Welcome

Thank you for choosing Freenove products!

Getting Started

When reading this, you should have downloaded the ZIP file for this product.
Unzip it and you will get a folder containing tutorials and related files. Please start with this PDF tutorial.

! Unzip the ZIP file instead of opening the file in the ZIP file directly.
! Do not move, delete or rename files in the folder just unzipped.

Get Support

Encounter problems? Don't worry! Refer to "TroubleShooting.pdf" or contact us.

When there are packaging damage, quality problems, questions encountering in use, etc., just send us an email. We will reply to you within one working day and provide a solution.

support@freenove.com

Safety and Precautions

Please follow the following safety precautions when using or storing this product:
● Keep this product out of the reach of children under 6 years old.
● This product should be used only when there is adult supervision present as young children lack necessary judgment regarding safety and the consequences of product misuse.
● This product contains small parts and parts, which are sharp. This product contains electrically conductive parts. Use caution with electrically conductive parts near or around power supplies, batteries and powered (live) circuits.
● When the product is turned ON, activated or tested, some parts will move or rotate. To avoid injuries to hands and fingers, keep them away from any moving parts!
● It is possible that an improperly connected or shorted circuit may cause overheating. Should this happen, immediately disconnect the power supply or remove the batteries and do not touch anything until it cools down! When everything is safe and cool, review the product tutorial to identify the cause.
● Only operate the product in accordance with the instructions and guidelines of this tutorial, otherwise parts may be damaged or you could be injured.
● Store the product in a cool dry place and avoid exposing the product to direct sunlight.
● After use, always turn the power OFF and remove or unplug the batteries before storing.

Any concerns? ✉ support@freenove.com

About Freenove

Freenove provides open source electronic products and services worldwide.

Freenove is committed to assist customers in their education of robotics, programming and electronic circuits so that they may transform their creative ideas into prototypes and new and innovative products. To this end, our services include but are not limited to:

- Educational and Entertaining Project Kits for Robots, Smart Cars and Drones
- Educational Kits to Learn Robotic Software Systems for Arduino, Raspberry Pi and micro: bit
- Electronic Component Assortments, Electronic Modules and Specialized Tools
- **Product Development and Customization Services**

You can find more about Freenove and get our latest news and updates through our website:

http://www.freenove.com

Copyright

All the files, materials and instructional guides provided are released under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. A copy of this license can be found in the folder containing the Tutorial and software files associated with this product.

Freenove brand and logo are copyright of Freenove Creative Technology Co., Ltd. and cannot be used without written permission.

Other registered trademarks and their owners appearing in this document:
Arduino® is a trademark of Arduino LLC (https://www.arduino.cc/).
Raspberry Pi® is a trademark of Raspberry Pi Foundation (https://www.raspberrypi.org/).
Raspberry Pi Pico® is a trademark of Raspberry Pi Foundation (https://www.raspberrypi.org/).
micro:bit® is a trademark of Micro:bit Educational Foundation (https://www.microbit.org/).
ESPRESSIF® and ESP32® are trademarks of ESPRESSIF Systems (Shanghai) Co, Ltd (https://www.espressif.com/).

Any concerns? ⊠ support@freenove.com

Contents

Any concerns? ✉ support@freenove.com

Preface

Raspberry Pi Pico is a tiny, fast, and versatile board built using RP2040, a brand new microcontroller chip designed by Raspberry Pi in the UK. Supporting Python and C/C++ development, it is perfect for DIY projects. In this tutorial, we use Arduino to learn Pico. If you want to learn the Python version, please refer to another tutorial: python_tutorial.pdf.

Using Arduino IDE as the development environment for Raspberry Pi Pico allows users to learn Pico better and more quickly, which is just like developing Arduino programs. In addition, resources such as Arduino's libraries can be directly used to greatly improve the efficiency of development.

In this tutorial, we devide each project into 4 sections:
1, Component list: helps users to learn and find what components needed in each project.
2, Component knowledge: allows you to learn the features and usage of the components.
3, Circuit: assists to build circuit for each project.
4, Code and annotation: makes it easier for users to learn to use Raspberry Pi Pico and make secondary development.

After completing the projects in this tutorial, you can also combine the components in different projects to make your own smart homes, smart car, robot, etc., bringing your imagination and creativity to life with Raspberry Pi Pico.
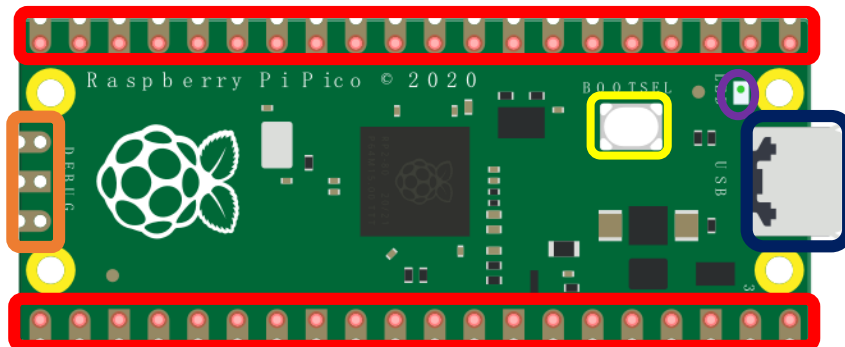
If you have any problems or difficulties using this product, please contact us for quick and free technical support: support@freenove.com

Raspberry Pi Pico

Before learning Pico, we need to know about it. Below is an imitated diagram of Pico, which looks very similar to the actual Pico.
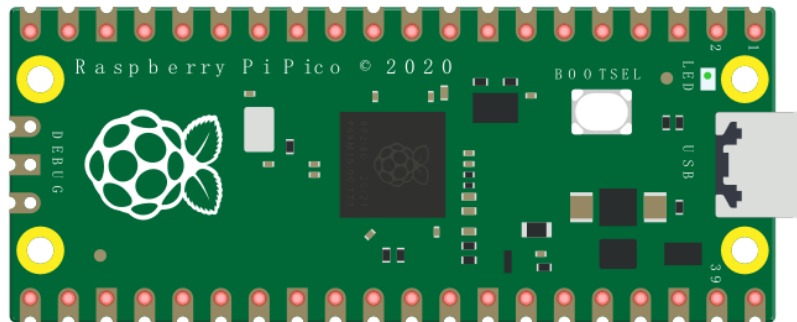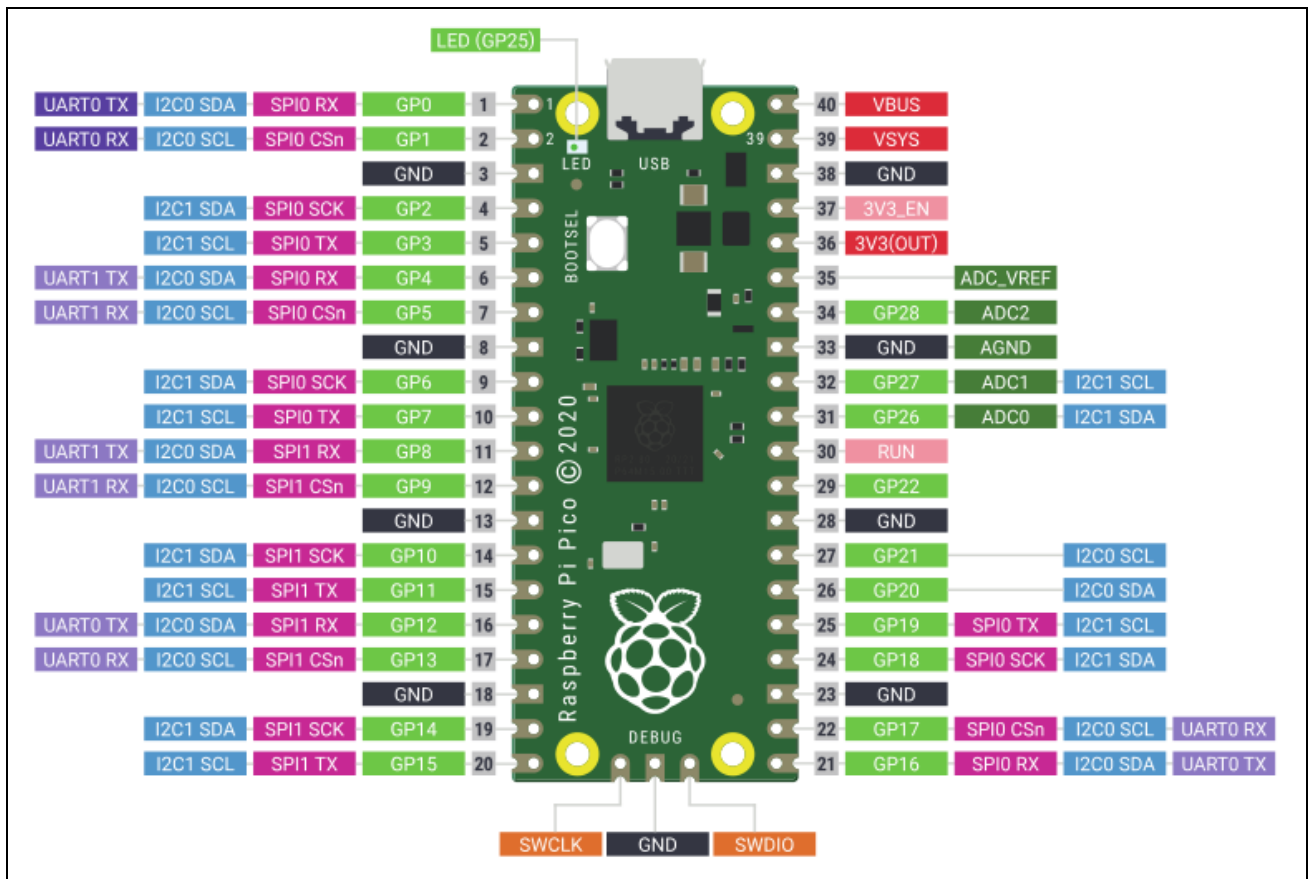

The hardware interfaces are distributed as follows:

Frame color	Description
	Pins
	BOOTSE button
	USB port
	LED
	Debugging

Function definition of pins:

Any concerns? ⊠ support@freenove.com

Color	Pins	Color	Pins
	GND		Power
	GPIO		ADC
	UART(defualt)		UART
	SPI		I2C
	System Control		Debugging

For details： https://datasheets.raspberrypi.org/pico/pico-datasheet.pdf

Any concerns? ✉ **support@freenove.com**

GND	Ground Pin
Power	VBUS(microUSB Voltage)、VSYS(2-5VDC Input)、3V3(3.3V OUT)、3V3_EN(Enables Pico)
System Control	Run(Start or disable RP2040 microcontroller or reset)
ADC	Raspberry Pi Pico has a total of 5 ADC with a resolution of 12 bits, which are ADC0(GP26), ADC1(GP27), ADC2(GP28), ADC3(GP29), ADC4 respectively. Among them, ADC3(GP29) is used to measure the VSYS on Pico board; ADC4 is derectly connected to the RP2040's built-in temperature sensor. ADC_VREF can connect to external accurate voltmeter as ADC reference. ADC_GND pin is used as the reference point for grounding.
PWM	There are 16 PWM channels on Raspberry Pi Pico, each of which can control frequency and duty cycle independently. The GPIO pins are switched to PWM function.
UART	There are 2 UART: UART0, UART1.
SPI	There are 2 SPI: SPI0, SPI1.
I2C	2 I2C: I2C0, I2C1.
Debugging	It is used when debugging code.

UART, I2C, SPI Defalt Pin

UART

Function	Default
UART_BAUDRATE	115200
UART_BITS	8
UART_STOP	1
UART0_TX	Pin 0
UART0_RX	Pin 1
UART1_TX	Pin 4
UART1_RX	Pin 5

I2C

Function	Default
I2C Frequency	400000
I2C0 SCL	Pin 9
I2C0 SDA	Pin 8
I2C1 SCL	Pin 7
I2C1 SDA	Pin 6

Any concerns? ✉ support@freenove.com

SPI

| Function	Default
SPI_BAUDRATE	1000000
SPI_POLARITY	0
SPI_PHASE	0
SPI_BITS	8
SPI_FIRSTBIT	MSB
SPI_SCK	Pin 2
SPI0_MOSI	Pin 3
SPI0_MISO	Pin 4
SPI1_SS	Pin 5

Chapter 0 Getting Ready (Important)

Before starting building the projects, you need to make some preparation first, which is so crucial that you must not skip.

Programming Software

Arduino Software (IDE) is used to write and upload the code for Arduino Board.
First, install Arduino Software (IDE): visit https://www.arduino.cc, click "Download" to enter the download page.
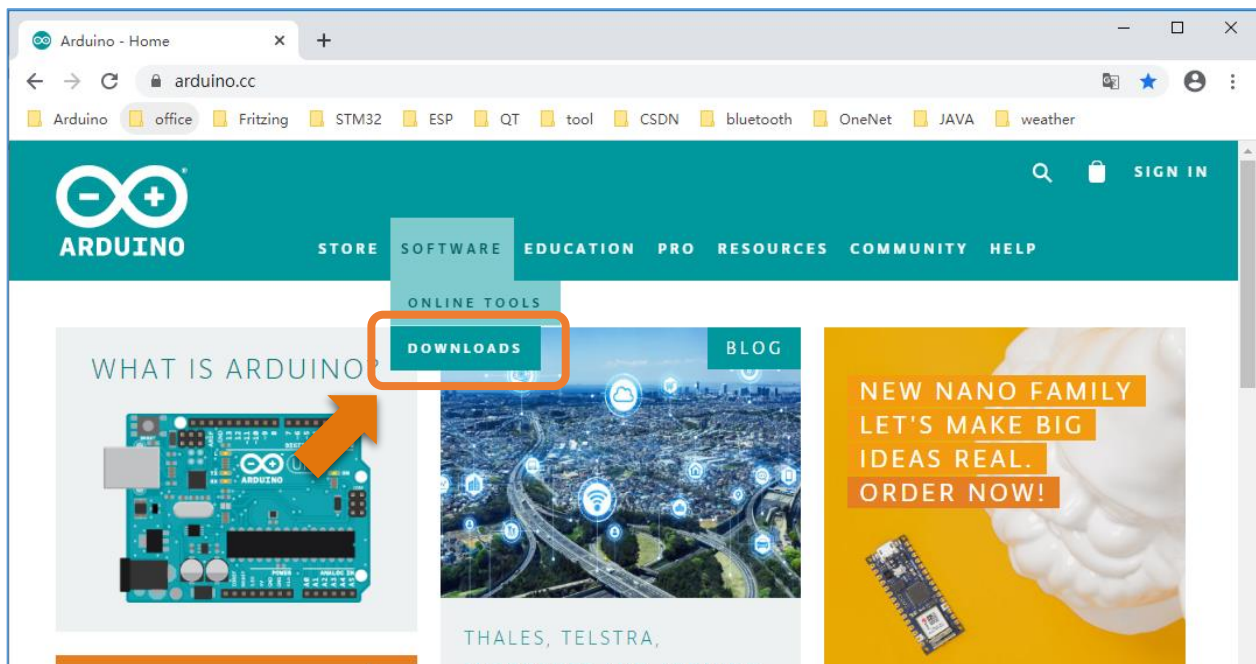
Select and download corresponding installer according to your operating system. If you are a windows user, please select the "Windows Installer" to download to install the driver correctly.

Any concerns? ✉ support@freenove.com

After the download completes, run the installer. For Windows users, there may pop up an installation dialog box of driver during the installation process. When it popes up, please allow the installation.
After installation is complete, an Arduino Software shortcut will be generated in the desktop. Run the Arduino Software.

The interface of Arduino Software is as follows:

Programs written with Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and saved with the file extension**.ino**. The editor has features for cutting/pasting and searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Verify
Check your code for compile errors .

Upload
Compile your code and upload them to the configured board.

New
Create a new sketch.

Open
Present a menu of all the sketches in your sketchbook. Clicking one will open it within the current window and overwrite its content.

Save
Save your sketch.

Serial Monitor
Open the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

Installation of Development Board Support Package

1,   Make sure your network is of good connection.
2,   Open Arduino IDE. Click Tools>**Board>Boards Manager**…on the menu bar.
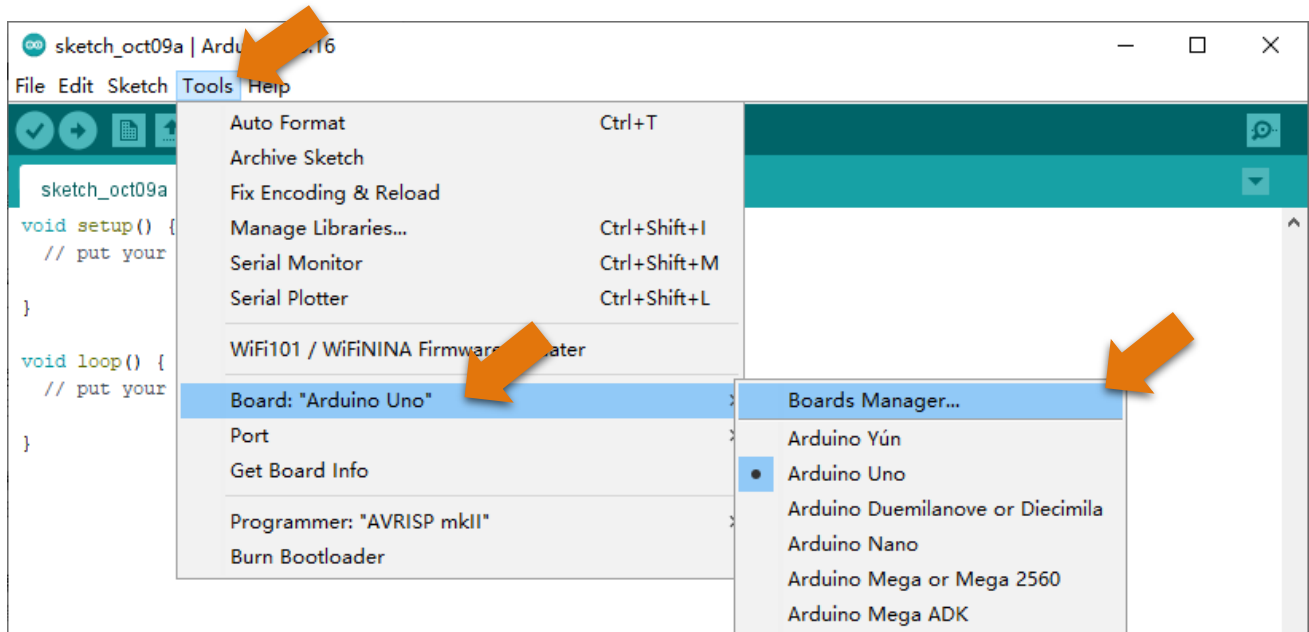
3,   Enter Pico in the searching box, select "Arduino Mbed OS RP2040 Boards" and click on Install.
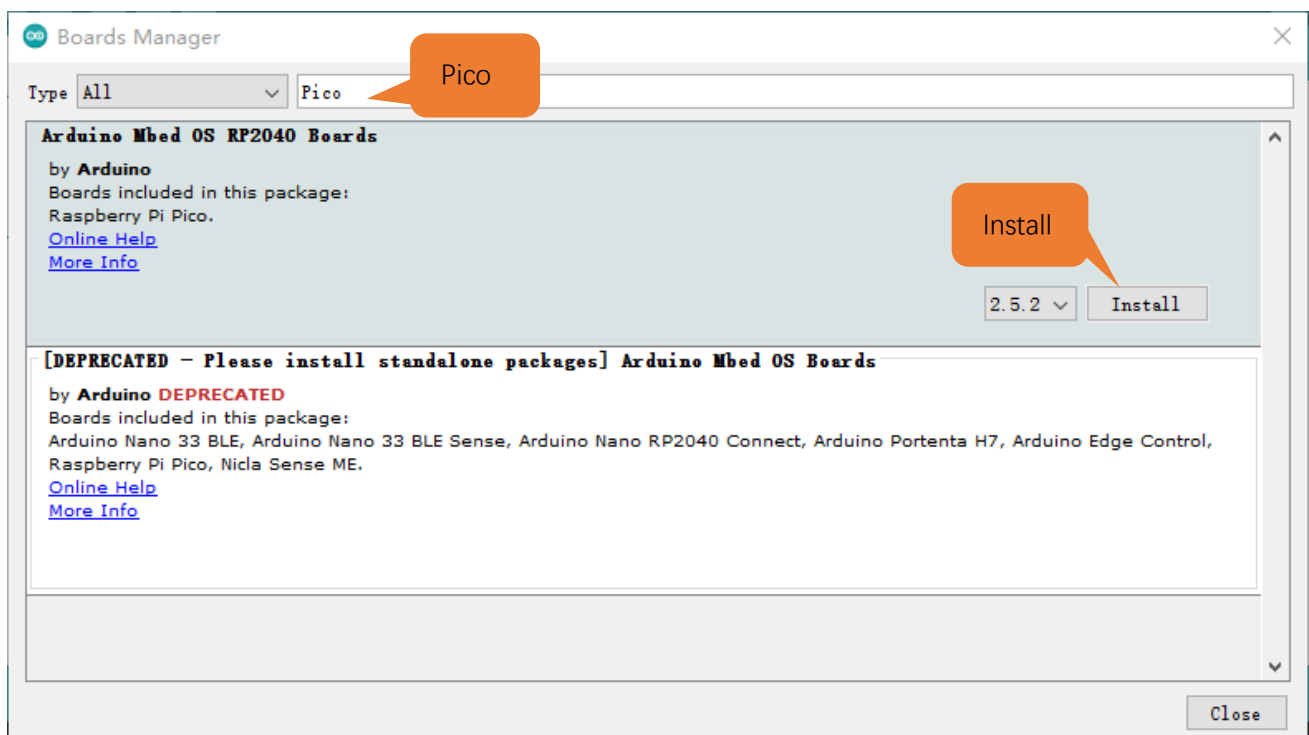
4,   Click Yes in the pop-up "dpinst-amd64.exe"installation window. (Without it, you will fail to communicate with Arduino.) Thus far, we have finished installing the development support package.
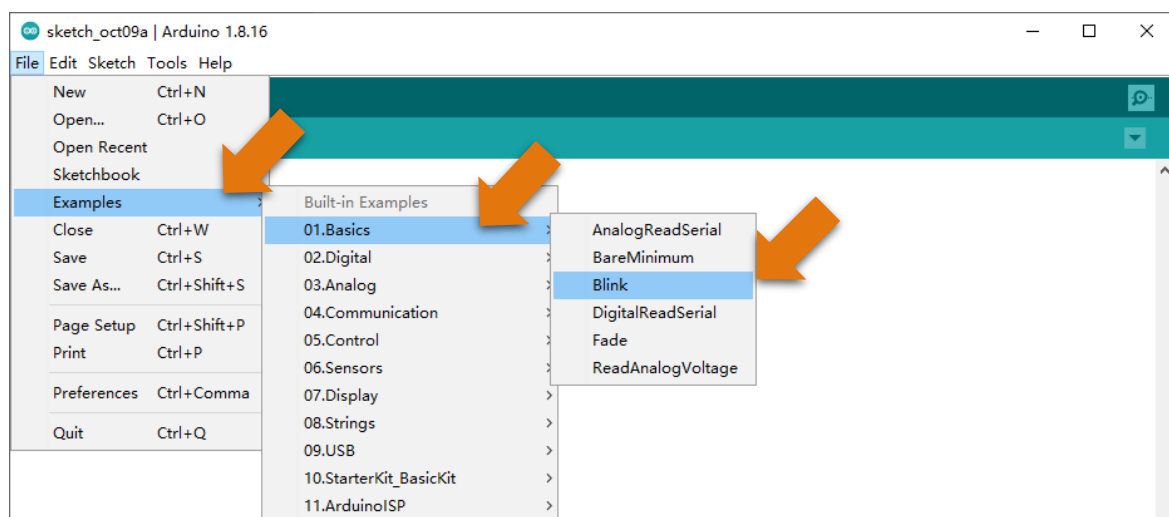
Any concerns? ⊠ **support@freenove.com**

Uploading Adruino-compatible Firmware for Pico

If your Pico is new and you want to use Arduino to learn and develop, you need to upload an Adruino-compatible Firmware for it. Please refer to the following steps to cinfigure.

1, Disconnect Pico from computer. Keep pressing the white button(BOOTSEL) on Pico, and connect Pico to computer before releasing the button. (Note: Be sure to keep pressing the button before powering the Pico, otherwise the firmware will not download successfully).
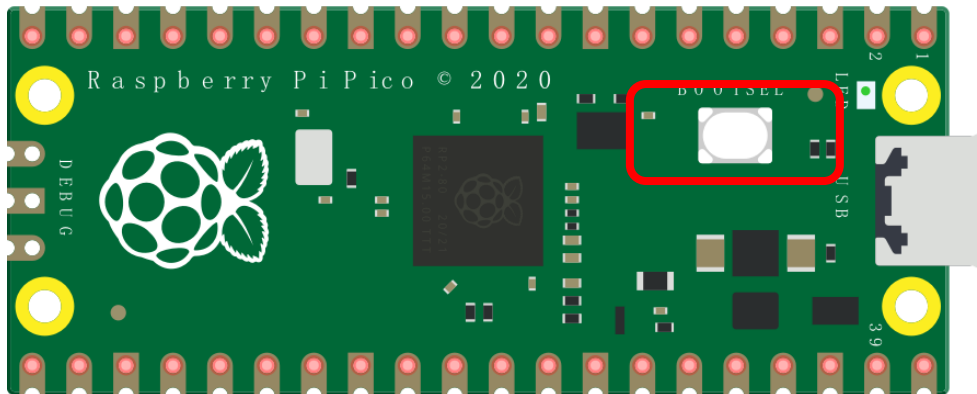
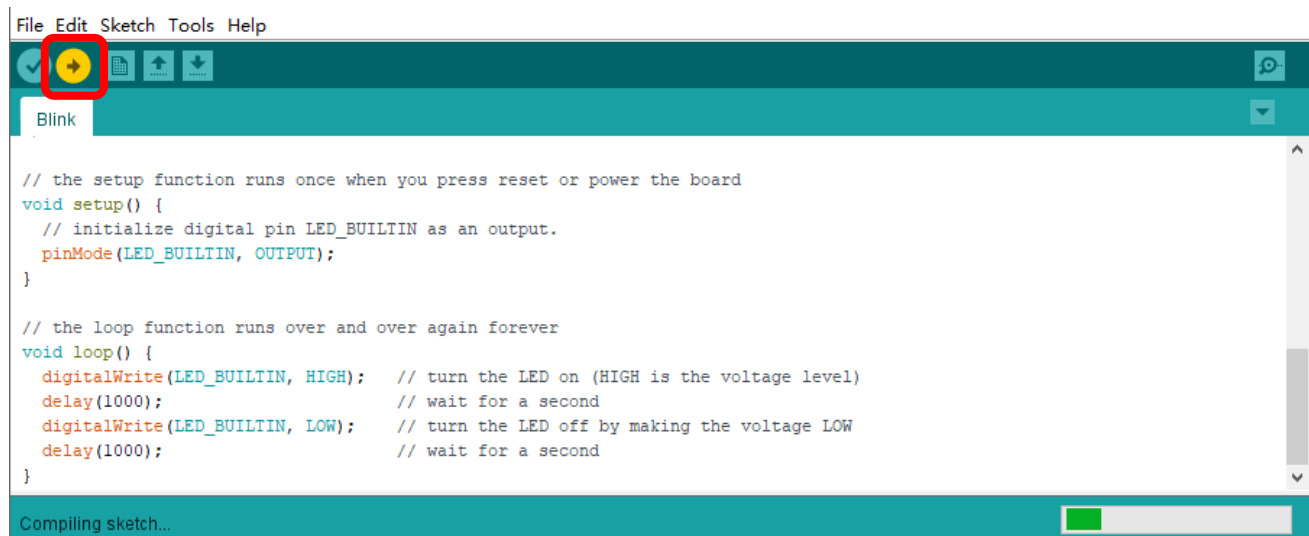2, Open Arduino IDE. Click File>Examples>01.Basics>Blink.

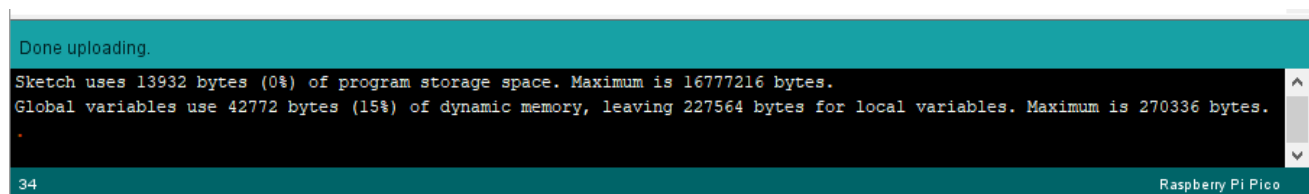3, Click **Tools>Board>Arduino Mbed OS RP2040 Boards>Raspberry Pi Pico.**
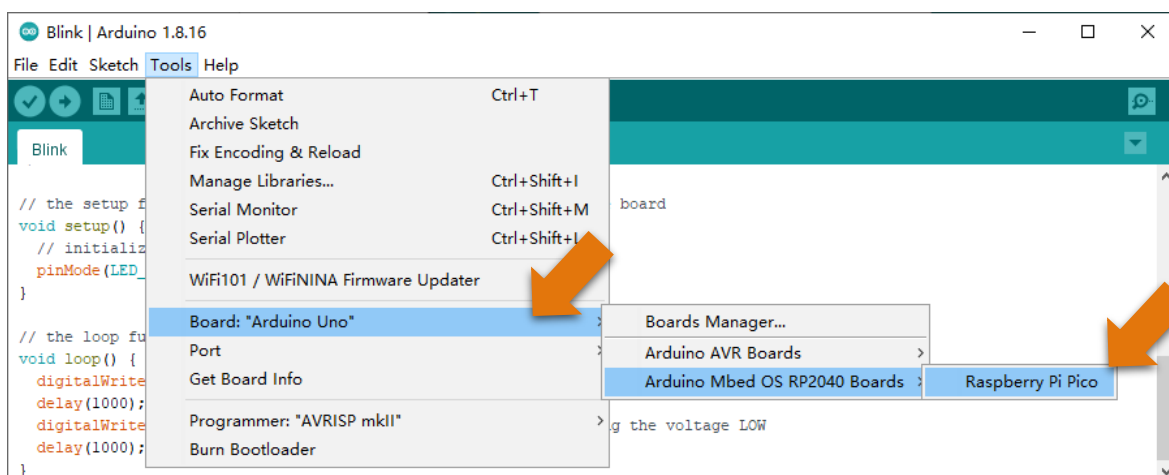
Any concerns? ✉ support@freenove.com
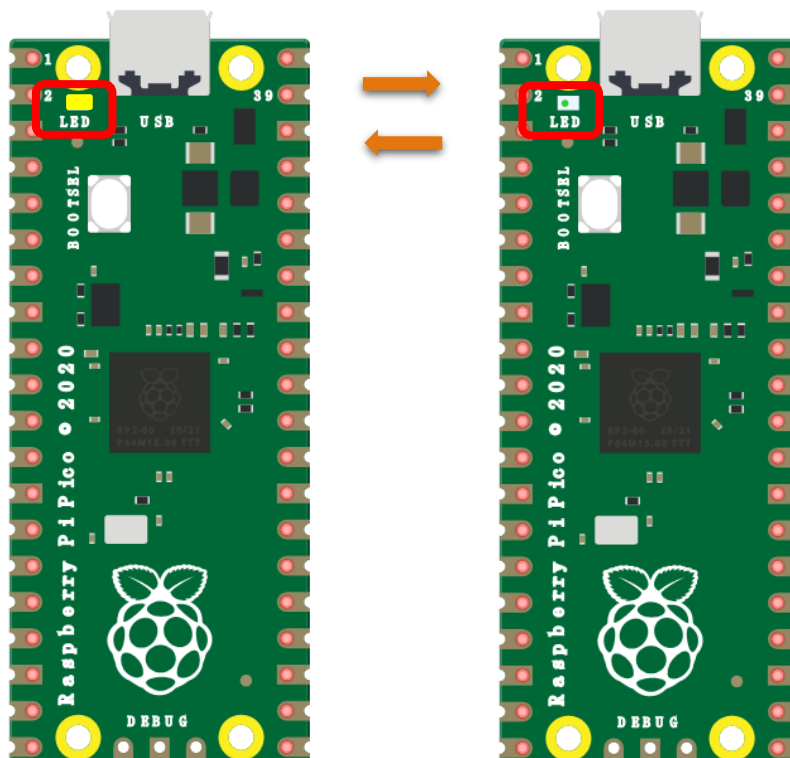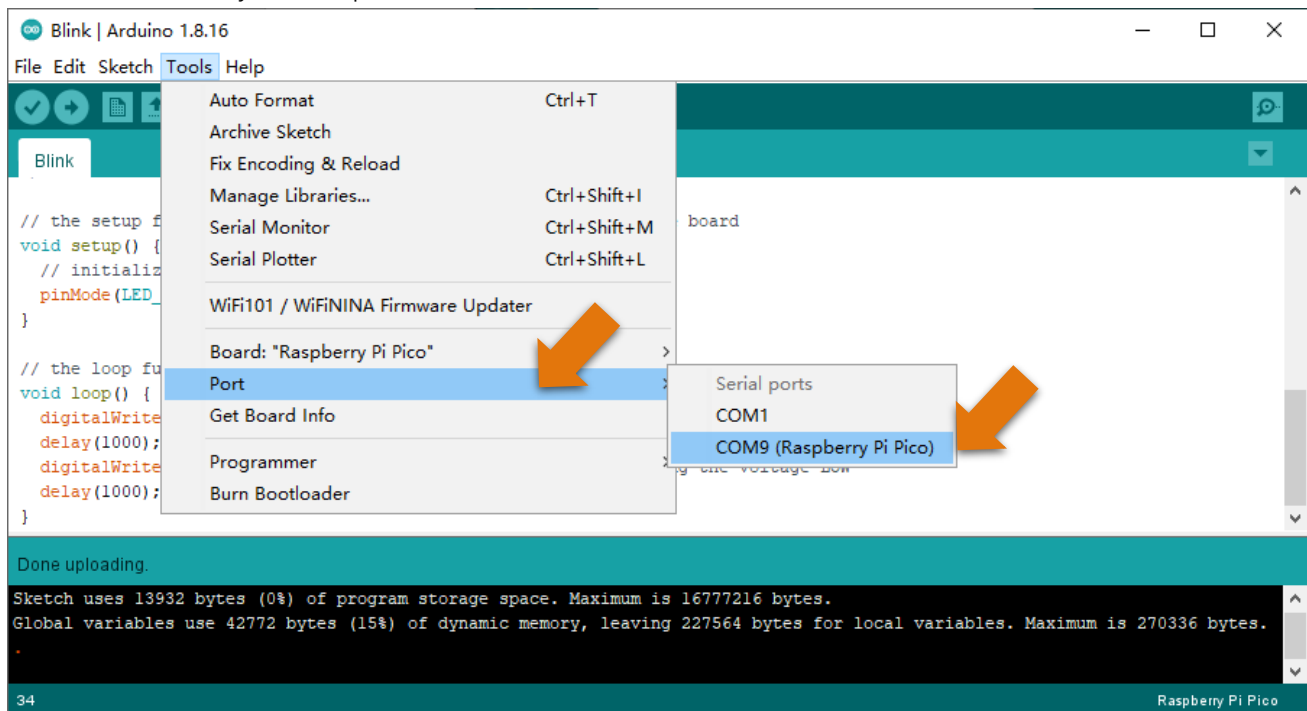
4， Upload sketch to Pico.

When the sketch finishes uploading, you can see the following prompt.

And the indicator on Pico starts to flash.

Any concerns? ✉ **support@freenove.com**

5, Click **Tools>Port>COMx(Raspberry Pi Pico)**. X of COMx varies from different computers. Please select the correct one on your computer. In our case, it is COM9.


Note:

1. At the first time you use Arduino to upload sketch for Pico, you don't need to select port. After that, each time before uploading sketch, please check whether the port has beed selected; otherwise, the downloading may fail.

2. Sometimes when using, Pico may lose firmware due to the code and fail to work. At this point, you can upload firmware for Pico as mentioned above.

Any concerns? ✉ support@freenove.com

Paste the Sticker on the Breadboard

It is not difficult to use the Pico. However, officially, the pin functions are printed on the back of the board, which makes it inconvenient to use. To help users finish each project in the tutorial faster and easier, we provide stickers of the pin functions as follows:
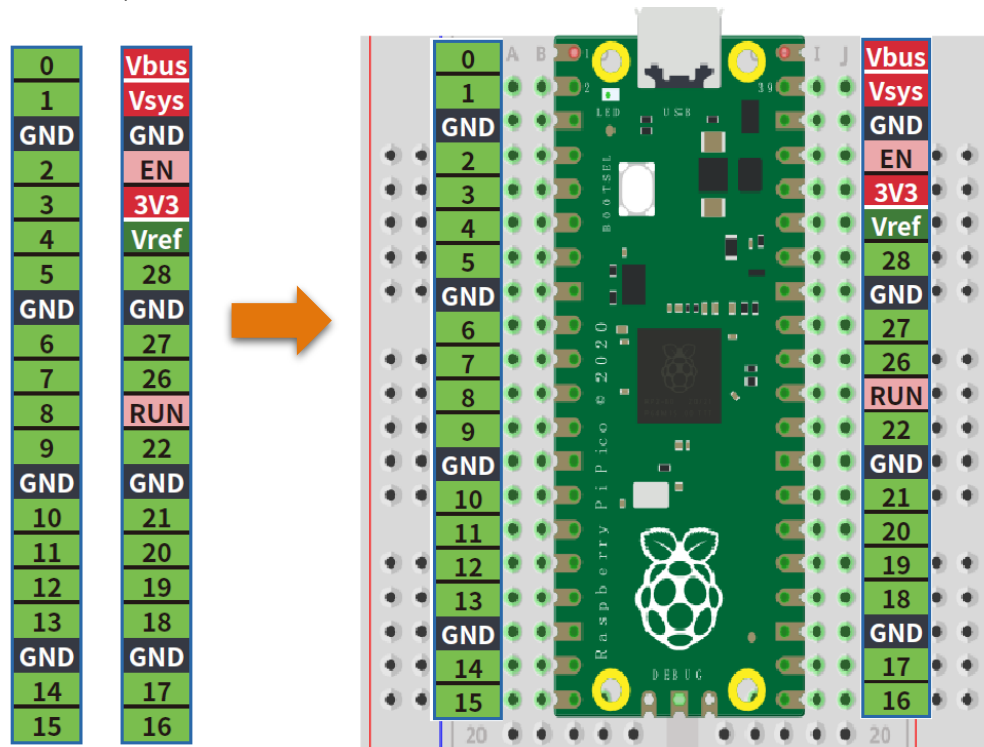
You can paste the sticker on the blank area of the breadboard as above.

To make the tutorial more intuitive, we've made some changes to the simulation diagram as below. The left one is the actual Pico and the right one is its simulation diagram. Please note that to avoid misunderstanding.

Any concerns? ⊠ **support@freenove.com**
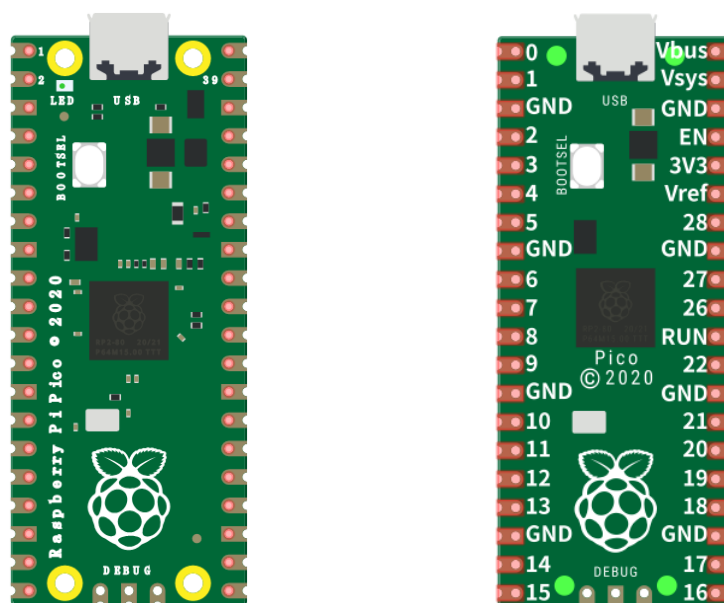
Chapter 1 LCD1602

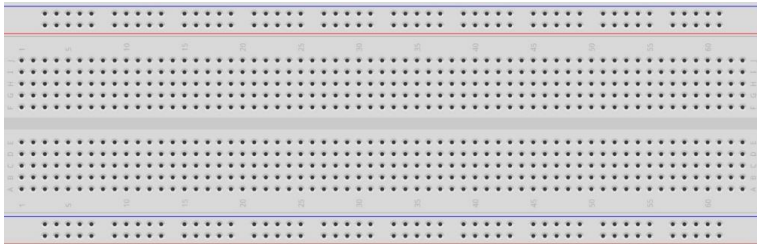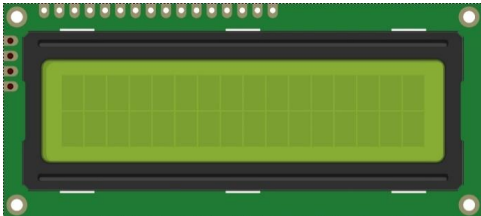In this chapter, we will learn about the LCD1602 Display Screen.
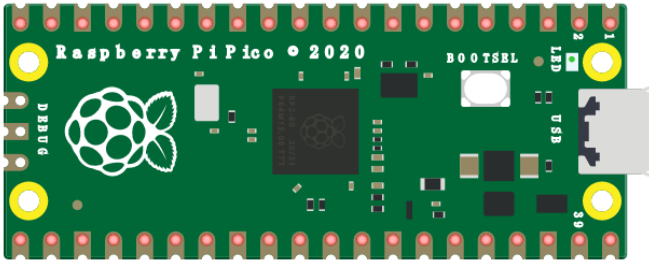
Project 1.1 LCD1602

In this section we learn how to use LCD1602 to display something.
If you haven't installed Arduino IDE, you can click Here.
If you haven't uploaded firmware for Pico, you can click Here to upload.

Component List

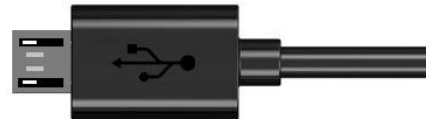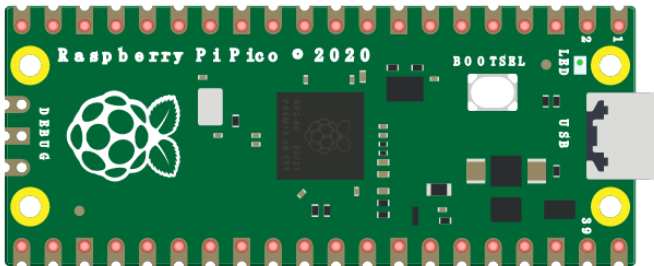| Raspberry Pi Pico x1	USB cable x1
Breadboard x1	
LCD1602 Module x1	Jumper

Component knowledge

Power

Raspberry Pi Pico requires 5V power supply. You can either connect external 5V power supply to Vsys pin of Pico or connect a USB cable to the onboard USB base to power Pico.

In this tutorial, we use USB cable to power Pico and upload sketches.

I2C communication

I2C (Inter-Integrated Circuit) is a two-wire serial communication mode, which can be used for the connection of micro controllers and their peripheral equipment. Devices using I2C communication must be connected to the serial data (SDA) line, and serial clock (SCL) line (called I2C bus). Each device has a unique address and can be used as a transmitter or receiver to communicate with devices connected to the bus.

LCD1602 communication

The LCD1602 Display Screen can display 2 lines of characters in 16 columns. It is capable of displaying numbers, letters, symbols, ASCII code and so on. As shown below is a monochrome LCD1602 Display Screen along with its circuit pin diagram.
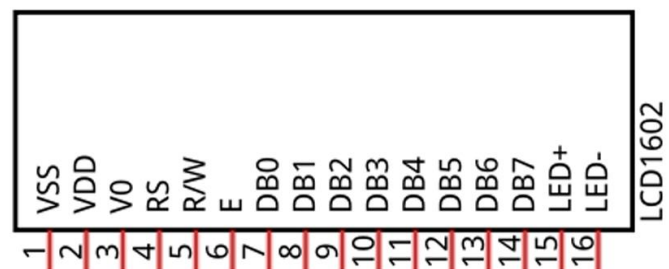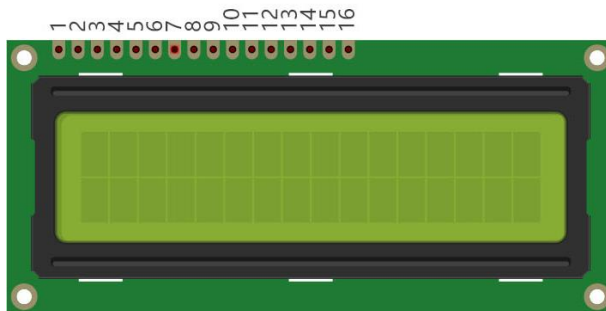
I2C LCD1602 Display Screen integrates an I2C interface, which connects the serial-input & parallel-output module to the LCD1602 Display Screen. This allows us to use only 4 lines to the operate the LCD1602.
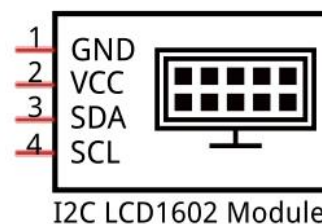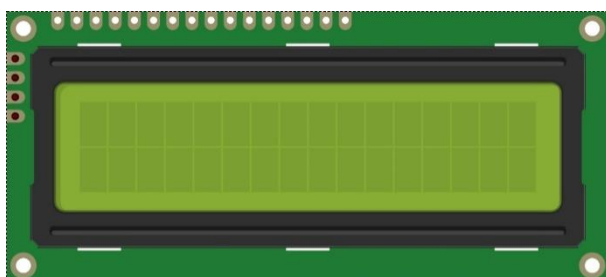
The serial-to-parallel IC chip used in this module is PCF8574T (PCF8574AT), and its default I2C address is 0x27(0x3F).

Any concerns? ✉ support@freenove.com

Below is the PCF8574 pin schematic diagram and the block pin diagram:

PCF8574 chip pin diagram | PCF8574 module pin diagram
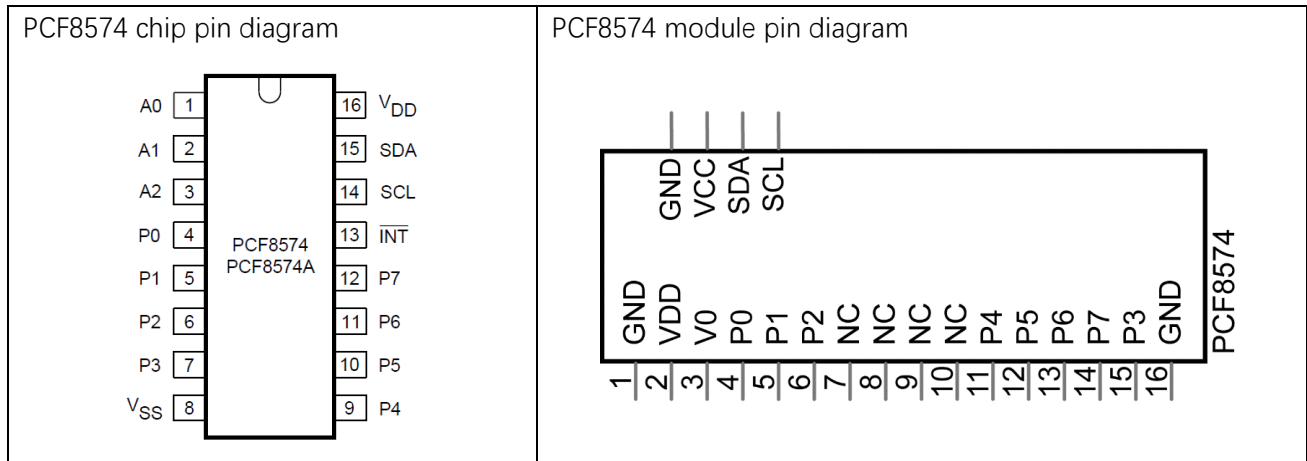
PCF8574 module pin and LCD1602 pin are corresponding to each other and connected with each other:
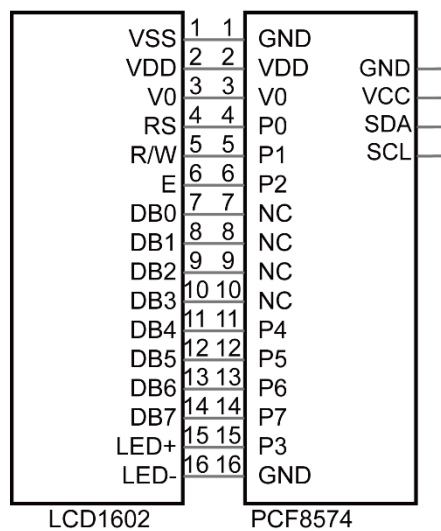
So we only need 4 pins to control the 16 pins of the LCD1602 Display Screen through the I2C interface.

In this project, we will use the I2C LCD1602 to display some static characters and dynamic variables.

Circuit

Schematic diagram

Hardware connection. If you need any support, please feel free to contact us via: support@freenove.com

Any concerns? ✉ support@freenove.com

Sketch

How to install the library

We use the third party library **LiquidCrystal I2C**. If you haven't installed it yet, please do so before learning.
The steps to add third-party Libraries are as follows: open arduino->Sketch->Include library-> Manage libraries. Enter " LiquidCrystal I2C" in the search bar and select " LiquidCrystal I2C " for installation.

There is another way you can install libraries.
Click "Add .ZIP Library…" and then find **LiquidCrystal_I2C.zip** in libraries folder (this folder is in the folder unzipped form the ZIP file we provided). This library can facilitate our operation of I2C LCD1602.

Use I2C LCD 1602 to display characters and variables.

Sketch_1.1_Display_the_string_on_LCD1602

```
8  /*
9  * note:If lcd1602 uses PCF8574T, IIC's address is 0x27,
10 * or lcd1602 uses PCF8574AT, IIC's address is 0x3F.
11 */
12 LiquidCrystal_I2C lcd(0x27,16,2);
13 void setup() {
14     lcd.init(); // LCD driver initialization
15     lcd.backlight(); // Open the backlight
16     lcd.setCursor(0,0); // Move the cursor to row 0, column 0
17     lcd.print("hello world"); // The print content is displayed on the LCD
18 }
19 void loop() {
20     lcd.setCursor(0,1); // Move the cursor to row 1, column 0
21     lcd.print("Counter:"); // The count is displayed every second
22     lcd.print(millis() / 1000);
23     delay(1000);
24 }
25
```
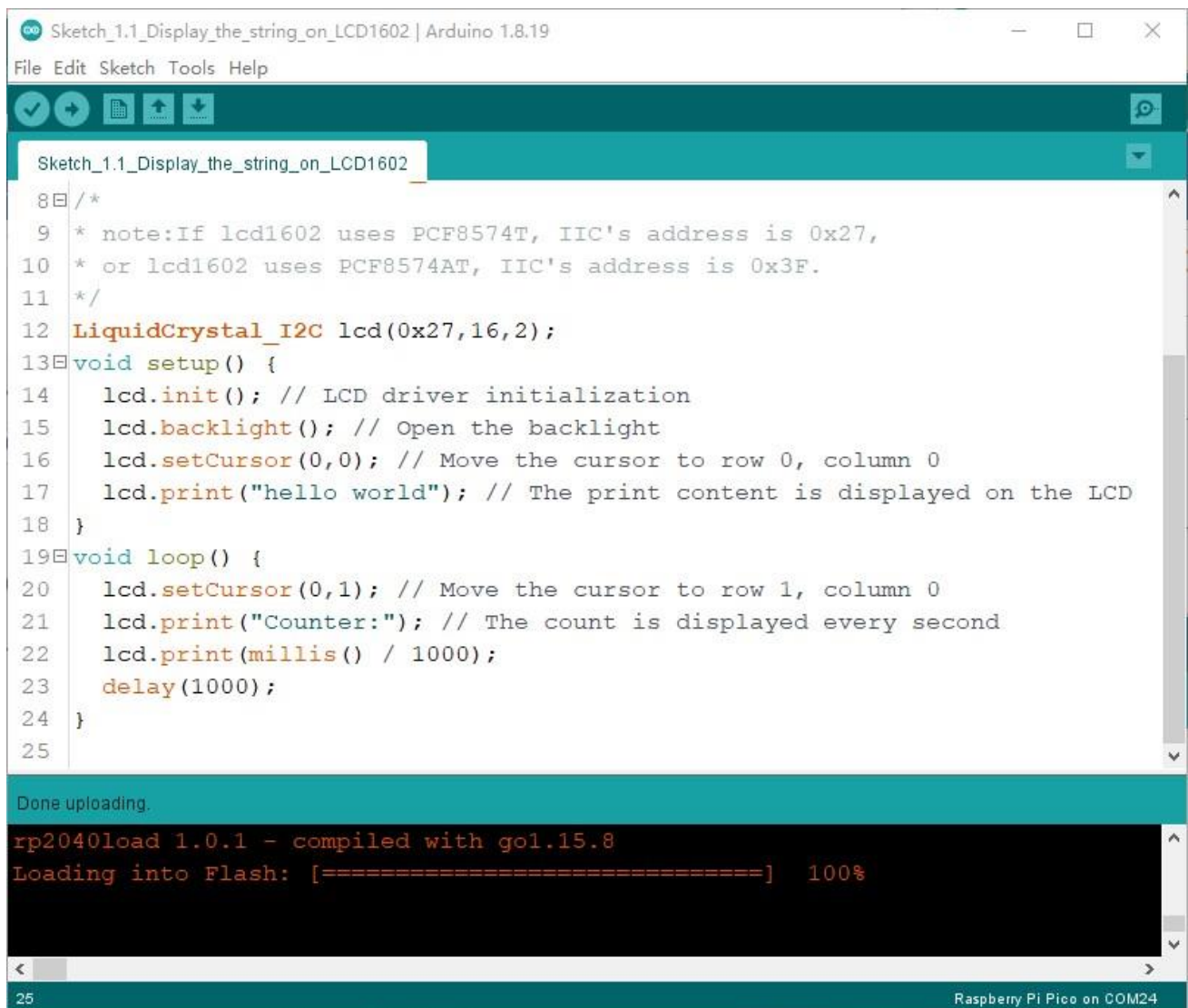
Done uploading.

```
rp2040load 1.0.1 - compiled with go1.15.8
Loading into Flash: [================================]  100%
```

25　　Raspberry Pi Pico on COM24

Compile and upload the code to Pico and the LCD1602 displays characters.

If you cannot see anything on the display or the display is not clear, try rotating the white knob on back of LCD1602 slowly, which adjusts the contrast, until the screen can display clearly.

Any concerns? ⊠ support@freenove.com

The following is the program code:
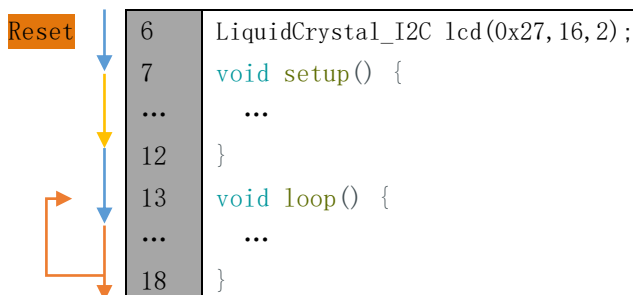
```
1    #include <LiquidCrystal_I2C.h>
2    /*
3    * note:If lcd1602 uses PCF8574T, IIC's address is 0x27,
4    * or lcd1602 uses PCF8574AT, IIC's address is 0x3F.
5    */
6    LiquidCrystal_I2C lcd(0x27,16,2);
7    void setup() {
8      lcd.init(); // LCD driver initialization
9      lcd.backlight(); // Open the backlight
10     lcd.setCursor(0,0); // Move the cursor to row 0, column 0
11     lcd.print("hello world"); // The print content is displayed on the LCD
12   }
13   void loop() {
14     lcd.setCursor(0,1); // Move the cursor to row 1, column 0
15     lcd.print("Counter:"); // The count is displayed every second
16     lcd.print(millis() / 1000);
17     delay(1000);
18   }
```

The Arduino IDE code usually contains two basic functions: void setup() and void loop().

After the board is reset, the setup() function will be executed firstly, and then the loop() function.

setup() function is generally used to write code to initialize the hardware. And loop() function is used to write code to achieve certain functions. loop() function is executed repeatedly. When the execution reaches the end of loop(), it will jump to the beginning of loop() to run again.

Reset

```
6    LiquidCrystal_I2C lcd(0x27,16,2);
7    void setup() {
...      ...
12   }
13   void loop() {
...      ...
18   }
```

Include header file of Liquid Crystal Display (LCD)1602.

```
1    #include <LiquidCrystal_I2C.h>
```

Instantiate the I2C LCD1602 screen. It should be noted here that if your LCD driver chip uses PCF8574T, set the I2C address to 0x27, and if uses PCF8574AT, set the I2C address to 0x3F.

```
6    LiquidCrystal_I2C lcd(0x27,16,2);
```

Initialize LCD1602 and turn on the backlight of LCD.

```
10     lcd.init(); // LCD driver initialization
11     lcd.backlight(); // Open the backlight
```

Move the cursor of LCD1602 to the first row, first column, and print out "Hello, world!"

```
11     lcd.move_to(0, 0)
12     lcd.putstr("Hello, world!")
```

Print the number on the second line of LCD1602.

```
13    void loop() {
14      lcd.setCursor(0,1); // Move the cursor to row 1, column 0
15      lcd.print("Counter:"); // The count is displayed every second
16      lcd.print(millis() / 1000);
17      delay(1000);
18    }
```

Reference

class **LiquidCrystal**

The LiquidCrystal class can manipulate common LCD screens. The first step is defining an object of LiquidCrystal, for example:

LiquidCrystal_I2C lcd(0x27,16,2);

    Instantiate the Lcd1602 and set the I2C address to 0x27, with 16 columns per row and 2 rows per column.

init();

    Initializes the Lcd1602's device

backlight();

    Turn on Lcd1602's backlight.

setCursor(column,row);

    Sets the screen's column and row.

    column： The range is 0 to 15.

    row： The range is 0 to 1.
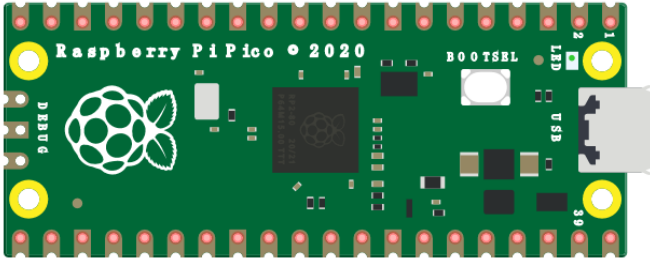
print(String);

    Print the character string on Lcd1602
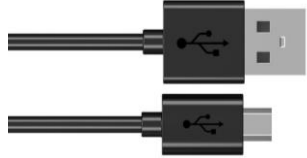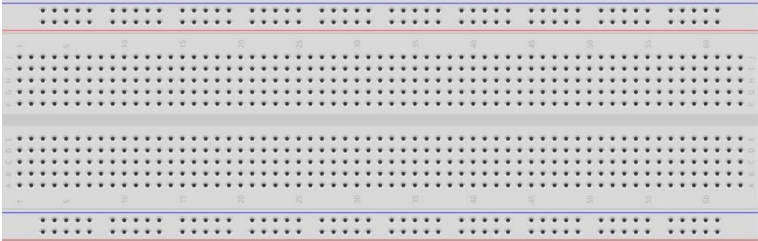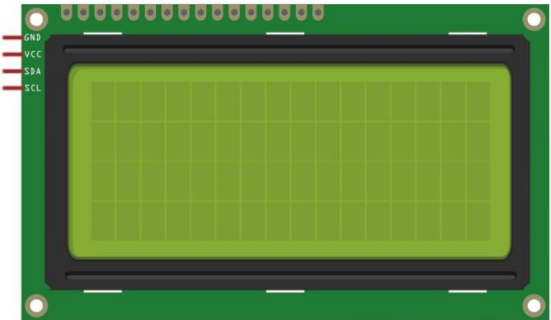
Any concerns? ✉ **support@freenove.com**

Chapter 2 LCD2004

In the previous chapter, we studied the LCD1602 display. In order to display more content,In this chapter, we will learn about the LCD2004 Display Screen.

Project 2.1 LCD2004

In this section we learn how to use LCD2004 to display something.

Component List

| Raspberry Pi Pico x1	USB cable x1
Breadboard x1	
LCD2004 Module x1	Jumper

Any concerns? ✉ support@freenove.com

Component knowledge

I2C communication

I2C (Inter-Integrated Circuit) is a two-wire serial communication mode, which can be used for the connection of micro controllers and their peripheral equipment. Devices using I2C communication must be connected to the serial data (SDA) line, and serial clock (SCL) line (called I2C bus). Each device has a unique address and can be used as a transmitter or receiver to communicate with devices connected to the bus.

LCD2004 communication

The LCD2004 display screen can display 4 lines of characters in 20 columns. It is capable of displaying numbers, letters, symbols, ASCII code and so on. As shown below is a monochrome LCD2004 display screen along with its circuit pin diagram.
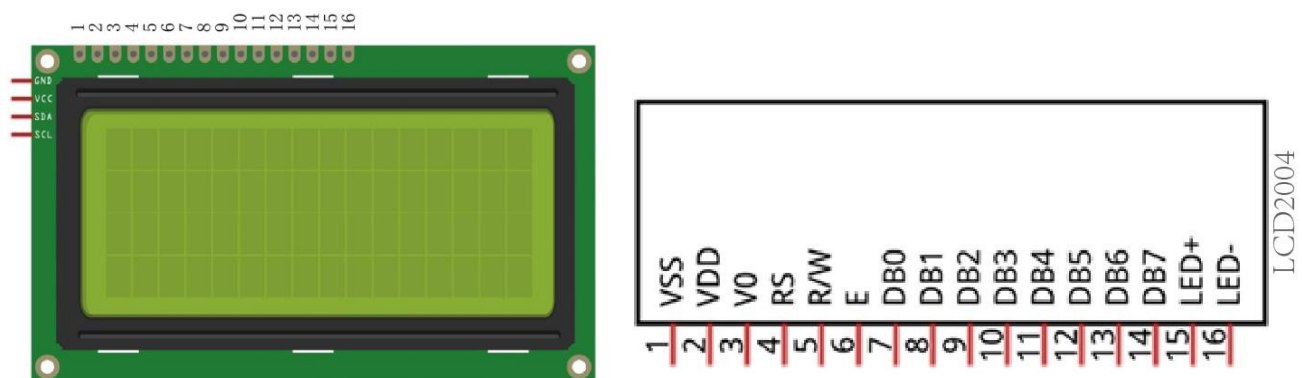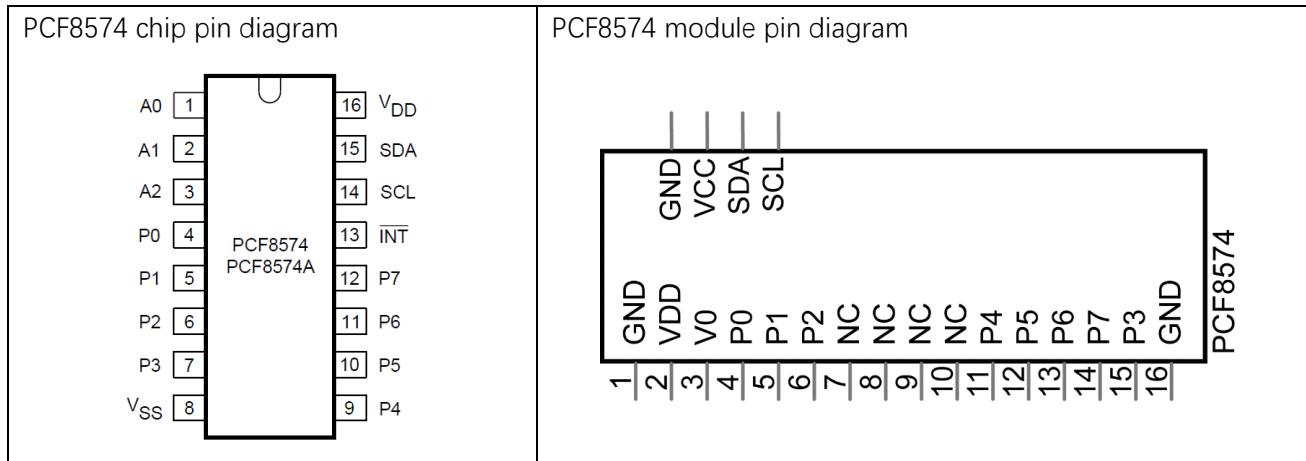

I2C LCD2004 display screen integrates a I2C interface, which connects the serial-input & parallel-output module to the LCD2004 display screen. This allows us to only use 4 lines to the operate the LCD2004.
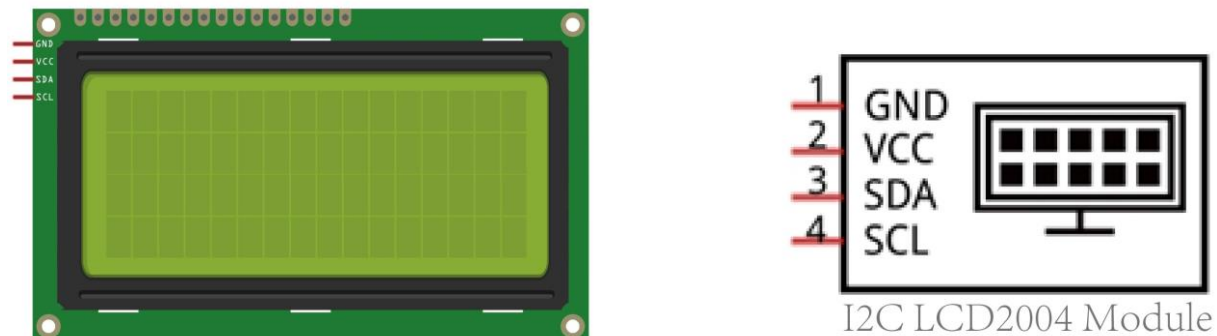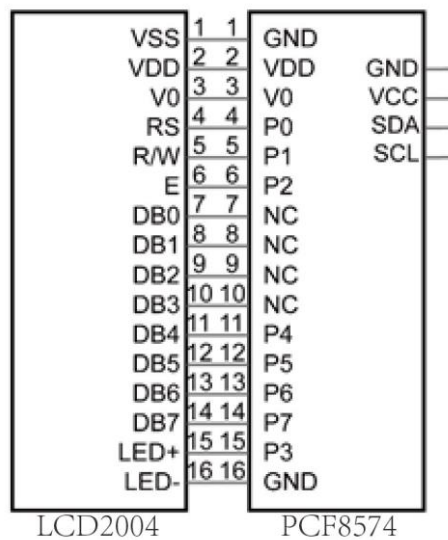

The serial-to-parallel IC chip used in this module is PCF8574T (PCF8574AT), and its default I2C address is 0x27(0x3F).

Below is the PCF8574 pin schematic diagram and the block pin diagram:

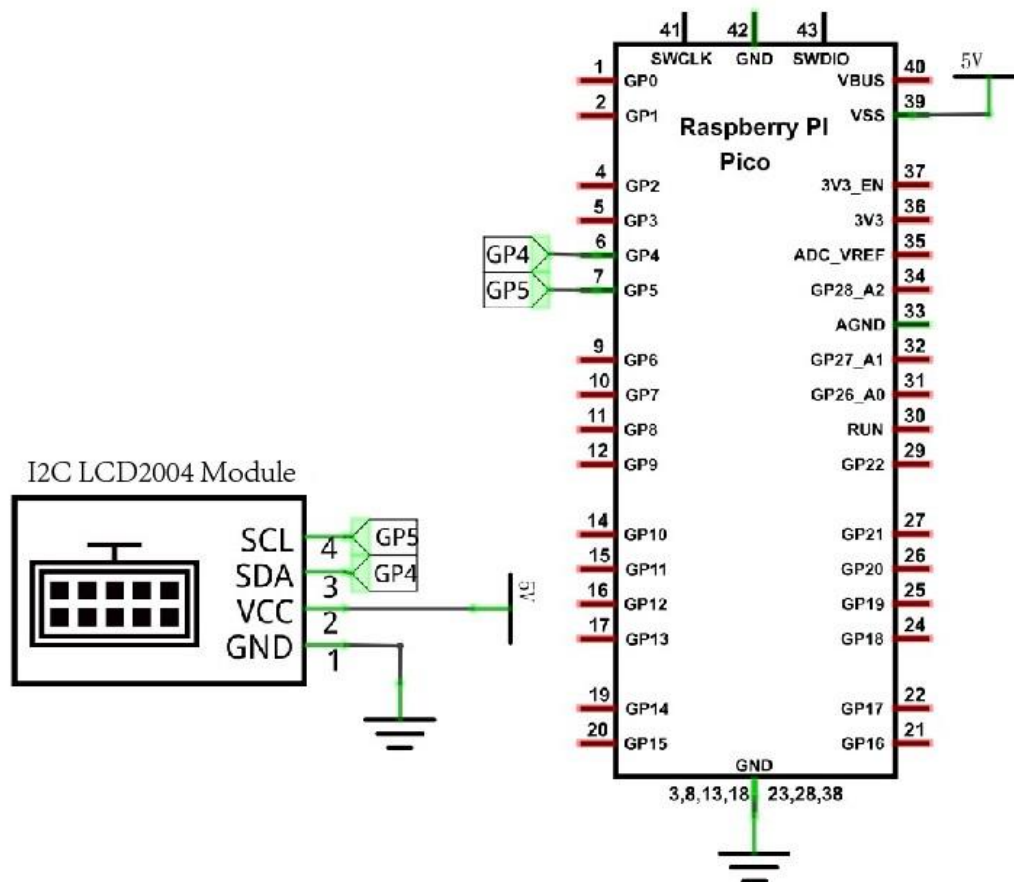| PCF8574 chip pin diagram	PCF8574 module pin diagram

PCF8574 module pin and LCD2004 pin are corresponding to each other and connected with each other:
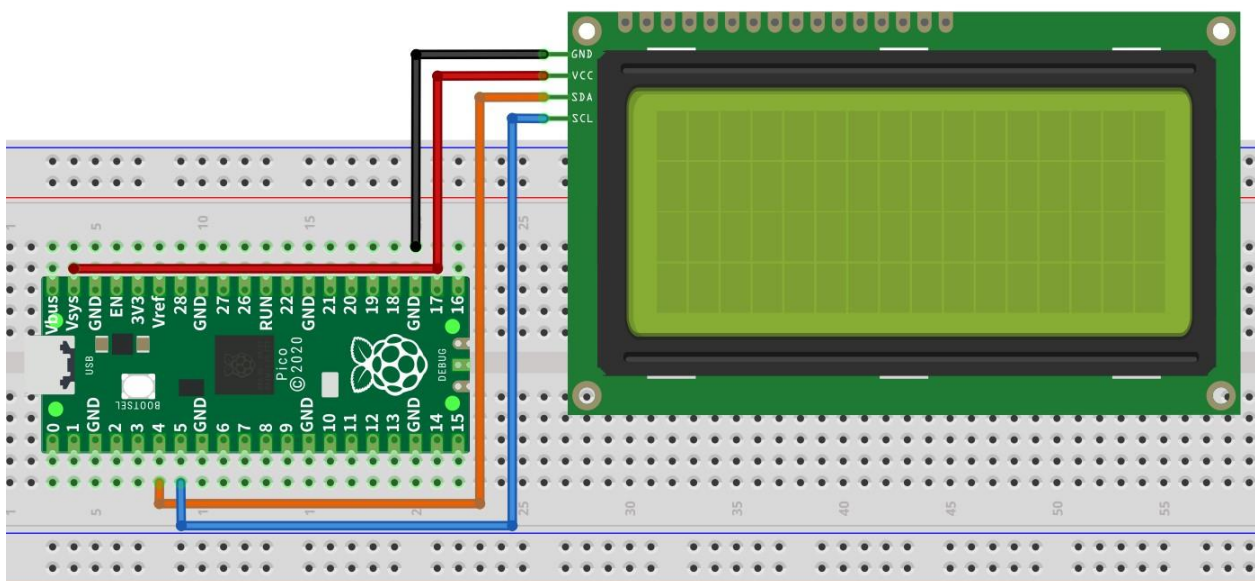

So we only need 4 pins to control the 16 pins of the LCD2004 Display Screen through the I2C interface.
In this project, we will use the I2C LCD2004 to display some static characters and dynamic variables.

Circuit

Schematic diagram

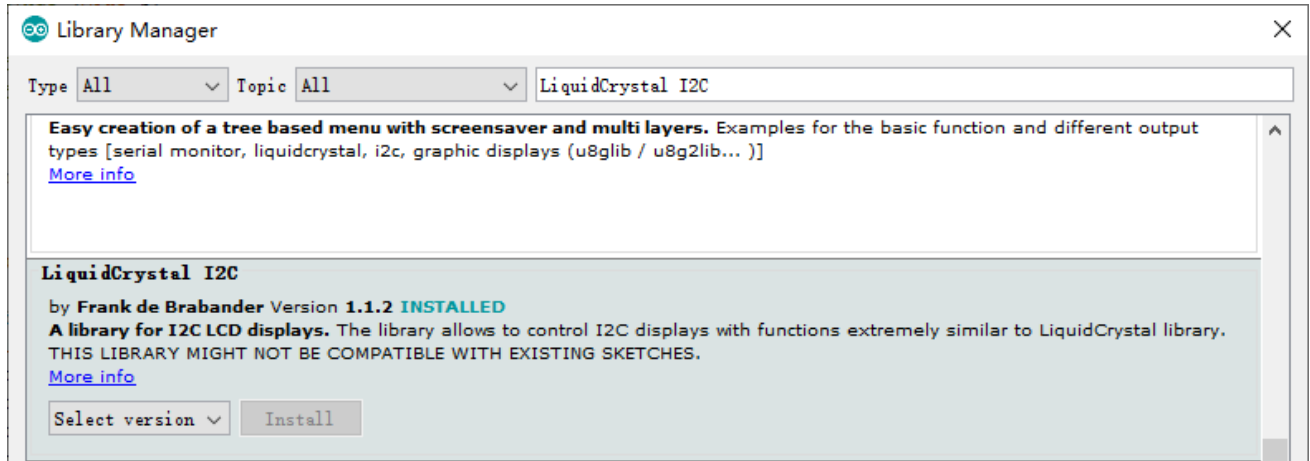Hardware connection. If you need any support, please feel free to contact us via: **support@freenove.com**

Any concerns? ✉ **support@freenove.com**

Sketch

How to install the library

We use the third party library **LiquidCrystal I2C**. If you haven't installed it yet, please do so before learning. The steps to add third-party Libraries are as follows: open arduino->Sketch->Include library-> Manage libraries. Enter " LiquidCrystal I2C" in the search bar and select " LiquidCrystal I2C " for installation.
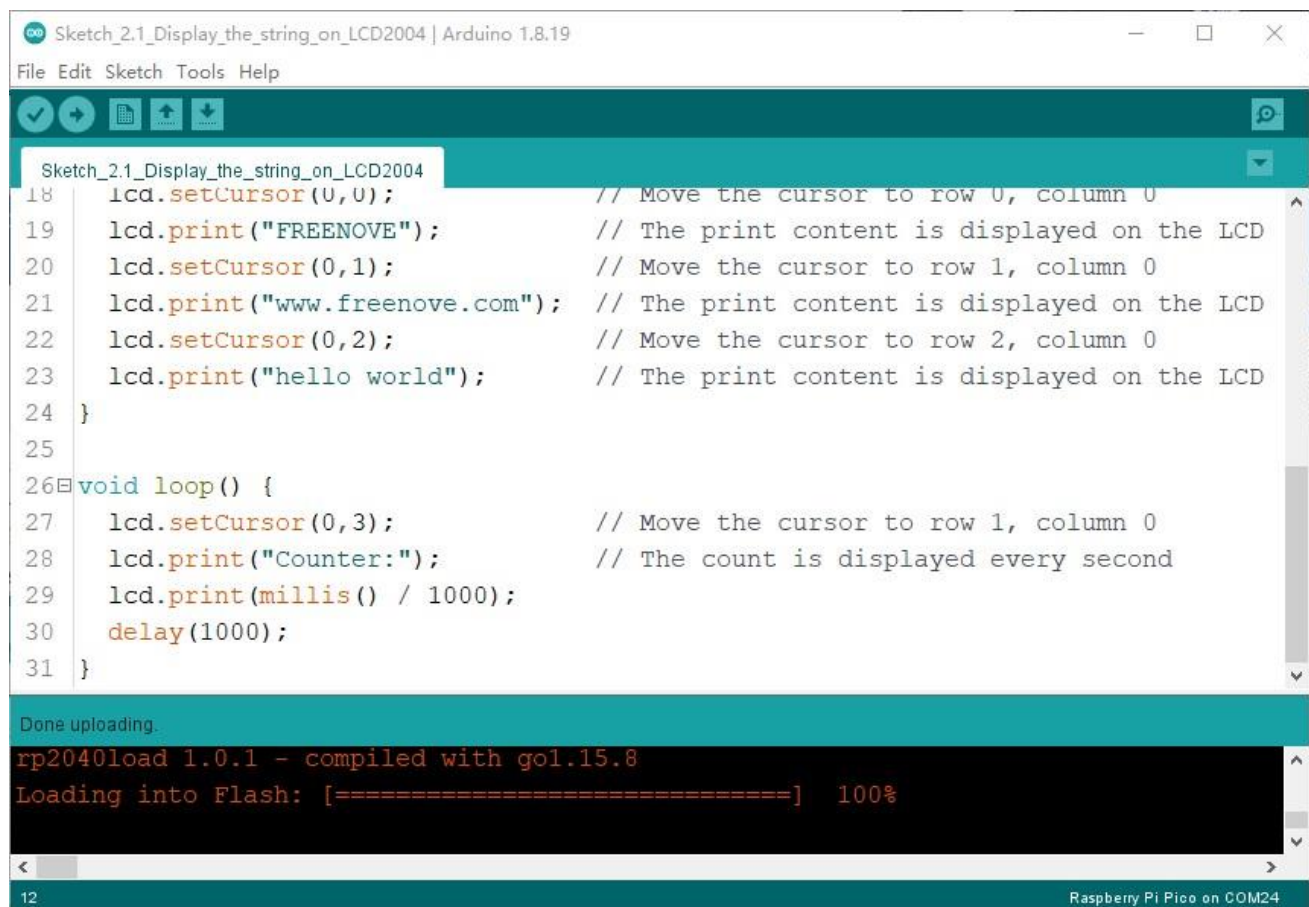

There is another way you can install libraries.

Click "Add .ZIP Library…" and then find **LiquidCrystal_I2C.zip** in libraries folder (this folder is in the folder unzipped form the ZIP file we provided). This library can facilitate our operation of I2C LCD2004.

Use I2C LCD 2004 to display characters and variables.

Sketch_2.1_Display_the_string_on_LCD2004

Compile and upload the code to Pico and the LCD2004 displays characters.

If you cannot see anything on the display or the display is not clear, try rotating the white knob on back of LCD2004 slowly, which adjusts the contrast, until the screen can display clearly.

Any concerns? ✉ **support@freenove.com**

The following is the program code:

```
1   #include <LiquidCrystal_I2C.h>
2   /*
3   * note:If lcd2004 uses PCF8574T, IIC's address is 0x27,
4   * or lcd2004 uses PCF8574AT, IIC's address is 0x3F.
5   */
6   LiquidCrystal_I2C lcd(0x27,20,4);
7   void setup() {
8     lcd.init(); // LCD driver initialization
9     lcd.backlight(); // Open the backlight
10    lcd.setCursor(0,0);              // Move the cursor to row 0, column 0
11    lcd.print("FREENOVE");          // The print content is displayed on the LCD
12    lcd.setCursor(0,1);             // Move the cursor to row 1, column 0
13    lcd.print("www.freenove.com");  // The print content is displayed on the LCD
14    lcd.setCursor(0,2);             // Move the cursor to row 2, column 0
15    lcd.print("hello world");       // The print content is displayed on the LCD
16  }
17  void loop() {
18    lcd.setCursor(0,3); // Move the cursor to row 3, column 0
19    lcd.print("Counter:"); // The count is displayed every second
20    lcd.print(millis() / 1000);
21    delay(1000);
22  }
```

Include header file of Liquid Crystal Display (LCD)2004 and I2C.

```
1   #include <LiquidCrystal_I2C.h>
```

Instantiate the I2C LCD2004 screen. It should be noted here that if your LCD driver chip uses PCF8574T, set the I2C address to 0x27, and if uses PCF8574AT, set the I2C address to 0x3F.

```
6   LiquidCrystal_I2C lcd(0x27,20,4);
```

Initialize LCD2004 and turn on the backlight of LCD.

```
8     lcd.init(); // LCD driver initialization
9     lcd.backlight(); // Open the backlight
```

Move the cursor of LCD2004 to the third row, first column, and print out "Hello, world!"

Any concerns? ✉ **support@freenove.com**

| 14 | `    lcd.move_to(0, 2)` |
| 15 | `    lcd.putstr("Hello, world!")` |

Print the number on the fourth line of LCD2004.

17	`void loop() {`
18	`  lcd.setCursor(0,3); // Move the cursor to row 3, column 0`
19	`  lcd.print("Counter:"); // The count is displayed every second`
20	`  lcd.print(millis() / 1000);`
21	`  delay(1000);`
22	`}`

Reference

class LiquidCrystal

The LiquidCrystal class can manipulate common LCD screens. The first step is defining an object of LiquidCrystal, for example:

`LiquidCrystal_I2C lcd(0x27,20,4);`

Instantiate the Lcd2004 and set the I2C address to 0x27, with 20 columns per row and 4 rows per column.

`init();`

Initializes the Lcd2004's device

`backlight();`

Turn on Lcd2004's backlight.

`setCursor(column,row);`

Sets the screen's column and row.

column: The range is 0 to 19.

row: The range is 0 to 3.

`print(String);`

Print the character string on Lcd2004.

What's Next?

THANK YOU for participating in this learning experience!

We have reached the end of this Tutorial. If you find errors, omissions or you have suggestions and/or questions about the Tutorial or component contents of this Kit, please feel free to contact us: **support@freenove.com**
We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.
http://www.freenove.com/

Thank you again for choosing Freenove products.

Any concerns? ✉ **support@freenove.com**