

## Overview

This is a 2-Channel RS485 module for Raspberry Pi Pico, SP3485 transceiver, UART To RS485.

## Features

- Standard Raspberry Pi Pico header supports Raspberry Pi Pico series.
- Dual-channel design, each channel leads to two interfaces, which is convenient for users to choose.
- On-board TVS (transient voltage suppression tube), RS485 communication can effectively suppress the surge voltage and transient peak voltage in the circuit, lightning protection and anti-static.
- Provide a complete supporting information manual (example programs such as Raspberry Pi Pico C/C++ and MicroPython).

## Specification

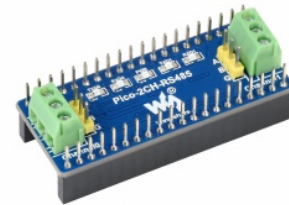
- Operating voltage: 3.3V
- Communication bus: UART
- RS485 transceiver: SP3485
- Baudrate: 300~500000bps
- Dimension: 52.5mm × 21mm

## Pinout

GP0	1	40	VBUS
GP1	2	39	VSYS
GND	3	38	GND
GP2	4	37	3V3_EN
GP3	5	36	3V3(OUT)
GP4	6	35	ADC_VREF
GP5	7	34	GP28
GND	8	33	GND
GP6	9	32	GP27
GP7	10	31	GP26
GP8	11	30	RUN
GP9	12	29	GP22
GND	13	28	GND
GP10	14	27	GP21
GP11	15	26	GP20
GP12	16	25	GP19

3V3(OUT)	Power supply	
GND	Ground	
GP0	TXD0	UART TX
GP1	RXD0	UART RX
GP4	TXD1	UART TX
GP5	RXD1	UART RX

### Pico-2CH-RS485



2-Channel RS485 Module for Raspberry Pi Pico, SP3485 Transceiver, UART To RS485

GP13	17	24	GP18
GND	18	23	GND
GP14	19	22	GP17
GP15	20	21	GP16

## Hardware connection

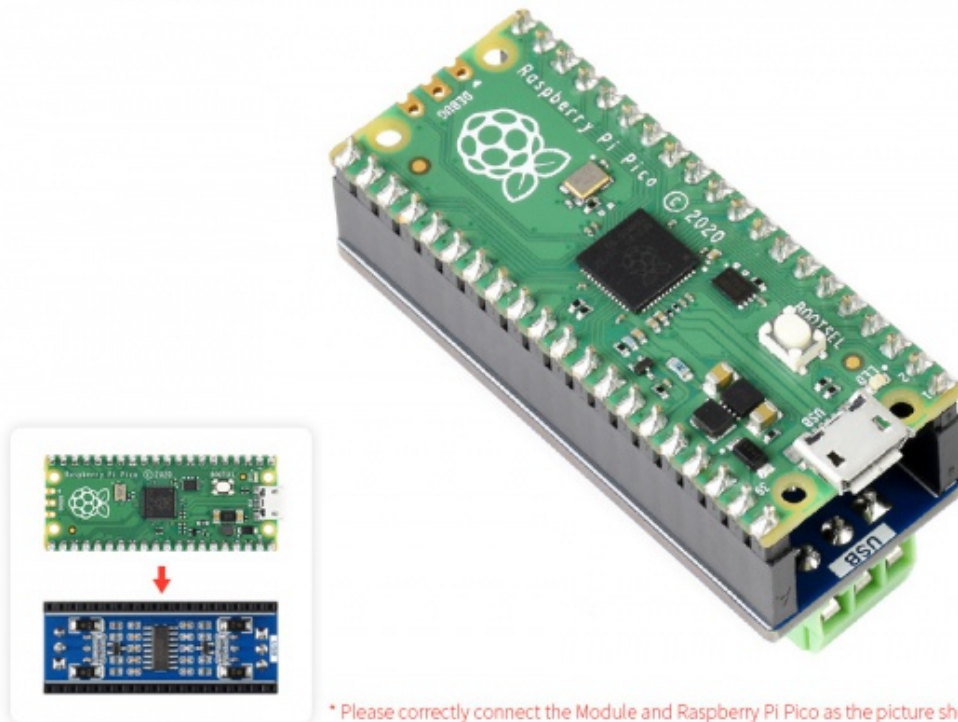
When connecting PICO, please pay attention not to connect in the opposite direction. The direction can be judged by observing the USB screen printing end of the module and the USB interface of the Pico (also can be judged by the pin label of the platoon master on the module and the pin label of the Pico).

You can also wire it according to the table below:

RS485	Pico	Description
VCC	VSYS	Power input
GND	GND	Ground
TX_CH0	GP0	UART transmit pin
RX_CH0	GP1	UART receive pin
TX_CH1	GP4	UART transmit pin
RX_CH1	GP5	UART receive pin

## Connection

The connection of the RS485 board is same as the Pico-2CH-RS232.



## Setup environment

To setup the Pico for working, please refer to the official guide:

<https://www.raspberrypi.org/documentation/pico/getting-started/>

## Download example

Open a terminal and run the follow command

```
sudo apt-get install p7zip-full
cd ~
sudo wget https://files.waveshare.com/upload/9/94/Pico_2CH_RS485.7z
7z x Pico_2CH_RS485.7z -o./Pico_2CH_RS485
cd ~/Pico_2CH_RS485
cd c/build/
```

## Run the examples

### C codes

---

- Go the example directory and build it.

```
cd ~/Pico_2CH_RS485/c/
```

Create the build folder and add SDK path to it:

By default, the ../../pico-sdk is the path of SDK.

We have created the build folder in examples, you can just enter it.

Note: If the actual path of your Pi is different, you need to write the correct path.

```
cd build
export PICO_SDK_PATH=../../pico-sdk
```

Run the cmake to generate Makefile file

```
cmake ..
```

Run the make command to build the codes and generate an executable file.

```
make -j9
```

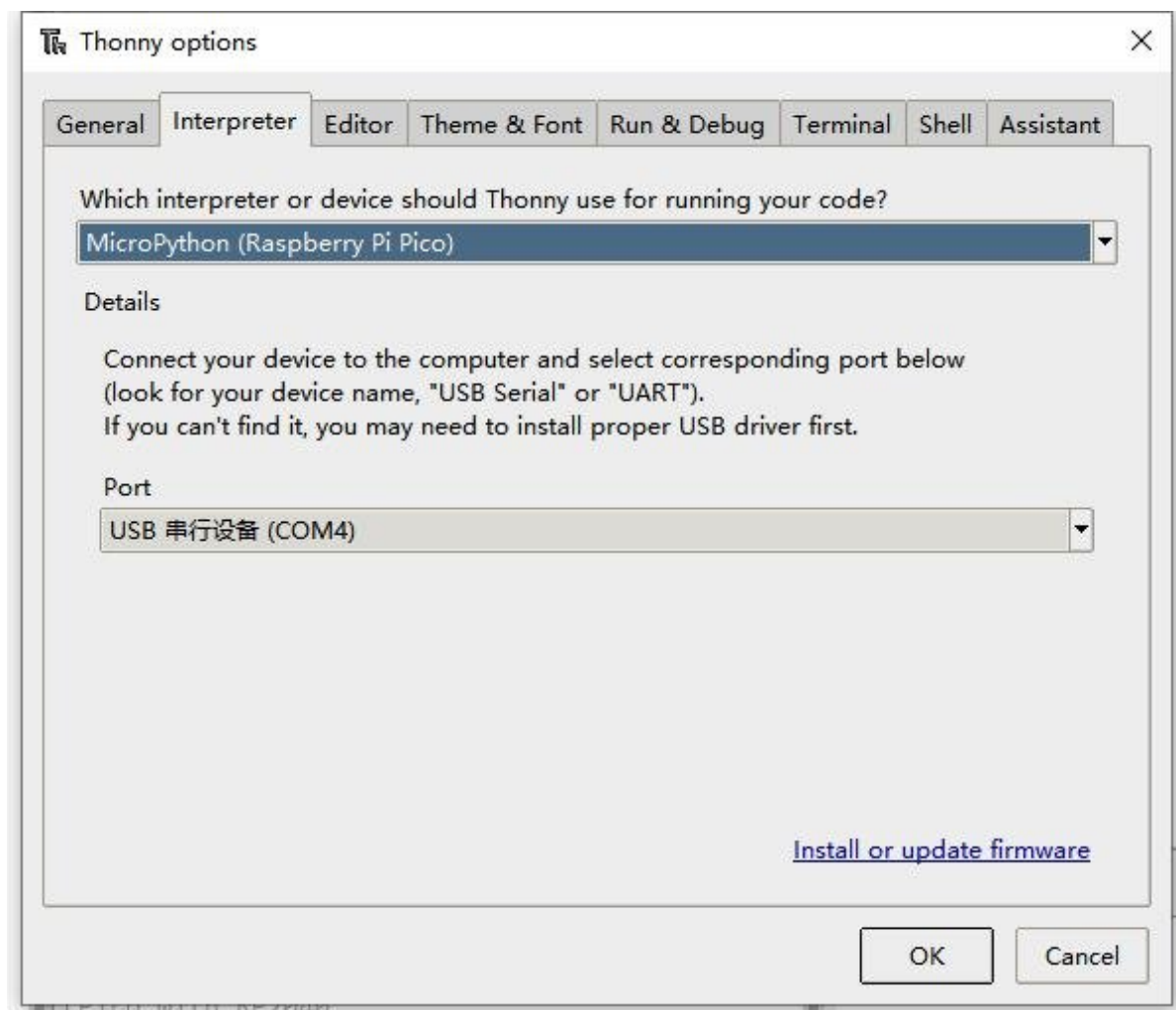
After building, an uf2 file is generated. Press and hold the key on the Pico board to connect the Pico to the Raspberry Pi's USB port via the Micro USB cable, then release the key. Once connected, Raspberry will automatically recognize a removable disk (RPI-RP2). Copy the file main.uf2 in the corresponding folder to the recognized removable disk (RPI-RP2).

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

## Python codes

### Use in Windows

- 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. Copy the rp2-pico-20210418-v1.15.uf2 file in the python directory to the recognized removable disk (RPI-RP2).
- 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 -> the corresponding .py file, click to run, as shown in the following figure:

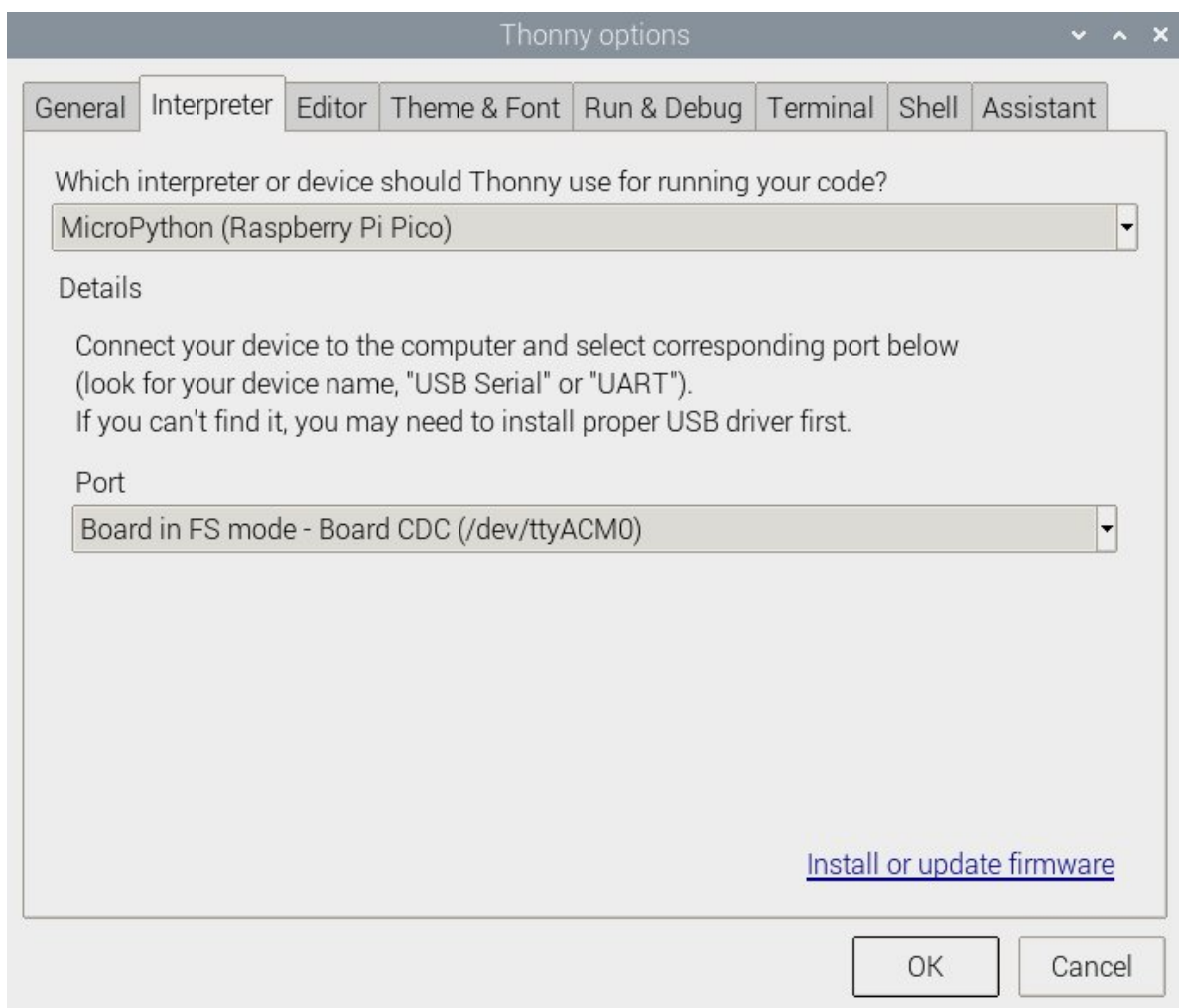
```
MicroPython v1.15 on 2021-04-18; Raspberry Pi Pico with RP2040
```

```
PI PICO WITH RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
```

This demo provides a simple program...

### Run in Raspberry Pi

- Hold the BOOTSET key of the Pico board, then connect the Pico to Raspberry Pi by USB cable, then release the key.
- Once the removable disk (RPI-RPI2) is recognized, copy the rp2-pico-20210418-v1.15.uf2 file to pico.
- Open the Thonny IDE in Raspberry Pi, update it if it doesn't support Pico.
- Configure the port by choosing MicroPython(Raspberry Pi and ttyACM0 port) in Tools -> Options... -> Interpreter.



If your Thonny doesn't support Pico, you can update it with the following command:

```
sudo apt upgrade thonny
```

- Choose File->Open...->python/ and select the corresponding .py file to run the codes.

## Test Demo Function

It can achieve self-receiving and sending functions.

## Resource

### Documents

- [Schematic](#)
- [SP2485 Datasheet](#)

### Demo Codes

- [Demo code](#)

### 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

---

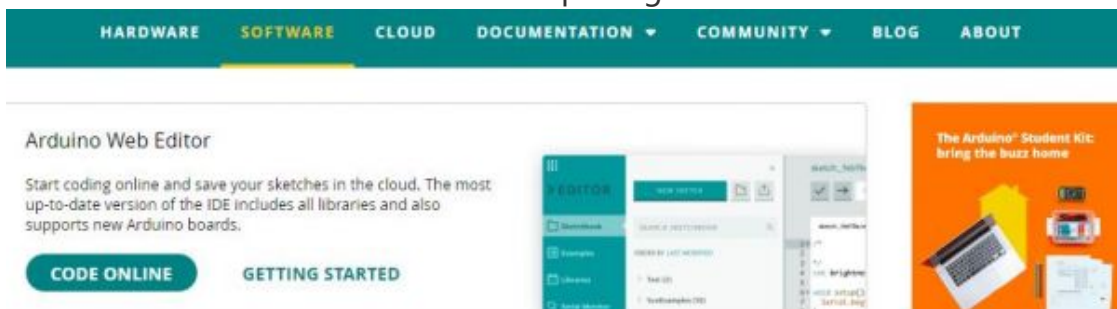
- [【C/C++】 Windows Tutorial 1 - Environment Setting](#)
- [【C/C++】 Windows Tutorial 1 - Create New Project](#)

## Arduino IDE Series

---

### Install Arduino IDE

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



## Downloads



### Arduino IDE 2.0.0

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE  
The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

#### DOWNLOAD OPTIONS

- Windows** Win 10 and newer, 64 bits
- Windows** MSI installer
- Windows** ZIP file
- Linux** AppImage 64 bits (X86-64)
- Linux** ZIP file 64 bits (X86-64)
- macOS** 10.14: "Mojave" or newer, 64 bits

2. Just click on "JUST DOWNLOAD".

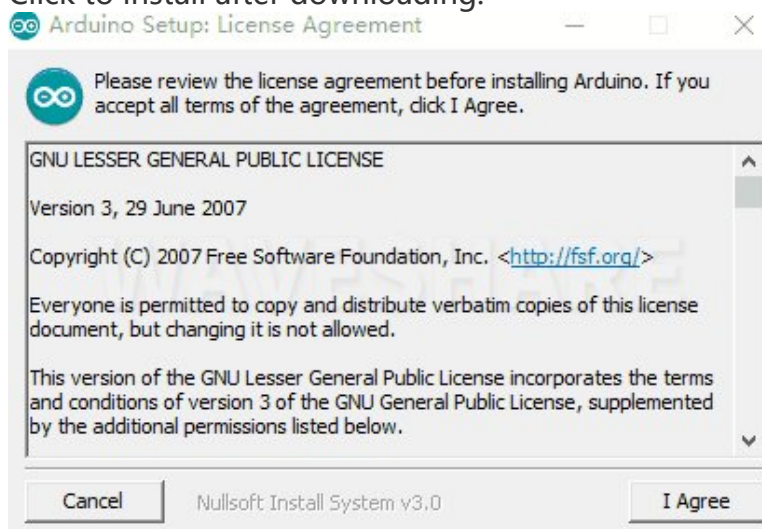
## Support the Arduino IDE

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".

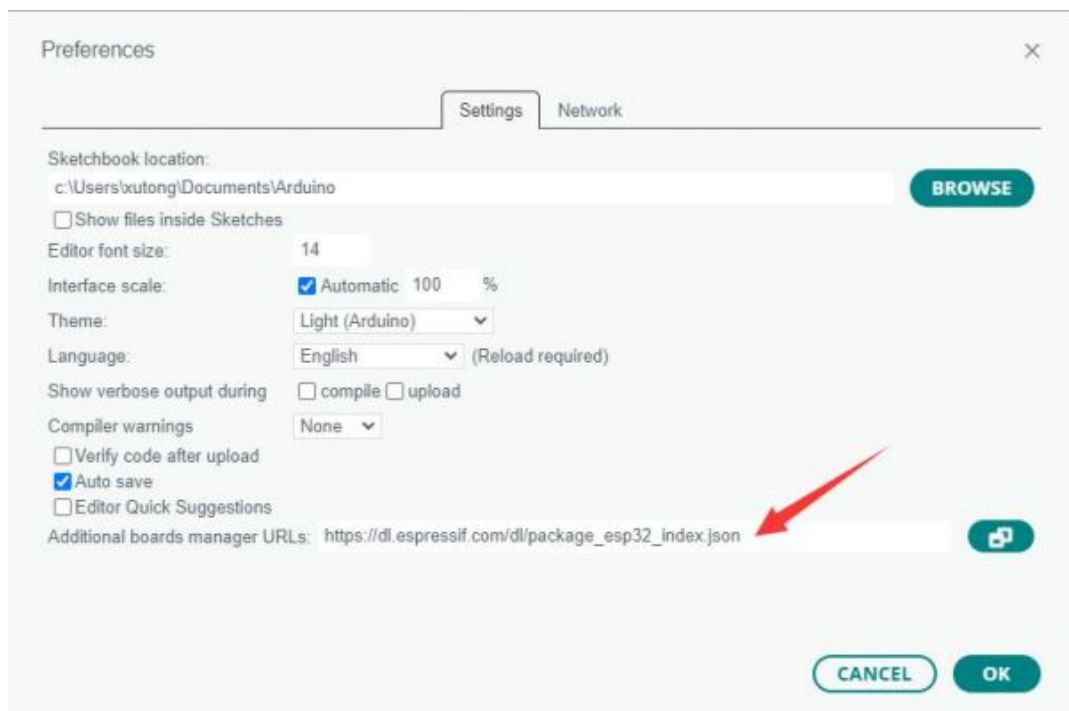






2. Add the following link in the additional development board manager URL, then click OK.

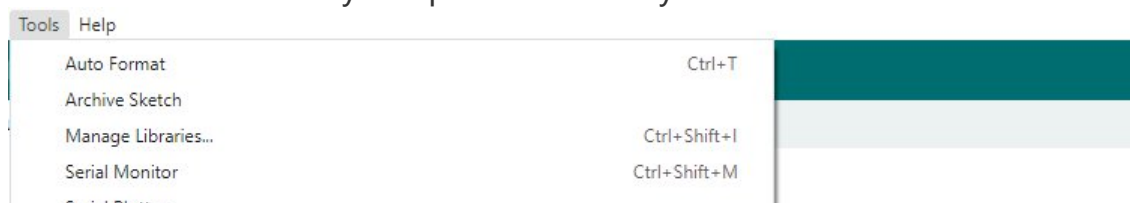
```
https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json
```

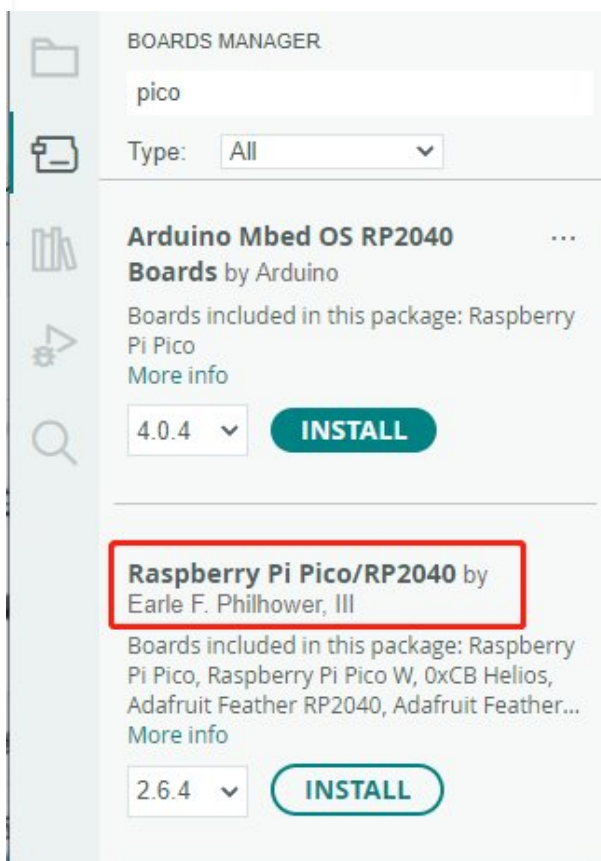
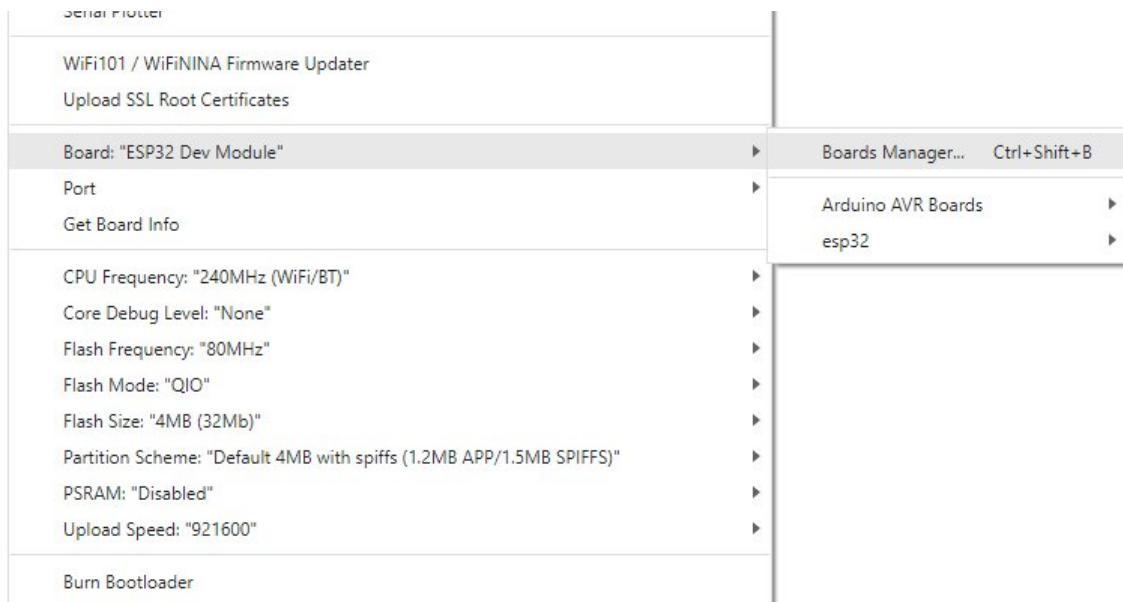


**Note: If you already have the ESP8266 board URL, you can separate the URLs with commas like this:**

```
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.





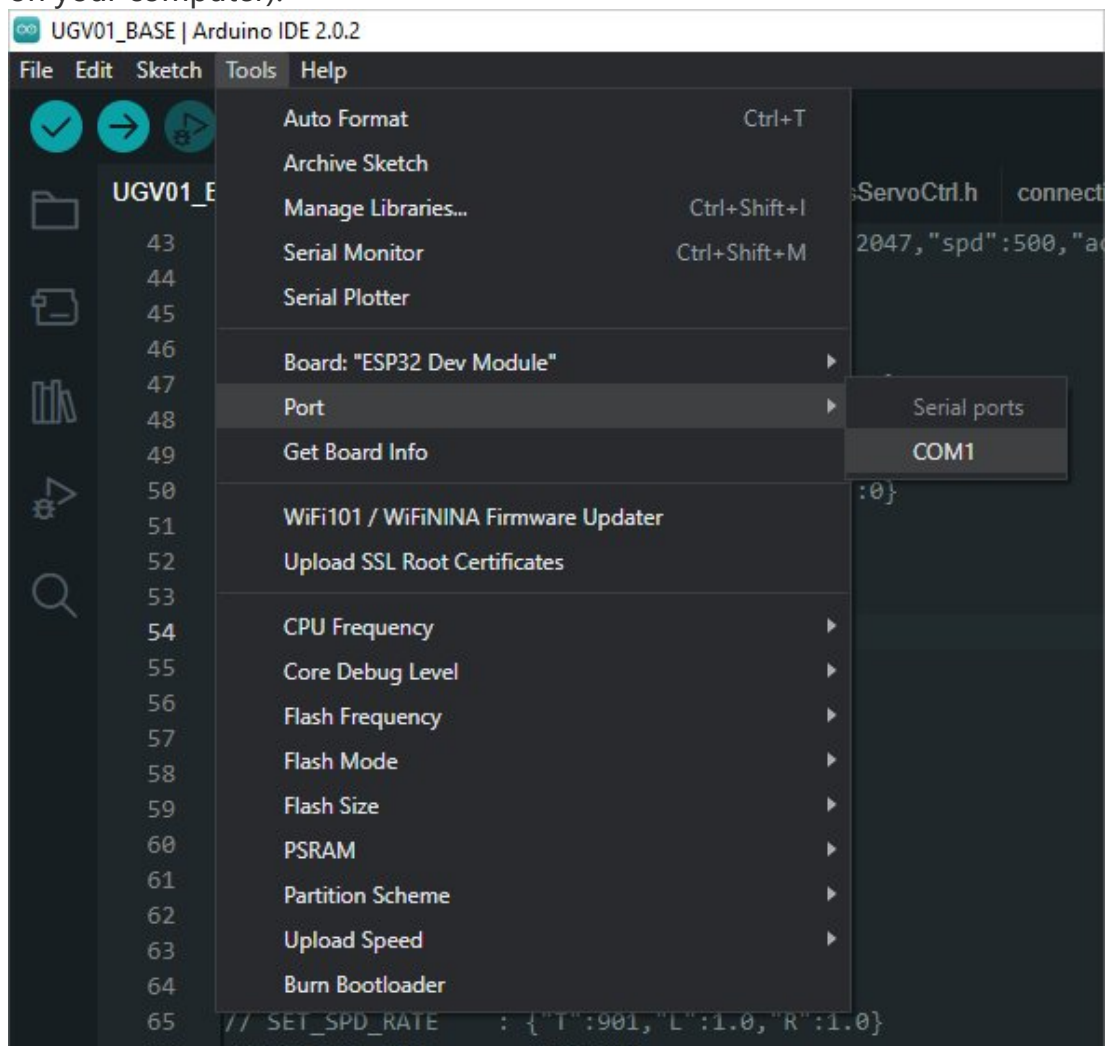
## Upload Demo At the First Time

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).

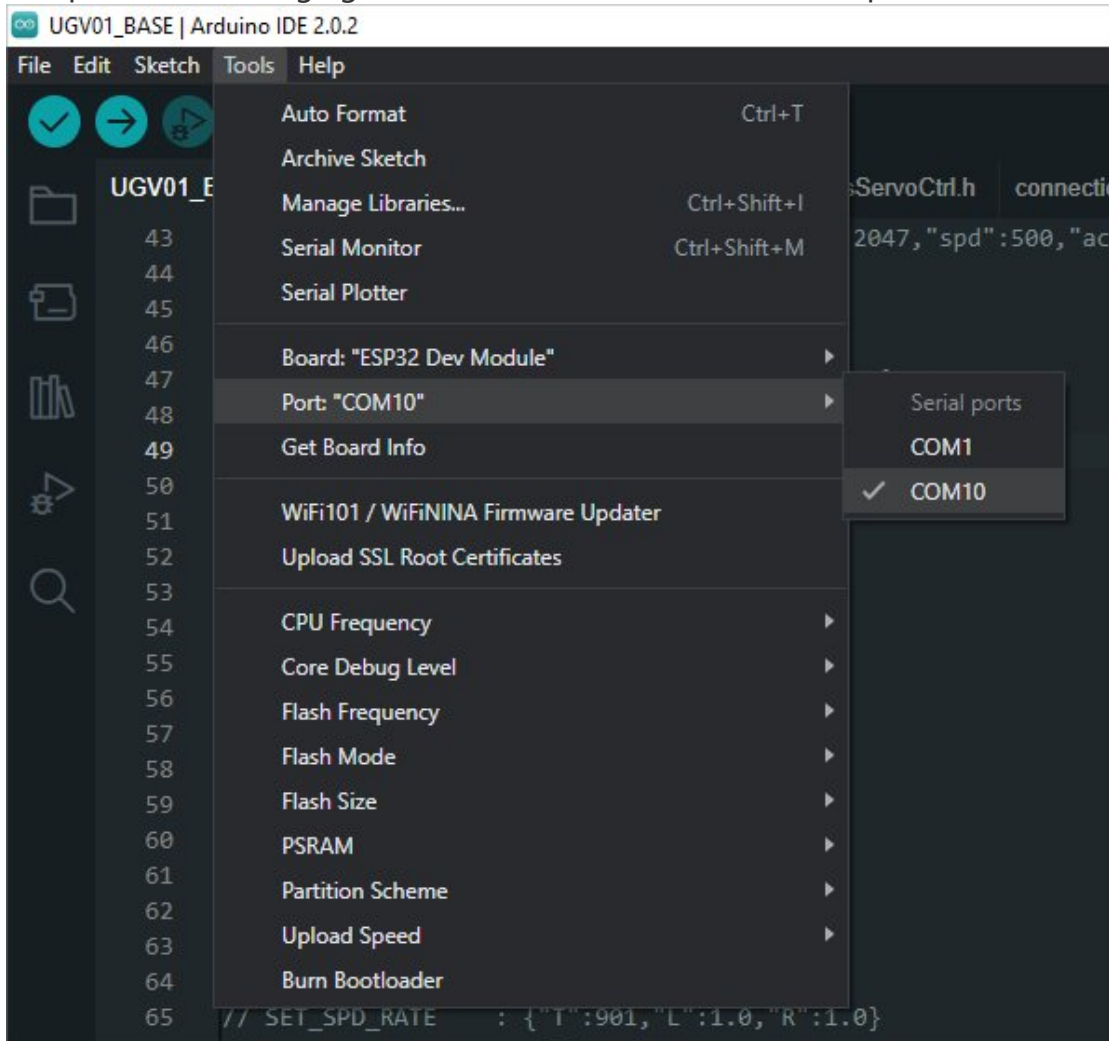




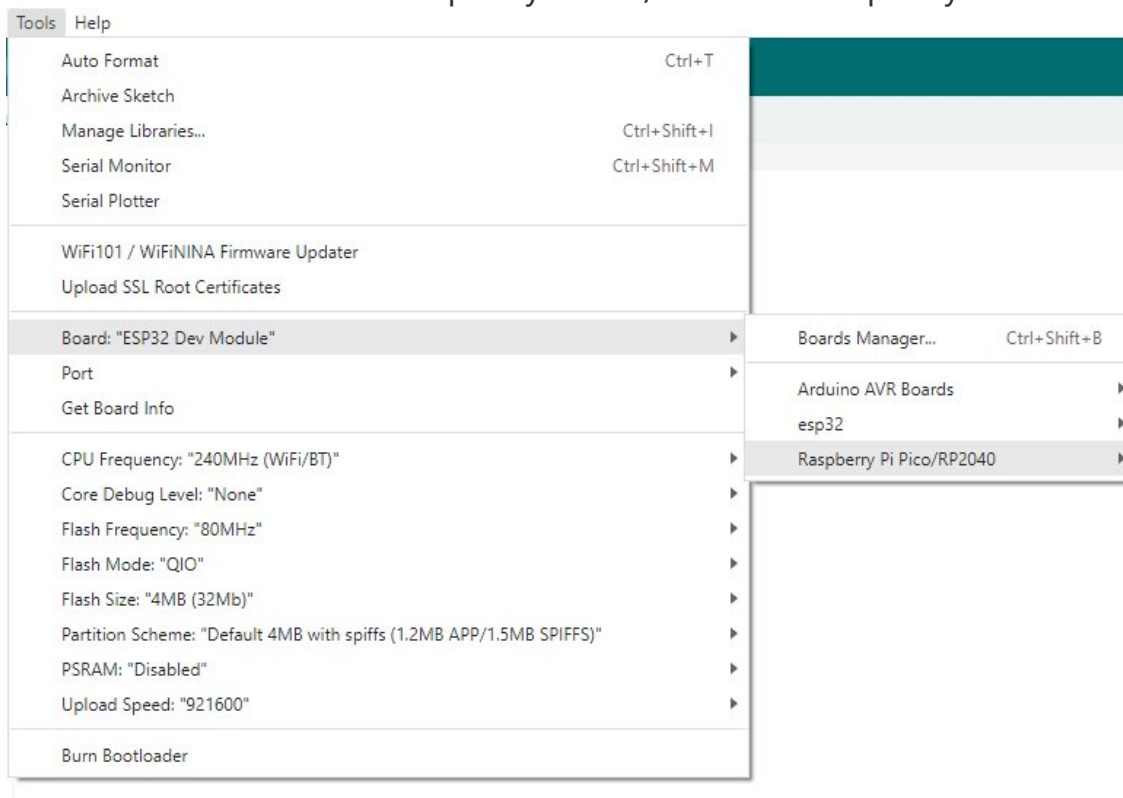
2. Download the demo, open arduino\PWM\D1-LED path under the D1-LED.ino.
3. Click Tools -> Port, remember the existing COM, do not need to click this COM (different computers show different COM, remember the existing COM on your computer).



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.



- 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](#)