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

	VBUS VBUS	2.111.52 () 0 (T	0 SPIO RX 12CO SDA UARTO TX
	VSYS Vsys	C	1 SPIO CSN 12C0 SCL UARTO RX
	GND G G 3V3_EN D EN		D 2 SPIO SCK 12C1 SDA
	3V3(OUT) 3V3 9	GP	3 SPI0 TX 12C1 SCL
ADC.	0C2 GP28 0 28		4 SPI0 RX 12C0 SDA UART1 1X 5 SPI0 CSn 12C0 SCL UART1 RX
AG	SND GND G	G C GN	D
I2C1 SCL AL	0C1 GP27 27	Ran 6 C GP	6 SPI0 SCK I2C1 SDA
12C1 SDA AL	RUN RUN		7 SPI0 TX 12CT SCL 8 SPI1 RX 12C0 SDA UART1 TX
	GP22 0 22 R	P2040-ETH 9 🚾 🔤 GP	9 SPI1 CSn I2C0 SCL UART1 RX
Power	Ground	UART / UART (defa	ult) 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.

Image: The second seco		<u> </u>		×
File Edit View Run Tools Help				
<untiled> X</untiled>	Variables 🗶]		
	Name		Value	A.

Shell X Python 3.7.9 (bundled)	
>>> 	
	Python 3.7.9

- 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.

The Thonny File Edit View Run Tools Help			×
Shell 🗶			
		1	
MicroPytho	n (Raspl	berry Pi	Pico)
The Thonny options	3	×	
General Interpreter Editor Theme & Font Run & Debug Terminal Shell Assist	ant		

MicroPython (Raspberry Pi Pico)	
Details	
Connect your device to the computer a (look for your device name, "USB Seria If you can't find it, you may need to ins	nd select corresponding port below l" or "UART"). tall proper USB driver first.
Port	
USB 串行设备 (COM7)	-
	Install or update firmware
	Install or update firmware

- 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 (Acsll 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
```

С

- 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 compil
e for the first time
make -j9
#Compilatio 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:

			Thon	ny options			*	^
General	Interpreter	Editor	Theme & Font	Run & Debug	Terminal	Shell	Assistant	
Which	interpreter o	r device	should Thonny	use for running	your code?			
Micro	Python (Ras	oberry Pi	i Pico)					-
Conr (lool- If yo	s nect your dev (for your dev u can't find it	ice to th ice nam , you ma	e computer and e, "USB Serial" c ay need to instal	select correspo or "UART"). I proper USB dri	nding port ver first.	below		
Boa	rd in FS mod	e - Board	d CDC (/dev/tty/	ACM0)				-
					Install	or upda	ate firmwar	<u>e</u>

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

С

- 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 C ode, and compile again.
```

- Click to download the demo a, 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.

>	RP2040-ETH > RP2040_ETH_CODE >	C → RP2040-ETH-Demo		ٽ ~	<i>∕</i> ₹	E RP2040-ETH-Demo
^	名称	修改日期	类型	大小		
	📙 build	2023/4/23 14:58	文件夹			
	examples	2023/4/1 15:21	文件夹			
	📙 lib	2023/4/1 15:20	文件夹			
	uf2	2023/4/1 16:17	文件夹			
	CMakeLists.txt	2023/4/1 15:26	文本文档		1 KB	
	inain.c	2023/4/1 15:22	C 文件		1 KB	
	📋 pico_sdk_import.cmake	2021/2/4 14:48	CMAKE 文件		3 KB	
		排序方式(O) 分组依据(P) 刷新(E)		>		
		自定义文件夹(F).	•			
*		粘贴(P) 粘贴快速方式(S) く 使用 Visual Stud Git GUI Here Git Bash Here	lio 打开(V) rm terminal here			
		▲ 通过 Code 打开	in certaina nere			

• Choose the Compiler.



> > > > > M ⊡ Ⅲ Ⅲ Ⅲ	GCC 8.4 build GCC 9.2 examples Visual S lib visual S uf2 Visual S CMakeLists.tx Visual S main.c 2 pico_sdk_import.cmake ReadmeEN txt	.0 xtensa-esp .1 arm-none- tudio Commu tudio Commu tudio Commu tudio Commu 10	32s3-elf eabi Usin unity 2019 unity 2019 unity 2019 unity 2019	Using comj ng compiler 9 Release - 9 Release - 9 Release - 9 Release -	pilers: C = F s: C = C:\Prd - amd64 U - amd64_xi - x86 Using - x86_amd	:\Espressif\t ogram Files sing compi 86 Using co 9 compilers 64 Using co	tools\xtens: (x86)\GNU lers for 16.1 ompilers fo for 16.11.9 ompilers fo	a-esp32s Tools Ar 11.9 (x64 r 16.11.9 (x86 arcl r 16.11.9	3-elf\ m Em archit (x64 nitectu (x86			
- - - - - - - - - - - - - -	物 1 间线 目组件	问题 2 [main] E [build] [proc] E Project/ Pico/c/t [build]	编出 Building Startin Executin (1.28ind build" Warning	调试控制台 g folder ng build ng comma ch Touch config g: NMake	會 终端 :: c ind: "C:\ LCD/Den ; Debug - e does no ion	\Program no/1.28i target ot suppo	CMal Files\(nch_Touc all -j rt para)	cMake\ CMake\ ch_LCD 6 llel b	bin∖cm _Demo/ uilds.	→ ake.E 1.28i Igno	E" nch ring	ć To
<u>A</u> 2	① CMake: [Debug]: Ready	🗶 [Visual S	studio Cor	mmunity 20	19 Release	- amd64]	😂 Build	[all]		LF	С	W

• Start to compile.

	问题 2 编出 调试控制台 终端	CMake/Build	~ ≡ 6
	[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] [2%] Performing configure step for 'ELF2UF2P-11d'		
· Pondy	[build] The C compiler identification is MSV(9-19.29.30139.0	肉肉 ITE Swith BC	MIECW

• 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	4			
<pre>[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</pre>	Ļ	build] [37%] Built target GUI		
<pre>[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</pre>]	build] [54%] Built target LCD		
[build] [74%] Built target examples [build] [100%] Built target main [driver] Build completed: 00:00:02.568 [build] Build finished with exit code 0	[build] [57%] Built target Fonts		
[build] [100%] Built target main [driver] Build completed: 00:00:02.568 [build] Build finished with exit code 0	[build] [74%] Built target examples		
[driver] Build completed: 00:00:02.568 [build] Build finished with exit code 0]	build] [100%] Built target main		
[build] Build finished with exit code 0	[driver] Build completed: 00:00:02.568		
	[build] Build finished with exit code 0		
	l	build] Build finished with exit code 0		

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

A	and the second	and lines	1.00
名称	修改日期	类型	大小
.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
ain.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:

	Interpreter	Editor	Theme & Font	Run & Debug	Terminal	Shell	Assistant
	-			10			
Which	interpreter or	device :	should Thonny u	se for running yo	our code?		
Micro	Python (Raspl	berry Pi I	Pico)				<u> </u>
Details	5						
Con	act your devi	ice to the	computer and a	alact correspon	ding port k	alow	
(lool	c for your dev	vice name	e, "USB Serial" or	r "UART").	ung port i	501044	
If yo	u can't find it,	you may	need to install	proper USB driv	er first.		
Port	6						
USB	串行设备 (CO	M4)					-
					Install or	update	firmware
					Install or	update	firmware

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.

Name	IP	MAC		Ver				_
CH9120	192.168.1.200 2	50:54:7B:18	:D0:D2	13	Node:	TCP CLIER	T T	(?
					Local Port:	₩ Random	2000	(?
					Conn Type:	IP	•	(?
					Dest IP:	192 .16	8 . 1 .100	(?
					Dest Port:	1000		(?
-					Baud:	9600	•	(?
	1 Sear	ch			Data Bit:	8	•	(?
					Stop Bit:	1	•	(?
Reset	Load Co	onfig	Save C	onfig	Parity:	None	•	(?
Basic				-	Conn Lost:	✓ Close (Conn	(?
Name:	CH9120		(?)		Pack Len:	512	(<=512)	(?
DHCP:	□ On		(?)		Pack TimeOu	t: 0	(10ns)	(?
IP:	192 . 168 .	1 . 200	(?)		Reconnect:	□ Clear	Buff	(?
lask:	255 . 255 .	255 . 0	(?)		L			_
Gate¥ay:	192 . 168 .	1.1	(?)					
Serial Nego	E On		(?)		3	Set A	LL	

• 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]
[Expand]
[Expand]
[Expand]
[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 a.





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.



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.

Sketchbook location: c:\Users\xutong\Documents\Arduino	Settings Network
Sketchbook location: c:\Users\xutong\Documents\Arduino	
c:\Users\xutong\Documents\Arduino	
-	BROWSE
Show files inside Sketches	
Editor font size: 14	
Interface scale: Automatic 1	96
Theme: Light (Arduino)	*
Language: English	✓ (Reload required)
Show verbose output during 🛛 compile 🗋 up	pload
Compiler warnings None 🗸	
Verify code after upload	
Auto save	
C Editor Ould Supportions	

the URLs with commas like this:

https://dl.espressif.com/dl/package_esp32_index.json,https://github.co m/earlephilhower/arduino-pico/releases/download/global/package_rp2040_i ndex.json

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

lo	ols Help		12			
	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"		•	Boards Manager	Ctrl+Shift+B	
	Port		•	Arduine AV/P Peerde		
	Get Board Info			esp32		1
	CPU Frequency: "240MHz (WiFi/BT)"		۰E			
	Core Debug Level: "None"		۰I.			
	Flash Frequency: "80MHz"		۰I.			
	Flash Mode: "QIO"					
	Flash Size: "4MB (32Mb)"		× L			
	Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"	× L			
	PSRAM: "Disabled"		× L			
	Upload Speed: "921600"					
	Burn Bootloader					

Ph	BOARDS MANAGER				
	pico				
	Type: All 🗸				
	Arduino Mbed OS RP2040 ···· Boards by Arduino				
⊳ ©	Boards included in this package: Raspberry Pi Pico More info				
Q	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 V (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).

💿 U	GV01_	BASE Ar	duino	IDE 2.0.2					
File	Edit	Sketch	Tools	Help					
				Auto Format	Ctrl+T				
				Archive Sketch					
P		IGV01_E		Manage Libraries	Ctrl+Shift+I		Servo	Ctrl.h	connect
		43		Serial Monitor	Ctrl+Shift+M		2047,	"spd"	:500,"a
1-	3	44 45		Serial Plotter					
		46		Board: "ESP32 Dev Module"		Þ			
	0	47		Port		Þ	5	Serial po	rts
0.855		49		Get Board Info			(COM1	
1	>	50 51		WiFi101 / WiFiNINA Firmware Update	r		:0}		
0		52		Upload SSL Root Certificates					

Q	53			
	54	CPU Frequency	•	
	55	Core Debug Level	•	
	56	Flash Frequency	•	
	57			
	58	Flash Mode	•	
	59	Flash Size	•	
	60	PSRAM	۶.	
	61	Partition Scheme	5	
	62	Tarcadon Scheme		
	63	Upload Speed	۲	
	64	Burn Bootloader		
	65	// 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.

💿 U	GV01_BASE Ar	duino IDE 2.0.2				
File	Edit Sketch	Tools Help				
		Auto Format	Ctrl+T			
		Archive Sketch				
P	UGV01_E	Manage Libraries	Ctrl+Shift+I	Sei	rvoCtrl.h	connectio
	43	Serial Monitor	Ctrl+Shift+M	2.04	47,"spd":5	00,"ac
5	3 44	Serial Plotter				
	45					
D-0	40	Board: "ESP32 Dev Module"		•		
	48	Port: "COM10"		Þ.	Serial ports	
	49	Get Board Info			COM1	
1	> 50			~	COM10	
	51	WIFITOT / WIFINING FIRMware Opd	ater			
0	52	Upload SSL Root Certificates				
	54	CPU Frequency		×		
	55	Core Debug Level		•		
	56	Flash Frequency		•		
	57	Flash Mode		×		
	58	Flash Size		•		
	60	PSRAM				
	61	Destaling Calence		3		
	62	Partition Scheme				
	63	Upload Speed		•		
	64	Burn Bootloader				
	65	<pre>7/ SET_SPD_RATE : {"T":90</pre>	1,"L":1.0,"R":	1.0}		

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

				_
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"	•	Boards Manager	Ctrl+Shift+B	

		eese see see see see see see see see se	
Port	•	Arduino AVR Boards	
Get Board Info		esp32	•
CPU Frequency: "240MHz (WiFi/BT)"	•	Raspberry Pi Pico/RP2040	•
Core Debug Level: "None"	• T		
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"	- E		
Burn Bootloader			

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.

Submit Now

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