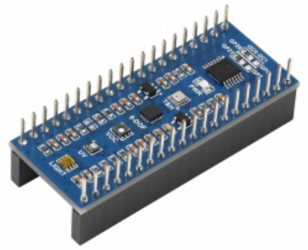
Overview

Pico Environment Sensor



The Pico-Environment-Sensor gives Raspberry Pi Pico the ability to collect environment data like temperature & humidity, air pressure, ambient light intensity, VOC, UV ray, etc. It can also be used to build a robot that can detect motion and orientation.

Features

- Standard Raspberry Pi Pico header, supports Raspberry Pi Pico series.
- Onboard TSL25911FN digital ambient light sensor, for measuring IR and visible light.
- Onboard BME280 sensor, for measuring temperature, humidity, and air pressure.
- Onboard ICM20948 motion sensor, accelerometer, gyroscope, and magnetometer.
- Onboard LTR390-UV-1 sensor, for measuring UV rays.
- Onboard SGP40 sensor, for detecting ambient VOC.
- I2C bus allows retrieving data by just using two wires.

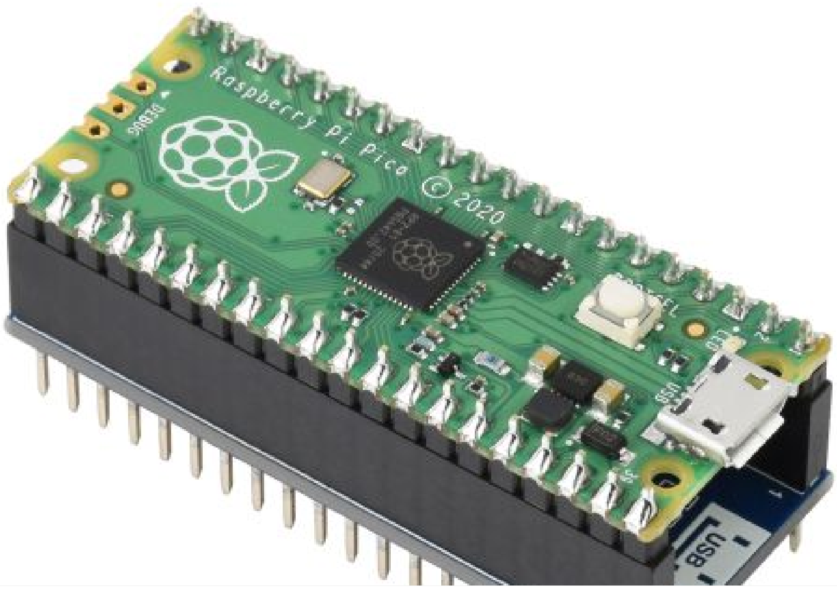
Specifications

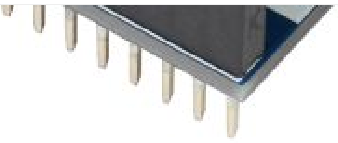
TSL25911FN DIGITAL AMBIENT LIGHT SENSOR	
I2C address	0 x 29
Effective range	0~88000 Lux
BME280 TEMPERATURE, HUMIDITY, AND AIR PRESSURE SENSOR	
I2C address	0 x 76
Temperature detection	-40 ~ 85℃ (0.01℃ resolution, ±1℃ accuracy)
Humidity detection	0~100%RH (0.008%RH resolution, ±3%RH accuracy, 1s response time, ≤2%RH delay)
Air presure detection	300~1100hPa (0.18Pa resolution, ±1hPa accuracy)
ICM20948 MOTION SENSOR **(9-DOF: 3-AXIS ACCELEROMETER, 3-AXIS GYROSCROPE, 3-AXIS**	

MAGNETOMETER)	
I2C address	0 x 68
Accelerometer resolution	16-bit
Accelerometer range (configurable)	2, ±4, ±8, ±16g
Gyroscope resolution	16-bit
Gyroscope range (configurable)	±250, ±500, ±1000, ±2000°/sec
Magnetometer resolution	16-bit
Magnetometer range	±4900µT
LTR390-UV-1 UV SENSOR	
I2C address	0 x 53
Response wavelength	280nm - 430nm
SGP40 VOC Sensor	
I2C address	0 x 59
Measuring range	0 ~ 1,000 ppm ethanol equivalent
Limit condition	< 0.05 ppm ethanol equivalent OR < 10 % preset concentration point (the larger one should prevail)
Response time	< 10 s (tau 63 %)
Start time	< 60 s
On-chip humidity compensation support	

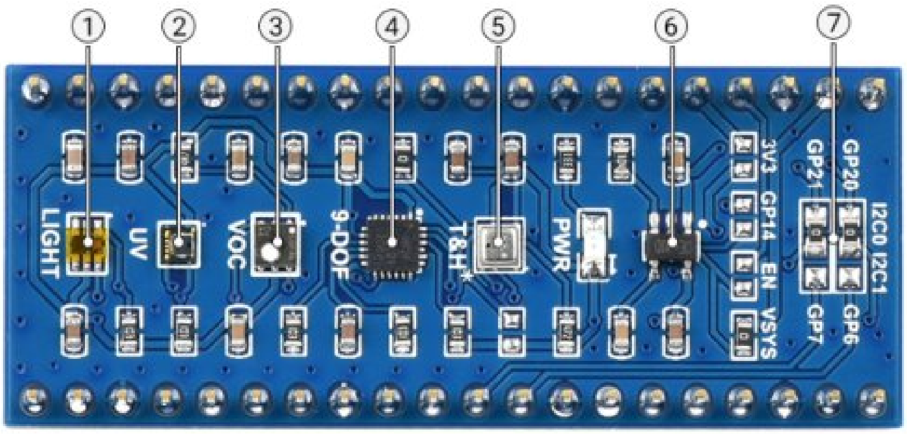
How to Connect

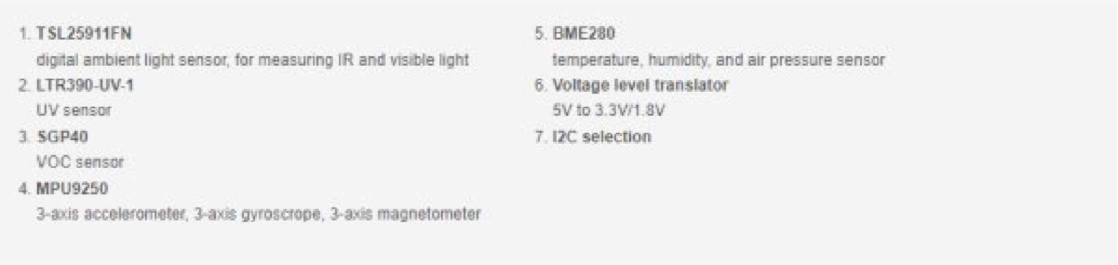
As shown below: directly connect to the GPIO headers of the Jetson Nano.





Onboard Resource





1. **TSL25911FN**
   digital ambient light sensor, for measuring IR and visible light
2. **LTR390-UV-1**
   UV sensor
3. **SGP40**
   VOC sensor
4. **MPU9250**
   3-axis accelerometer, 3-axis gyroscrope, 3-axis magnetometer
5. **BME280**
   temperature, humidity, and air pressure sensor
6. **Voltage level translator**
   5V to 3.3V/1.8V
7. **I2C selection**

Sensor

Ambient Light Sensor

The TSL2591 is an optical intensity digitizer based on IIC bus communication. The sensor combines a broadband photodiode (visible and IR light) and an IR-responsive photodiode on a single CMOS integrated circuit capable of providing near-light adaptive response over an effective 16-bit dynamic range (16-bit resolution). Two integral ADCs convert the photodiode current to a digital output representing the irradiance measured on each channel. This digital output can be fed to a microprocessor, which uses an empirical formula to derive an illuminance (ambient light level) in lux to approximate the human eye response.
The sensor address is 0X29.
The following points need to be noted when using this sensor:

1. The default is turned on the interrupt, if you do not need to comment out the corresponding code;
2. The default range of the routine is small, if the light suddenly changes a lot, please increase the range;

3. There is a difference with the light intensity reader on the market, those are added to the lens;

Temperature, Humidity, and Ari Pressure Sensor

BME280 is for temperature, humidity, and air pressure sensor with the characteristics of low power consumption, high accuracy, and stability. It is suitable for environmental monitoring, weather prediction, altitude monitoring, and Internet of Things application scenarios.
Sensor address: 0 x 76.
Measuring range:
Temperature range: 40~85°C (resolution is 0.01°C, error ±1°C).
Humidity range: 0~100%RH (resolution is 0.008%RH, error ±3% RH)
Pressure range: 300~1100 hPa (resolution is 0.18Pa, error ±1 hPa)
Note:

1. The measuring range should not exceed the measuring range, or it may be inaccurate.
2. The pressure measured is atmospheric pressure, which normally fluctuates around the standard atmospheric pressure (1.013x 10^5 = 1013hPa).
3. The measured temperature may be biased to the temperature above the jetson nano. If possible, fans can be added to avoid this situation during use.
4. The higher the air temperature is, the lower the air pressure is. Because the higher the air temperature is, the air near the ground will expand and rise due to heat, resulting in a decrease in air density and air pressure. Instead, the pressure rises.
5. Atmospheric pressure decreases with altitude, but there is no proportional relationship. The higher the altitude, the slower the decrease.
6. The relationship between atmospheric pressure and altitude is as follows: as altitude increases, atmospheric pressure decreases. Theoretically, within the range of 3000M, atmospheric pressure decreases by 1mmHg, or about 133Pa, for every 12M increase.

9-axis Sensor

MPU9250 is a multi-chip module with 9-axis motion tracking, 3-axis acceleration, 3-axis gyroscope, and 3-axis magnetometer, and a built-in digital motion processing engine to reduce complex fusion algorithm data and reduce the load on the processor.
Sensor address: 0x68
Parameters:
Accelerometer characteristics:

- Resolution: 16 bit
- Measurement range (Optional): ±2, ±4, ±8, ±16g
- Operating current: 450uA

Gyroscope characteristics:

- Resolution: 16 bit
- Measurement range (Optional): ±250, ±500, ±1000, ±2000°/sec
- Operating current: 3.2mA

Magnetometer characteristics:

- Resolution: 16 bit
- Measurement range: ±4900µT
- Operating current: 280uA

Note:

1. Acceleration (LSB, convert to g)
2. Gyroscope angle velocity (LSB, convert to °/second)
3. Magnetic electronic compass angle (°)

UV Sensor

The LTR390-UV-01 is a multifunctional optical measurement module with a focus on measuring ultraviolet light. The sensor supports the measured ambient light intensity.
The sensor address is: 0X53
The use of this sensor requires attention to the following points.

1. The principle of operation is actually a pair of diodes, one sensitive to normal light and one sensitive to ultraviolet light, and the result is obtained by the calculation of both.
2. The value readout is the original ADC data, if you want to calibrate can refer to the manual formula and combine it with the actual light intensity value, UV for correction.
3. The data can be set differently by gain, which needs to be set according to the actual application environment.
4. The manual does not specify the detection range, the actual measurement in the sun has a better effect or the best effect with a UV pen in 50cm thought.

VOC Sensor

SGP40 Sensirion's new digital VOC (Volatile Organic Compounds) sensor, easily integrated into air handling devices and air quality monitors, features a temperature-controlled miniature heating plate and a humidity-compensated indoor air quality signal that provides a complete sensor system on a single chip.

The sensor address is: 0X59

The following points need to be noted when using this sensor.

1. The default Python demo outputs the VOC-processed algorithm value.
2. The sensor work needs to be stabilized within 1 minute, as the internal is slowly heating up.
3. Can be fed back to the SGP40 by measuring the current temperature and humidity so that the accuracy of the output will be higher.
4. The sensor measurement range of 0 to 1,000 ppm ethanol equivalent.

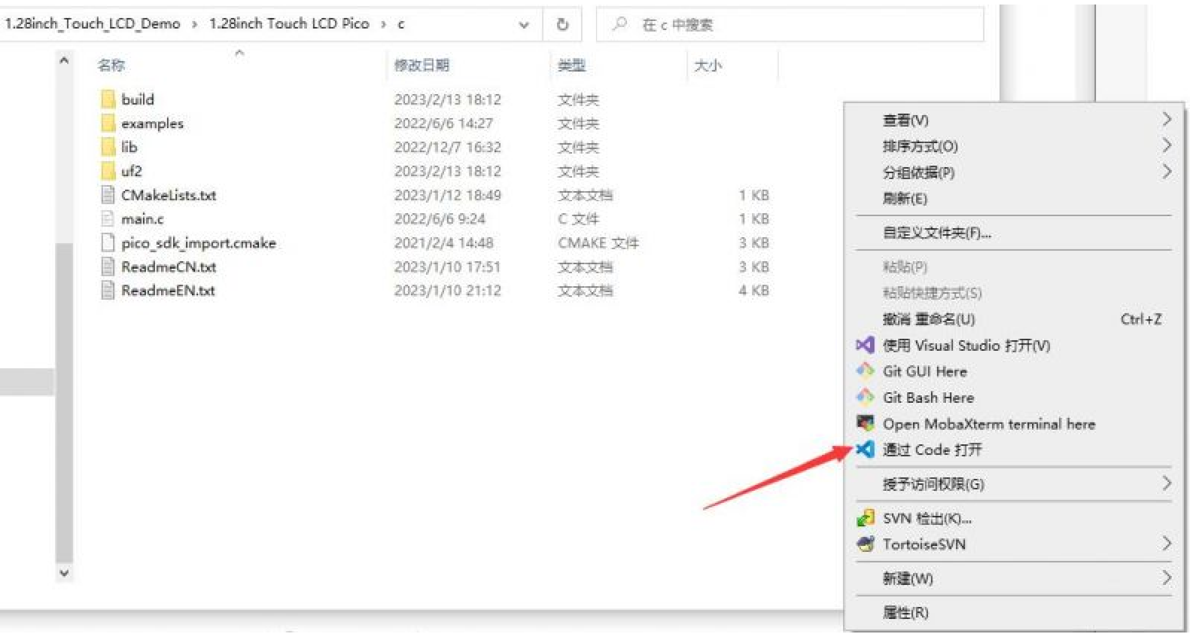
C

C/C++ Development Environment Installation

- Before using the demos and tutorials, you need to set up the development environment and learn the basic usage of the project:
  - Pico-Get-Start-Windows
  - Pico-Get-Start-RPI

Windows

- Click here to download the demo, unzip it and enter the folder of Pico-Environment-Sensor-code\c.
- After entering Pico-Environment-Sensor-code\c, you can open the project with vs code.

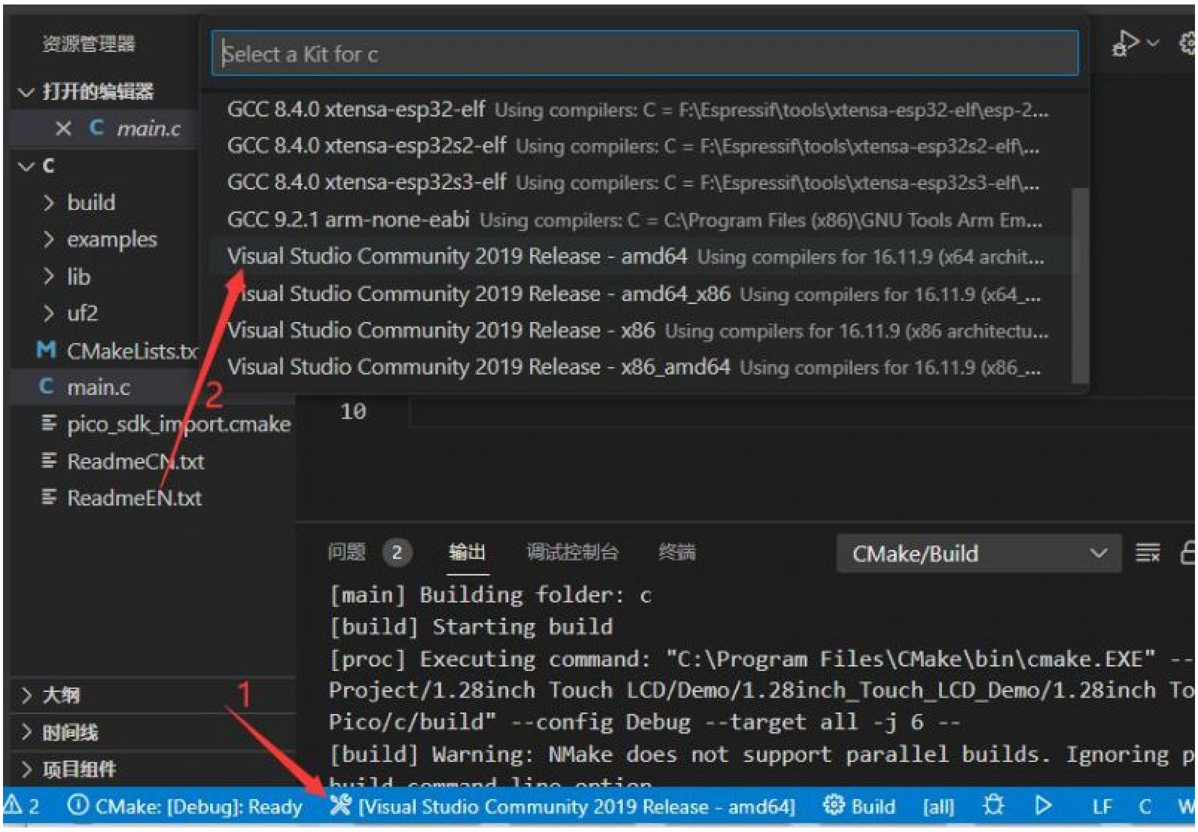


1.28inch_Touch_LCD_Demo › 1.28inch Touch LCD Pico › c

在 c 中搜索

名称	修改日期	类型	大小
build	2023/2/13 18:12	文件夹	
examples	2022/6/6 14:27	文件夹	
lib	2022/12/7 16:32	文件夹	
uf2	2023/2/13 18:12	文件夹	
CMakeLists.txt	2023/1/12 18:49	文本文档	1 KB
main.c	2022/6/6 9:24	C 文件	1 KB
pico_sdk_import.cmake	2021/2/4 14:48	CMAKE 文件	3 KB
ReadmeCN.txt	2023/1/10 17:51	文本文档	3 KB
ReadmeEN.txt	2023/1/10 21:12	文本文档	4 KB

- Choose the Compiler.

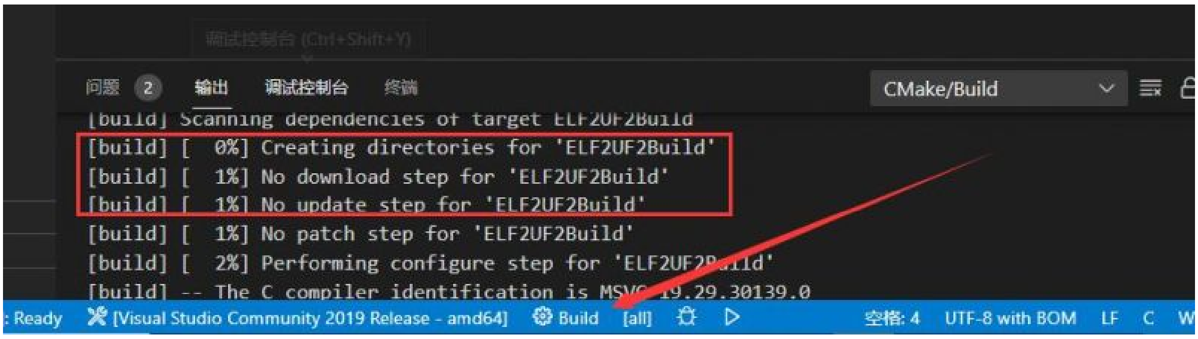


Select a Kit for c

- GCC 8.4.0 xtensa-esp32-elf Using compilers: C = F:\Espressif\tools\xtensa-esp32-elf\esp-2...
- GCC 8.4.0 xtensa-esp32s2-elf Using compilers: C = F:\Espressif\tools\xtensa-esp32s2-elf\...
- GCC 8.4.0 xtensa-esp32s3-elf Using compilers: C = F:\Espressif\tools\xtensa-esp32s3-elf\...
- GCC 9.2.1 arm-none-eabi Using compilers: C = C:\Program Files (x86)\GNU Tools Arm Em...
- Visual Studio Community 2019 Release - amd64 Using compilers for 16.11.9 (x64 archit...
- Visual Studio Community 2019 Release - amd64_x86 Using compilers for 16.11.9 (x64_...
- Visual Studio Community 2019 Release - x86 Using compilers for 16.11.9 (x86 architectu...
- Visual Studio Community 2019 Release - x86_amd64 Using compilers for 16.11.9 (x86_...

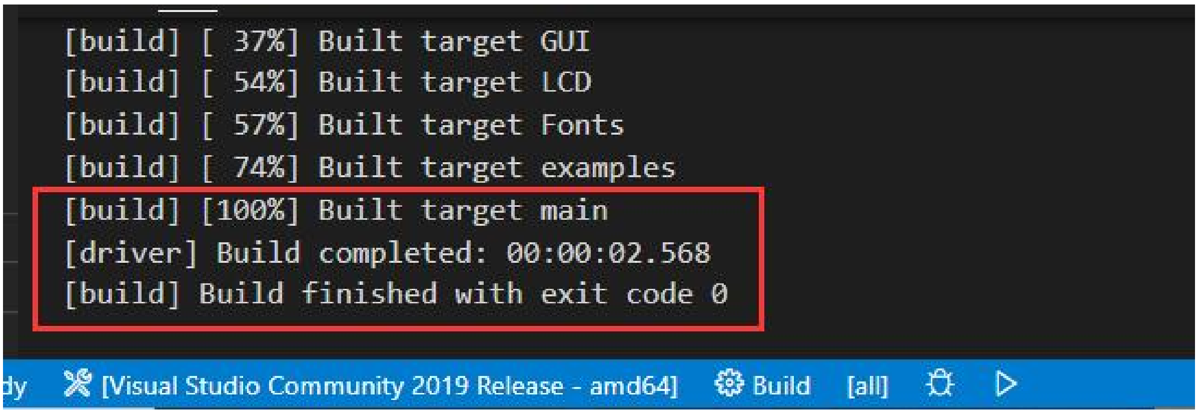
```
[main] Building folder: c
[build] Starting build
[proc] Executing command: "C:\Program Files\CMake\bin\cmake.EXE" --
Project/1.28inch Touch LCD/Demo/1.28inch_Touch_LCD_Demo/1.28inch To
Pico/c/build" --config Debug --target all -j 6 --
[build] Warning: NMake does not support parallel builds. Ignoring p
build command line option
```

- Start to compile.



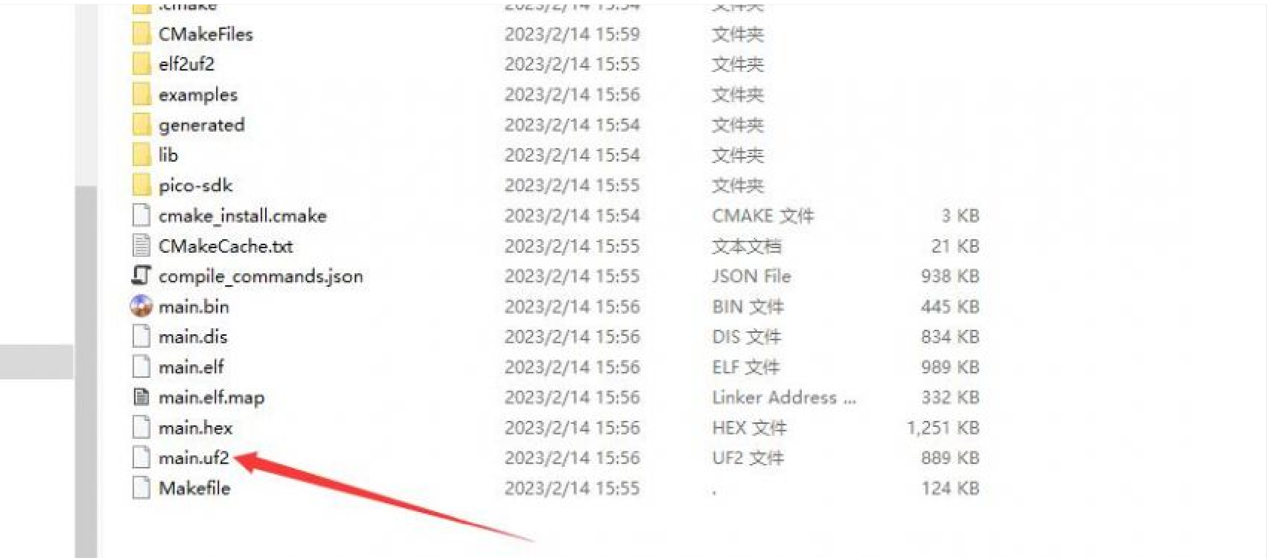
```
[build] Scanning dependencies of target ELF2UF2Build
[build] [  0%] Creating directories for 'ELF2UF2Build'
[build] [  1%] No download step for 'ELF2UF2Build'
[build] [  1%] No update step for 'ELF2UF2Build'
[build] [  1%] No patch step for 'ELF2UF2Build'
[build] [  2%] Performing configure step for 'ELF2UF2Build'
[build] -- The C compiler identification is MSVC 19.29.30139.0
```

- Finish.



```
[build] [ 37%] Built target GUI
[build] [ 54%] Built target LCD
[build] [ 57%] Built target Fonts
[build] [ 74%] Built target examples
[build] [100%] Built target main
[driver] Build completed: 00:00:02.568
[build] Build finished with exit code 0
```

- Copy the Pico_Environment_Sensor.uf2 in build to Pico, and then it can automatically run the demo.



.cmake	2023/2/14 15:54	文件夹	
CMakeFiles	2023/2/14 15:59	文件夹	
elf2uf2	2023/2/14 15:55	文件夹	
examples	2023/2/14 15:56	文件夹	
generated	2023/2/14 15:54	文件夹	
lib	2023/2/14 15:54	文件夹	
pico-sdk	2023/2/14 15:55	文件夹	
cmake_install.cmake	2023/2/14 15:54	CMAKE 文件	3 KB
CMakeCache.txt	2023/2/14 15:55	文本文档	21 KB
compile_commands.json	2023/2/14 15:55	JSON File	938 KB
main.bin	2023/2/14 15:56	BIN 文件	445 KB
main.dis	2023/2/14 15:56	DIS 文件	834 KB
main.elf	2023/2/14 15:56	ELF 文件	989 KB
main.elf.map	2023/2/14 15:56	Linker Address ...	332 KB
main.hex	2023/2/14 15:56	HEX 文件	1,251 KB
main.uf2	2023/2/14 15:56	UF2 文件	889 KB
Makefile	2023/2/14 15:55	.	124 KB

In Raspberry Pi

- Open the Raspberry Pi and run:

```
sudo apt-get install p7zip-full
cd ~
sudo wget https://files.waveshare.com/upload/f/f9/Pico-Environment-Sensor-code.7
z
7zr x Pico-Environment-Sensor-code.7z -r -o./Pico-Environment-Sensor-code
cd ~/Pico-Environment-Sensor-code
cd c/build/
```

Demo Use

- Please operates the following tutorial on Raspberry Pi, but due to the multi-platform and portable nature of CMake, it can also be compiled successfully on PC, but the operation is slightly different and requires your own judgment.
- To perform the compilation, make sure that in the c directory:

```
cd ~/Pico-Environment-Sensor-code/c/
```

Create and enter the build directory, and add the SDK: ../../pico-sdk is the directory of your SDK. There is a build in our sample application, just enter it directly.

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

Execute cmake to automatically generate the Makefile file:

```
cmake ..
```

Execute make to generate the executable file, you may wait for a long time as it is the first time to be compiled.

```
make -j9
```

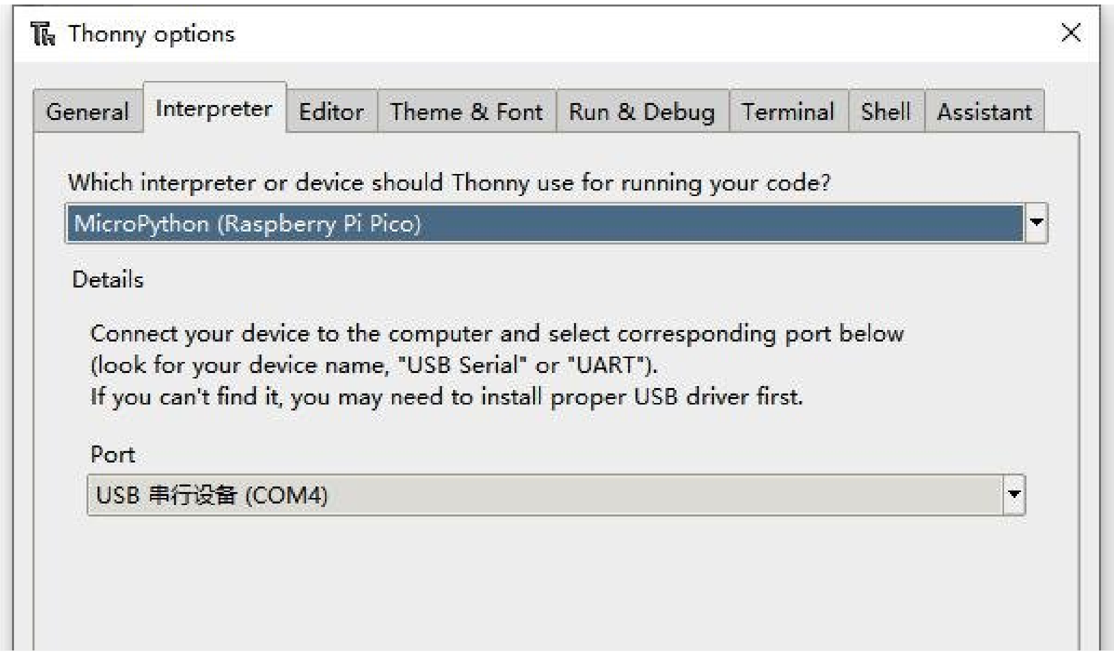
After compiling, the uf2 file will generate. Press the button on the Pico board, Pico can connect to the USB port of the Raspberry Pi via a Micro USB cable, and then release the buttons. After connecting, Raspberry Pi will automatically identify a movable disk (RPI-RP2), and copy main.uf2 in the build file to the recognizable movable disk (RPI-RP2).

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

Python

Windows

1. Download the demo.
2. 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 disk (RPI-RP2).
3. Copy the pico_micropython_20221125.uf2 file from the UF2 directory to the recognized removable disk (RPI-RP2).
4. Open the Thonny IDE (note: use the latest version of Thonny, otherwise there is no support package for Pico, the current latest version for Windows is v3.3.3).
5. Click Tools -> Settings -> Interpreter, and select Pico and the corresponding port as shown in the picture.



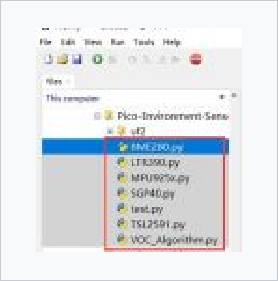
Which interpreter or device should Thonny use for running your code?
MicroPython (Raspberry Pi Pico)

Details

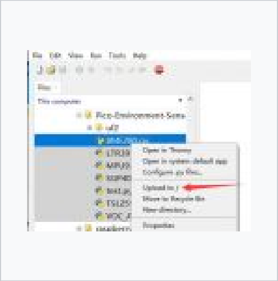
Connect your device to the computer and select corresponding port below
(look for your device name, "USB Serial" or "UART").
If you can't find it, you may need to install proper USB driver first.

Port
USB 串行设备 (COM4)

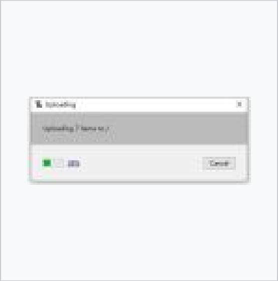
6. After connecting to Pico, save all the downloaded code to Pico, then run test.py, Thonny will print the sensor data as follows:
(1) Find the path of your downloaded program in the Thonny IDE.
(2) Upload all the Python to Pico.
(3) After uploading, open test.py for module testing.
(4) No save and the output value is normal, then the function is OK.



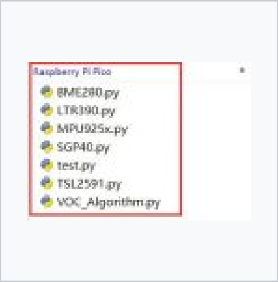
Find the downloaded directory



Righ click to upload



Wait for uploading



When finished, the files are all in Pico

Effect shown

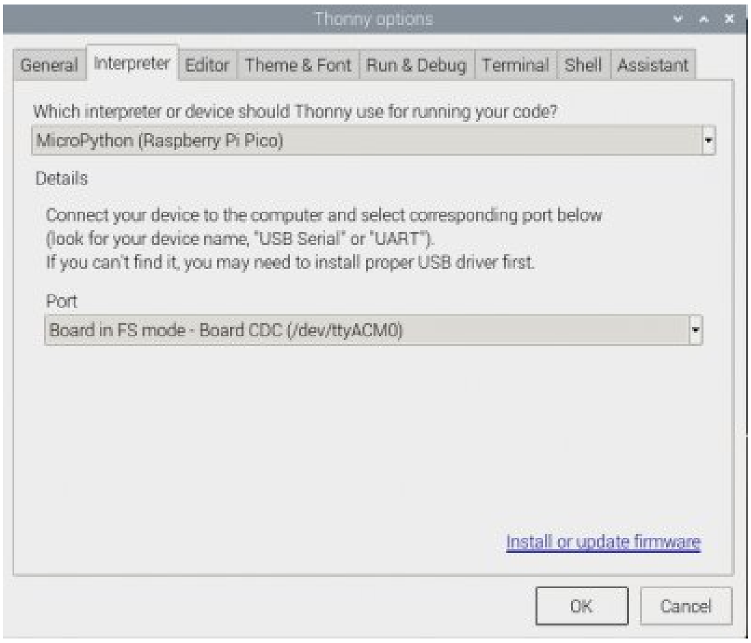
Raspberry Pi

- Demo download:

```
sudo apt-get install p7zip-full
cd ~
sudo wget  https://files.waveshare.com/upload/f/f9/Pico-Environment-Sensor-code.
```

```
7z
7z x Pico-Environment-Sensor-code.7z -o./Pico-Environment-Sensor-code
cd ~/Pico-Environment-Sensor-code
```

1. The process of flashing the firmware is the same as on Windows, you can choose to copy the pico_micropython_20210121.uf2 file into pico on your PC or Raspberry Pi.
2. Open the Thonny IDE in Raspberry Pi Mountain (click Raspberry logo -> Programming -> Thonny Python IDE), you can check the version information in Help -> About Thonny.
To make sure your version has the Pico support package, also you can click on Tools -> Options... -> Interpreter to select MicroPico. -> Interpreter to select MicroPython (Raspberry Pi Pico and ttyACM0 port).
As shown in the image:



Which interpreter or device should Thonny use for running your code?
MicroPython (Raspberry Pi Pico)

Details

Connect your device to the computer and select corresponding port below
(look for your device name, "USB Serial" or "UART").
If you can't find it, you may need to install proper USB driver first.

Port
Board in FS mode - Board CDC (/dev/ttyACM0)

If your current Thonny version has no Pico support package, you should enter the following commands to update Thonny IDE:

```
sudo apt upgrade thonny
```

3. Click File -> Open... -> test.py, and run the script.

Resource

Documents

- Schematic
- BME280 Specification
- TSL25911 Specification

- ICM20948 Datasheet
- LTR-390UV Specification
- SGP40 Specification

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 [Expand]

- 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

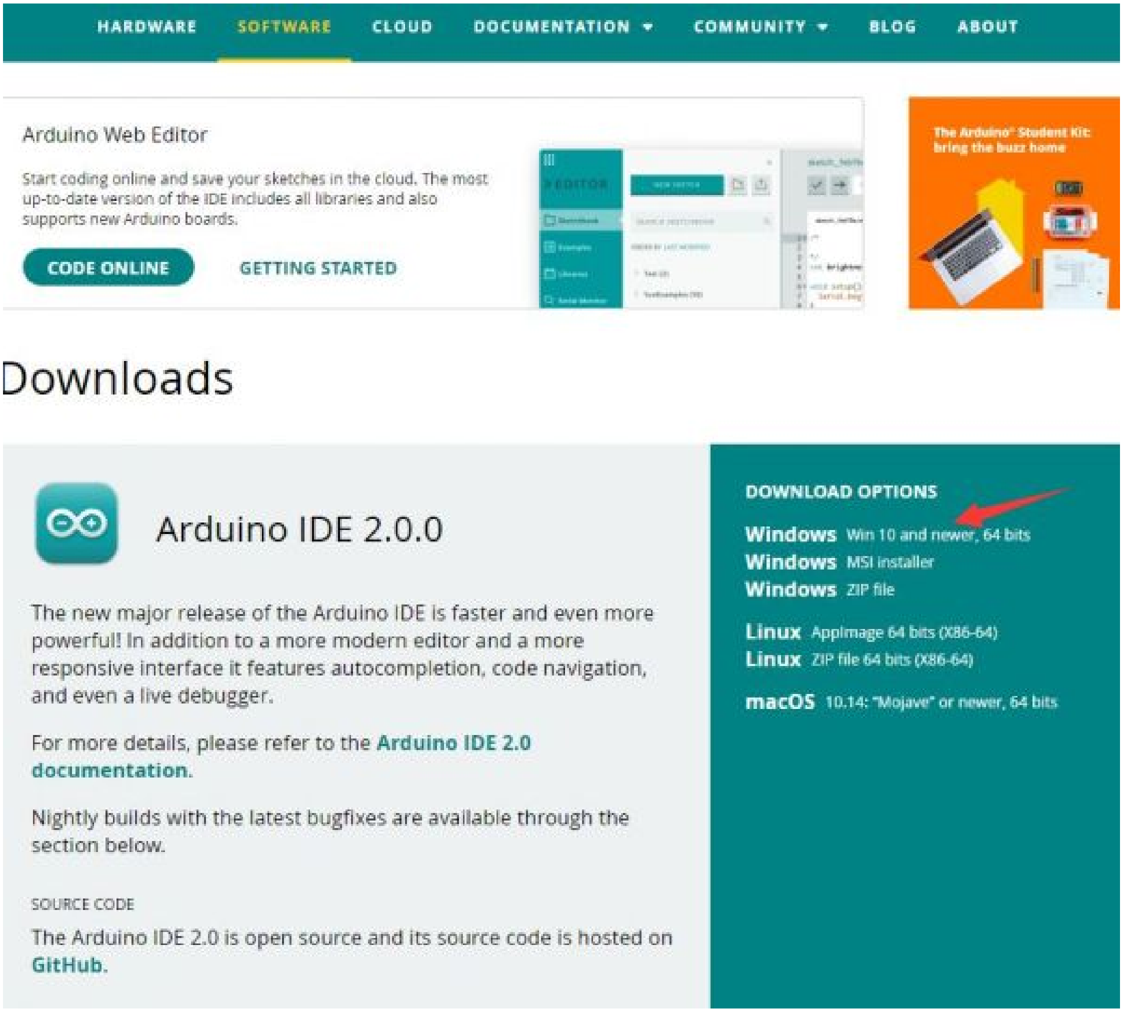
C/C++ Series

- 【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.



Arduino Web Editor

Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

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.**

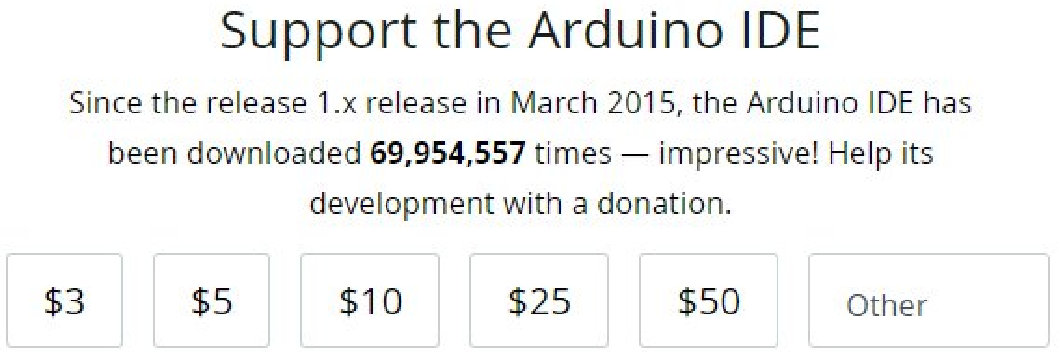
DOWNLOAD OPTIONS

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)

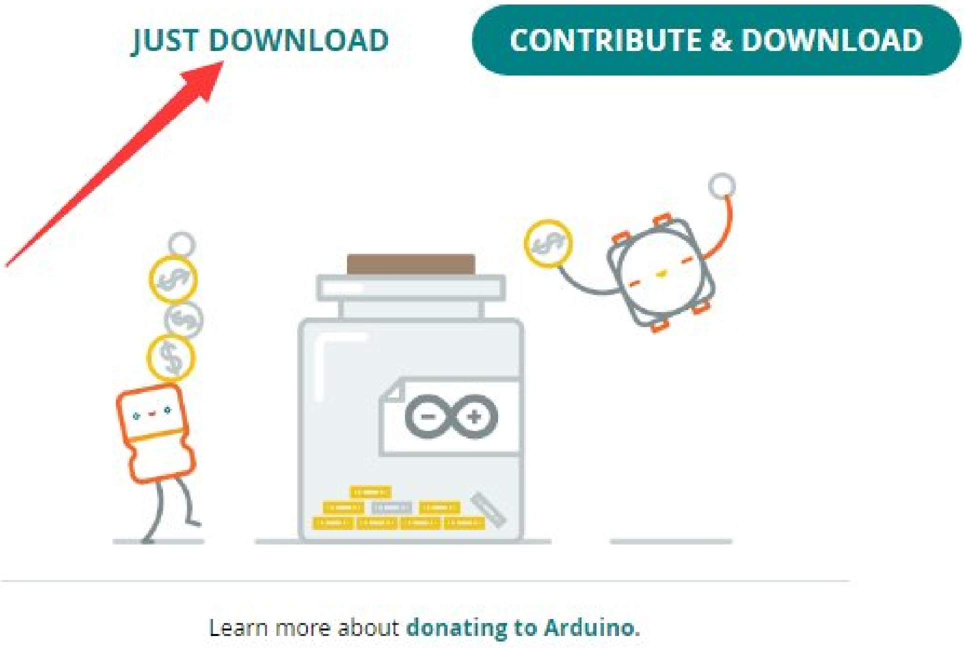
macOS 10.14: "Mojave" or newer, 64 bits

2. Just click on "JUST DOWNLOAD".



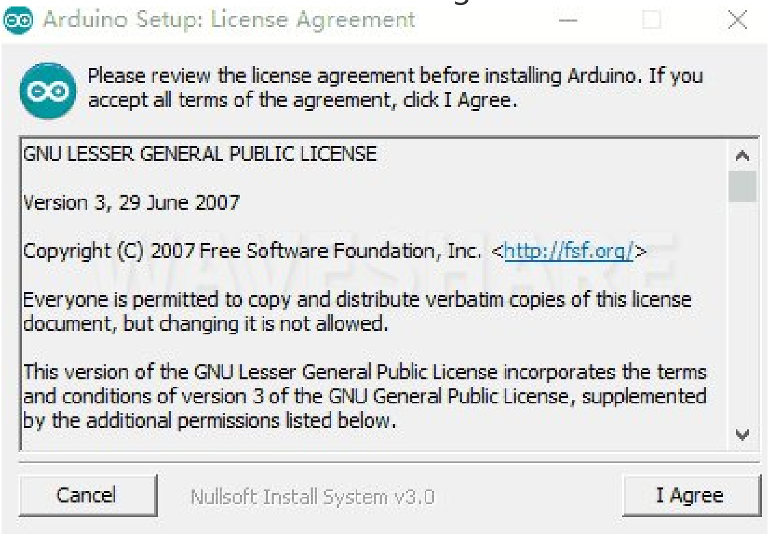
Support the Arduino IDE

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.



Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

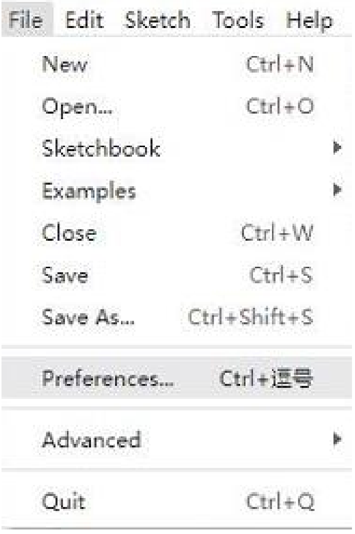
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

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

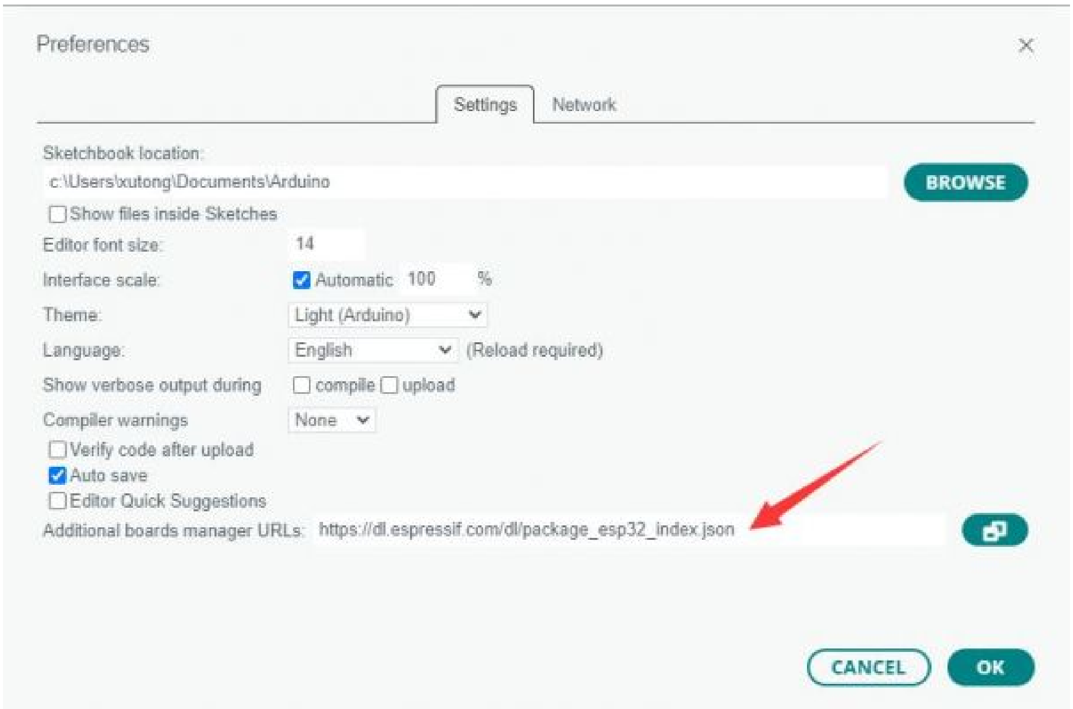
Install Arduino-Pico Core on Arduino IDE

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

Settings | Network

Sketchbook location: c:\Users\xutong\Documents\Arduino

□ Show files inside Sketches

Editor font size: 14

Interface scale: ☑ Automatic 100 %

Theme: Light (Arduino)

Language: English (Reload required)

Show verbose output during □ compile □ upload

Compiler warnings None

□ Verify code after upload

☑ Auto save

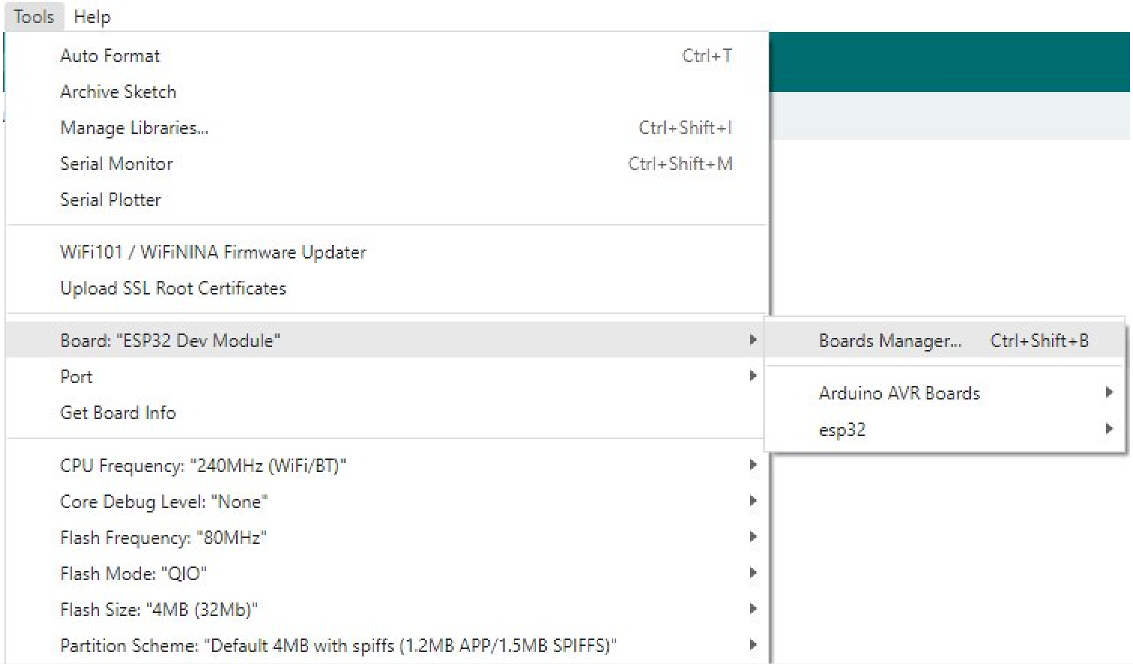
□ Editor Quick Suggestions

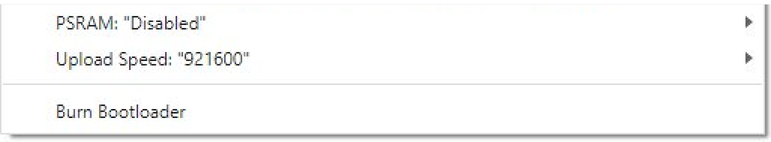
Additional boards manager URLs: https://dl.espressif.com/dl/package_esp32_index.json

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.

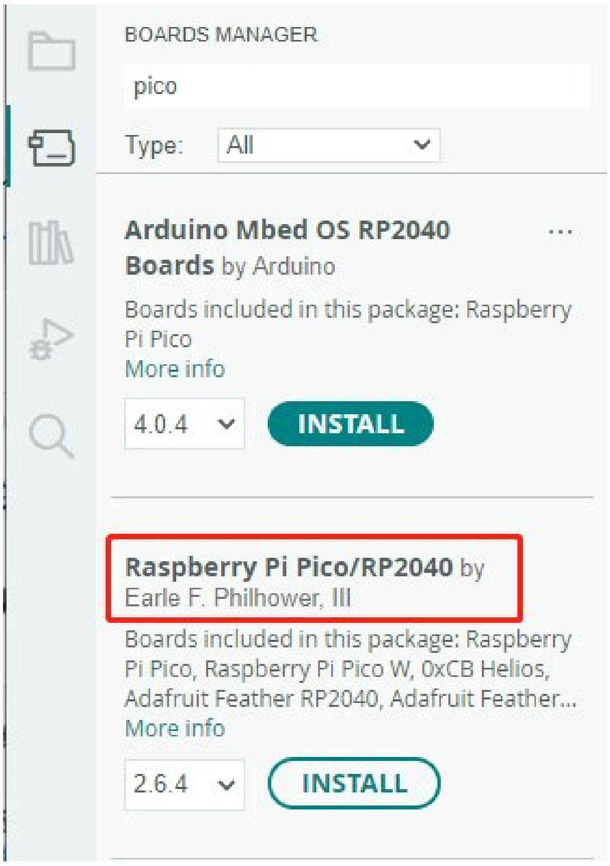

Tools
Help
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Board: "ESP32 Dev Module"
Port
Get Board Info
CPU Frequency: "240MHz (WiFi/BT)"
Core Debug Level: "None"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Boards Manager... Ctrl+Shift+B
Arduino AVR Boards
esp32



PSRAM: "Disabled"

Upload Speed: "921600"

Burn Bootloader



BOARDS MANAGER

pico

Type: All

Arduino Mbed OS RP2040 Boards by Arduino

Boards included in this package: Raspberry Pi Pico

More info

4.0.4 INSTALL

Raspberry Pi Pico/RP2040 by Earle F. Philhower, III

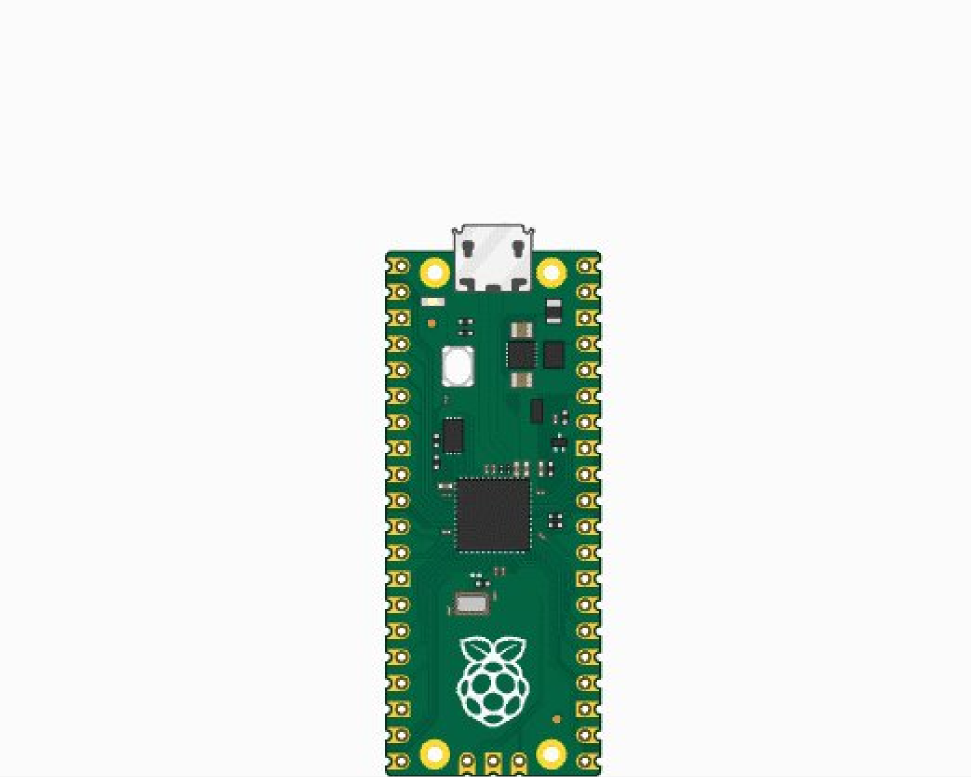
Boards included in this package: Raspberry Pi Pico, Raspberry Pi Pico W, 0xCB Helios, Adafruit Feather RP2040, Adafruit Feather...

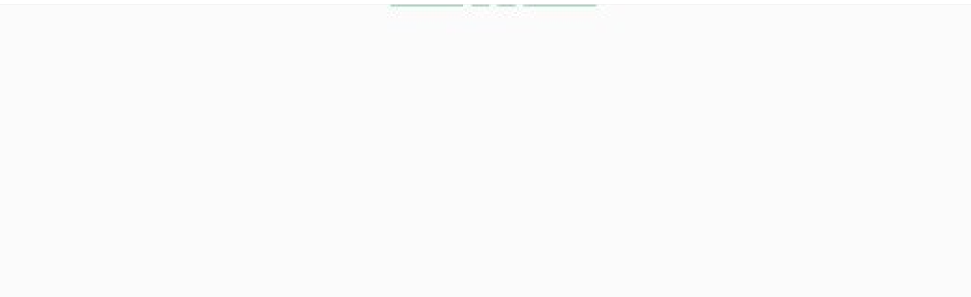
More info

2.6.4 INSTALL

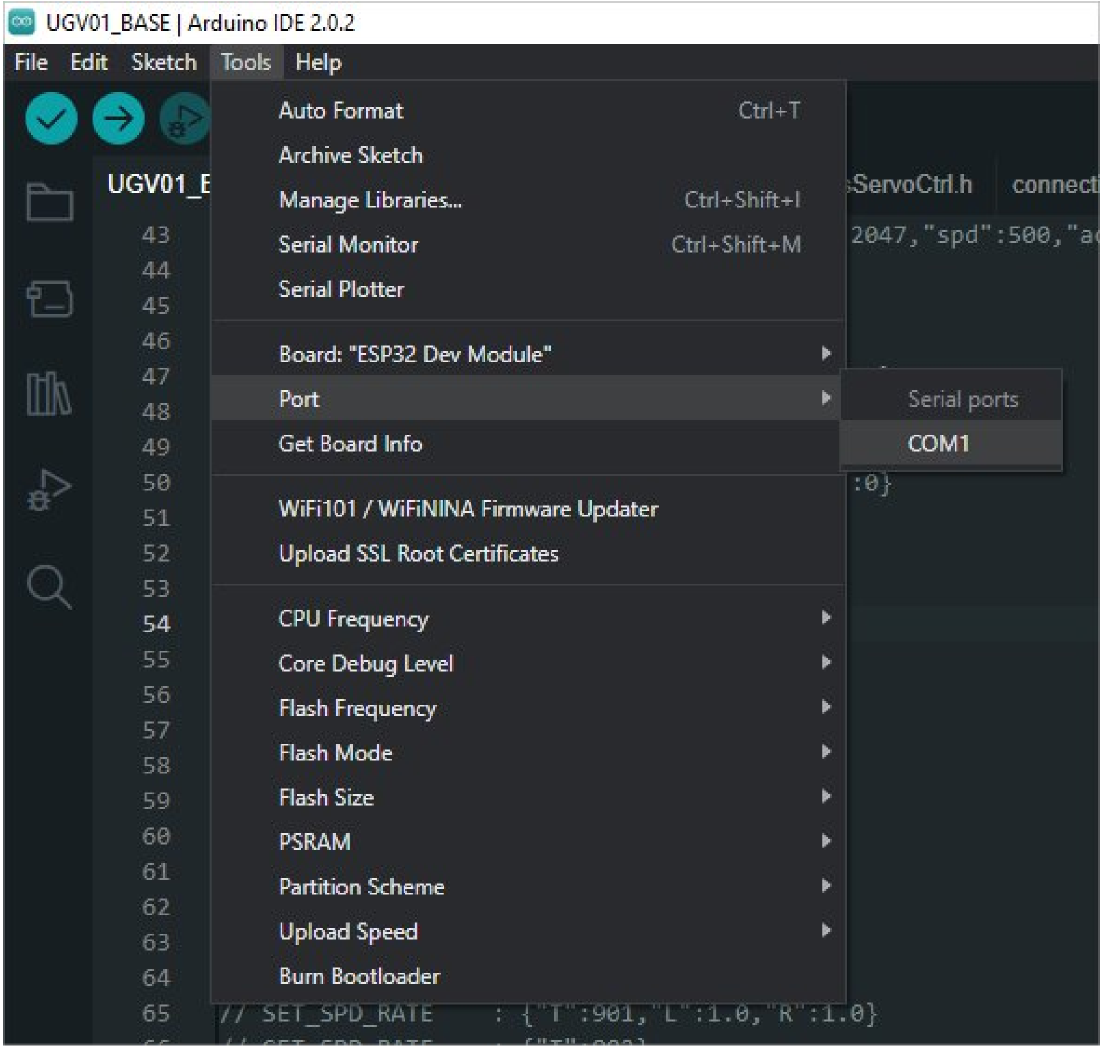
Upload Demo At the First Time

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).

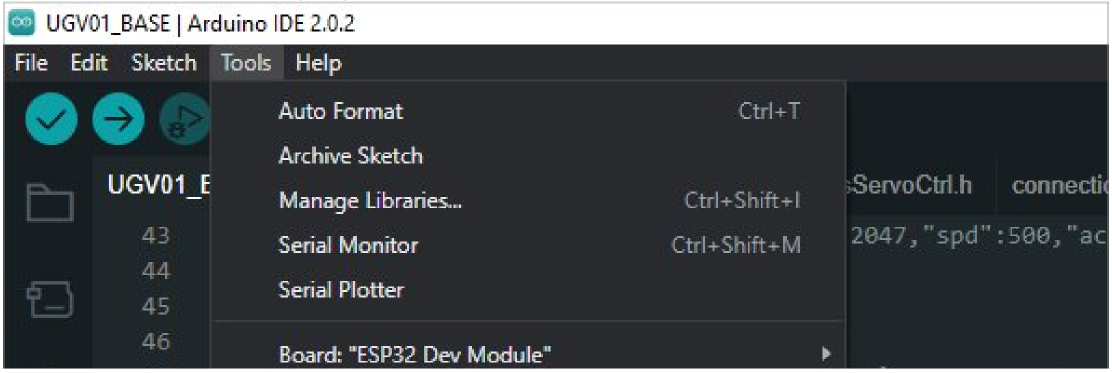


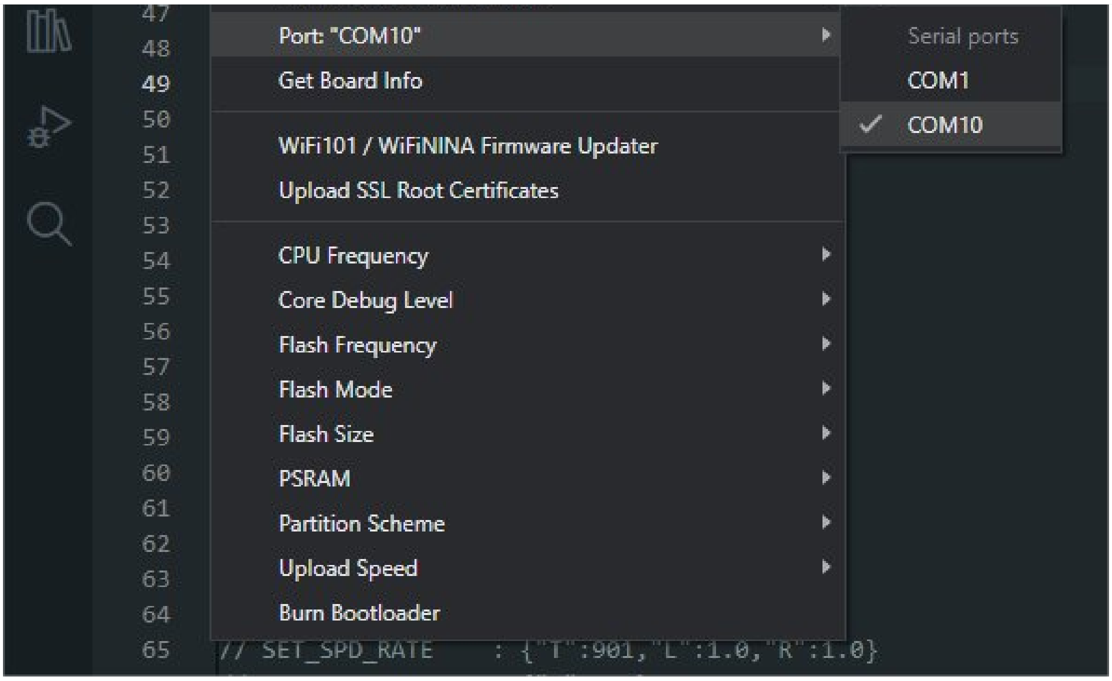


2. Download the demo, open arduino\PWM\D1-LED path under the D1-LED.ino.
3. Click Tools -> Port, remember the existing COM, do not need to click this COM (different computers show different COM, remember the existing COM on your computer).


UGV01_BASE | Arduino IDE 2.0.2
File Edit Sketch Tools Help
UGV01_E
sServoCtrl.h
connecti
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Board: "ESP32 Dev Module"
Port
Get Board Info
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
CPU Frequency
Core Debug Level
Flash Frequency
Flash Mode
Flash Size
PSRAM
Partition Scheme
Upload Speed
Burn Bootloader
Serial ports
COM1
2047,"spd":500,"a
:0}
// SET_SPD_RATE : {"T":901,"L":1.0,"R":1.0}

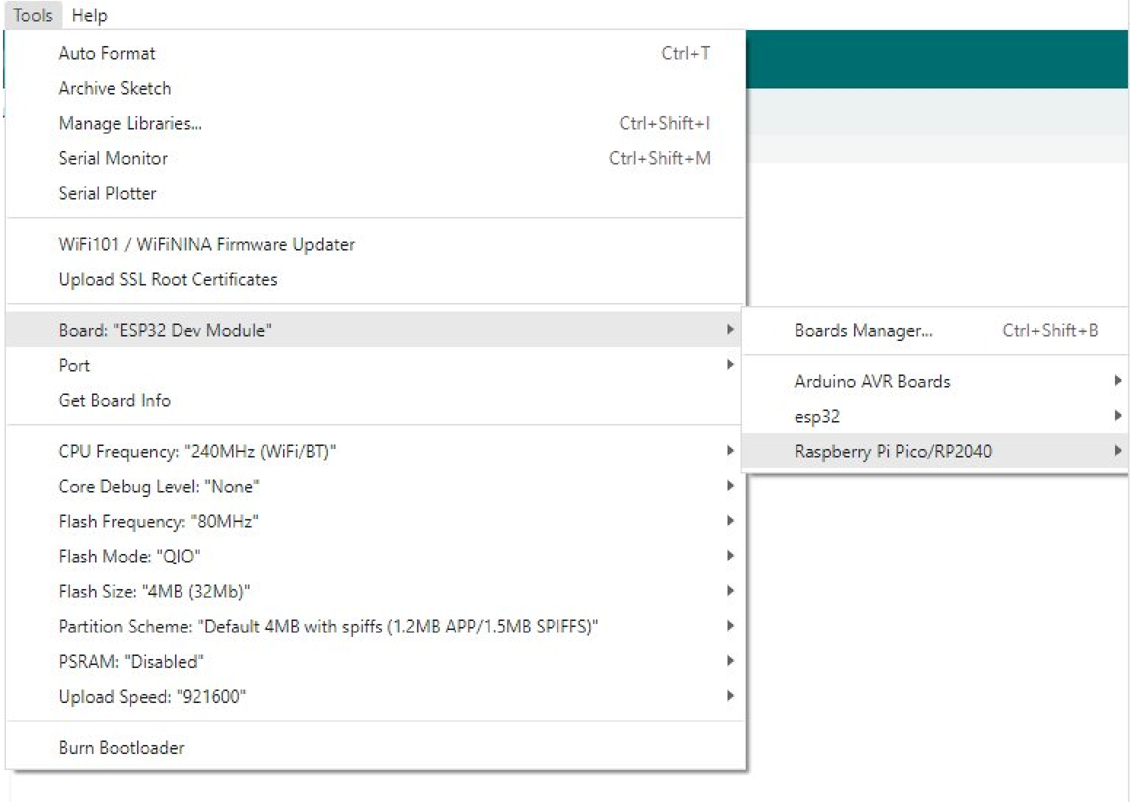
4. Connect the driver board to the computer with a USB cable, then click Tools -> Ports, select uf2 Board for the first connection, and after the upload is complete, connecting again will result in an additional COM port.


UGV01_BASE | Arduino IDE 2.0.2
File Edit Sketch Tools Help
UGV01_E
sServoCtrl.h
connecti
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Board: "ESP32 Dev Module"
2047,"spd":500,"ac



```
// SET_SPD_RATE    : {"T":901,"L":1.0,"R":1.0}
```

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


Tools
Help
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Port
Get Board Info
Burn Bootloader
Boards Manager... Ctrl+Shift+B
Arduino AVR Boards
esp32
Raspberry Pi Pico/RP2040

Board: "ESP32 Dev Module"

CPU Frequency: "240MHz (WiFi/BT)"

Core Debug Level: "None"

Flash Frequency: "80MHz"

Flash Mode: "QIO"

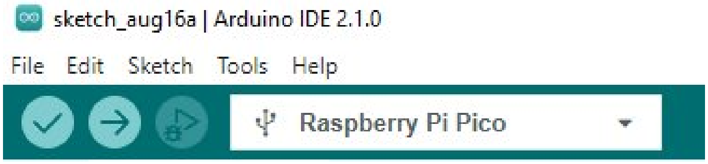
Flash Size: "4MB (32Mb)"

Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"

PSRAM: "Disabled"

Upload Speed: "921600"

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