micropython?

Como puedes ver en este vídeo en 21:20 Python se compila **dentro del microcontrolador** es decir, dentro del ESP32. A diferencia con otros lenguajes, como el C++, el ordenador tiene el compilador, y se lo da ya en binario.



Fuente vídeo Exploring the Arduino Nano ESP32 | MicroPython & IoT

## ¿Y a mi qué más me da?

No, para nada, si programas ESP32 con Arduino IDE o Arduino Cloud (que está basado en C++) **te has cargado el compilador Python** del ESP32 luego si quieres programar en Python, tienes que instalar el compilador dentro del ESP32 (y da problemas, [ver Modo Bootloader](#))

## ¿Cómo meto el compilador MicroPython dentro del ESP32?

Para meter el programa compilador de MicroPython dentro del ESP32 (cuando un programa se instala en un hardware entonces se llama **firmware**) tenemos que

1. Poner la placa en modo bootloader ([ver página siguiente](#))
2. Instalar **micropython** en el ESP32 con el programa **Arduino Micropython Installer** de esta página <https://labs.arduino.cc/en/labs/micropython-installer>

## MicroPython Installer

Descargamos el programa y ejecutamos teniendo conectado el ESP32 del Alvik, (no hace falta encender el robot, pues sólo trabajamos con el ESP32) lo detecta y simplemente le damos a Instalar Micropython dentro del chip



### Detected Boards



Arduino Nano ESP32

Reload

**INSTALL MICROPYTHON**

## ¿Y al revés pasar de Micropython a ArduinoIDE hay problemas?

Pues al revés, si ejecutamos Arduino IDE, da error, pues el compilador microPython instalado dentro del ESP32 impide que se ejecuten los programas de Arduino IDE (sale el error No DFU capable USB device available Failed uploading: uploading error: exist status 74) se soluciona tan sencillo como volver a poner el ESP32 en modo Bootloader

## ¿Y si me paso de Arduino IDE a Micropython?

Pues vuelve a

1. Poner la placa en modo bootloader
2. Instalar micropython en el ESP32 con el programa Arduino Micropython Installer

## ¿Y con esto ya puedo crear mis programas con Micropython?

No, con esto tienes el compilador interpretador dentro del chip, pero necesitas un editor en tu PC y que se comunique con el Micropython del chip

## Arduino Lab for Micropython

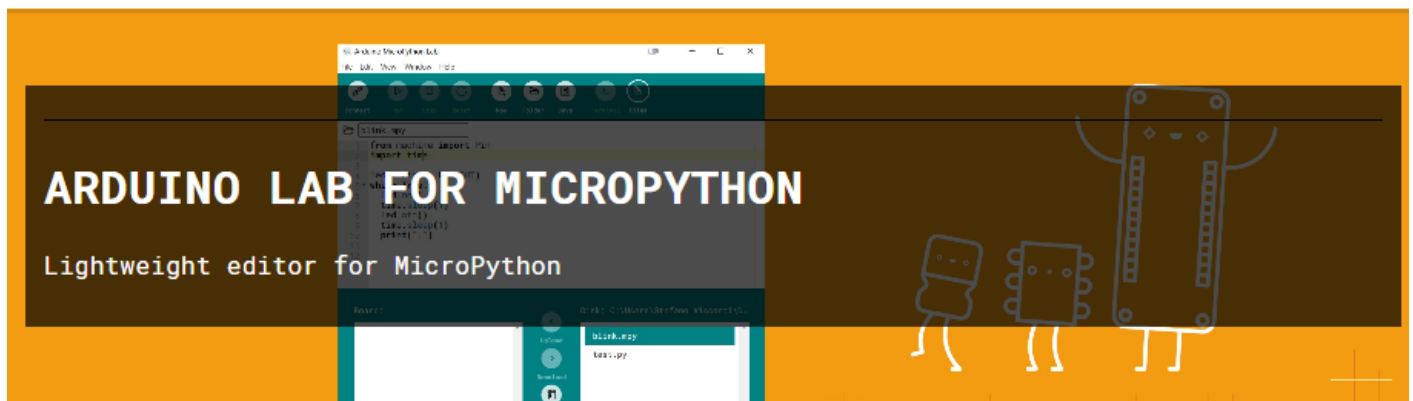
El lenguaje de programación más idóneo para el Arduino Alvik es el **MICROPYTHON**, de momento Arduino Alvik no esta disponible en programación con bloques. Al ser hardware libre, si se convierte popular, no es de extrañar que lo incorporen los diferentes programadores de entornos de lenguaje en bloques para ser más accesible a niveles educativos de primaria.

Tal y como dice la página <https://docs.arduino.cc/micropython/> hay dos editores para cargar MicroPython en el Arduino Alvik

- **Arduino Lab for Micropython** <https://labs.arduino.cc/en/labs/micropython>
- **OpenMW** <https://openmv.io/pages/download>

Nosotros en este curso elegimos **Arduino Lab for Micropython** por su sencillez y adaptación al Arduino Alvik

Tal y como dice aquí **ES UN PROGRAMA PORTABLE**, es decir, no hay que instalarlo, simplemente descomprimir y ejecutar



Arduino Lab for MicroPython is a lightweight editor for MicroPython programs, supporting connection with a board, code upload, file transfer, interactive REPL shell.

This project is sponsored by Arduino, based on original work by Murilo Polese. This is an experimental pre-release software, please direct any questions only to Github issues.

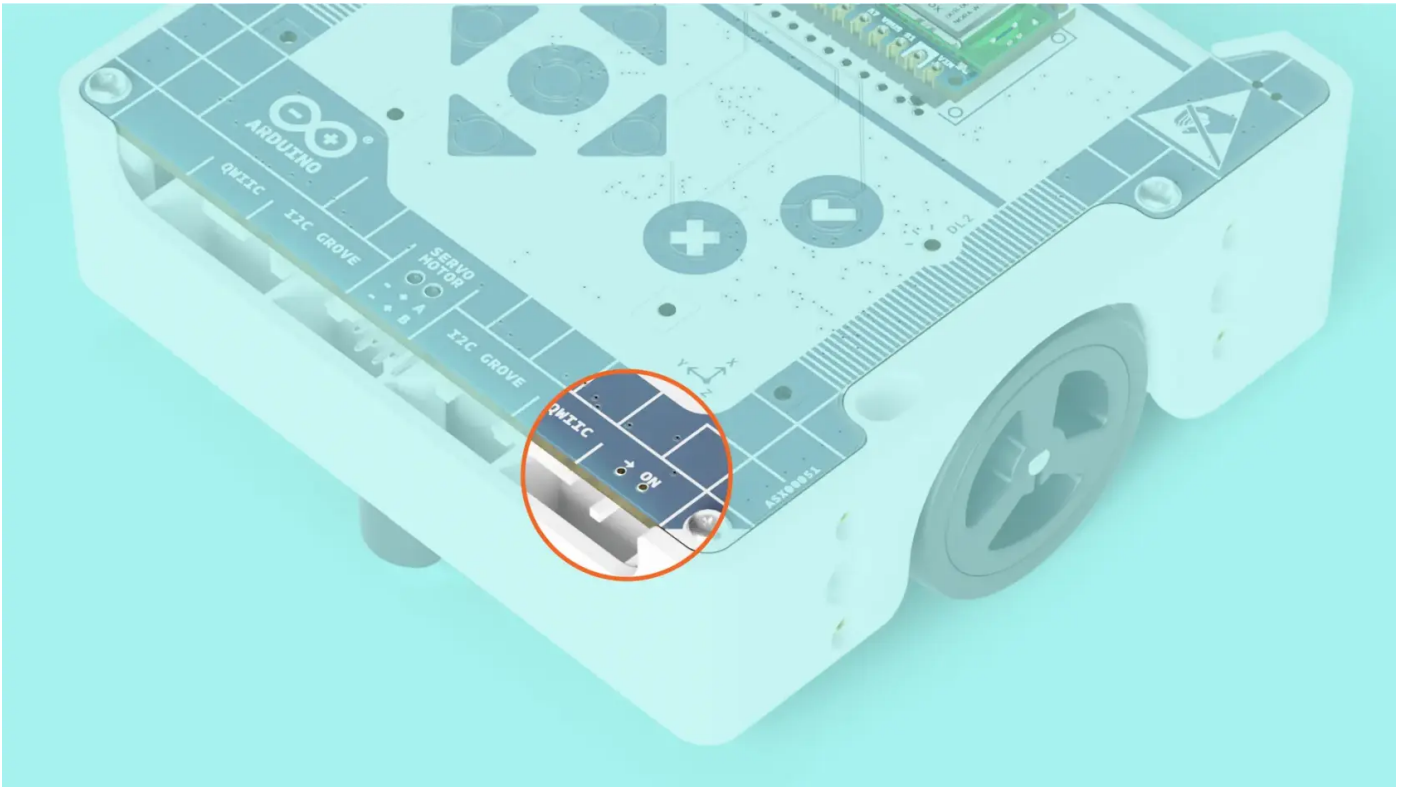
Download the latest version of the editor here:

- [For Linux \(x64\)](#)
- [For macOS \(Intel-based\)](#)
- [For macOS \(Apple silicon\)](#)
- [For Windows \(x64\)](#)

Install by simply unzipping in a folder the compressed file (based on your platform) and launching the executable file inside the folder.

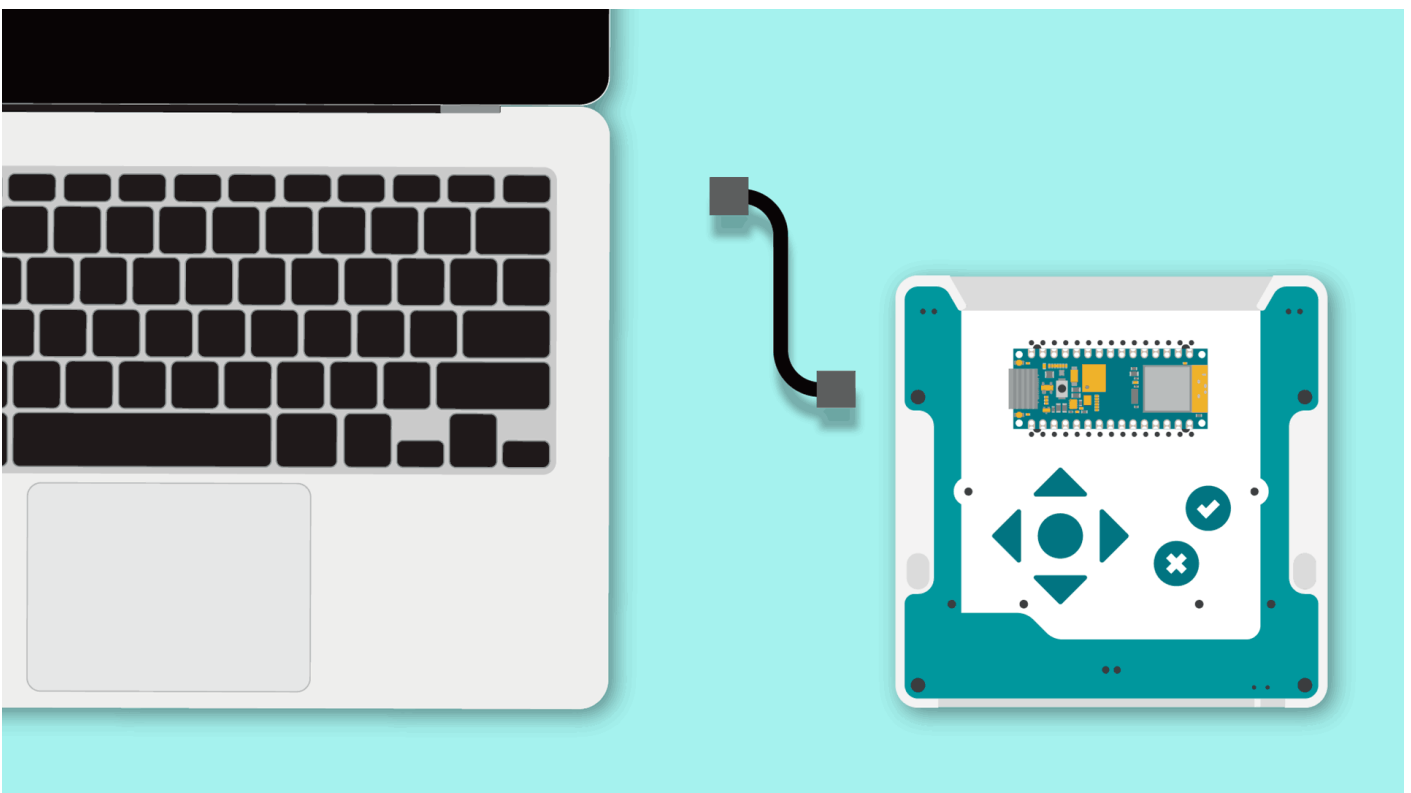
You can find the source code in this [repository](#). Previous releases can be found [here](#).

Ahora, apagamos Arduino Alvik



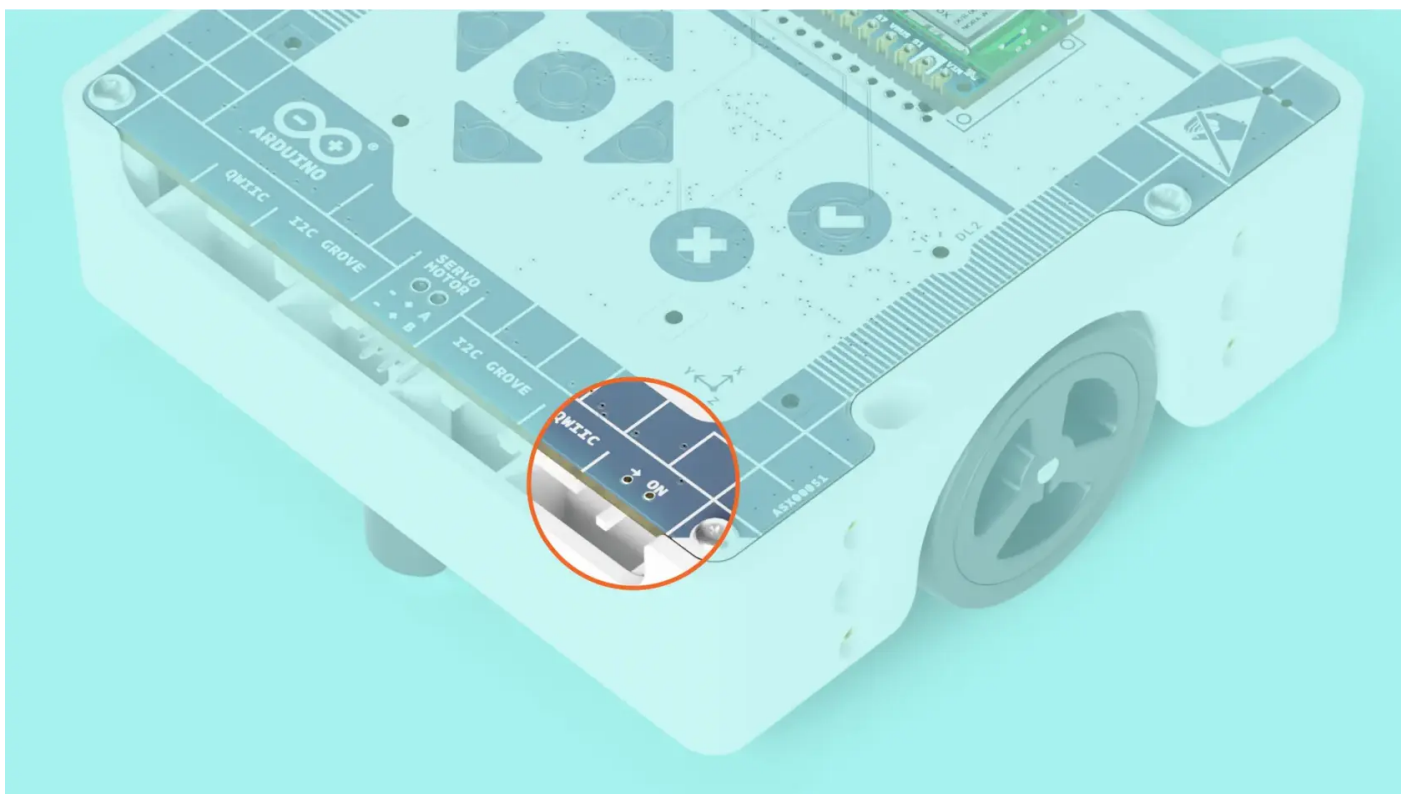
Licencia CC-BY-NC-SA origen <https://courses.arduino.cc/explore-robotics-micropython/lessons/getting-started/>

Luego lo conectamos por cable



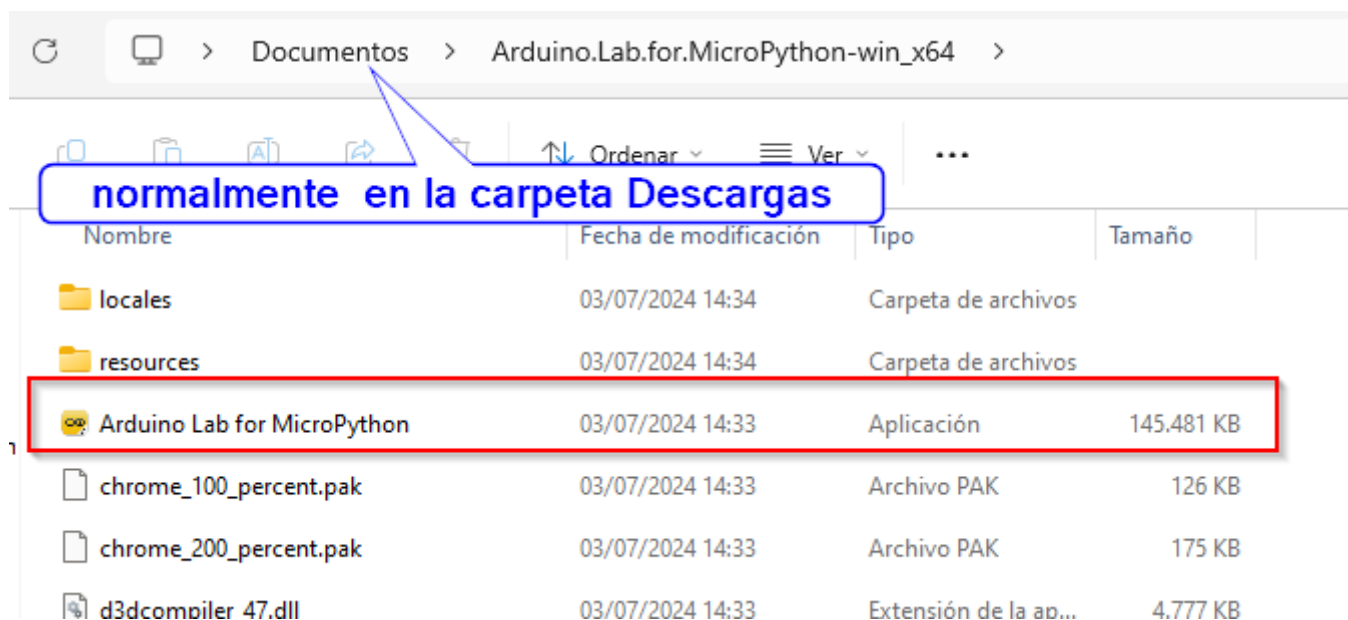
Licencia CC-BY-NC-SA origen <https://courses.arduino.cc/explore-robotics-micropython/lessons/getting-started/>

Lo encendemos

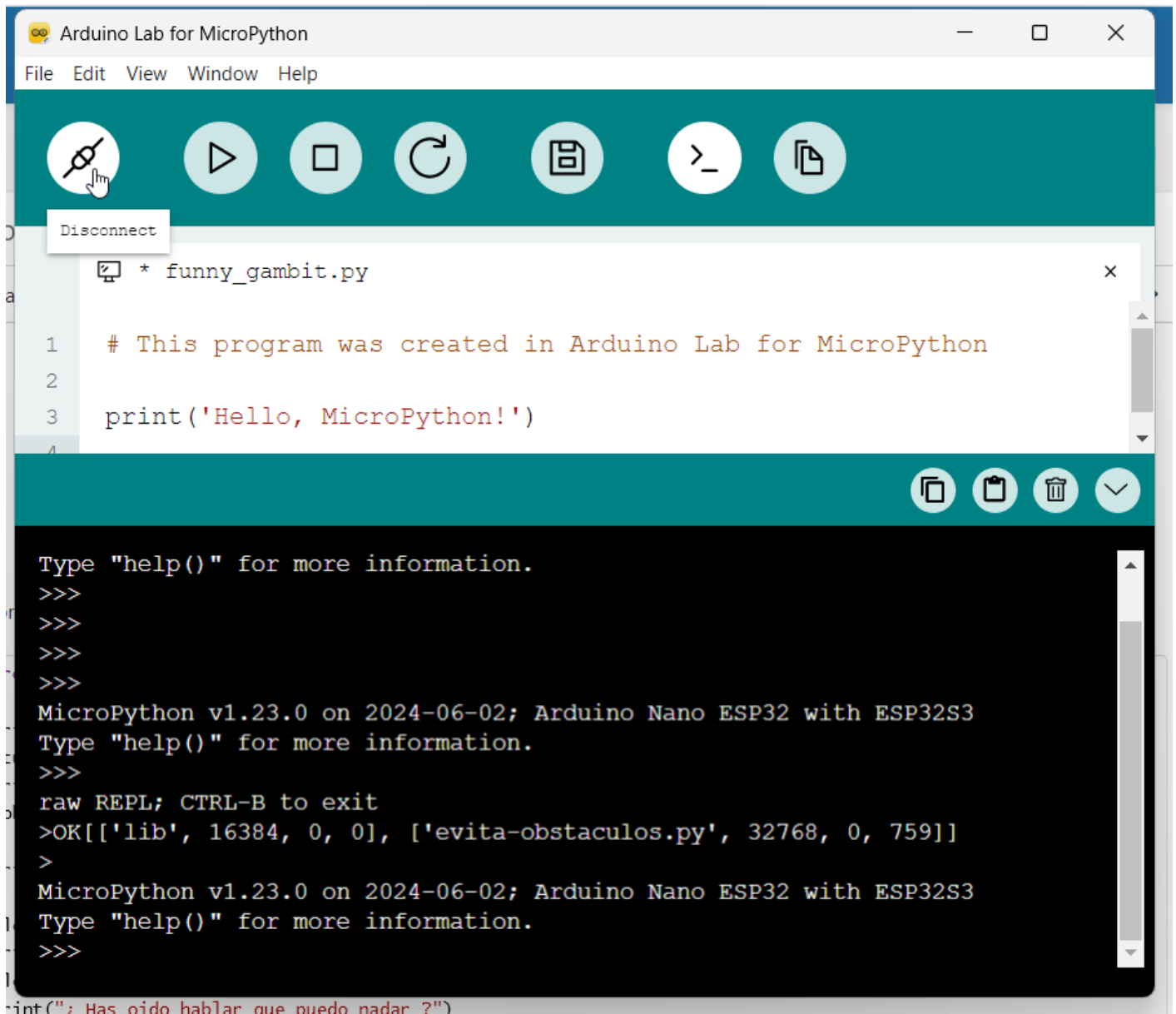


Licencia CC-BY-NC-SA origen <https://courses.arduino.cc/explore-robotics-micropython/lessons/getting-started/>

Ejecutamos el programa que hemos descomprimido:



Al ejecutar el programa, le podemos dar al botón de conectar con la placa y veremos que se comunica correctamente con el compilador instalado dentro del chip



Revision #9

Created 4 July 2024 18:39:03 by Javier Quintana

Updated 14 September 2024 13:34:46 by Javier Quintana