

# Pico Servo Driver

## Overview

Servo Driver Module For Raspberry Pi Pico, 16-Channel Outputs, 16-Bit Resolution.

### Features

- Standard Raspberry Pi Pico header, supports Raspberry Pi Pico series boards.
- Up to 16-Channel servo/PWM outputs, 16-bit resolution for each channel.
- Integrates 5V regulator, up to 3A output current, allows battery power supply from the VIN terminal.
- Standard servo interface, supports commonly used servos such as SG90, MG90S, MG996R, etc.
- Adapting unused pins of Pico, easy expansion.
- Comes with online resources and manuals (including Raspberry Pi Pico C/C++ and MicroPython demo).

### Specification

- Operating voltage: 5V (Pico) or 6~12V (VIN terminal)
- Logic voltage: 3.3V
- Servo voltage level: 5V
- Control interface: GPIO
- Mounting hole size: 3.0mm
- Dimensions: 65 × 56mm

### Pinout



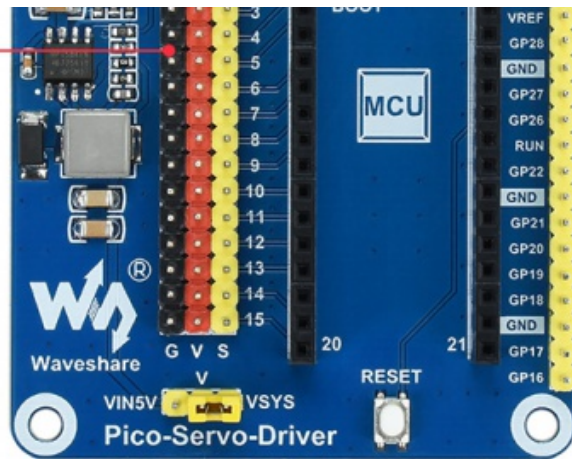
Channel 0~15, for concurrently connecting up to 16x servo



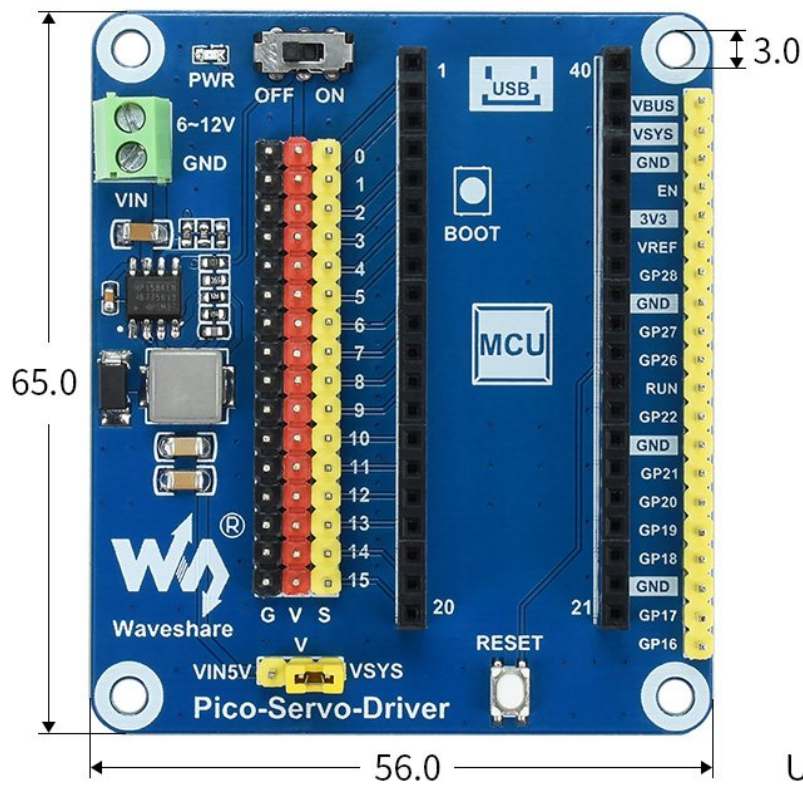
16x channels servo headers

- G pins: GND
- V pins: 5V power
- S pins: PWM signal

\* please correctly connect the servo, it would not work if the connection were reversed.



## Dimensions

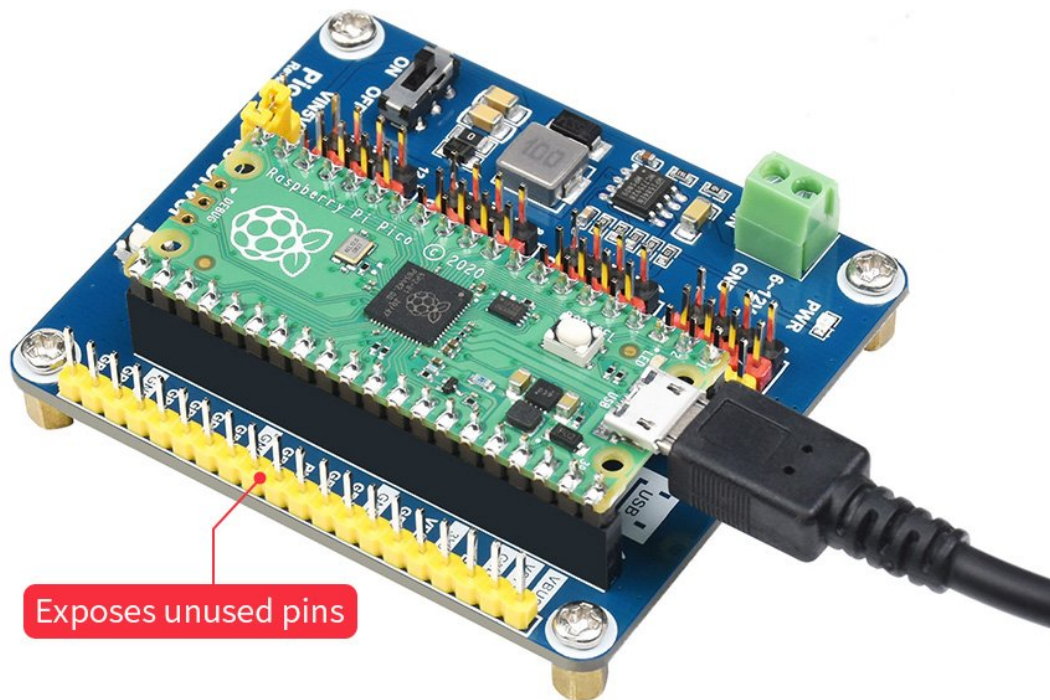


Unit :mm

## User Guide

## Hardware Connection

When connecting the Pico, please be careful not to reverse the corresponding direction. You can observe the end of the module with USB silkscreen and the end of Pico's USB interface to determine the direction (you can also judge the pin number of the motherboard on the module and the pin number of Pico).



## Demo Download

### Raspberry Pi

1. Open a terminal of Raspberry Pi
2. Download and unzip the demo codes to directory Pico C/C++ SDK

```
sudo apt-get install p7zip-full
cd ~
sudo wget https://files.waveshare.com/upload/3/31/Pico_Servo_Driver_Code.7z
7z Pico_Servo_Driver_Code.7z -o./Pico_Servo_Driver_Code.7z
cd ~/Pico_Servo_Driver_Code
```

### C

- The following tutorials are operated on Raspberry Pi. As CMake has multi-platforms and can be ported, you also can successfully compile on a PC in different ways.

To compile, make sure you are in the c directory:

```
cd ~/Pico_Servo_Driver_Code/c/
```

Create and enter the "build" directory in the folder, and add the SDK:

where ../../pico-sdk is the directory of your SDK.

There is a "build" in our sample program, just enter it directly.

```
cd build
export PICO_SDK_PATH=../../pico-sdk
(Note: Be sure to write the path where your own SDK is located)
```

Execute cmake to automatically generate Makefile files

```
cmake..
```

Execute make to generate executable files, the first compilation time is relatively long.

```
make -j9
```

After the compilation is complete, the uf2 file will be generated.

Press and hold the button on the Pico board, connect the pico to the USB port of the Raspberry Pi through the Micro USB cable, and release the button. After connecting, the Raspberry Pi will automatically recognize a removable disk (RPI-RP2) and copy the main.uf2 file in the build folder to the recognized removable disk (RPI-RP2).

```
cp main.uf2 /media/pi/RPI-RP2/
```

## python

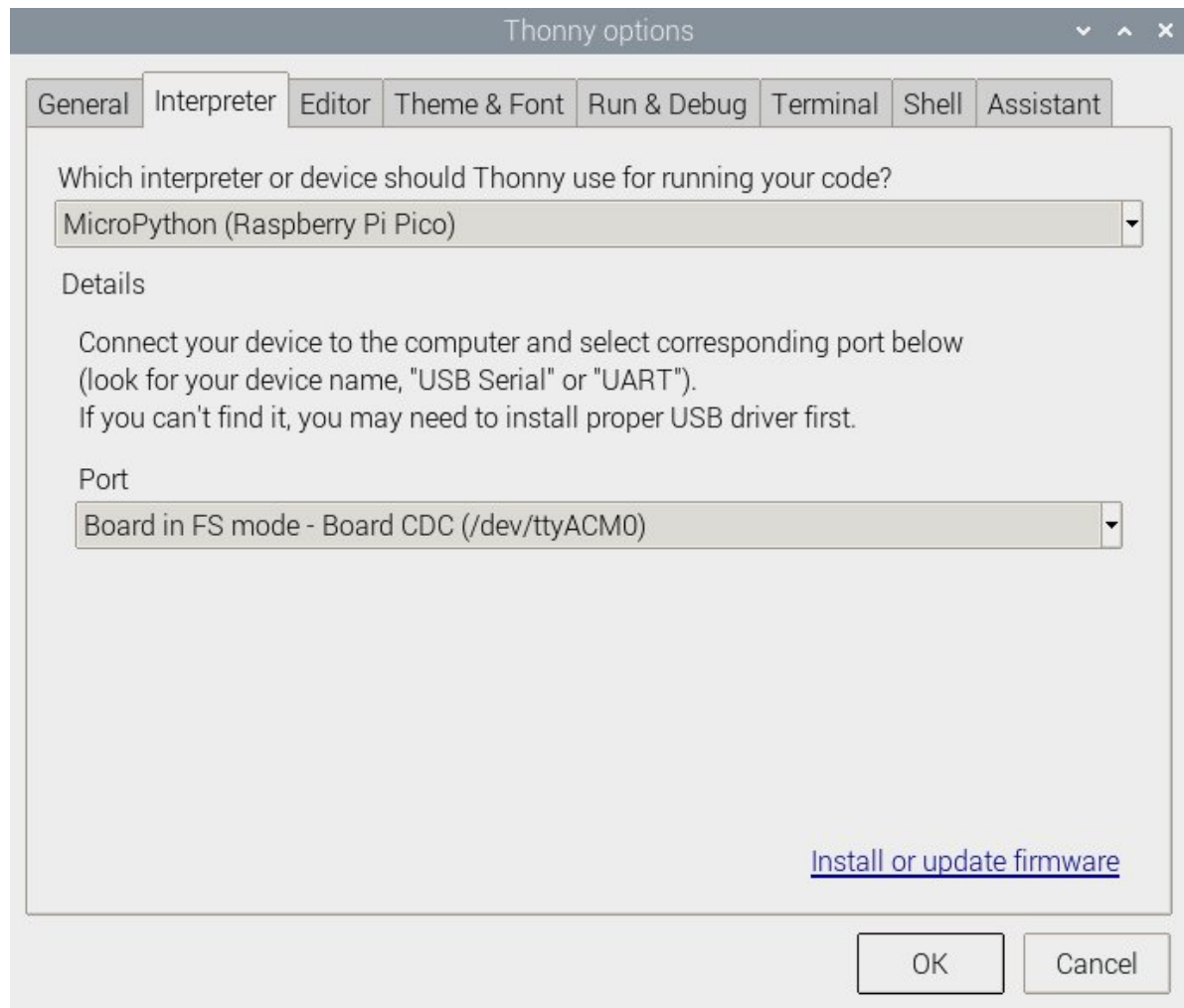
1. Flash the Micropython firmware and copy the pico\_micropython\_XXXXX.uf2 file into pico (detailed in the Windows tutorial below).

[official firmware download](#)

2. Open Thonny IDE on the Raspberry Pi (click the Raspberry Pi logo -> Programming -> Thonny Python IDE ), you can check the version information in Help->About Thonny.

- To make sure your version has Pico support package, also you can click Tools -> Options... -> Interpreter to select MicroPython (Raspberry Pi Pico and ttyACM0 port).

As shown:



If your current version of Thonny does not have the pico support package, enter the following command to update the Thonny IDE.

```
sudo apt upgrade thonny
```

3. Click File->Open...->python/Pico\_Servo\_Driver\_Code/python/servo.py to run the script.

As soon as the experimental phenomenon channel is connected to the servo, it will rotate from 0 degrees to 180 degrees, and cycle three times.

## Windows

### C

- [Install in Windows](#)

### Download the demo

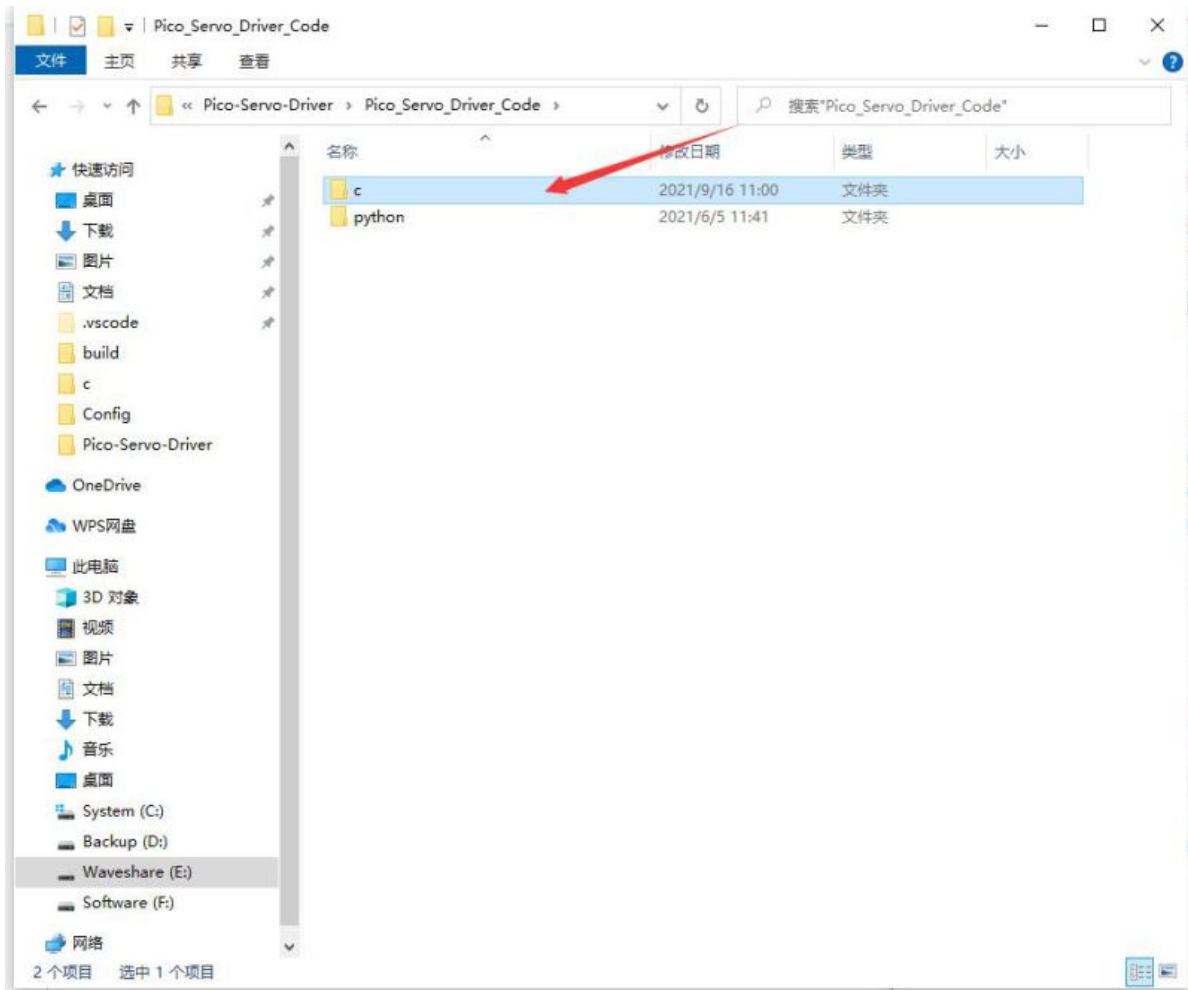
---

- [Pico Servo Driver Demo](#)

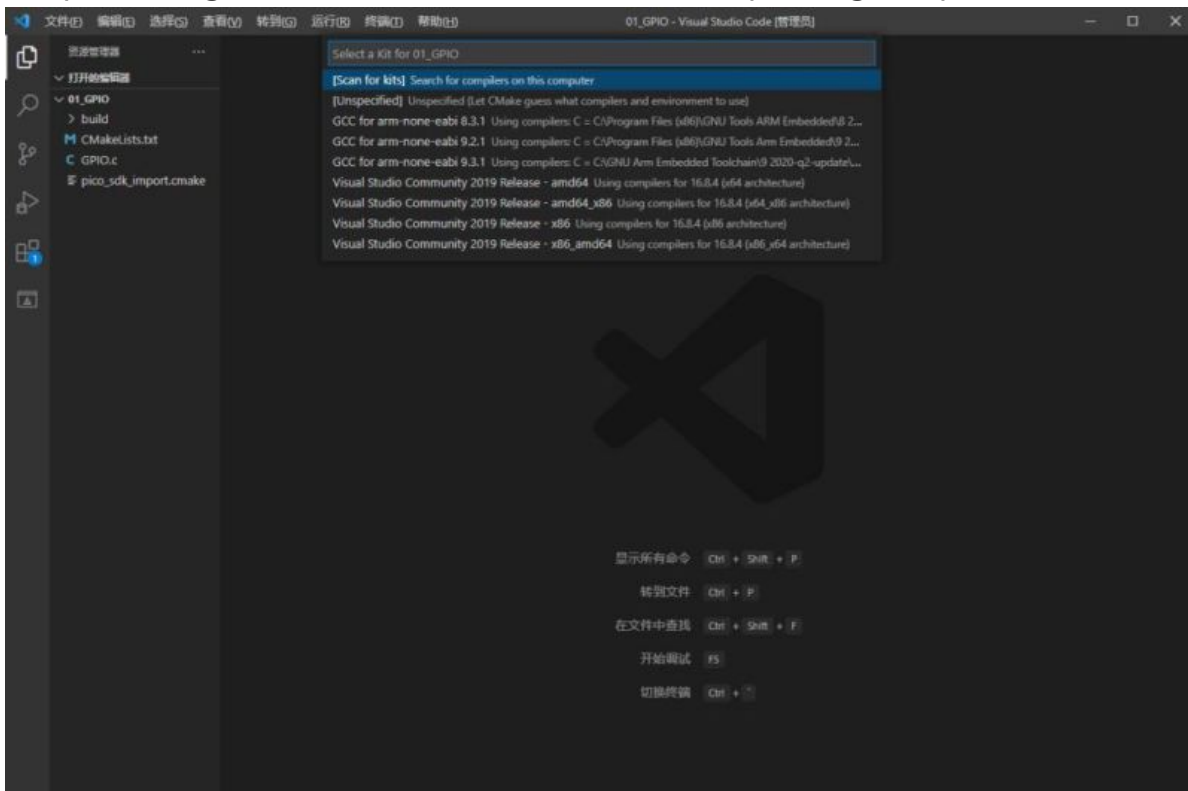
## How to Use the Demo

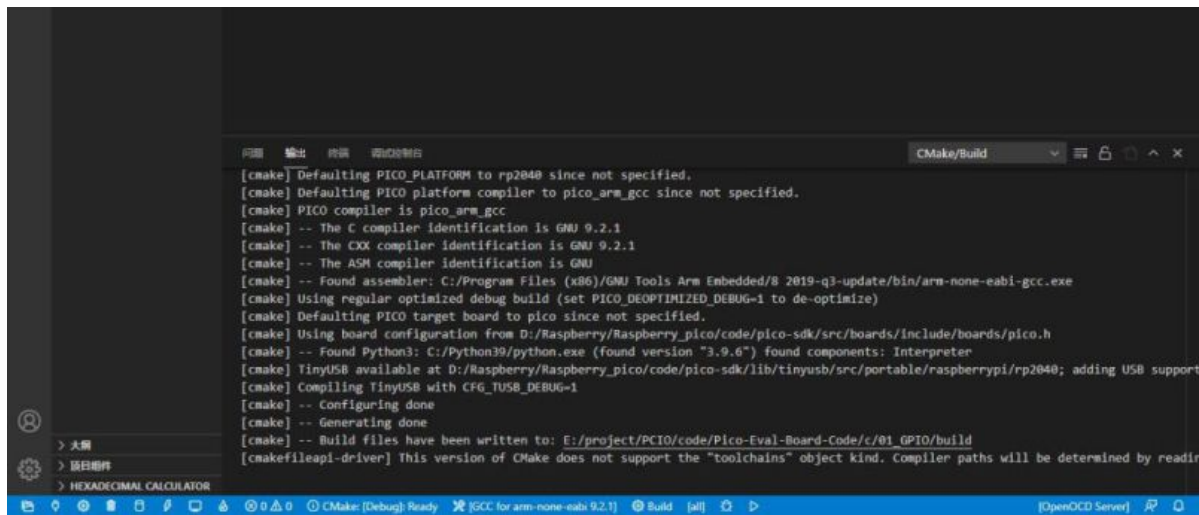
Note: The pictures below are for reference only, the steps are the same.

1. Open the corresponding C program folder.

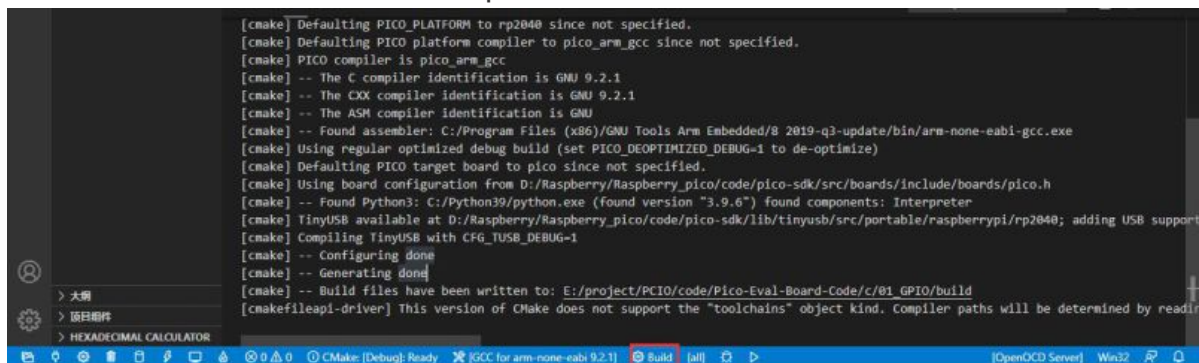


2. Open through the "Vs" coed and select the corresponding compilation tool.

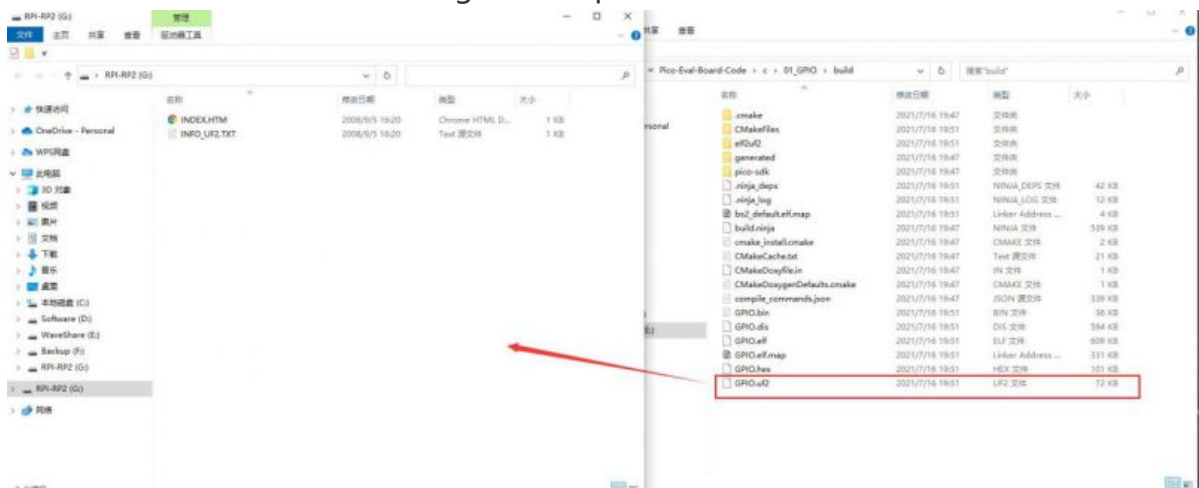




3. Click the "build" button to compile.



4. Press the "Reset" button on the Pico-Eval-Board to reset the Pico, first press the BOOTSEL button and then the RUN button and then release the Reset button, the Pico can enter the disk mode without plugging and unplugging the Pico, and the UF2 file under the build file Drag and drop to the RPI-RP2 drive letter.



5. By now, Pico starts to run the corresponding program.

## About the Demo

### Underlying hardware interface

- We have the underlying hardware package. Due to different hardware platforms, the internal implementation is different. If you need to know the internal implementation, you can refer to the corresponding directory.

You can see many definitions in DEV\_Config.c(h), in the directory: ...\\c\\lib\\Config.

- data type:

```
#define UBYTE uint8_t
#define UWORD uint16_t
#define UDOUBLE uint32_t
```

- Processing of module initialization and exit:

```
void DEV_Module_Init(void);
void DEV_Module_Exit(void);
```

- PWM initialization processing:

```
void PWM_initialization();
```

- Interrupt handler:

```
void on_pwm_wrap();
```

- Define the channel used:

```
#define CHANNE_N 0xFFFF // 0x0001 means 0 channel is open, 0x0000 means all channels are closed
```

- Rotation angle:

```
#define ROTATE_0 1700 //Rotate to 0° position
#define ROTATE_45 3300 //Rotate to 45° position
#define ROTATE_90 4940 //Rotate to 90° position
#define ROTATE_135 6600 //Rotate to 135° position
#define ROTATE_180 8250 //Rotate to 180° position
```

## Python

1. Press and hold the BOOTSET button on the Pico board, connect the pico to the USB port of the computer through the Micro USB cable, and release the button after the computer recognizes a removable hard disk (RPI-RP2).
2. Download the pico\_micropython\_xxxxx.uf2 file and copy it to the recognized removable disk (RPI-RP2)

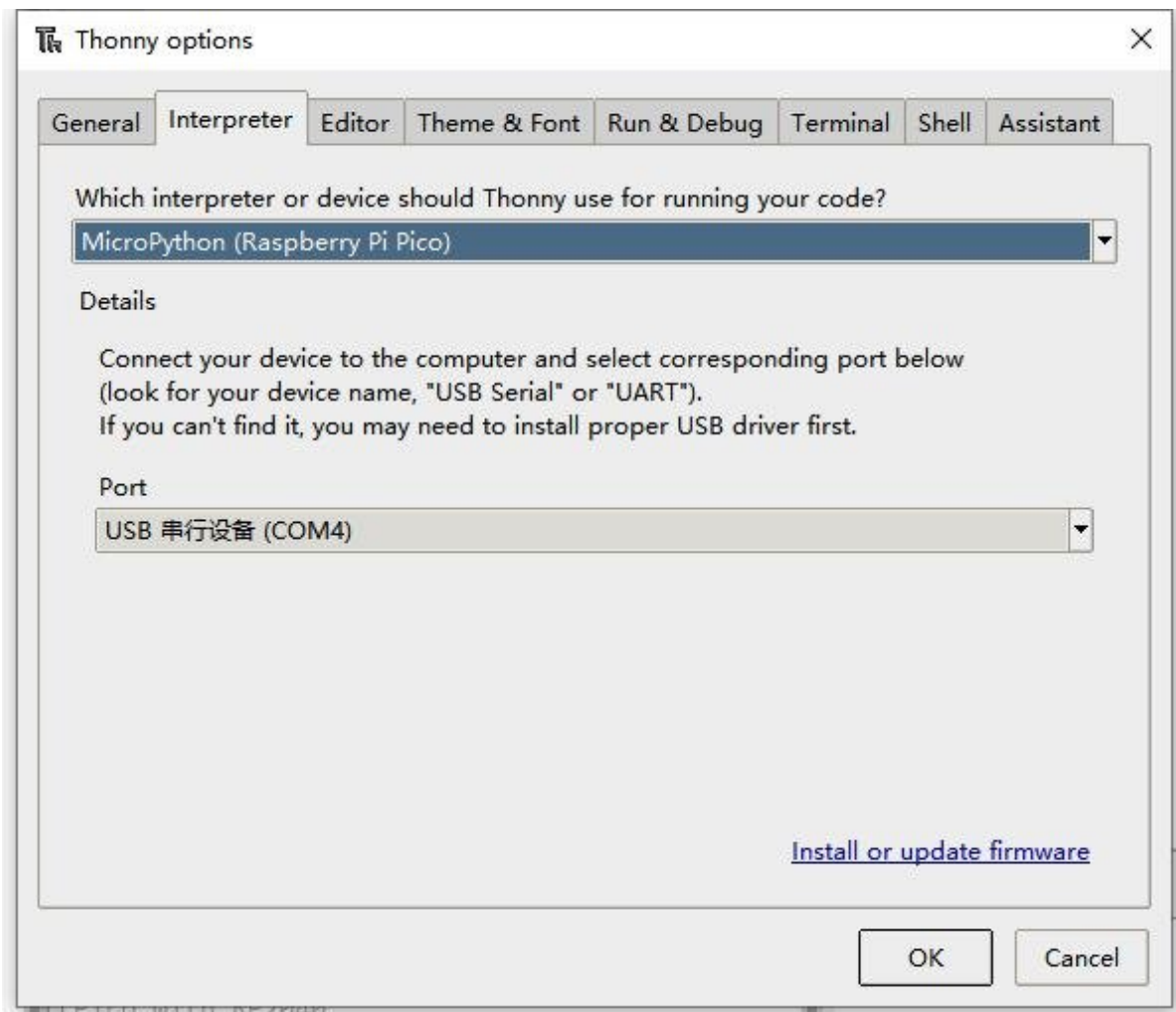
[official firmware download](#) .

3. Open Thonny IDE (Note: Use the latest version of Thonny, otherwise there is no



Pico support package, the latest version under Windows is v3.3.3).

4. Click Tools->Settings->Interpreter, select Pico and the corresponding port as shown in the figure.



5. File->Open->servo.py, click to run, as shown in the figure, the program has been run:



The experimental phenomenon is exactly the same as the c program, and will not be repeated here.

## Resource

### Document

- [Schematic](#)

### Demo codes

- [Demo codes](#)

## Development Software

- [Thonny Python IDE \(Windows V3.3.3\)](#)
- [Zimo221.7z](#)
- [Image2Lcd.7z](#)

## Pico Quick Start

### Download Firmware

---

- [MicroPython Firmware Download](#)
- [C\\_Blink Firmware Download](#) [\[Expand\]](#)

### Video Tutorial

---

- [Pico Tutorial I - Basic Introduction](#)
- [Pico Tutorial II - GPIO](#) [\[Expand\]](#)
- [Pico Tutorial III - PWM](#) [\[Expand\]](#)
- [Pico Tutorial IV - ADC](#) [\[Expand\]](#)
- [Pico Tutorial V - UART](#) [\[Expand\]](#)
- [Pico Tutorial VI - To be continued...](#) [\[Expand\]](#)

### MicroPython Series

---

- [【MicroPython】 machine.Pin Function](#)
- [【MicroPython】 machine.PWM Function](#)
- [【MicroPython】 machine.ADC Function](#)
- [【MicroPython】 machine.UART Function](#)
- [【MicroPython】 machine.I2C Function](#)
- [【MicroPython】 machine.SPI Function](#)
- [【MicroPython】 rp2.StateMachine](#)

### C/C++ Series

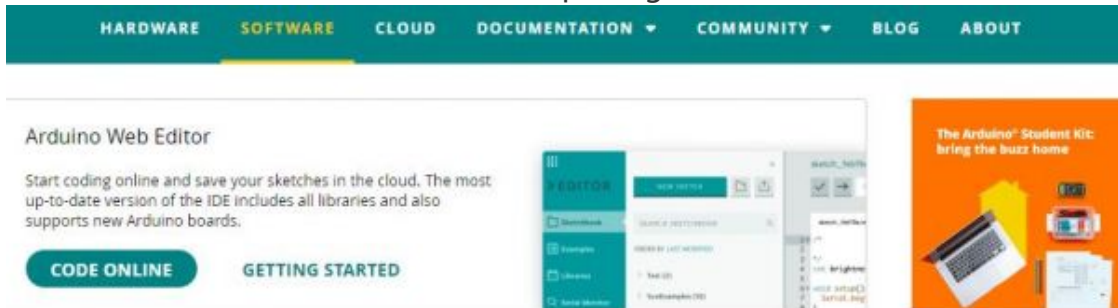
---

- [【C/C++】 Windows Tutorial 1 - Environment Setting](#)
- [【C/C++】 Windows Tutorial 1 - Create New Project](#)

# Arduino IDE Series

## Install Arduino IDE

1. Download the Arduino IDE installation package from [Arduino website](#).



## Downloads



2. Just click on "JUST DOWNLOAD".

## Support the Arduino IDE

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **69,954,557** times — impressive! Help its development with a donation.





Learn more about [donating to Arduino](#).

### 3. Click to install after downloading.



### 4. **Note: You will be prompted to install the driver during the installation process, we can click Install.**

## Install Arduino-Pico Core on Arduino IDE

### 1. Open Arduino IDE, click the File on the left corner and choose "Preferences".

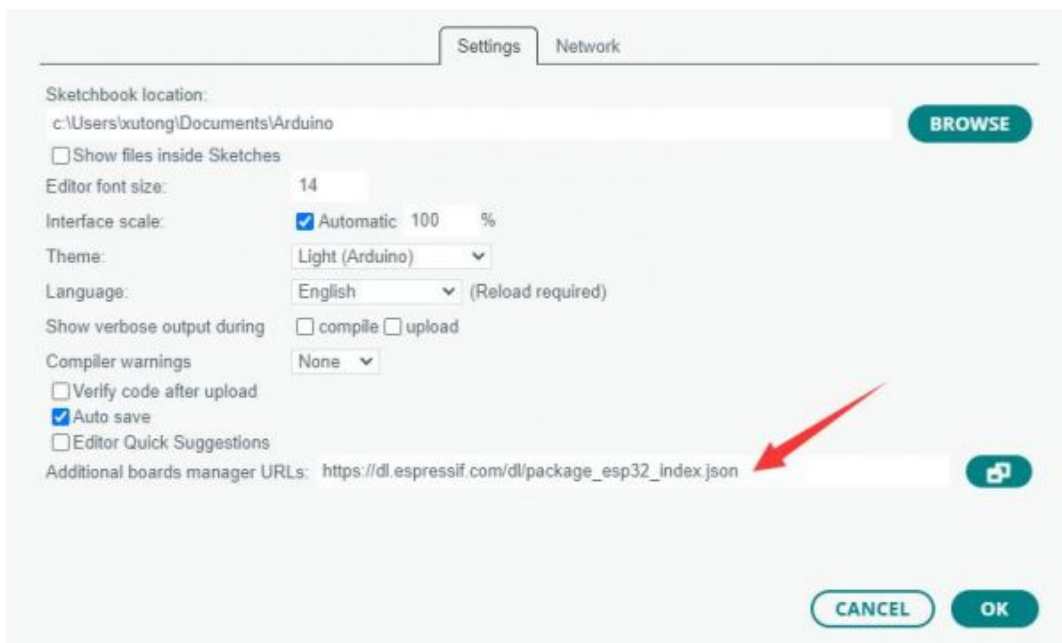


### 2. Add the following link in the additional development board manager URL, then click OK.

```
https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json
```

Preferences

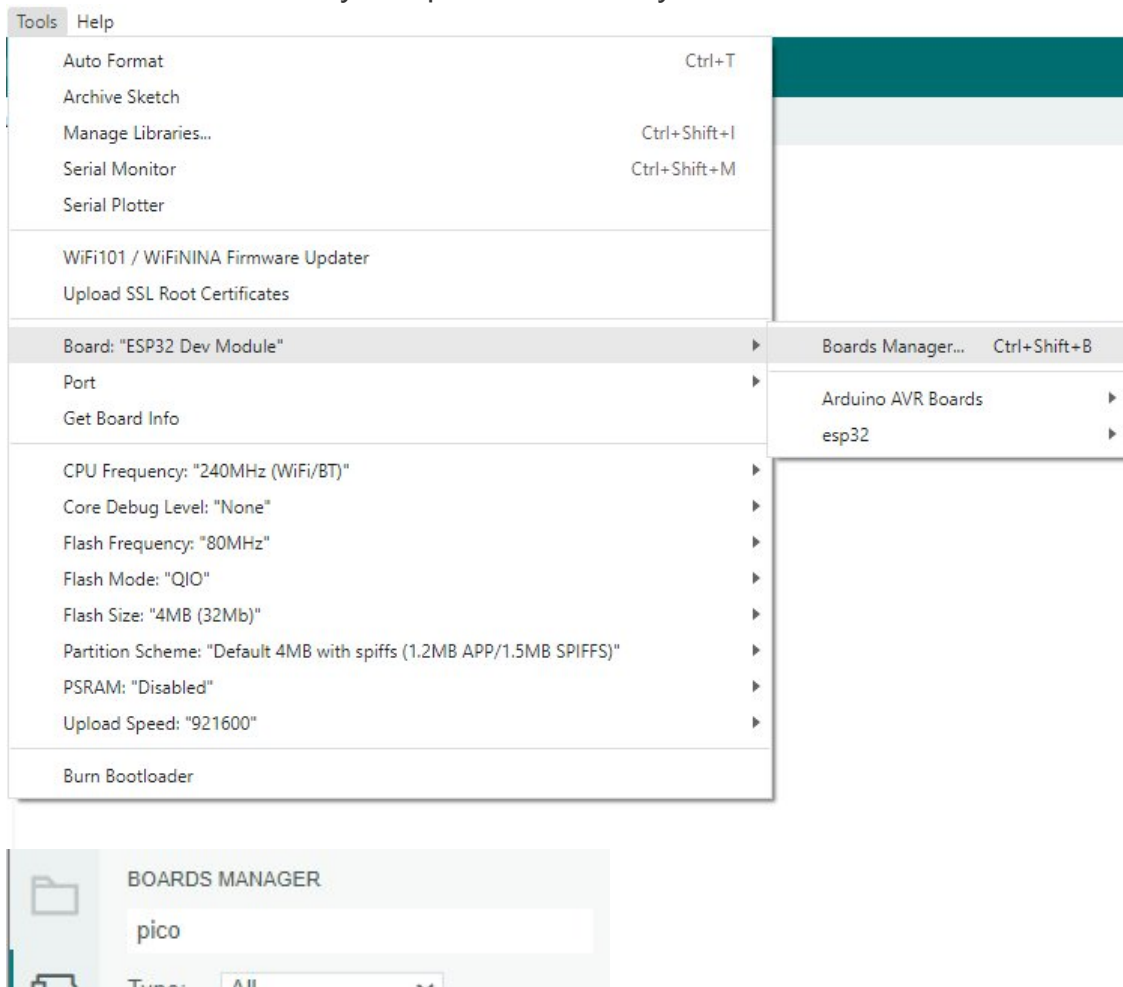


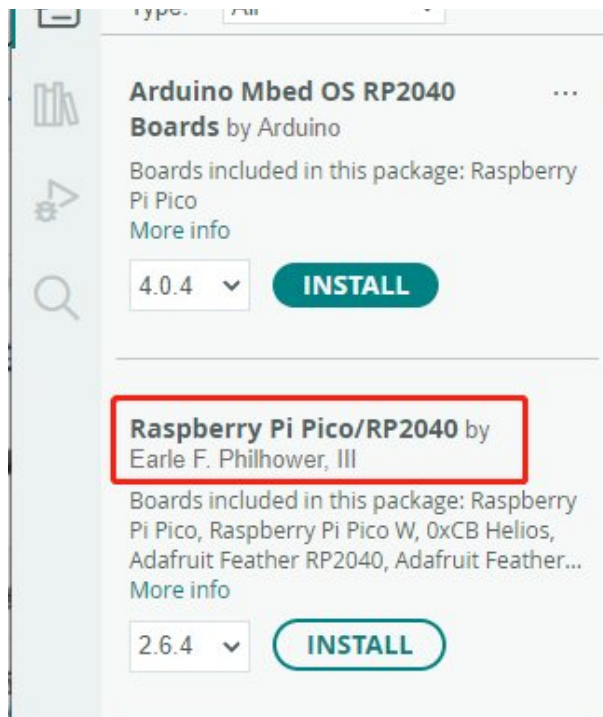


**Note: If you already have the ESP8266 board URL, you can separate the URLs with commas like this:**

```
https://dl.espressif.com/dl/package_esp32_index.json,https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json
```

3. Click on Tools -> Dev Board -> Dev Board Manager -> Search for pico, it shows installed since my computer has already installed it.



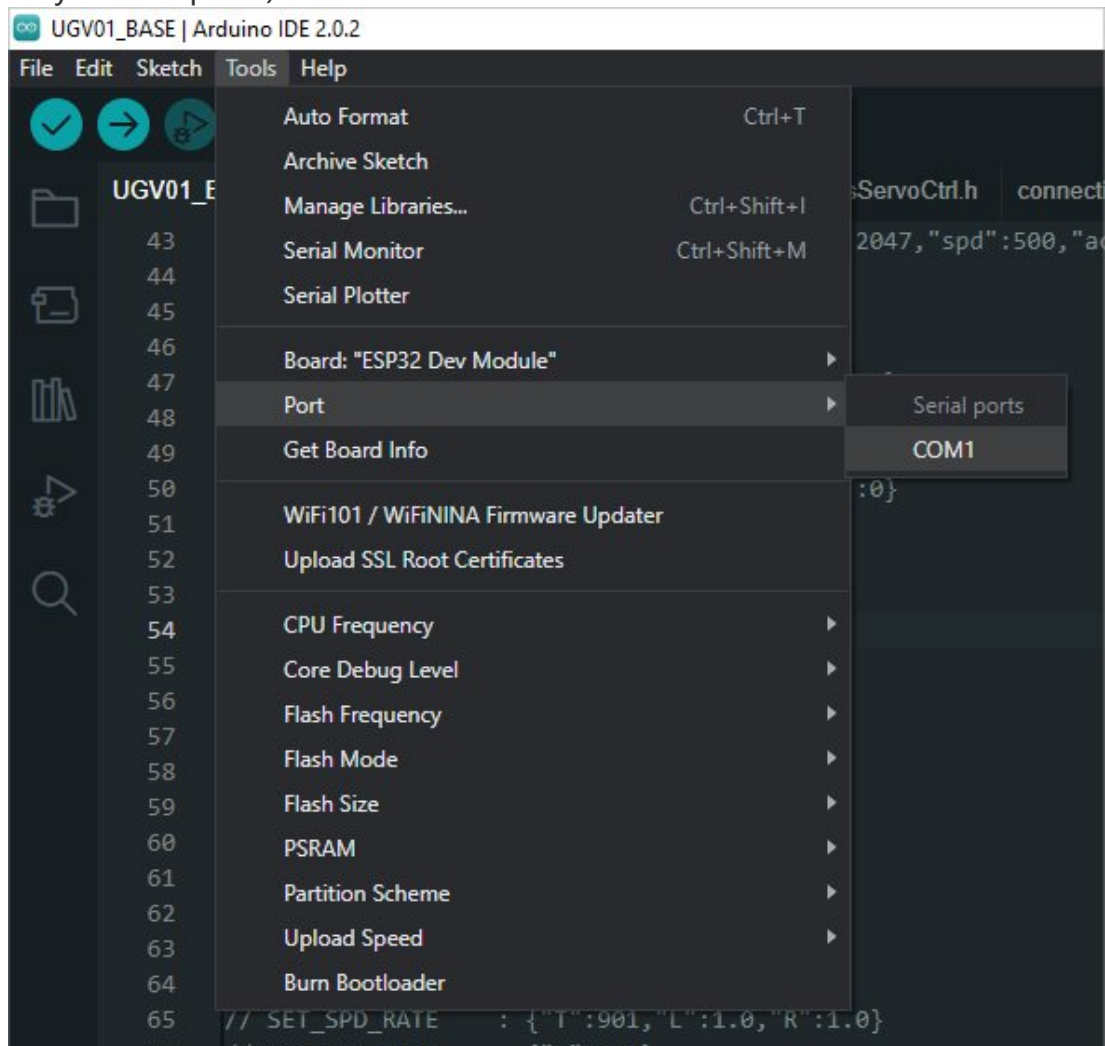


### Upload Demo At the First Time

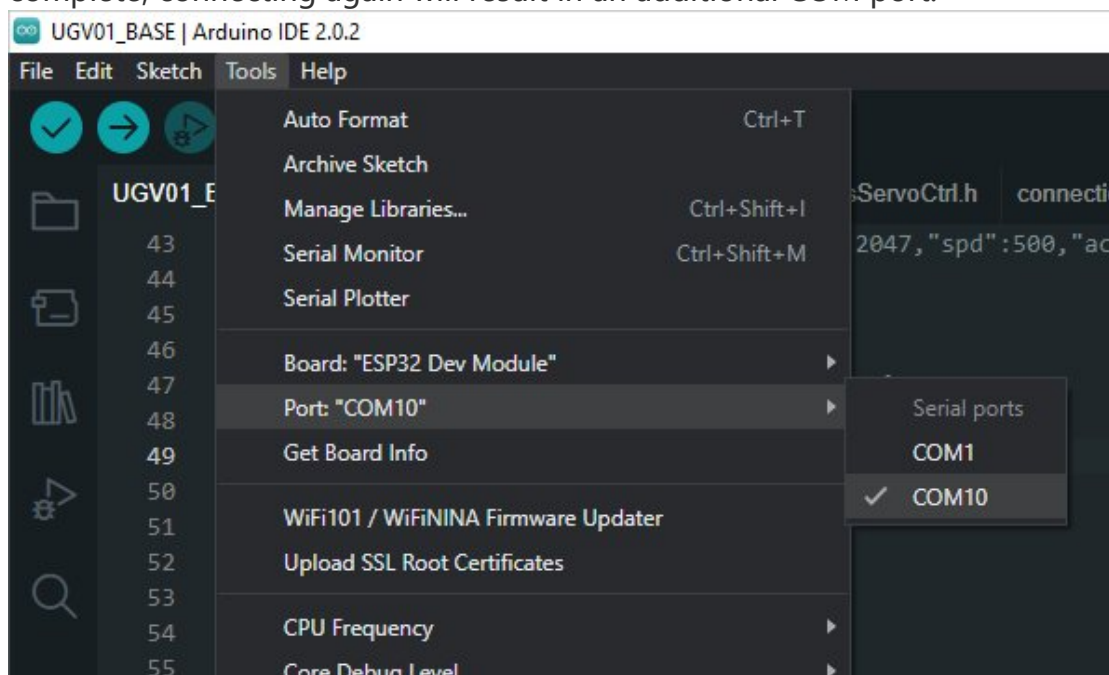
1. Press and hold the BOOTSET button on the Pico board, connect the Pico to the USB port of the computer via the Micro USB cable, and release the button when the computer recognizes a removable hard drive (RPI-RP2).

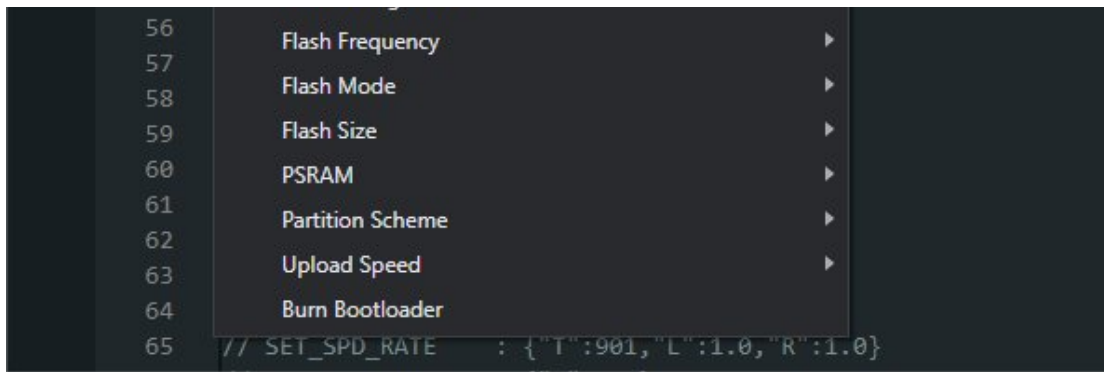


2. Download the demo, open arduino\PWM\D1-LED path under the D1-LED.ino.
3. Click Tools -> Port, remember the existing COM, do not need to click this COM (different computers show different COM, remember the existing COM on your computer).

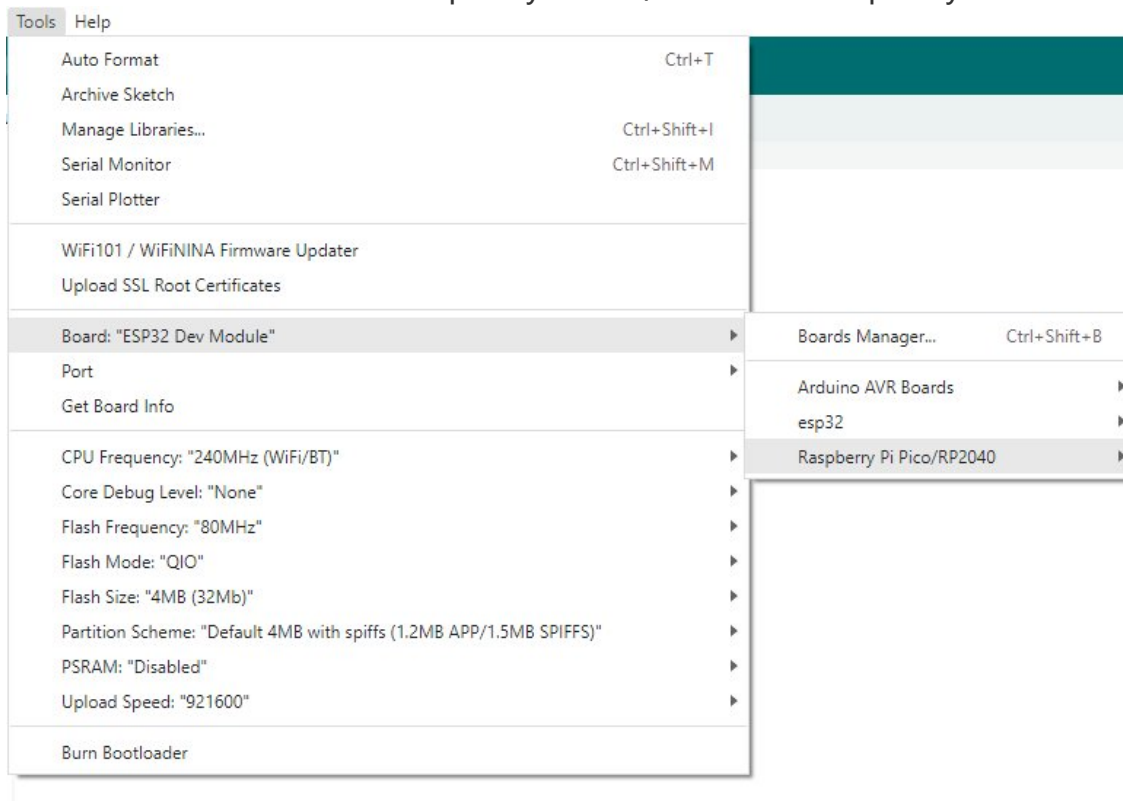


4. Connect the driver board to the computer with a USB cable, then click Tools -> Ports, select uf2 Board for the first connection, and after the upload is complete, connecting again will result in an additional COM port.

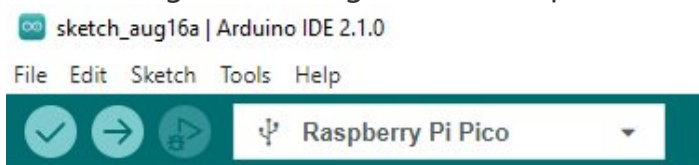




5. Click Tool -> Dev Board -> Raspberry Pi Pico/RP2040 -> Raspberry Pi Pico.



6. After setting, click the right arrow to upload.







- o If you encounter problems during the period, you need to reinstall or replace the Arduino IDE version, uninstall the Arduino IDE needs to be uninstalled cleanly, after uninstalling the software you need to manually delete all the contents of the folder C:\Users\[name]\AppData\Local\Arduino15 (you need to show the hidden files in order to see it) and then reinstall.

## Pico-W Series Tutorial (To be continued...)

## Open Source Demo

- [MicroPython Demo \(GitHub\)](#)



- [MicroPython Firmware/Blink Demo \(C\)](#) 
- [Official Raspberry Pi C/C++ Demo](#) 
- [Official Raspberry Pi MicroPython Demo](#) 
- [Arduino Official C/C++ Demo](#) 

## Support

### Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.

Working Time: 9 AM - 6 AM GMT+8 (Monday to Friday)

[Submit Now](#)