

Overview

Provide Pico C demo.

Introduction

The Pico-Audio is an audio expansion module designed for Raspberry Pi Pico, which incorporates PCM5101A low power stereo decoder, and uses I2S bus for transmitting the audio signals. It supports a wide range of audio sampling rates, and allows headphone and speaker signal output at the same time.

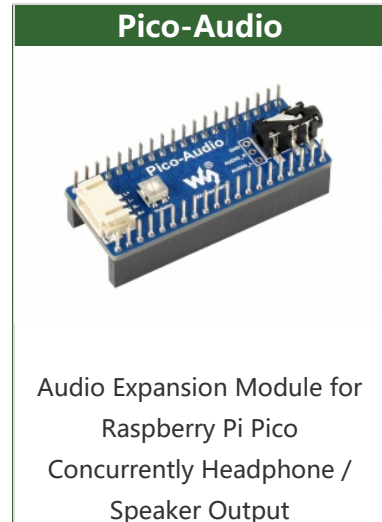
Rev2.1 version: This product is an audio module designed based on Pico. It adopts a CS4344 low-power stereo decoder, an I2S interface to transmit audio, and supports a wide range of audio sampling frequencies. It also can be output by headphones and speakers at the same time.

Features

- Support audio sampling frequency of 8~384,000Hz.
- Using 3-wire I2S, effectively reducing EMI.
- With undervoltage protection and a double mute system, it can automatically mute (stop output) when there is a clock error or power failure.
- Provides 20dB of out-of-band noise.
- Onboard standard 3.5mm headphone and 4-wire speaker output interface, support audio output.
- Provides Pico sound card, driver-free driver, compatible with Windows, Linux and other systems.
- Pico audio player provided.

Rev2.1:

- Support audio sampling frequency of 2~200KHz.
- The chip has a built-in PLL, and the audio clock pin can obtain the main clock screen and automatically reduce the frequency.
- Onboard standard 3.5mm headphone and 4-wire speaker output interface,



support audio output.

- Provides Pico sound card driver-free program, compatible with Windows, Linux and other systems.
- Pico audio player provided.

Specification

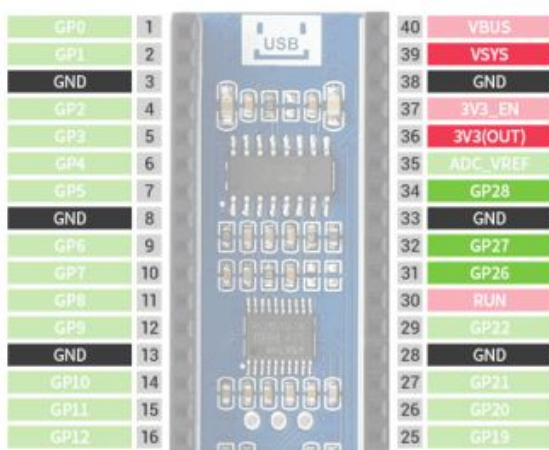
- Operating voltage: 5V
- Logic voltage: 3.3V
- Audio Decoder: PCM5101A
- DAC SNR: 106 dB
- Logic level: 3.3V
- Audio bus: I2S
- Speaker driving: 2.6W/channel (4Ω BTL)
- Earphone driving: 40W per channel (16@Ω 3.3V)

Rec2.1

- Operating voltage: 5V
- Logic voltage: 3.3V
- Audio Decoder: CS4344
- DAC SNR: 105 dB
- Logic level: 3.3V
- Audio bus: I2S
- Speaker driving: 2.6W/channel (4Ω BTL)
- Earphone driving: 40W per channel (16@Ω 3.3V)

Hardware Resource

Initial Version



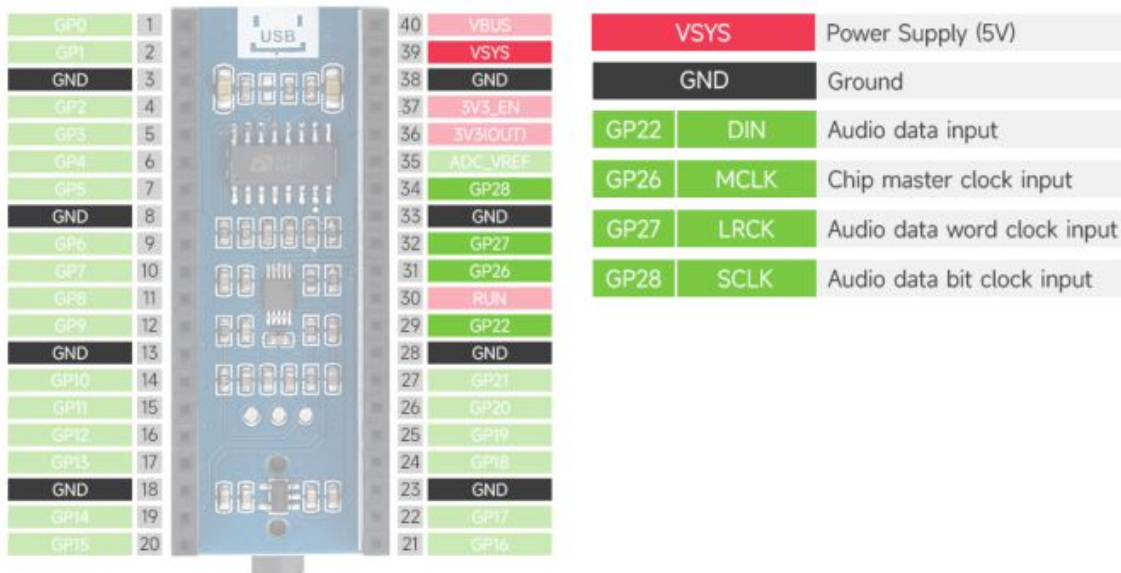
GP0	1	40	VBUS	VSYS	Power supply	
GP1	2	39	VSYS	3V3(OUT)	Logic power	
GND	3	38	GND	GND	Ground	
GP2	4	37	3V3_EN	GP26	DIN	Audio data input
GP3	5	36	3V3(OUT)	GP27	BCK	Audio data bit clock input
GP4	6	35	ADC_VREF	GP28	LRCK	Audio data word clock input
GP5	7	34	GP28			
GND	8	33	GND			
GP6	9	32	GP27			
GP7	10	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			



L+ and L- correspond to the positive and negative poles of the left speaker respectively; R+ and R- correspond to the positive and negative poles of the right speaker respectively.

Pin Function	Pico Pins (BCM)	Description
VSYS	VSYS	Power supply (5V power input)
3V3(OUT)	3V3(OUT)	Logic Power (3.3V Power Input)
GND	GND	Power Ground
DIN	GPIO26	Audio data input
BCK	GPIO27	Audio data bit clock input
LRCK	GPIO28	Audio data word clock input

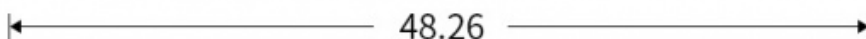
Rev2.1 Version

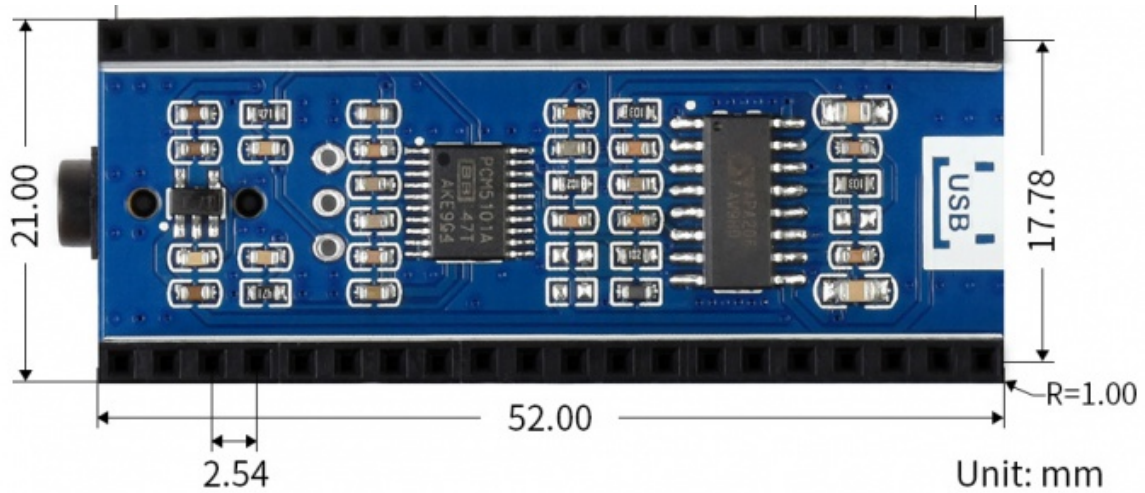


L+ and L- correspond to the positive and negative poles of the left speaker respectively; R+ and R- correspond to the positive and negative poles of the right speaker respectively.

Function Pin	Pico Pin (BCM)	Description
VSYS	VSYS	Power supply (5V)
GND	GND	Ground
DIN	GPIO22	Audio data input
MCLK	GPIO26	Chip main clock input
LRCK	GPIO27	Audio data word clock input
SCLK	GPIO28	Audio data bit clock input

Dimension





I2S



The most important thing to note about the I2S protocol is that it takes the second clock after the change of the sampling pulse as the first bit of data, and the first clock after the change of the following sampling pulse as the last bit of data.

LRCK: Audio data word clock input, when it is low level for the left channel data acquisition when it is a high level for the right channel data acquisition.

BCK: Audio data word clock input, when it produces a rising edge jump to collect data, the frequency must be 32 or 48 or 64 times the frequency of LRCK.

DATA: Audio data input, note that the data is signed data, data transmission can choose 16-bit, 24-bit, 32-bit data, high in the front, low in the back.

Get Started with Pico

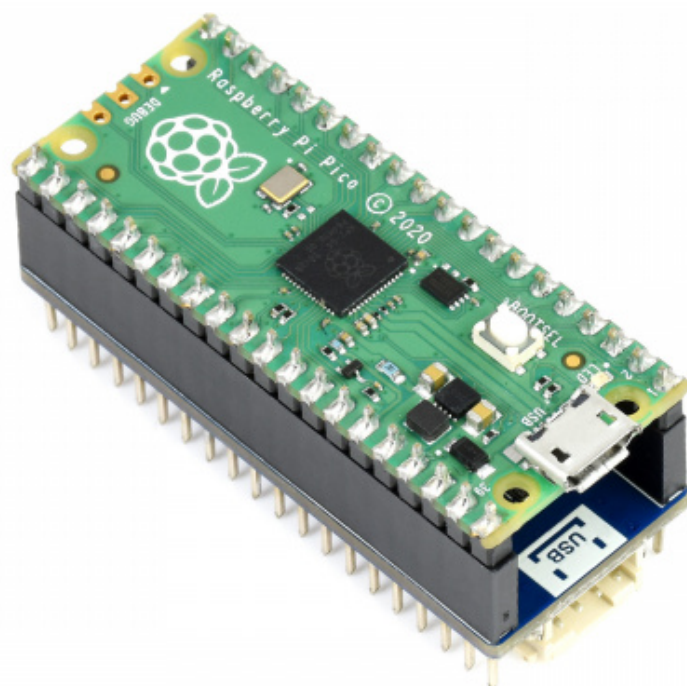
Hardware Connection

Please take care of the direction when connecting the board to Pico according to the USB port.

You can also wire it according to the table:

PCM5101A	Pico	Description
VSYS	VSYS	Power input (5V)
3V3(OUT)	3V3(OUT)	Logic level (3.3V)
GND	GND	Ground
DIN	GPIO26	Audio data input
BCK	GPIO27	Audio data bit clock input
LRCK	GPIO28	Audio data word clock input
Pico-Audio Rev2.1	Pico	Description
VSYS	VSYS	Power supply (5V)
GND	GND	Ground

DIN	GPIO22	Audio data input
MCLK	GPIO26	Chip main clock input
LRCK	GPIO27	Audio data word input
SCLK	GPIO28	Audio data bit input



Download Examples

Open a terminal of Raspberry Pi and run the following commands:

```
sudo apt-get install p7zip-full
cd ~
sudo wget https://files.waveshare.com/upload/7/7c/Pico_Audio.7z
7z x Pico_Audio.7z -o./Pico-Audio
cd ~/Pico_Audio
```

Please use the following command for Rev2.1:

```
sudo apt-get install p7zip-full
cd ~
sudo wget https://files.waveshare.com/upload/9/92/Pico-Audio-V2.7z
7z x Pico-Audio-V2.7z -o./Pico-Audio-V2
cd ~/Pico-Audio-V2
```

There are two folders in the Pico_Audio file, the audio output program is stored in the Pico_Audio folder, and the uf2 file of the sound card is stored in the usb_sound_card folder, which can be copied directly into Pico.

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

Run Examples

C Codes

- The following tutorials are operated on the Raspberry Pi, but due to the multi-platform and portable characteristics of CMake, it can be compiled successfully on the PC, but the operation is slightly different, and you need to judge by yourself.
- Go into the Pico-Audio directory and build the codes.

```
cd ~/Pico_Audio/Pico-Audio/
```

Please use the following command for Rev2.1.

```
cd ~/Pico-Audio-V2/Pico-Audio/
```

Create a "build" folder and enter, add the path of SDK:

../pico-sdk is the default path of SDK, you need to modify it if the actual path is different in your pi.

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

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

Run cmake to generate Makefile file:

```
cmake ..
```

Run make command to build the codes:

```
make -j9
```

After building, the uf2 file is generated.

Press and hold the key on the Pico board to connect the Pico to the USB port of the computer through the Micro USB cable, then release the key. After plugging in, the computer will automatically recognize a removable disk (RPI-RP2). Copy the audio_firmware.uf2 file in the build folder to the removable disk (RPI-RP2) recognized.

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

Resource

Documents

- [Schematic Diagram](#)
- [Schematic-Rec2.1](#)
- [PCM5101A Datasheet](#)
- [APA2068 Datasheet](#)
- [CS4344](#)

Demo Codes

- [Demo code](#)
- [Example-Rev2.1](#)

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](#)

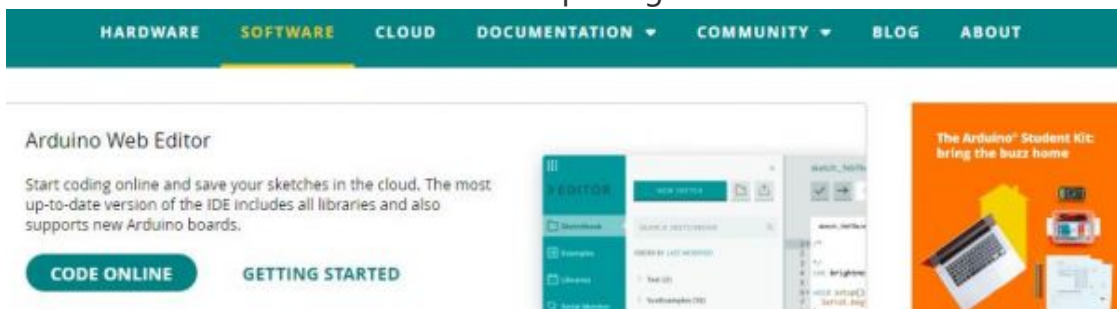
C/C++ Series

- [【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

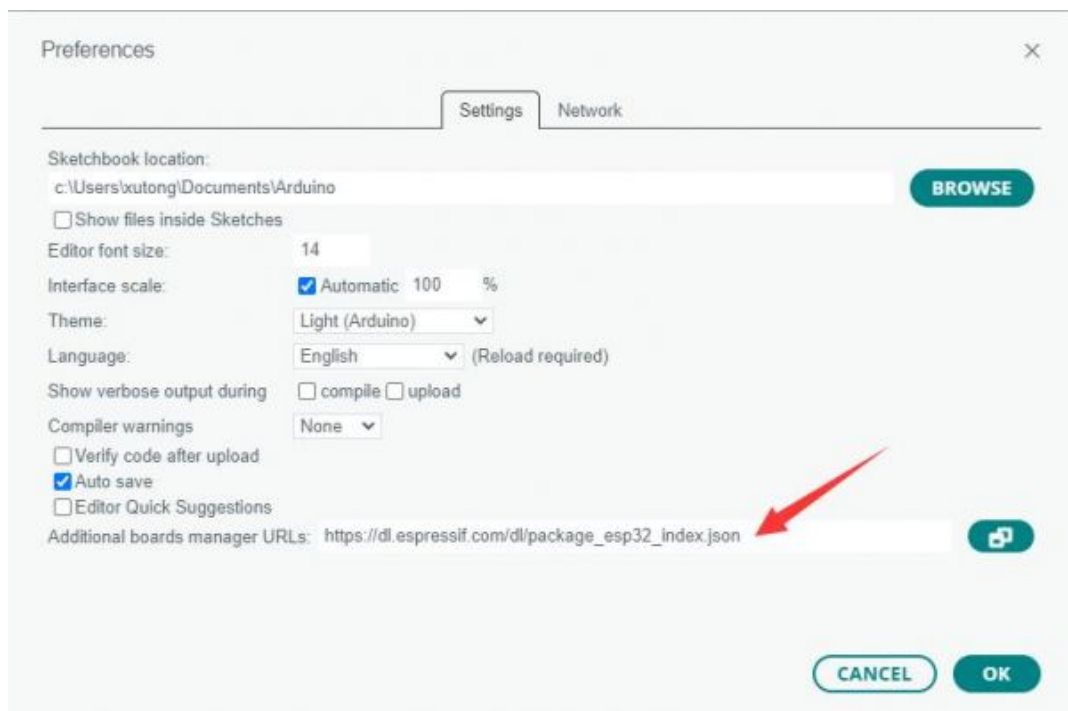
1. Open Arduino IDE, click the File on the left corner and choose "Preferences".





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

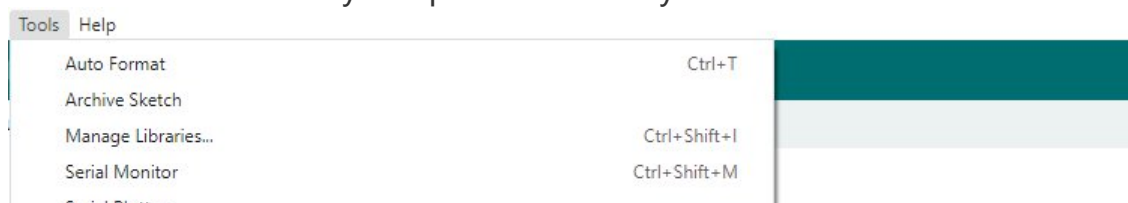
```
https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json
```

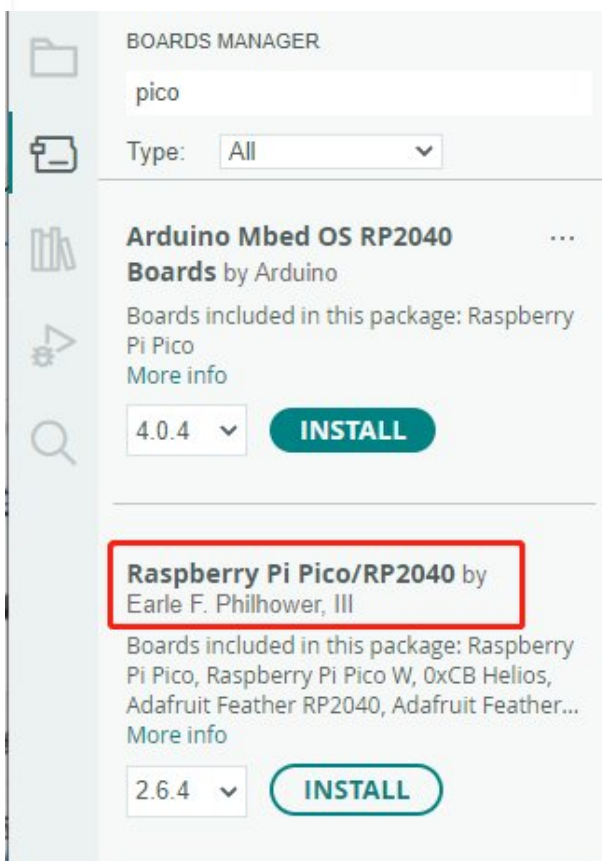
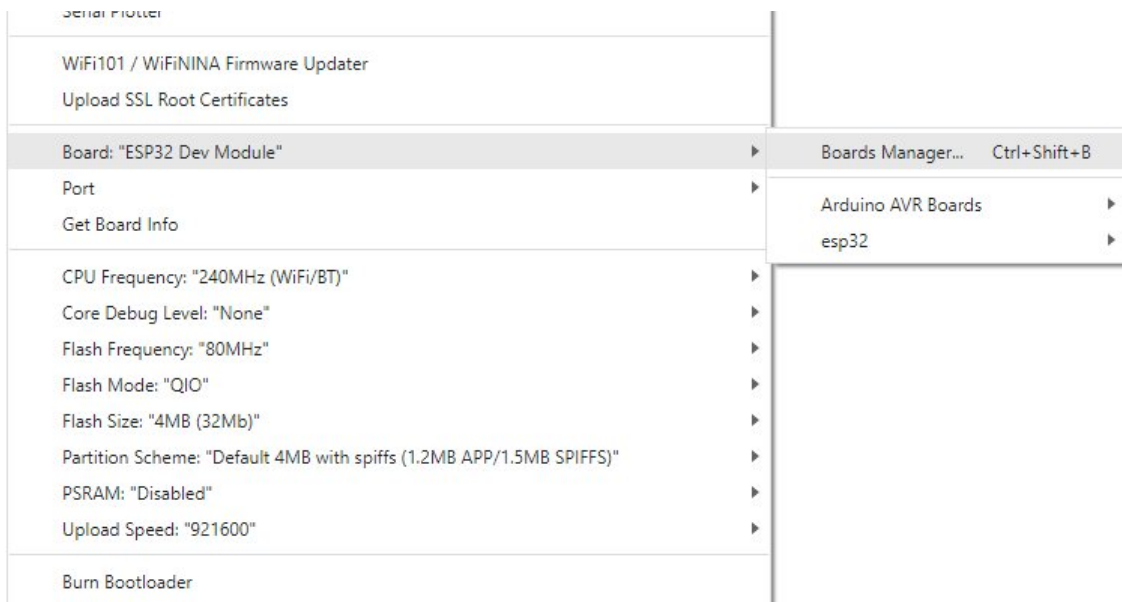


Note: If you already have the ESP8266 board URL, you can separate the URLs with commas like this:

```
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.





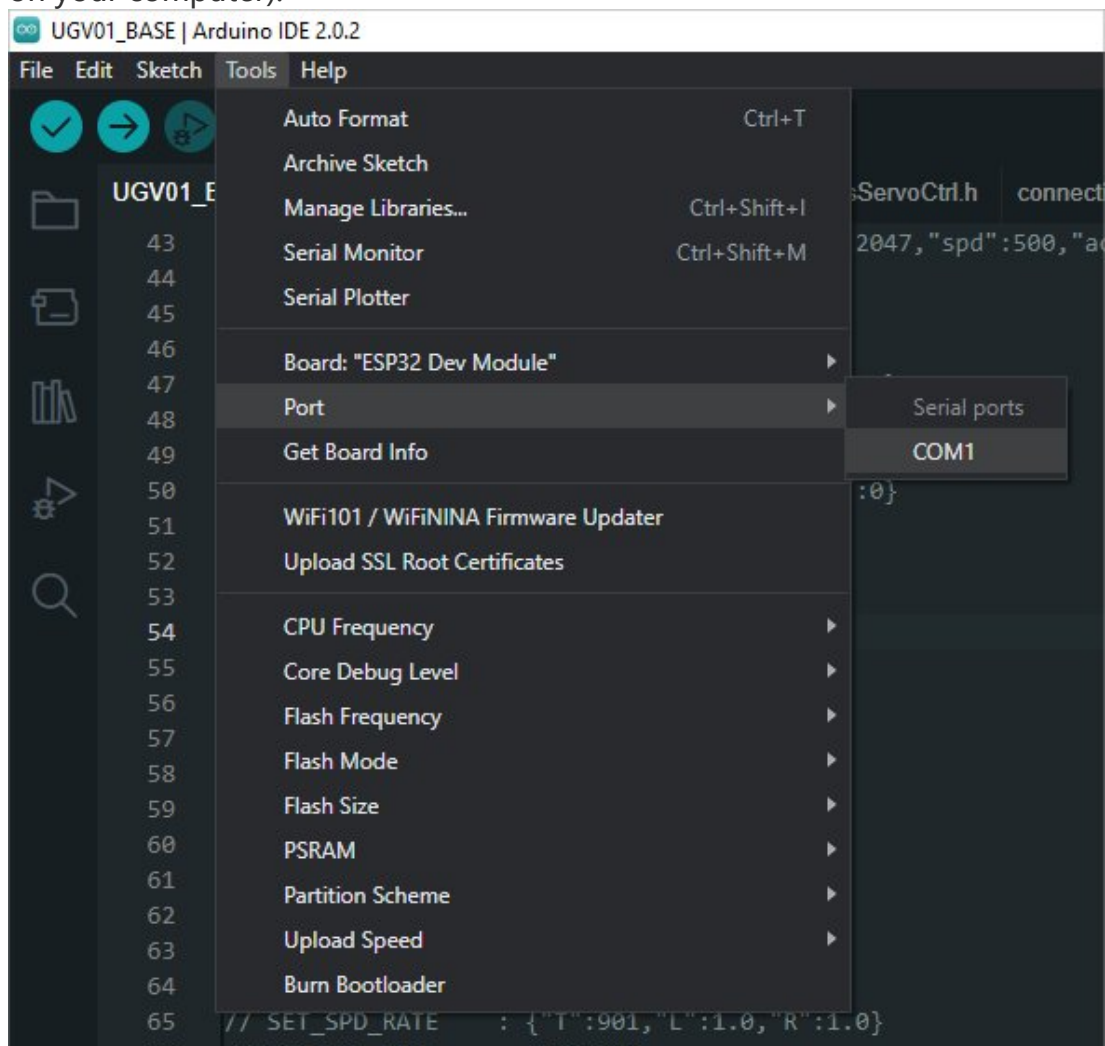
Upload Demo At the First Time

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).

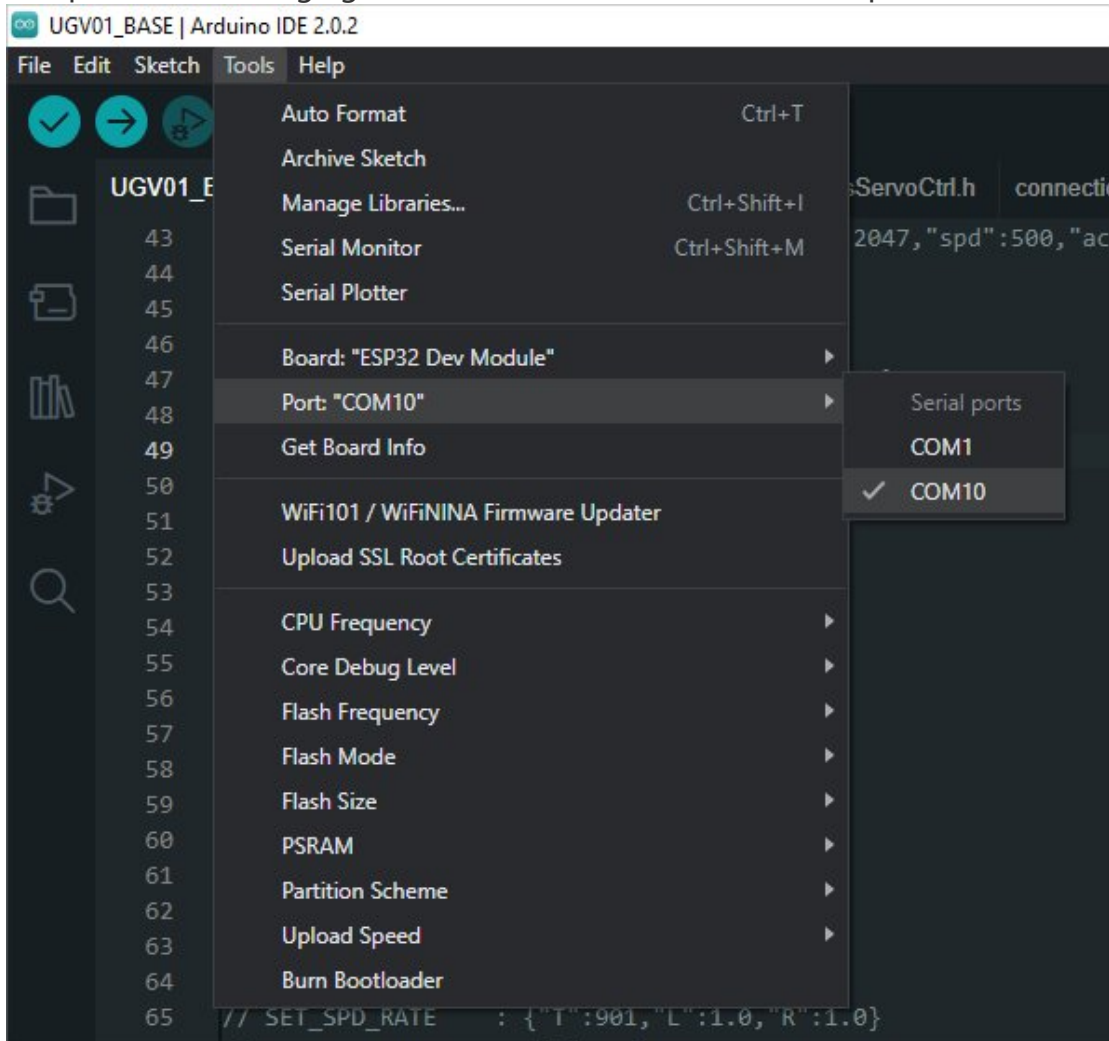




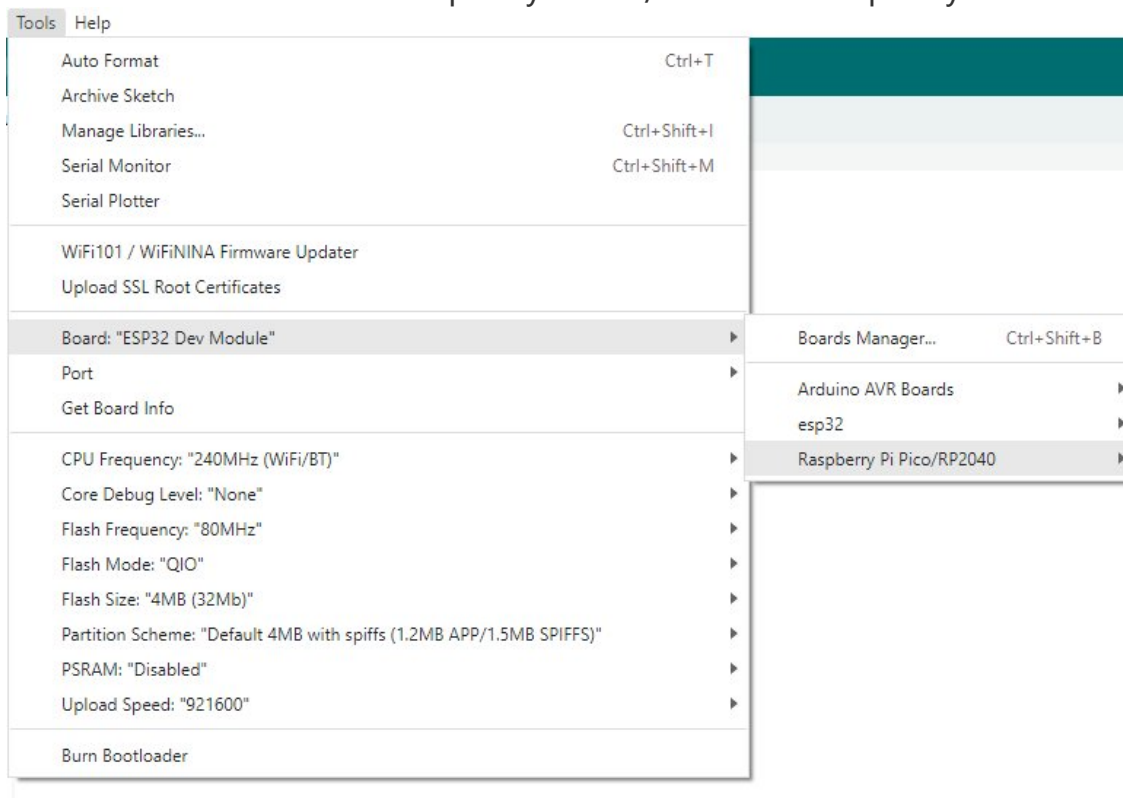
2. Download the demo, open arduino\PWM\D1-LED path under the D1-LED.ino.
3. Click Tools -> Port, remember the existing COM, do not need to click this COM (different computers show different COM, remember the existing COM on your computer).



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](#)