

Overview

Introduction

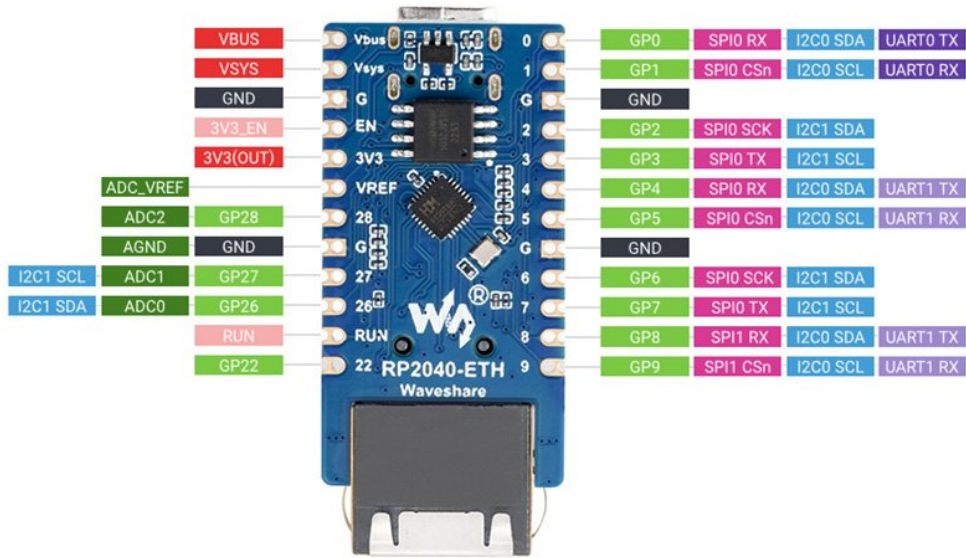
RP2040-ETH is a mini RP2040-ETH development board, which integrates TCP/IP protocol stack for network communication. 14 × multi-function GPIO pins and the castellated holes on its PCB edges allow easy and quick integration into projects.



Features

- Adopts RP2040 micro-controller chip designed by the official Raspberry Pi.
- Dual-core Arm Cortex M0+ processor, flexible clock running up to 133 MHz.
- 264KB of SRAM, and 4MB of onboard Flash memory.
- Onboard CH9120 with integrated TCP/IP protocol stack.
- With the host software and serial commands, you can set the network parameters such as chip operating mode, port, and IP.
- The castellated module allows soldering directly to carrier boards.
- Drag-and-drop programming using mass storage over USB.
- 14 × multi-function GPIO pins, compatible with some Pico HATs.
- More hardware peripherals:
 - 1x SPI (SPI0)
 - 2x I2C
 - 2x UART
 - 13x Controllable PWM channels
- Temperature sensor.
- 8 × Programmable I/O (PIO) state machines for custom peripheral support.

Pinout Definition



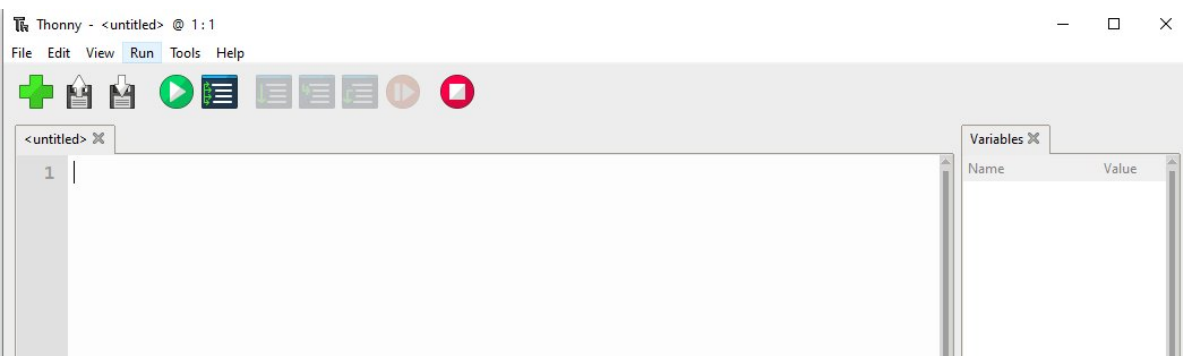
■ Power	■ Ground	■ UART / UART (default)	■ GPIO, PIO, and PWM
■ ADC	■ SPI	■ I2C	■ System Control

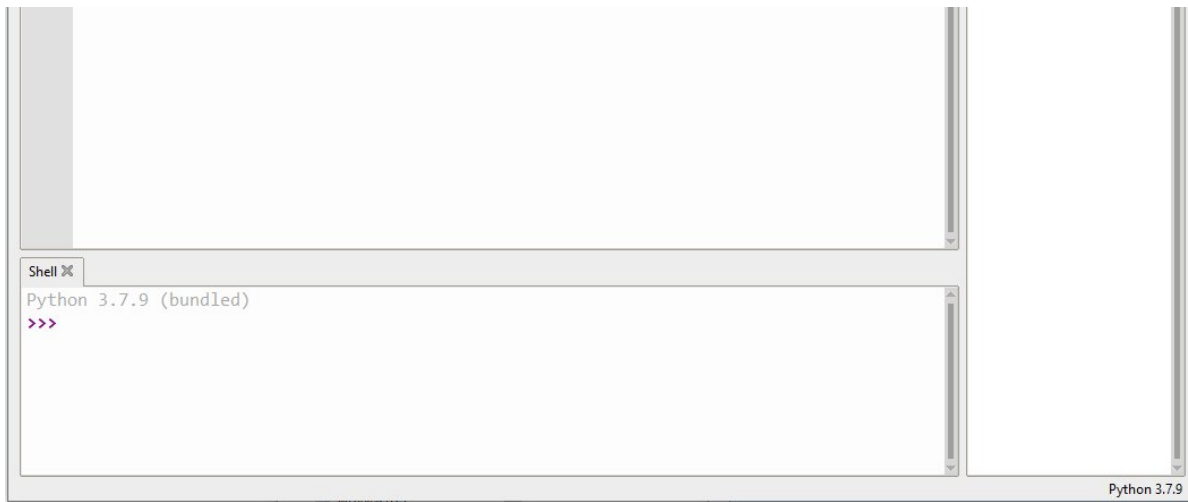
RP2040-ETH User Manual

Environment Building

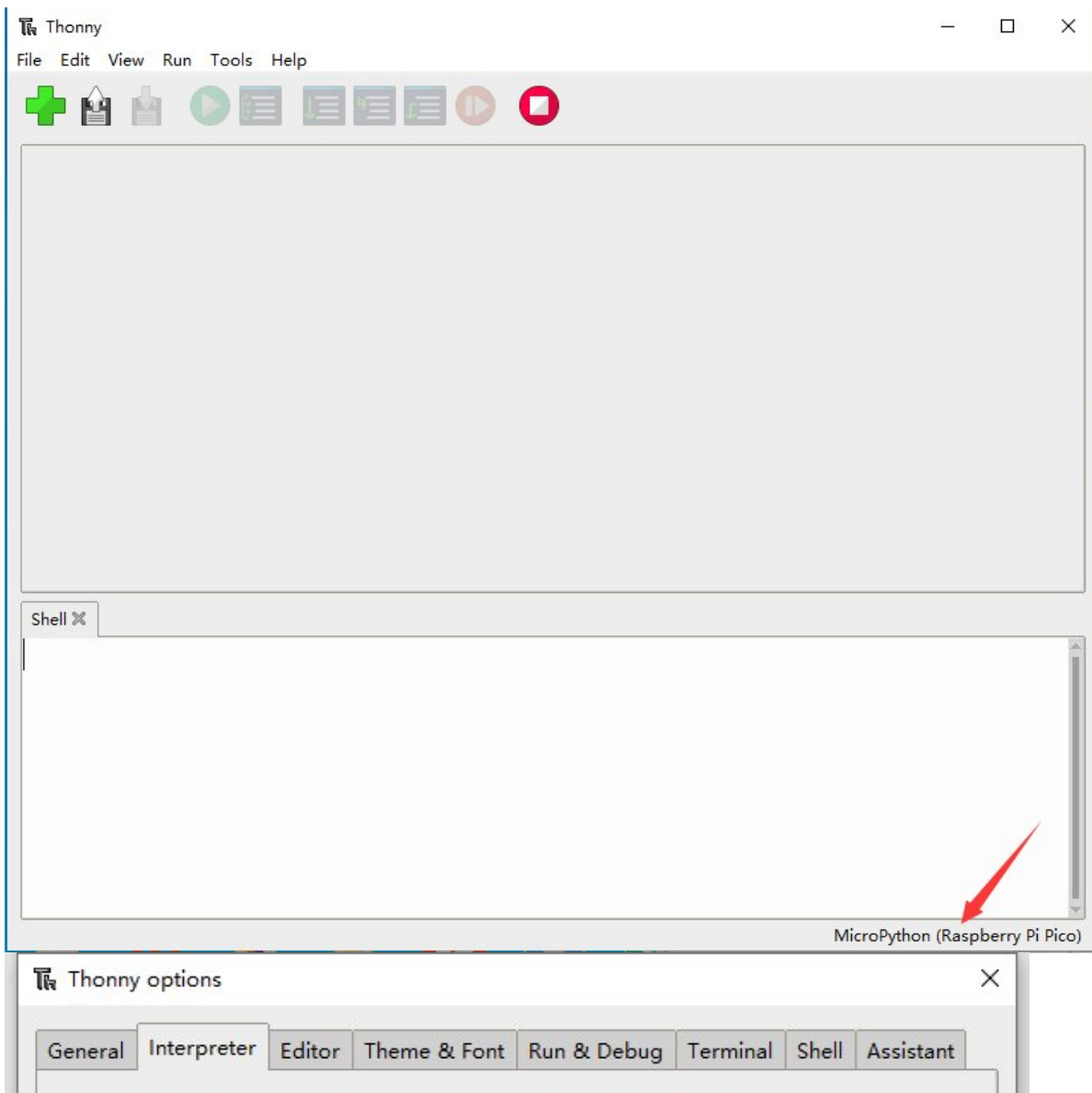
- C/C++ Development Environment Installation
 - [Pico-Get-Start-Windows](#)
 - [Pico-Get-Start-RPI](#)
- MicroPython Development Environment Installation:
 - Download the software according to your system from [Thonny](#).

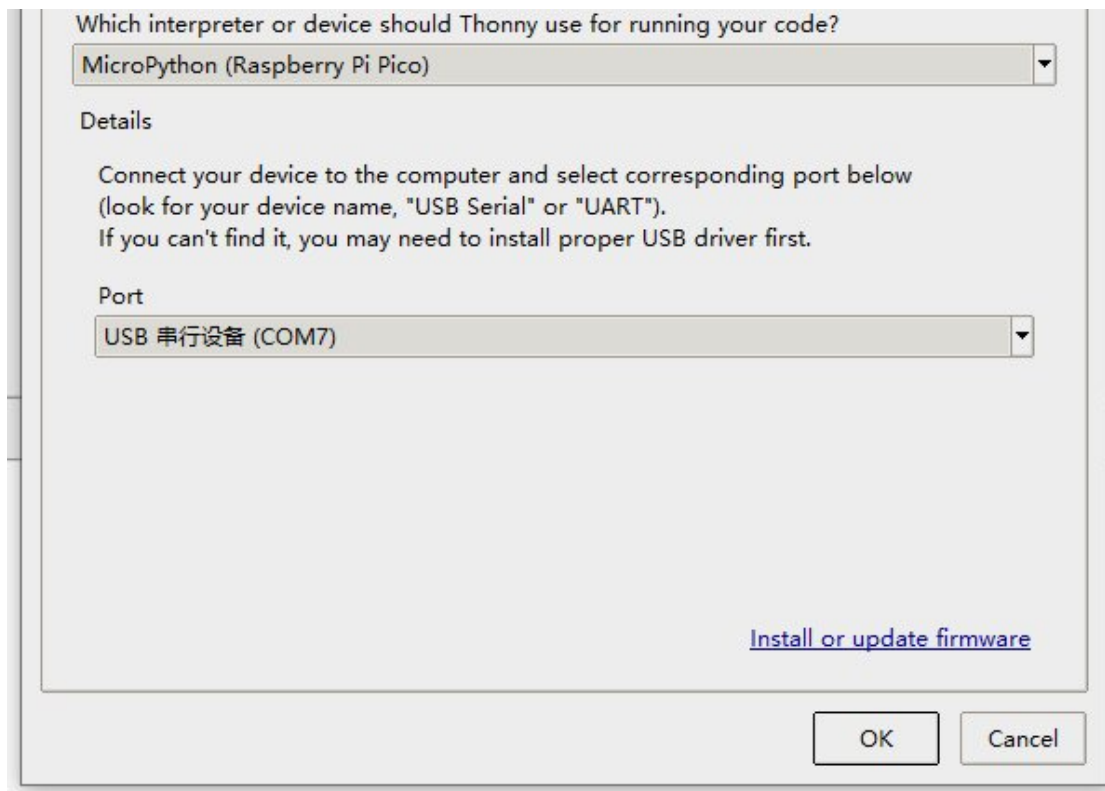
*After installing, please configure the language and the environment for the first time. Note that we should choose the Raspberry Pi option in the board environment.





- Configure the Micrpython environment and select the Pico port.
 - First connect the Raspberry Pi Pico to the computer, left-click on the configuration environment option in the lower right corner of Thonny --> select configure an interpreter.
 - In the pop-up window bar, select MicroPython (Raspberry Pi Pico), and select the corresponding port.





- Click OK to return to the main interface of Thonny, download the [firmware library](#) to Pico, and then click the stop button to display the currently used environment in the Shell window.
- Pico download firmware library method in Windows: Press and hold the BOOT button and connect to the computer, release the BOOT button, and a removable disk will appear on the computer and copy the firmware library into it.
- **The Raspberry Pi system is similar to the Windows system, the Raspberry Pi needs to install the latest version of the system with the desktop.**
- Arduino IDE Development Environment Installation:
 - Open Arduino IDE, enter File -> Preferences

In the pop-up dialog box, enter the following URL in the "Additional Boards Manager URLs" field:

```
https://github.com/earlephilhower/arduino-pico/releases/download/global/package\_rp2040\_index.json
```

Click "OK" to close the dialog box.

Go to Tools -> Board Manager in IDE.

Type "pico" in the search box and select "Add".

Hardware Connection

- The internal connection between CH9120 and RP2040 as shown below:

RP2040 connection pins correspondence

CH9120	RP2040	Function
RXD	GP21	UART data input
TXD	GP20	UART data output
TCPCS	GP17	In TCP client mode, indicates connection status, low level for successful connection
CFG0	GP18	Network configuration enabled pin, low level for serial debugging mode
RSTI	GP19	Reset, active LOW

Demo Usage

Demo Introduction

- In CH9120.c or CH9120.cpp file, you can configure C/C++ parameters such as the IP, gateway, subnet mask, port number, and UART baud rate according to your needs.

```
UCHAR CH9120_LOCAL_IP[4] = {192, 168, 1, 200}; // LOCAL IP
UCHAR CH9120_GATEWAY[4] = {192, 168, 1, 1}; // GATEWAY
UCHAR CH9120_SUBNET_MASK[4] = {255, 255, 255, 0}; // SUBNET MASK
UCHAR CH9120_TARGET_IP[4] = {192, 168, 1, 10}; // TARGET_IP
UWORD CH9120_PORT1 = 1000; // LOCAL PORT1
UWORD CH9120_TARGET_PORT = 2000; // TARGET PORT
UDOUBLE CH9120_BAUD_RATE = 115200; // BAUD RATE
#Set the CH9120 operating mode in the void CH9120_init(void) function, combined
with the modes defined in the header file, and select the desired one:
CH9120_TX_4_bytes(TCP_CLIENT, Mode1); //Select the mode: 0x00:TCP server 0x01:TC
P client 0x02:UDP server 0x03:UDP client
#the header file is defined, you can directly use
#define TCP_SERVER 0
#define TCP_CLIENT 1
#define UDP_SERVER 2
#define UDP_CLIENT 3
```

- In the py file, you can configure micro python parameters such as the operating mode, IP, gateway, subnet mask, port number, and UART baud rate.

```
MODE = 1 #0:TCP server 1:TCP client 2:UDP server 3:UDP client
GATEWAY = (192, 168, 1, 1) # GATEWAY
TARGET_IP = (192, 168, 1, 1)# TARGET_IP
LOCAL_IP = (192, 168, 1,200) # LOCAL_IP
SUBNET_MASK = (255,255,255,0) # SUBNET_MASK
LOCAL_PORT1 = 1000 # LOCAL_PORT1
TARGET_PORT = 2000 # TARGET_PORT
BAUD_RATE = 115200 # BAUD_RATE
```

- After the first step is set, it will enter the send and receive mode, and the CH9120 will send back the received information (Acsl code string only).

In Raspberry Pi

- Open the Raspberry Pi terminal, and run:

```
cd ~
sudo wget https://files.waveshare.com/upload/8/88/RP2040_ETH_CODE.zip
unzip RP2040_ETH_CODE.zip
cd ~/RP2040_ETH_CODE
```

C

- The following guide is operated in the **Raspberry Pi**. As CMake is multi-platform and portable, it can also be compiled on the PC. But the operation steps are different, you can explore it by yourselves.
- Before compiling, you should make sure it is in C/RP2040-ETH-Demo/build directory:
- Enter the build directory and add SDK:
- .../.../pico -sdk is your SDK directory.

```
cd C/RP2040-ETH-Demo/build/
export PICO_SDK_PATH=../../pico-sdk
#Note: write the correct SDK path
cmake ..
#Execute make to generate the executable file and it takes a long time to compile for the first time
make -j9
#Compilation is finished, uf2 file is created.
```

- After connecting RP2040-ETH to the Raspberry Pi, you should press the BOOT and RESET keys at the same time. Release the RESET key first, and then the BOOT key. A portable disk will appear on the computer, and you can copy the firmware to it.

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

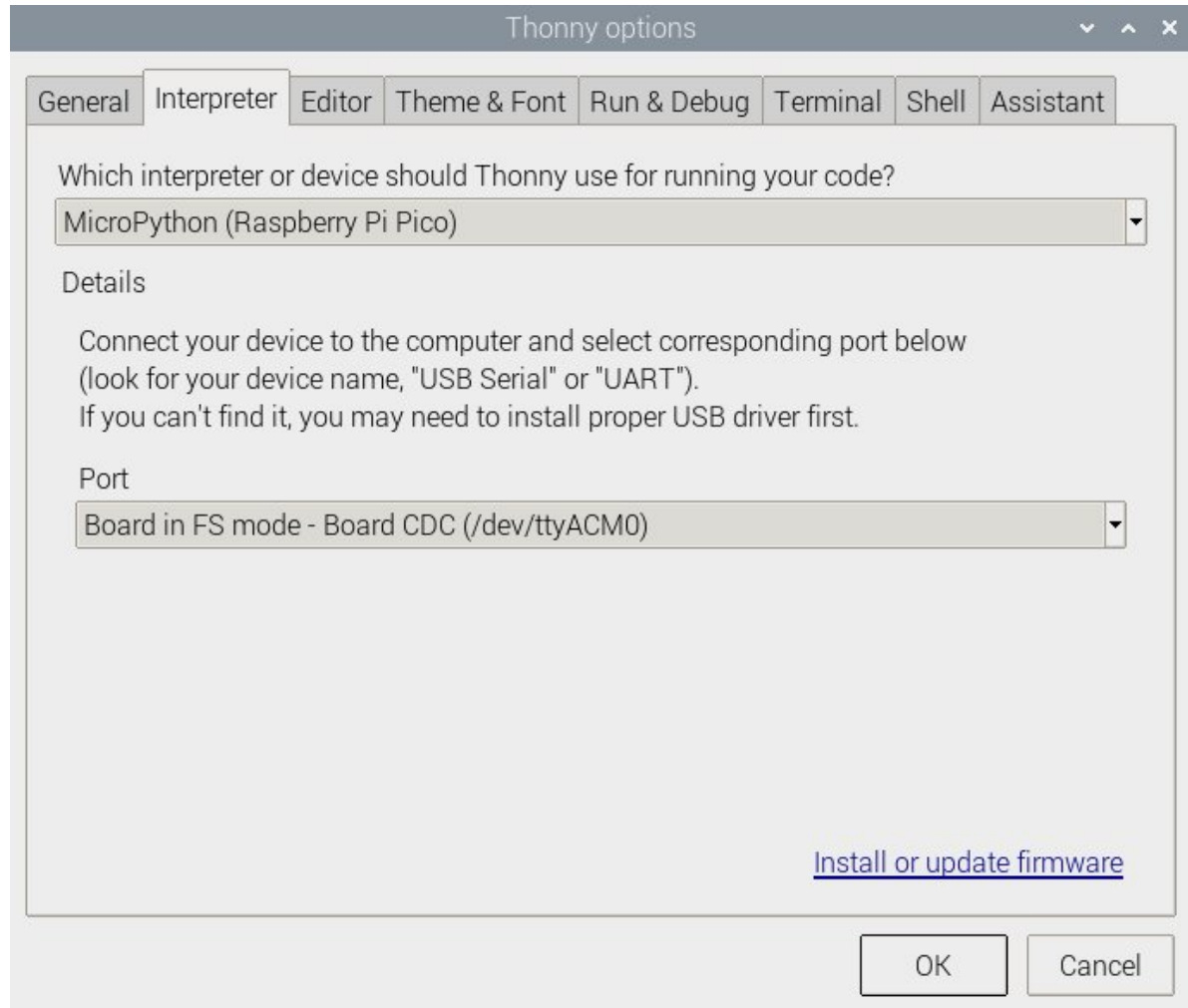
Python

- After connecting the RP2040-ETH to the Raspberry Pi, press the BOOT key and the RESET key at the same time. Release the RESET key, then the BOOT key and a removable disk are shown on the computer.
- Copy ~/RP2040_ETH_CODE/Python/rp2-pico-20230209-unstable-v1.19.1.uf2 file to the RP2040.
- Open the Thonny IDE on your Raspberry Pi (click on the 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 below:



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

```
sudo apt upgrade thonny
```

3. Click File -> Open... -> RP2040_ETH_CODE/Python/RP2040-ETH-Demo.py, run the demo.

In Windows Environment

C

- Firstly, install [Visual Studio Code](#).
- After installing, open the Visual Studio Code expansion interface (Ctrl + Shift + X).

- Type CMake Tools and install.
- Press F1 and type open settings UI to open the settings UI screen.
- Search for CMake.configureEnvironment.
- Click Add item.

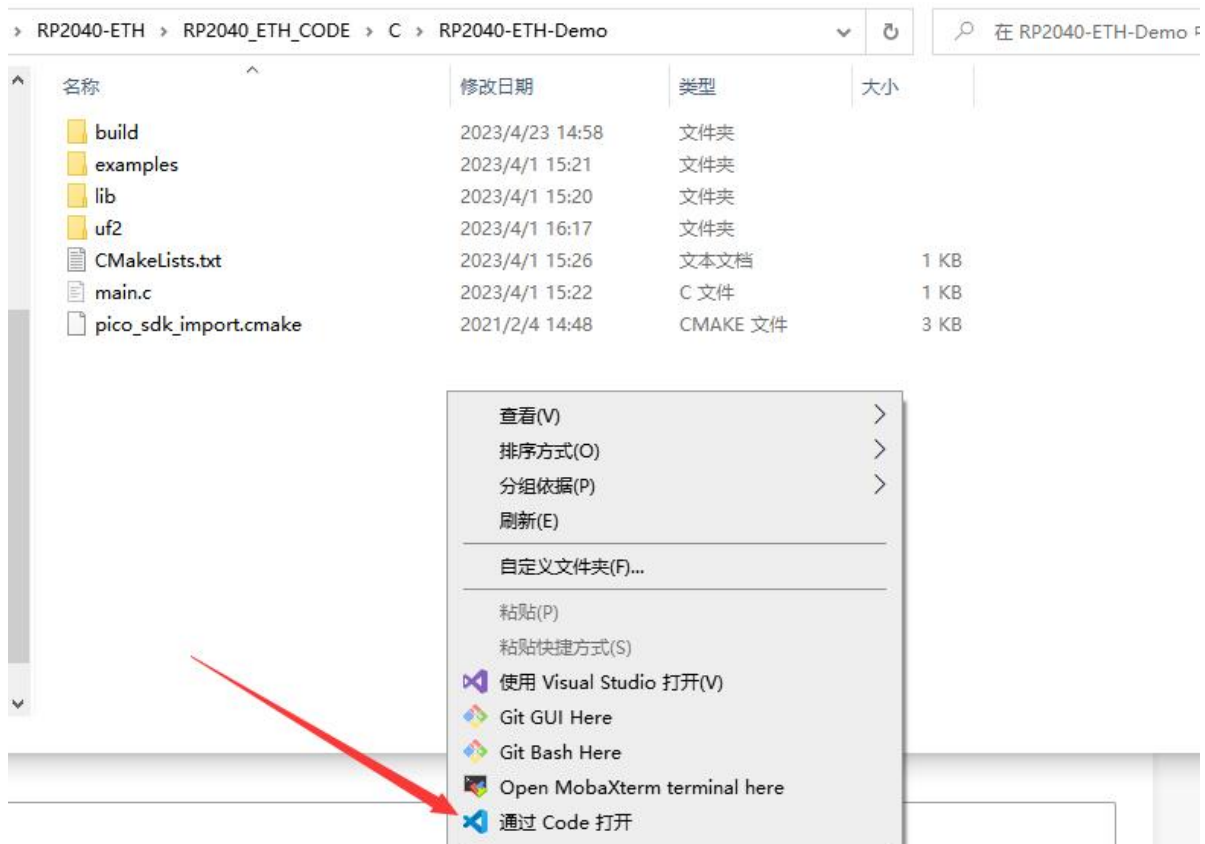
Item	Value
PICO_SDK_PATH	[storage PICO-SDK path]\pico-sdk

- "[storage PICO-SDK path]\pico-sdk" is the path you store SDK, and it can not be compiled when the path is wrong.
- Search cmake.generator, and fill in:

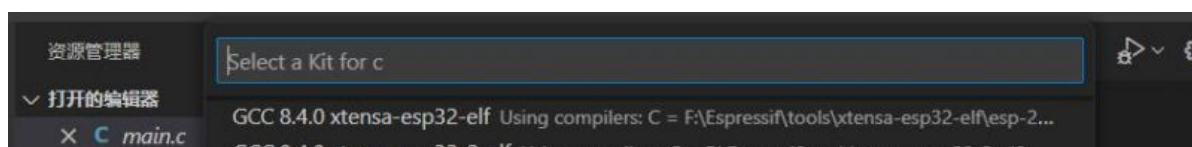
NMake Makefiles

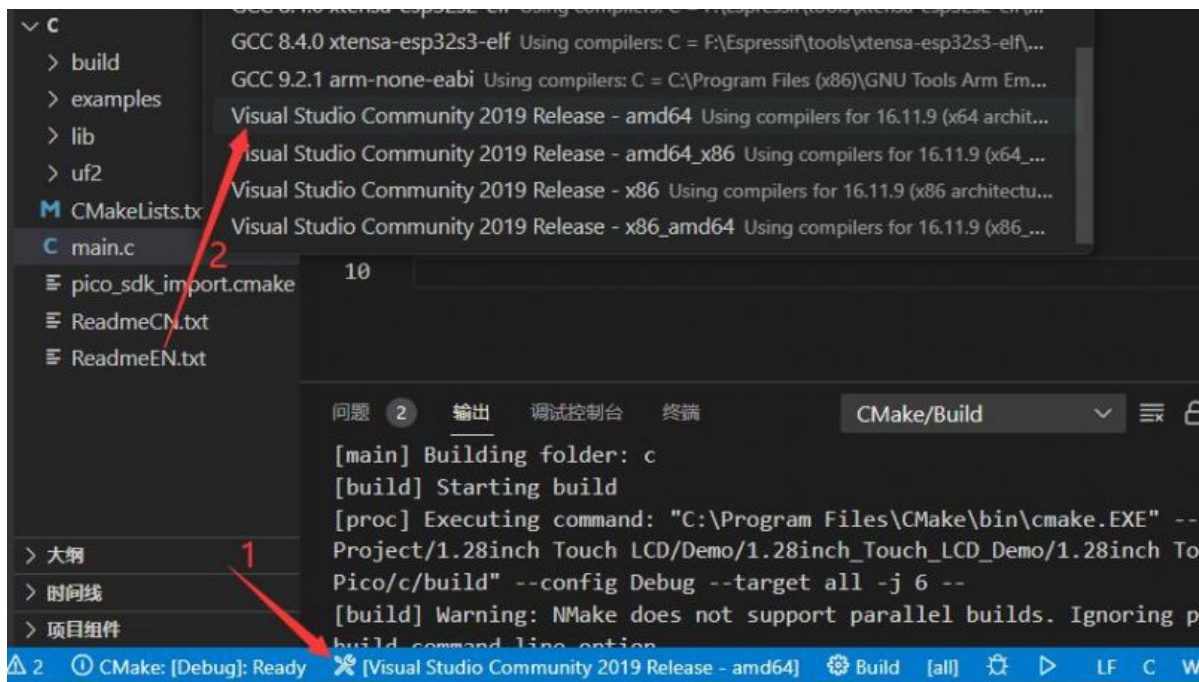
#Ctrl+S to save the setting, clear the build file folder, reboot Visual Studio Code, and compile again.

- Click to download the [demo](#), unzip and then enter the RP2040_ETH_CODE file folder.
- After entering RP2040_ETH_CODE\C\RP2040-ETH-Demo, open the project with vs code.

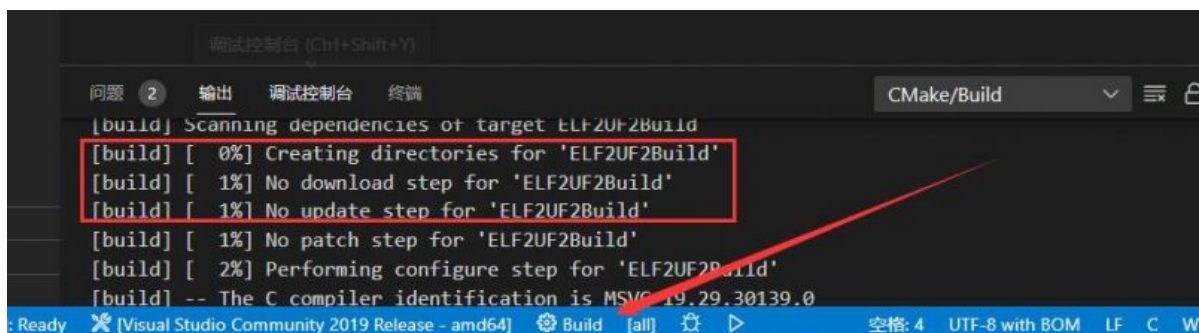


- Choose the Compiler.

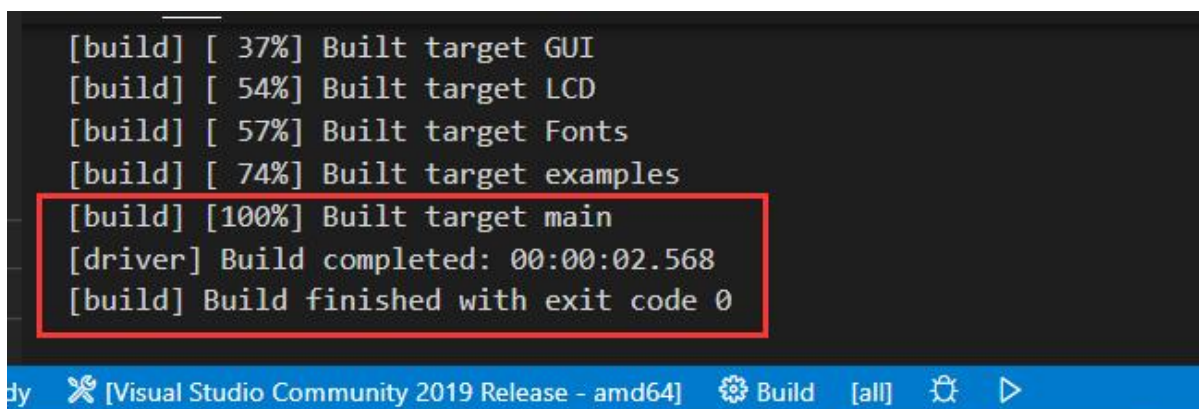




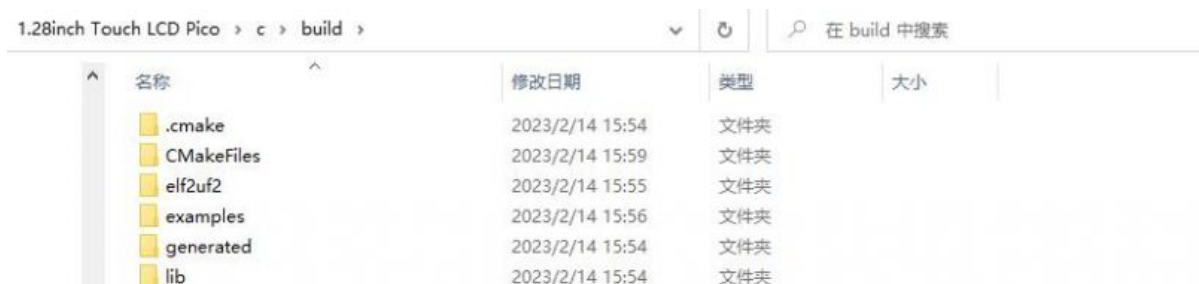
- Start to compile.



- Finish.



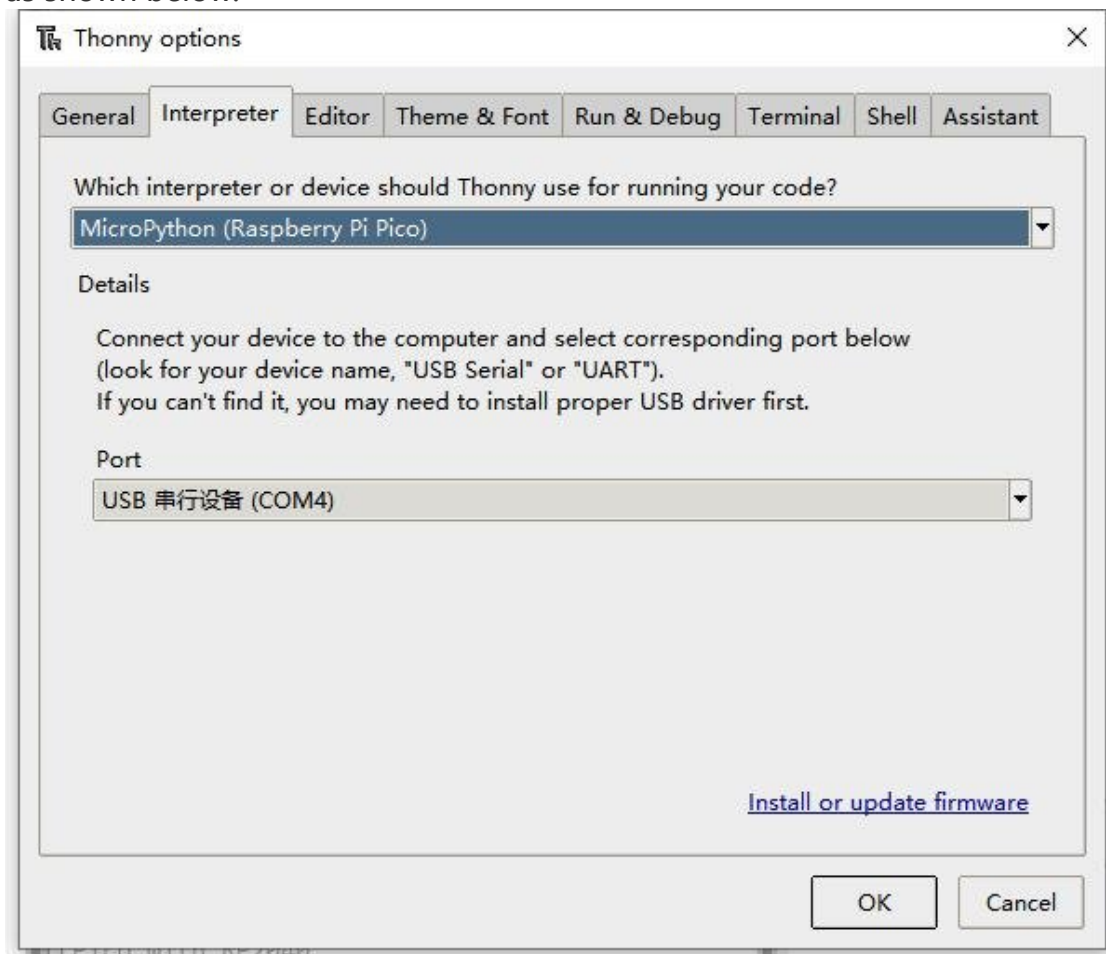
- Copy the main.uf2 file in build to Pico, and then it can automatically run the demo.



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

Python

1. Press the BOOTSET button on the Pico and connect the Pico to the USB port of the computer with a Micro USB cable. Release the button when the computer identifies a movable disk (RPI-RP2).
2. Copy the rp2-pico-20230209-unstable-v1.19.1.uf2 file in the Python directory to the recognizable movable disk (RPI-RP2).
3. Open Thonny IDE (Note: please use the latest version of Thonny, otherwise there is no Pico supporting package. Currently, the newest version in Windows is v3.3.3.)
4. Click Tool -> Setting -> Explainer. select the Pico and the corresponding port as shown below:



5. File -> Open -> RP2040-ETH-Demo.py, click to run, the effect is shown below.

```

MicroPython v1.19.1-859-g41ed01f13 on 2023-02-09; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR CONTENT

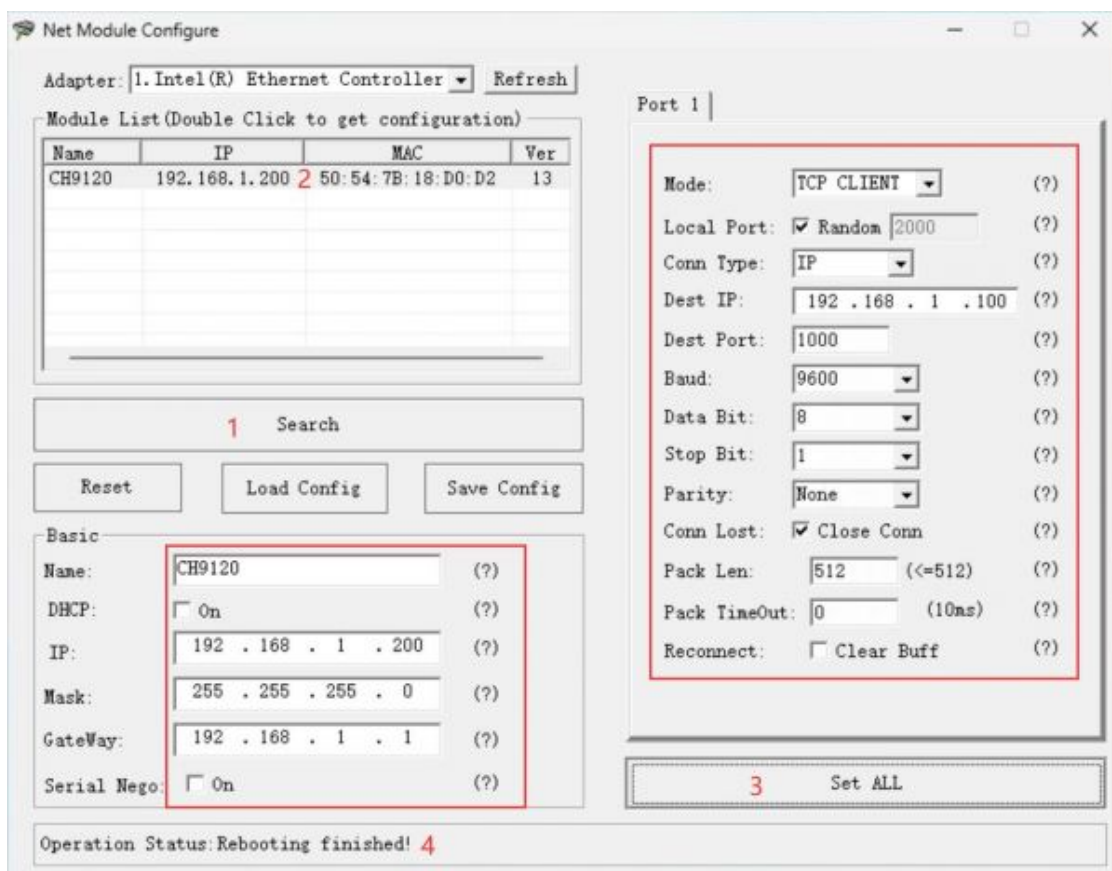
```

>>>

We provide a simple demo for you...

Host Debugging

- If you want to control with the host configuration rather than the serial commands, you can refer to the following method, and this method is only for Windows systems.
 - Download [Network configuration tool](#).
1. Power on RP2040-ETH and the Ethernet.
 2. Open the network debugging tool.
 3. Search the device -> double click the searched device -> set the information you required -> Configure the device parameters -> wait for rebooting.



- Follows steps 1 and 2 in the above figure, and configure the parameters in the red block. Please wait for 4 after step 3.

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](#)

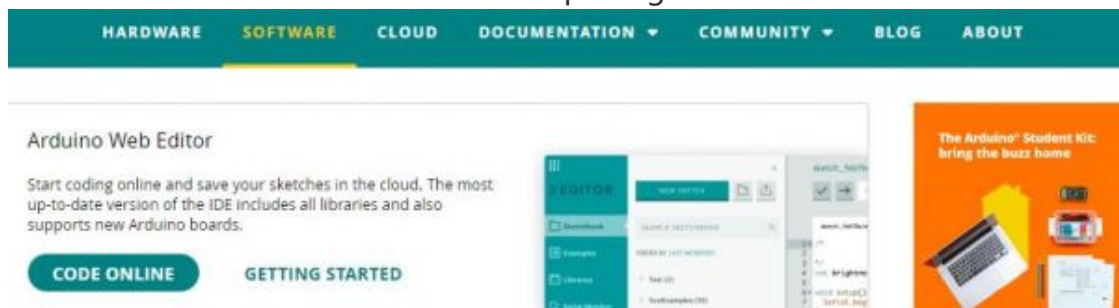
C/C++ Series

- [\[C/C++\] 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](#).



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](#).

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.

JUST DOWNLOAD **CONTRIBUTE & DOWNLOAD**



The illustration shows a small orange robot on the left stacking a tower of gold coins. In the center is a large grey jar with the Arduino logo on it, containing gold coins. On the right, another orange robot is holding a gold coin, ready to contribute.

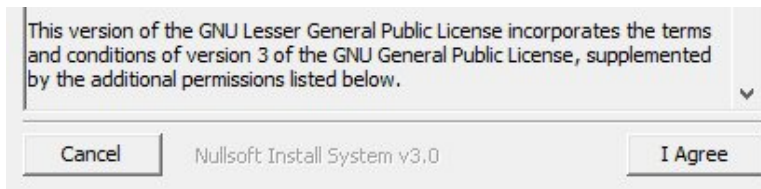
Learn more about [donating to Arduino](#).

3. Click to install after downloading.

 Arduino Setup: License Agreement

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.



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

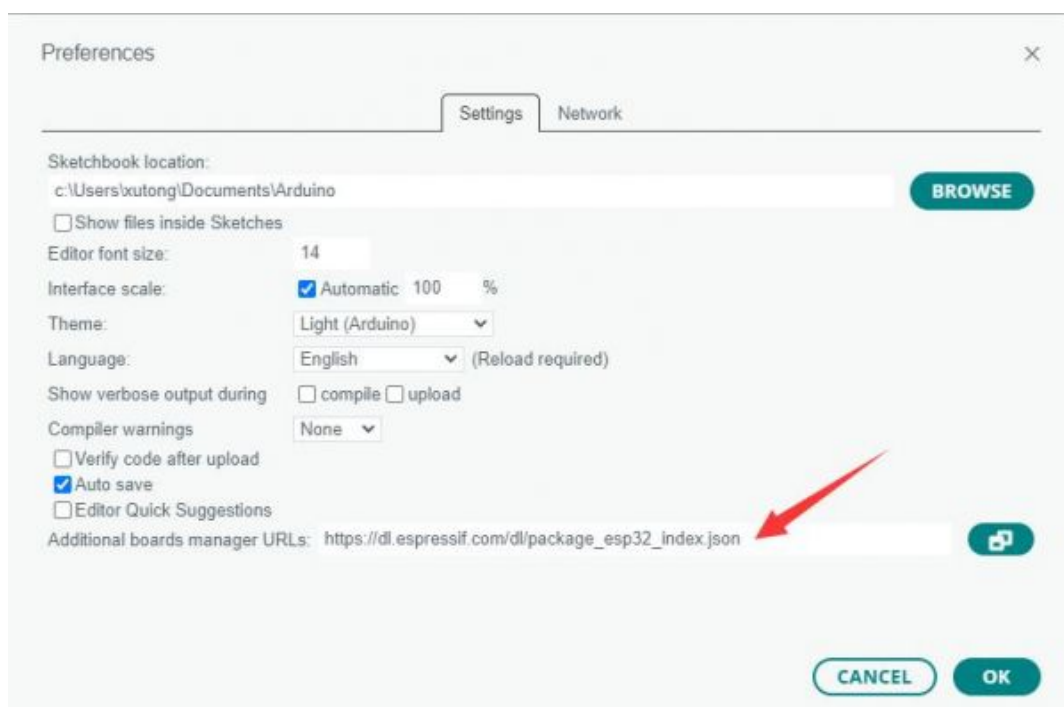
Install Arduino-Pico Core on Arduino IDE

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.

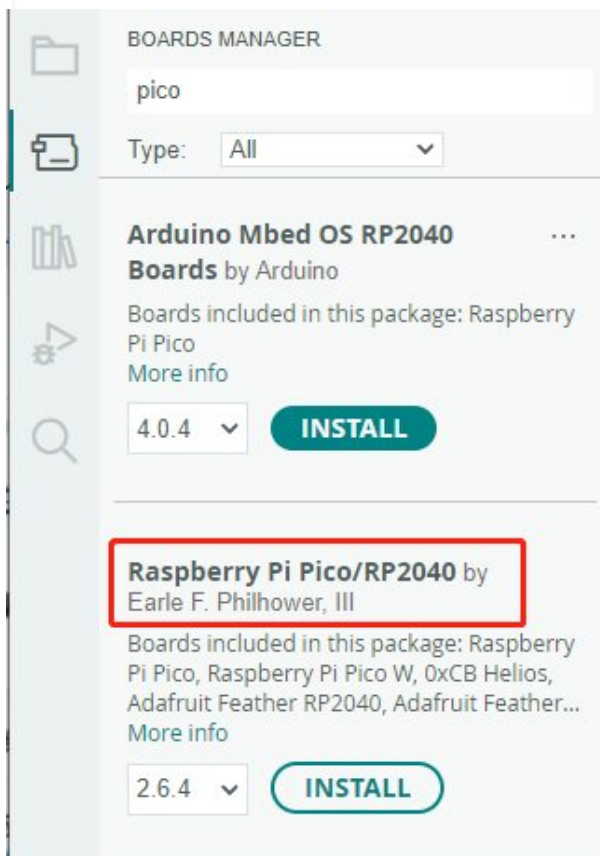
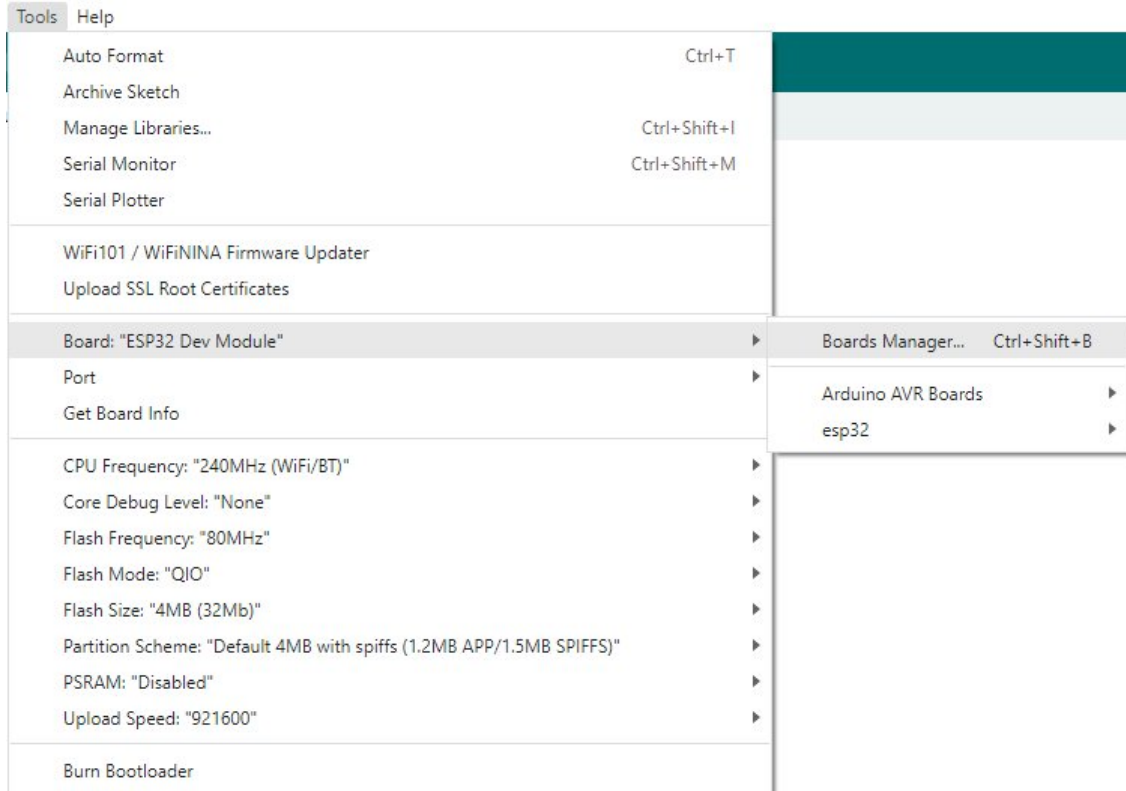
```
https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json
```



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

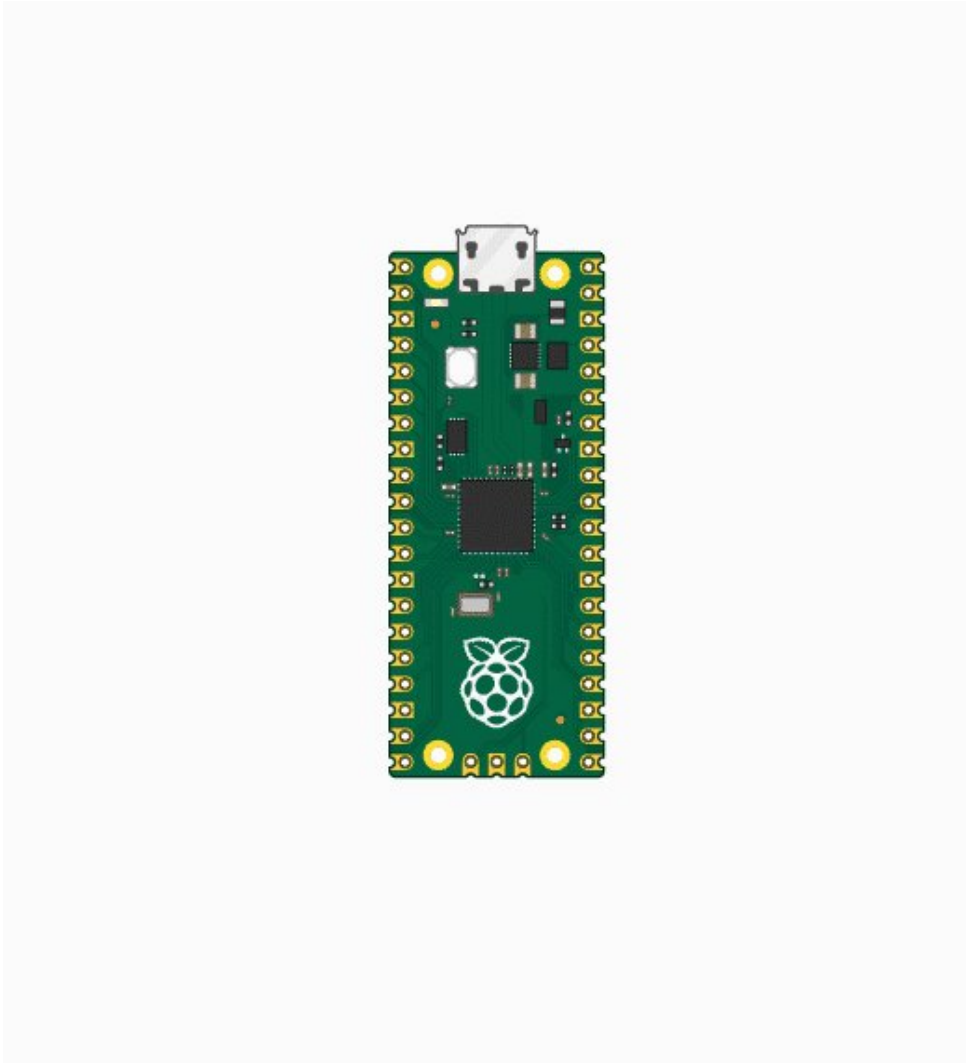
https://dl.espressif.com/dl/package_esp32_index.json,https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json

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

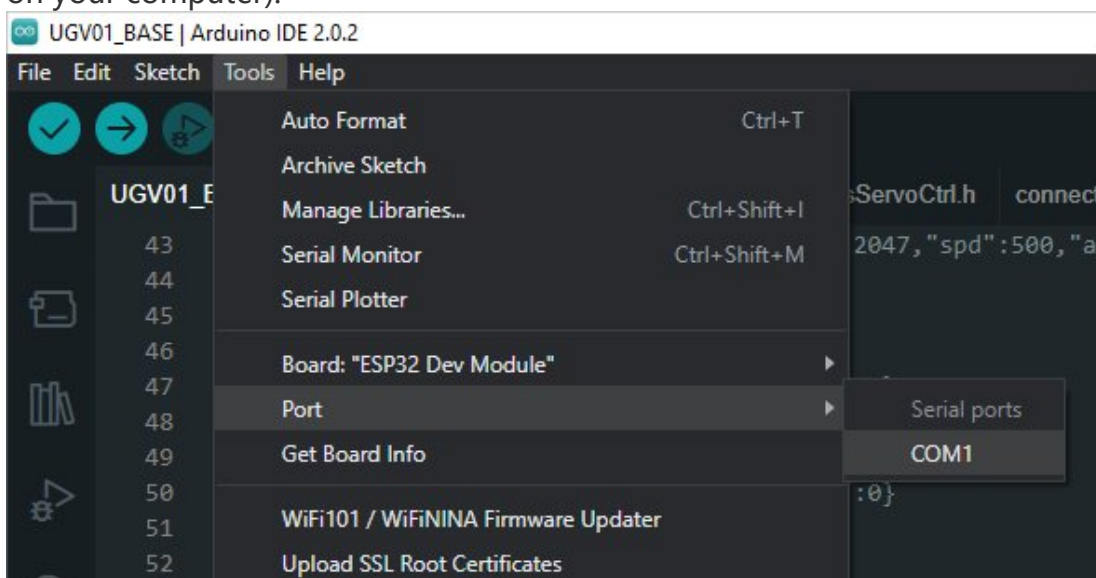


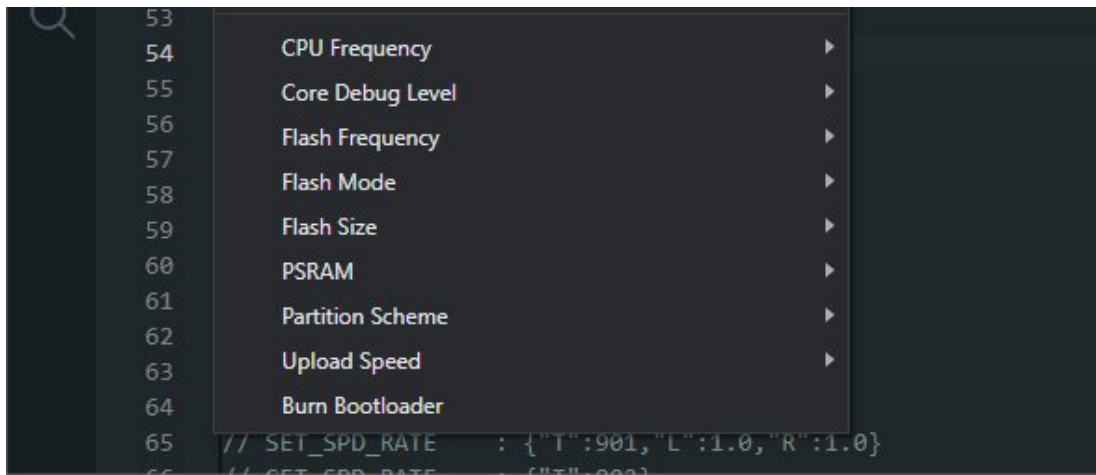
Upload Demo At the First Time

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).

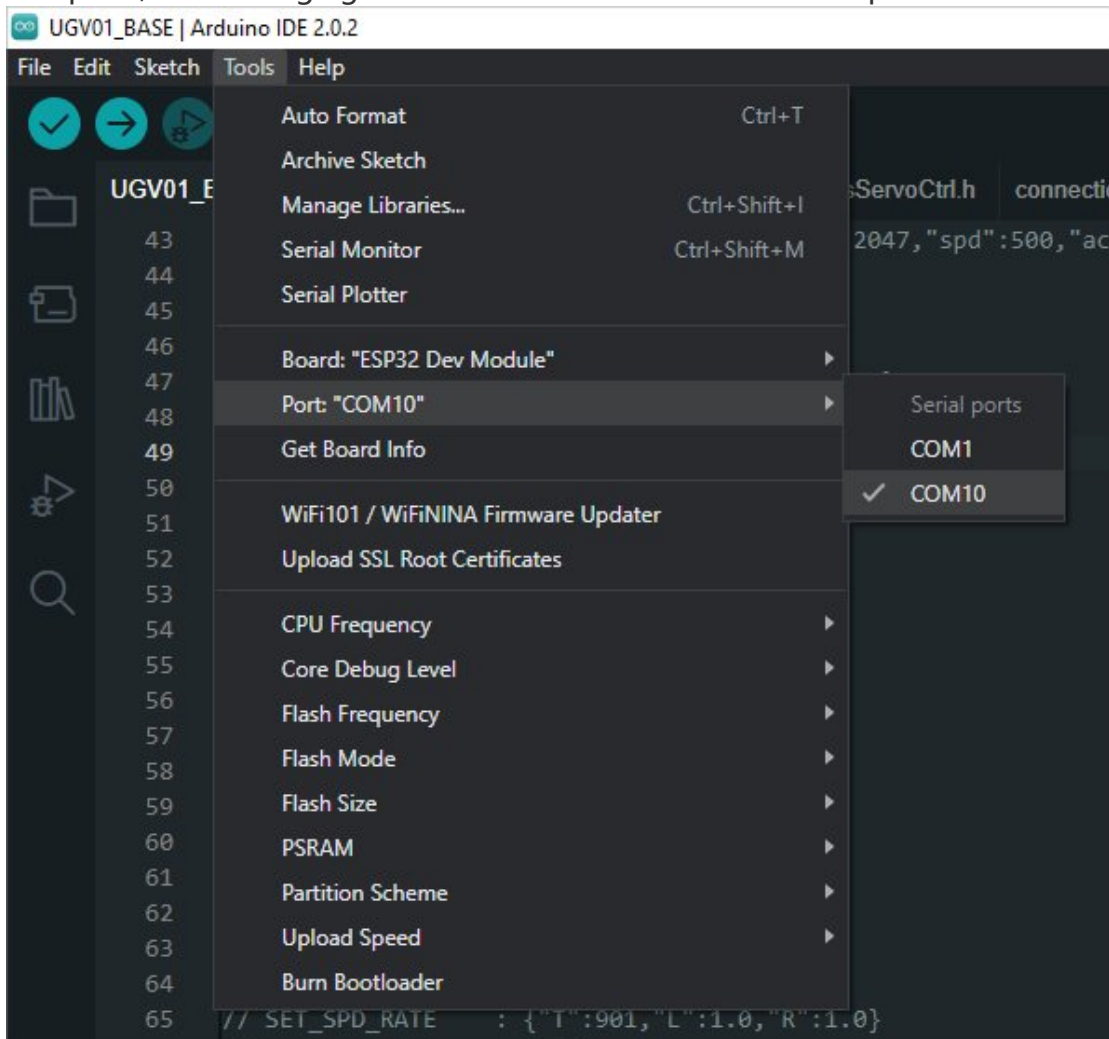


2. Download the demo, open arduino\PWM\D1-LED path under the D1-LED.ino.
3. Click Tools -> Port, remember the existing COM, do not need to click this COM (different computers show different COM, remember the existing COM on your computer).

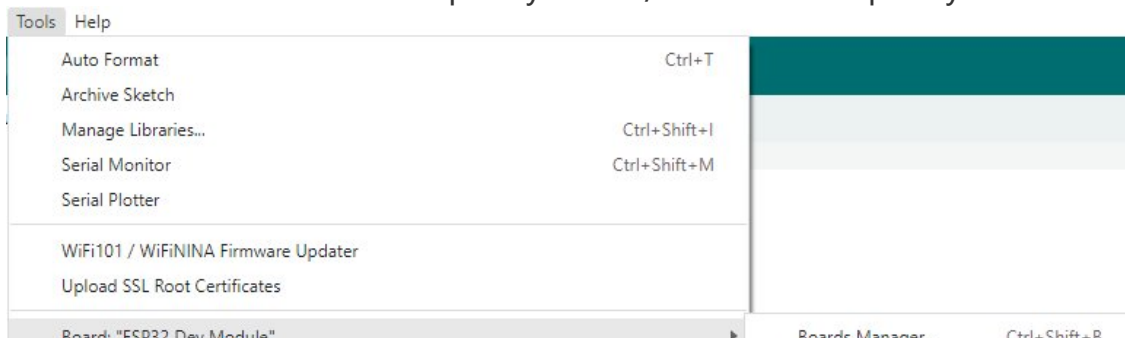


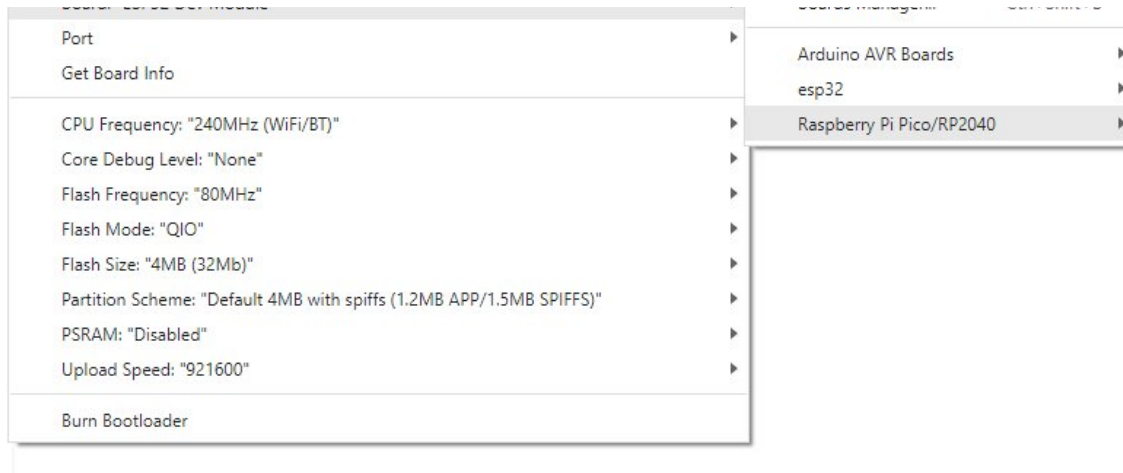


4. Connect the driver board to the computer with a USB cable, then click Tools -> Ports, select uf2 Board for the first connection, and after the upload is complete, connecting again will result in an additional COM port.

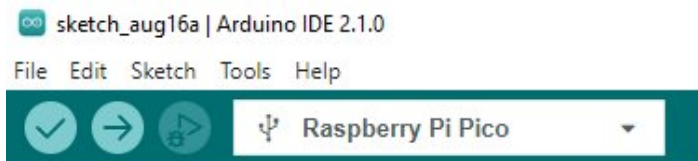


5. Click Tool -> Dev Board -> Raspberry Pi Pico/RP2040 -> Raspberry Pi Pico.





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](#)

Resource

Document

- [CH9120DS1_EN](#)
- [CH9120 Serial Instruction Set](#)
- [RP2040-ETH Schematic](#)
- [RP2040-ETH-STEP File \(3D model\)](#)

Demo

- [RGB LED test demo](#)
- [RP2040-ETH test demo](#)

Tool

- [Network configuration tool](#)
- [SSCOM Serial Assistant](#)

Raspberry Pi

- [Raspberry Pi Pico Get Started with MicroPython on Raspberry Pi Pico](#)
- [Official website of Pico](#)
- [Getting started with Pico](#)
- [Pico C SDK](#)
- [Pico Python SDK](#)
- [Pico Pinout](#)
- [Pico Datasheet](#)
- [RP2040 Datasheet](#)
- [RP2040 Hardware design](#)

Examples

- [Raspberry Pi C/C++ demo \(github\)](#)
- [Raspberry Pi micropython demo \(github\)](#)

Software

- [Thonny](#)
- [Zimo221.7z](#)
- [Image2Lcd.7z](#)

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](#)