Chapter 1 LCD1602

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

Project 1.1 LCD1602

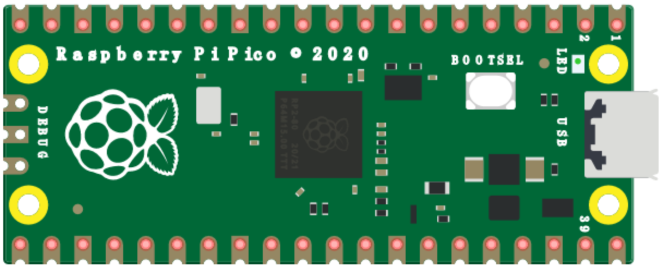
In this section we learn how to use LCD1602 to display something.

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.

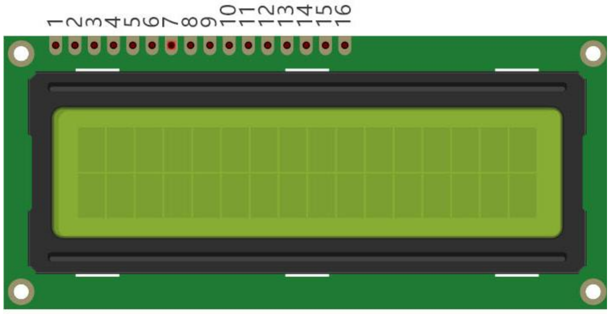


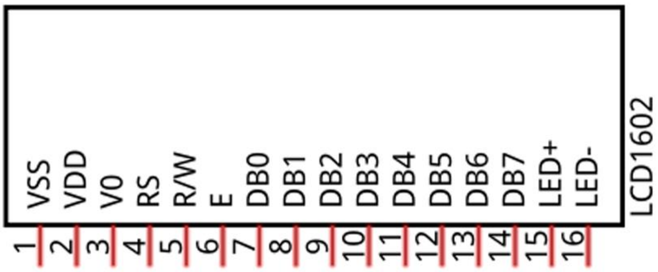
I2C communication

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.

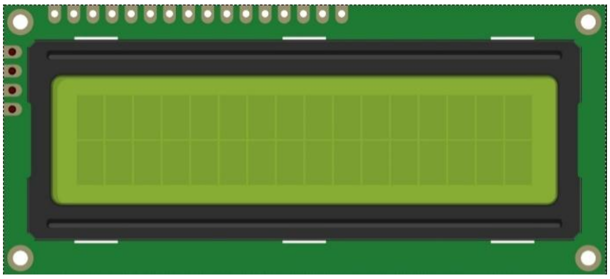
LCD1602 communication

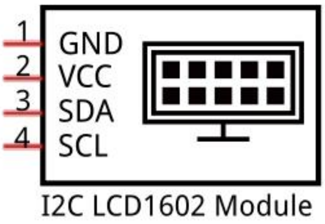
The LCD1602 Display Screen can display 2 lines of characters in 16 columns. It is capable of displaying numbers, letters, symbols, ASCII code and so on. As shown below is a monochrome LCD1602 Display Screen along with its circuit pin diagram.





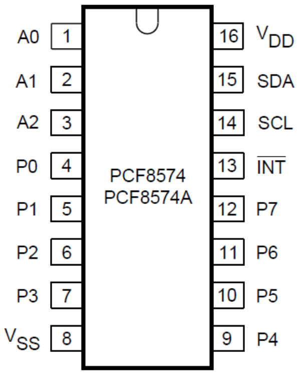
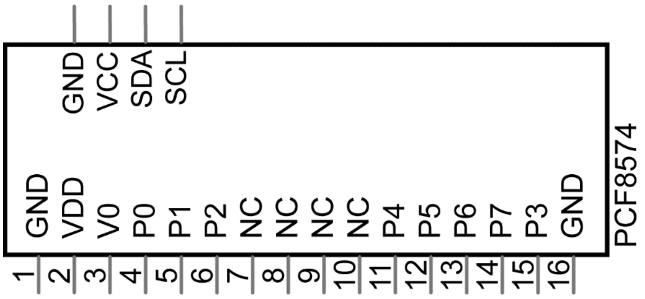
I2C LCD1602 Display Screen integrates an I2C interface, which connects the serial-input & parallel-output module to the LCD1602 Display Screen. This allows us to use only 4 lines to the operate the LCD1602.



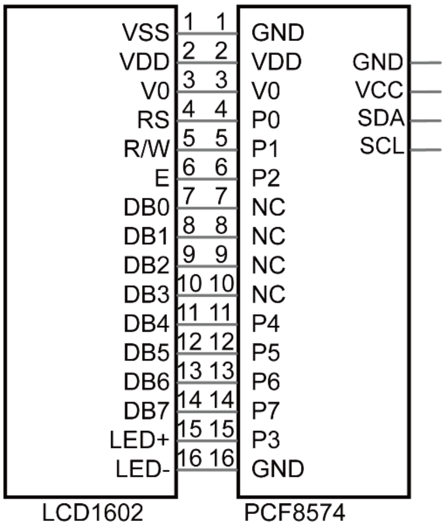


The serial-to-parallel IC chip used in this module is PCF8574T (PCF8574AT), and its default I2C address is 0x27(0x3F).

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

PCF8574 chip pin diagram	PCF8574 module pin diagram
	

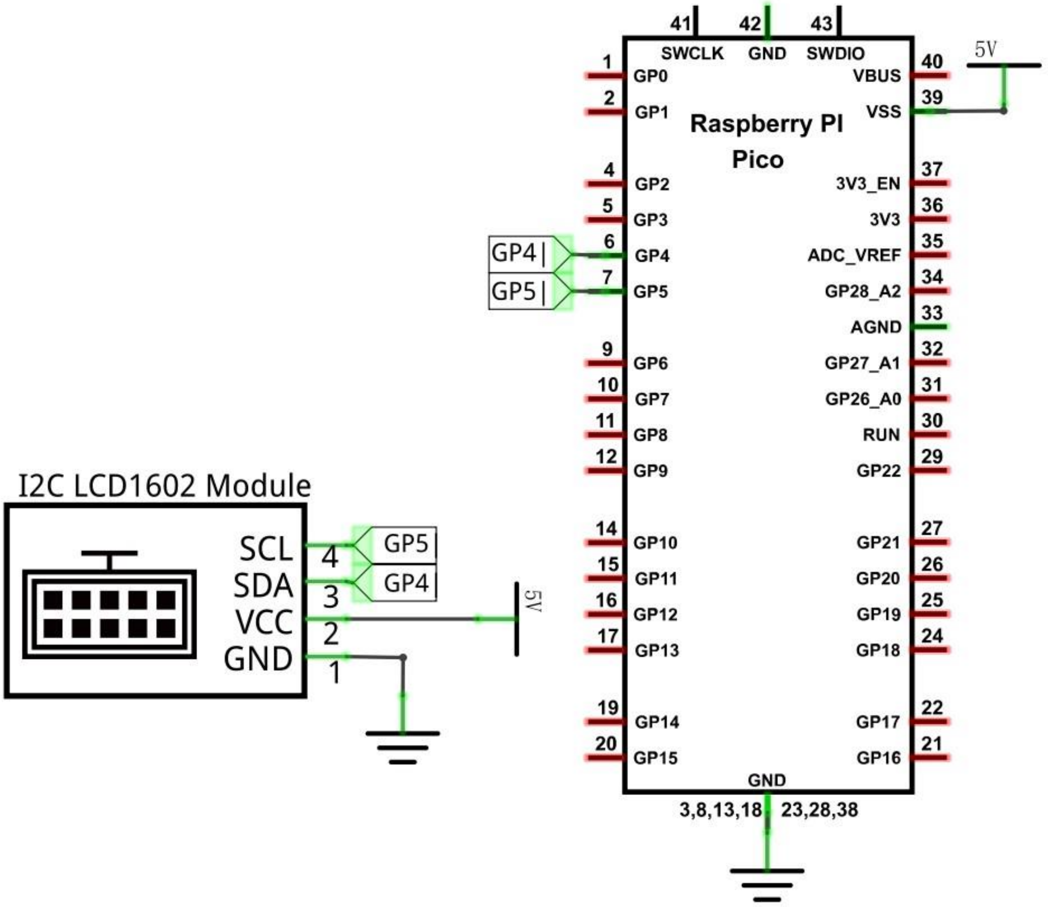
PCF8574 module pin and LCD1602 pin are corresponding to each other and connected with each other:



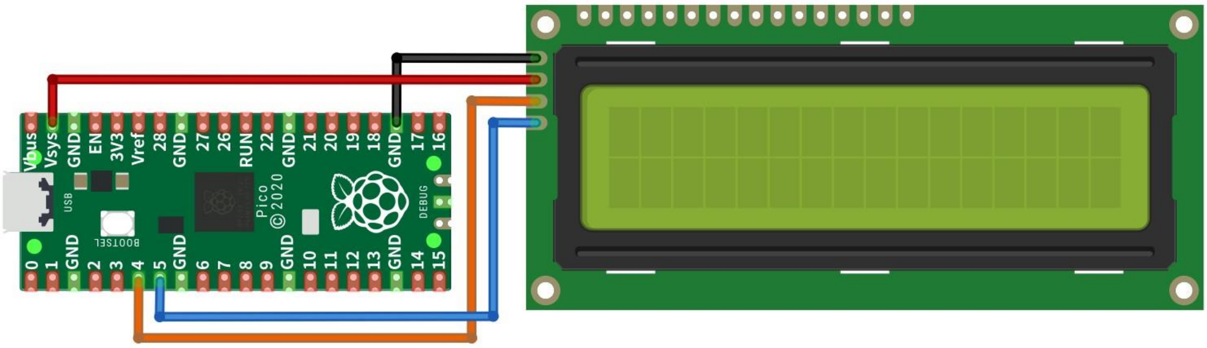
So we only need 4 pins to control the 16 pins of the LCD1602 Display Screen through the I2C interface.
In this project, we will use the I2C LCD1602 to display some static characters and dynamic variables.

Circuit

Schematic diagram



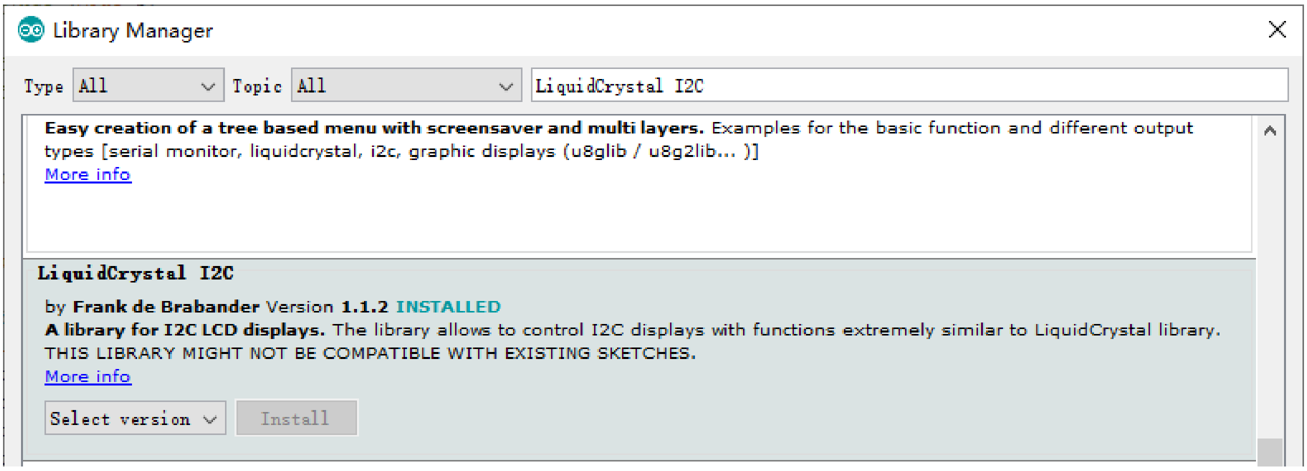
Hardware connection. If you need any support, please feel free to contact us via: support@freenove.com



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.



Easy creation of a tree based menu with screensaver and multi layers. Examples for the basic function and different output types [serial monitor, liquidcrystal, i2c, graphic displays (u8glib / u8g2lib...)]
More info

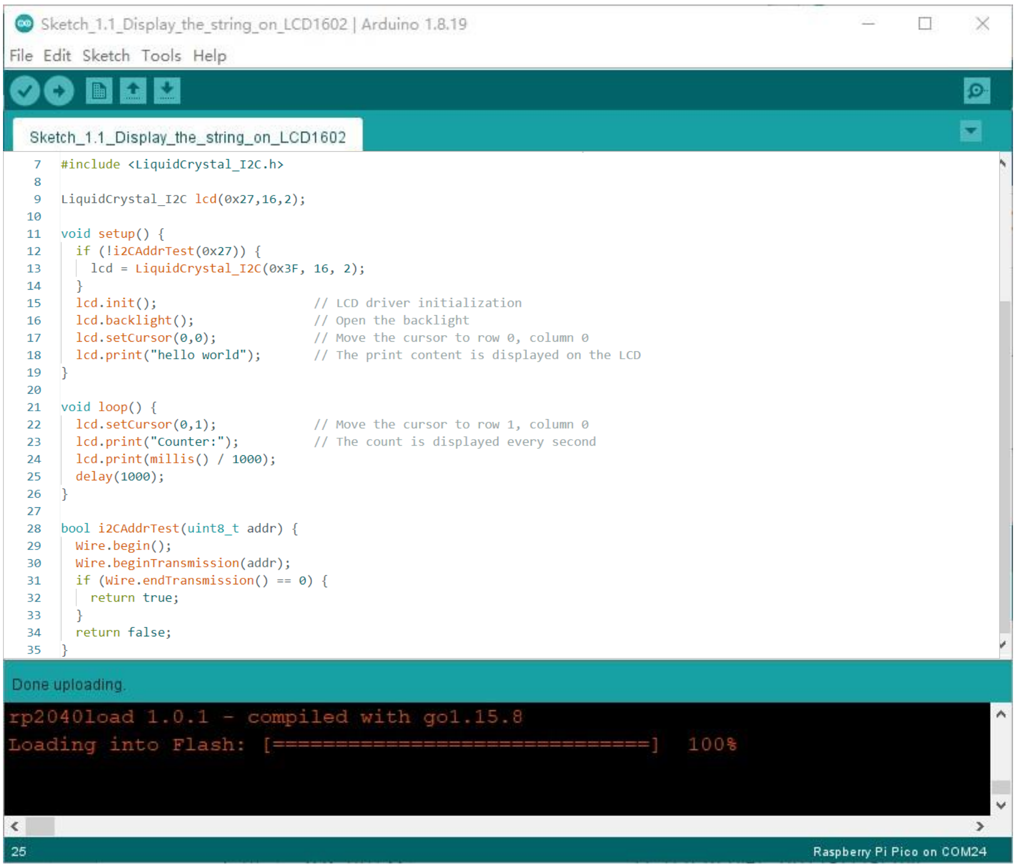
LiquidCrystal I2C
by **Frank de Brabander** Version **1.1.2** INSTALLED
A library for I2C LCD displays. The library allows to control I2C displays with functions extremely similar to LiquidCrystal library. THIS LIBRARY MIGHT NOT BE COMPATIBLE WITH EXISTING SKETCHES.
More info

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



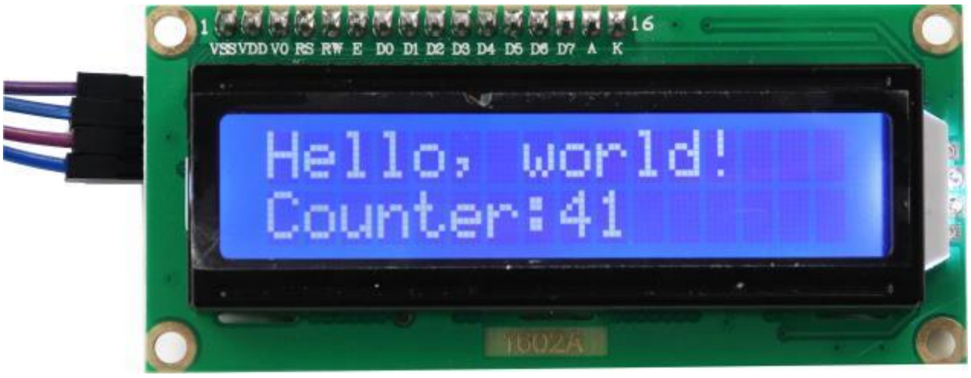
Sketch_1.1_Display_the_string_on_LCD1602 | Arduino 1.8.19
File Edit Sketch Tools Help
Sketch_1.1_Display_the_string_on_LCD1602

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);

void setup() {
  if (!i2CAddrTest(0x27)) {
    lcd = LiquidCrystal_I2C(0x3F, 16, 2);
  }
  lcd.init();                     // LCD driver initialization
  lcd.backlight();                // Open the backlight
  lcd.setCursor(0,0);             // Move the cursor to row 0, column 0
  lcd.print("hello world");       // The print content is displayed on the LCD
}

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

bool i2CAddrTest(uint8_t addr) {
  Wire.begin();
  Wire.beginTransmission(addr);
  if (Wire.endTransmission() == 0) {
    return true;
  }
  return false;
}
```

Done uploading.

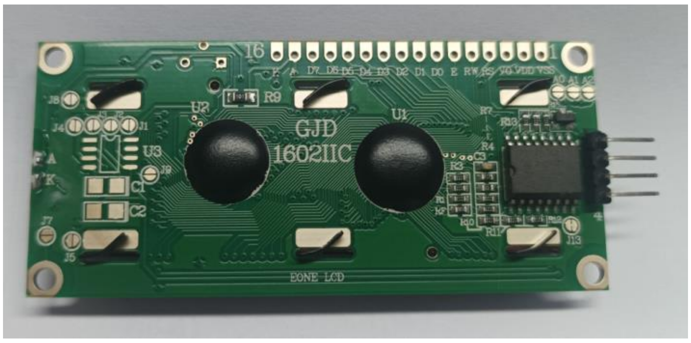
```
rp2040load 1.0.1 - compiled with go1.15.8
Loading into Flash: [==============================]  100%
```

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

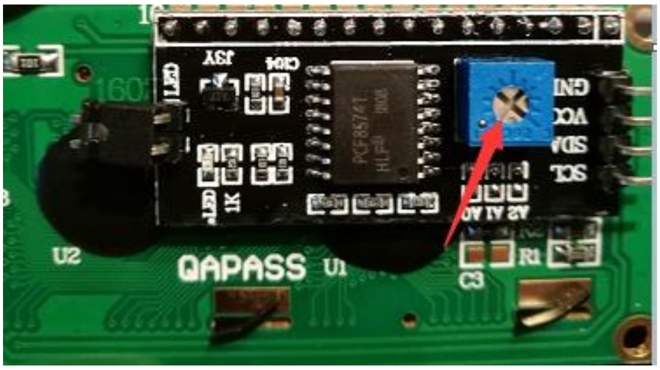


So far, at this writing, we have two types of LCD1602 on sale. One needs to adjust the backlight, and the other does not.

The LCD1602 that does not need to adjust the backlight is shown in the figure below.



If the LCD1602 you received is the following one, and 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.



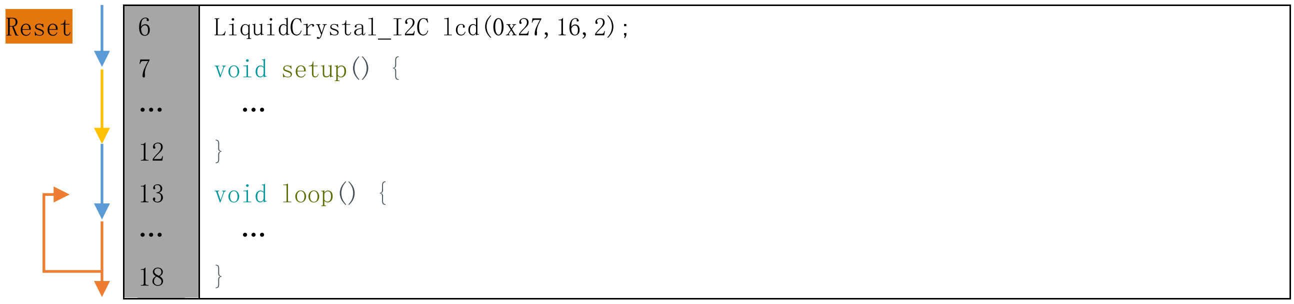
The following is the program code:

```
#include <LiquidCrystal_I2C.h>
/*
* note:If lcd1602 uses PCF8574T, IIC's address is 0x27,
* or lcd1602 uses PCF8574AT, IIC's address is 0x3F.
*/
LiquidCrystal_I2C lcd(0x27,16,2);
void setup() {
  if (!i2CAddrTest(0x27)) {
    lcd = LiquidCrystal_I2C(0x3F, 16, 2);
  }
  lcd.init(); // LCD driver initialization
  lcd.backlight(); // Open the backlight
  lcd.setCursor(0,0); // Move the cursor to row 0, column 0
  lcd.print("hello world"); // The print content is displayed on the LCD
}
void loop() {
  lcd.setCursor(0,1); // Move the cursor to row 1, column 0
  lcd.print("Counter:"); // The count is displayed every second
  lcd.print(millis() / 1000);
  delay(1000);
}
bool i2CAddrTest(uint8_t addr) {
  Wire.begin();
  Wire.beginTransmission(addr);
  if (Wire.endTransmission() == 0) {
    return true;
  }
  return false;
}
```

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.



```
LiquidCrystal_I2C lcd(0x27,16,2);
void setup() {
  …
}
void loop() {
  …
}
```

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

```
#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.

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

Initialize LCD1602 and turn on the backlight of LCD.

```
if (!i2CAddrTest(0x27)) {
    lcd = LiquidCrystal_I2C(0x3F, 16, 2);
  }
  lcd.init(); // LCD driver initialization
  lcd.backlight(); // Open the backlight
```

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

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

Print the number on the second line of LCD1602.

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

Check whether the I2C address exists.

```
bool i2CAddrTest(uint8_t addr) {
  Wire.begin();
  Wire.beginTransmission(addr);
  if (Wire.endTransmission() == 0) {
    return true;
  }
  return false;
}
```

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

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.

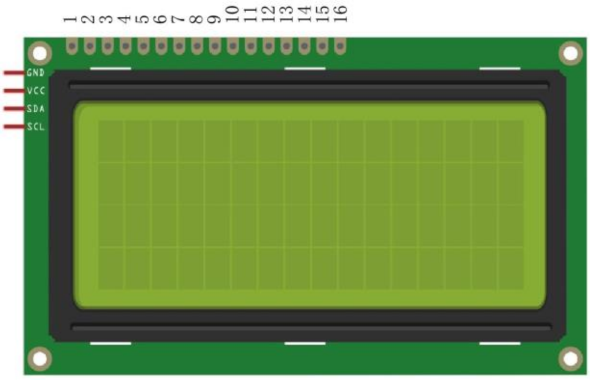
Component knowledge

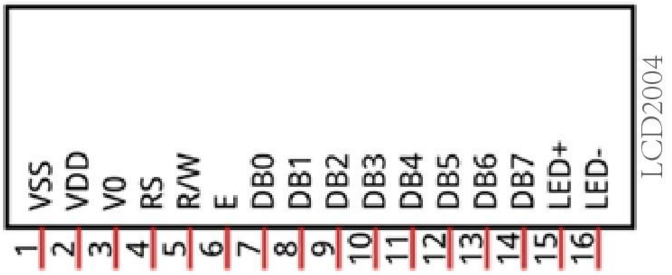
I2C communication

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.

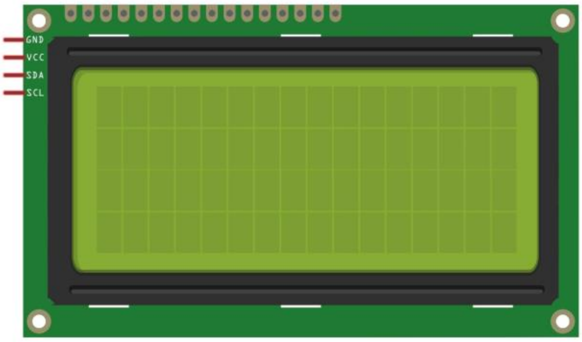
LCD2004 communication

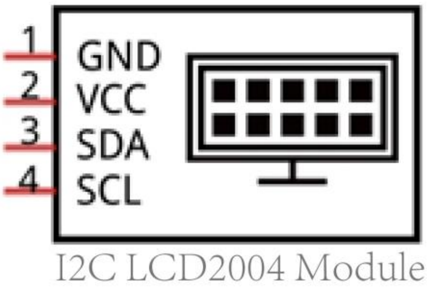
The LCD2004 display screen can display 4 lines of characters in 20 columns. It is capable of displaying numbers, letters, symbols, ASCII code and so on. As shown below is a monochrome LCD2004 display screen along with its circuit pin diagram.





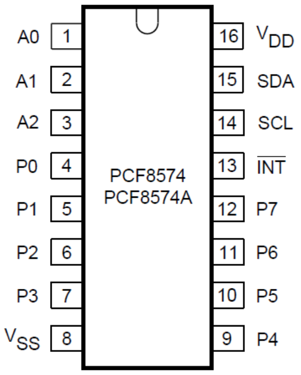
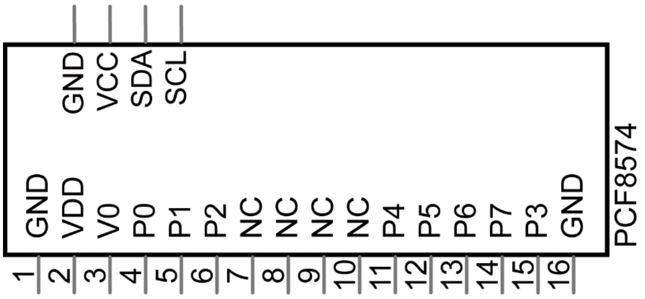
I2C LCD2004 display screen integrates a I2C interface, which connects the serial-input & parallel-output module to the LCD2004 display screen. This allows us to only use 4 lines to the operate the LCD2004.



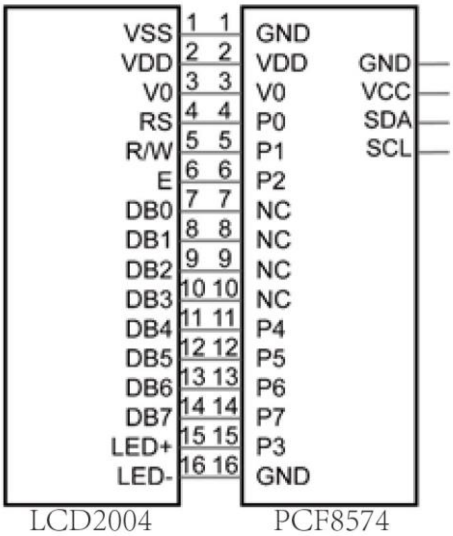


The serial-to-parallel IC chip used in this module is PCF8574T (PCF8574AT), and its default I2C address is 0x27(0x3F).

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

PCF8574 chip pin diagram	PCF8574 module pin diagram
	

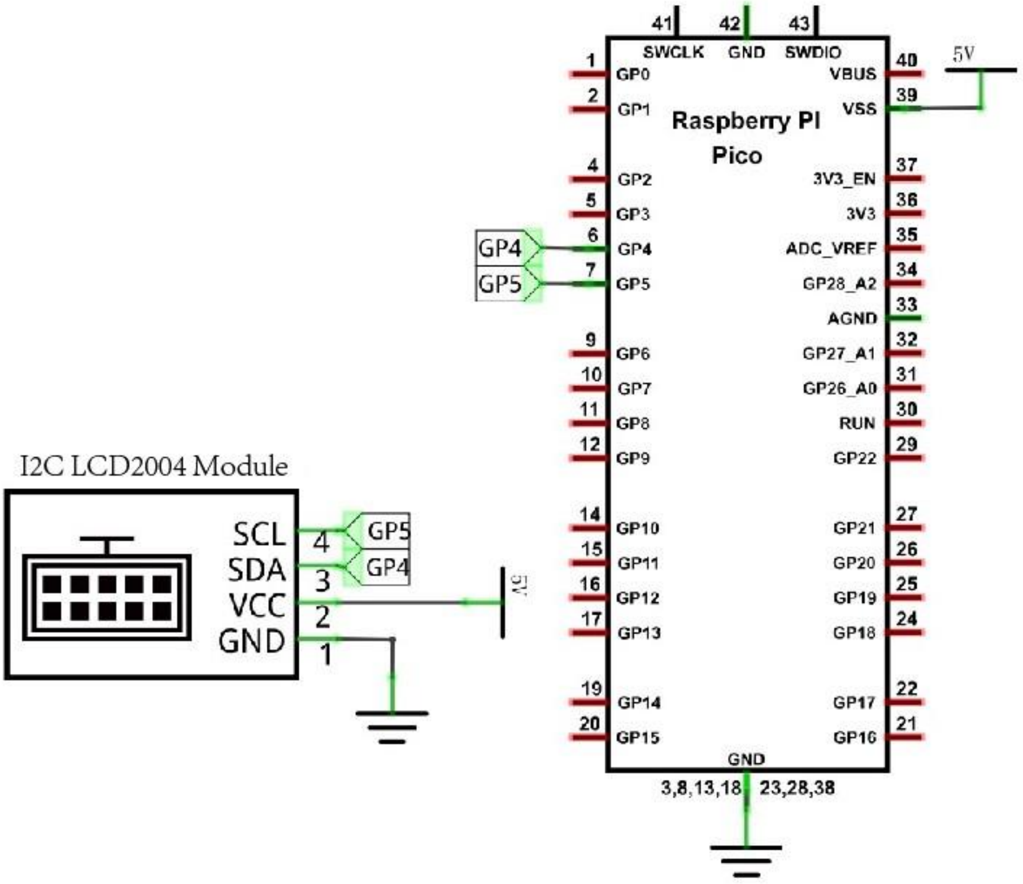
PCF8574 module pin and LCD2004 pin are corresponding to each other and connected with each other:



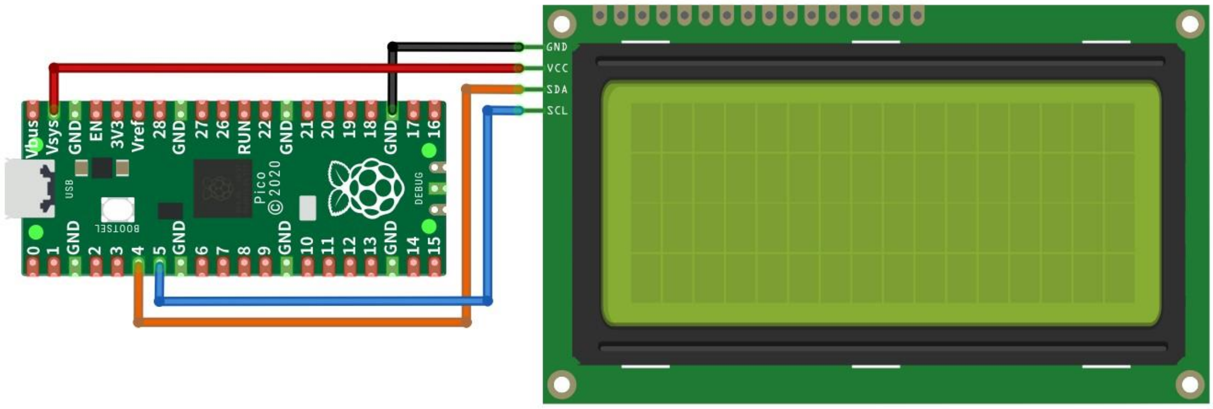
So we only need 4 pins to control the 16 pins of the LCD2004 Display Screen through the I2C interface.
In this project, we will use the I2C LCD2004 to display some static characters and dynamic variables.

Circuit

Schematic diagram



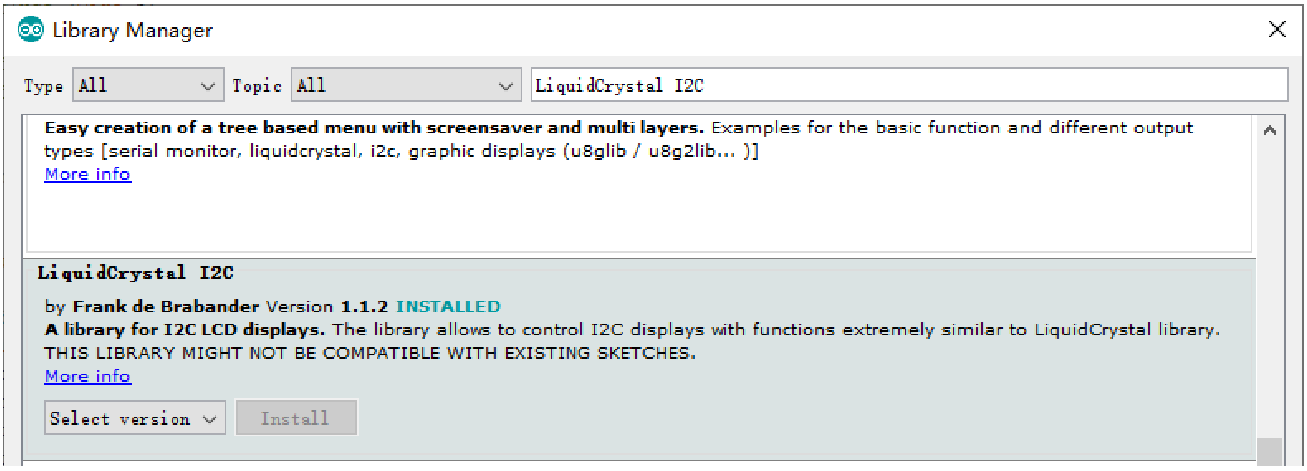
Hardware connection. If you need any support, please feel free to contact us via: support@freenove.com



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.



Easy creation of a tree based menu with screensaver and multi layers. Examples for the basic function and different output types [serial monitor, liquidcrystal, i2c, graphic displays (u8glib / u8g2lib...)]

LiquidCrystal I2C

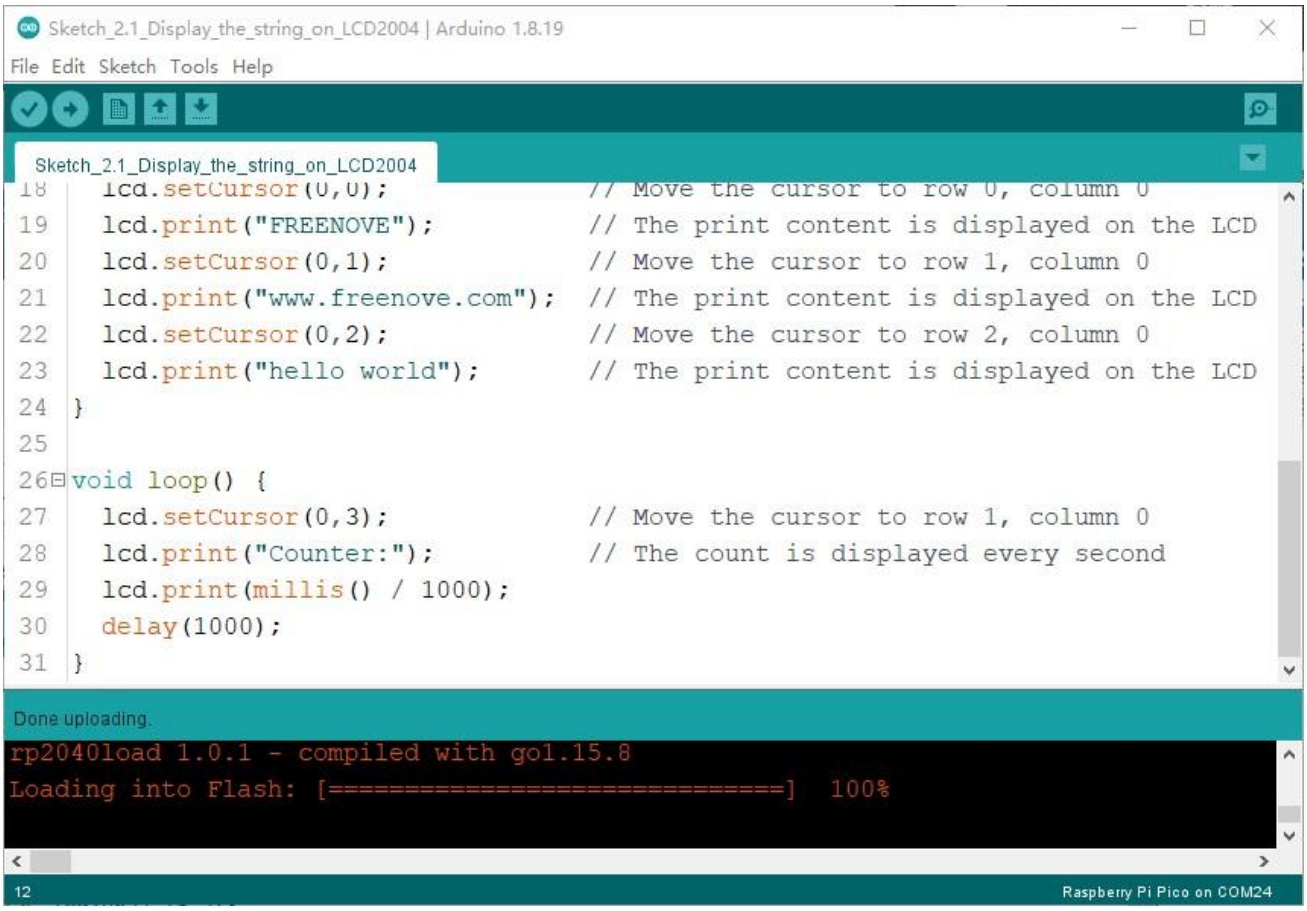
by **Frank de Brabander** Version **1.1.2** INSTALLED
A library for I2C LCD displays. The library allows to control I2C displays with functions extremely similar to LiquidCrystal library. THIS LIBRARY MIGHT NOT BE COMPATIBLE WITH EXISTING SKETCHES.

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

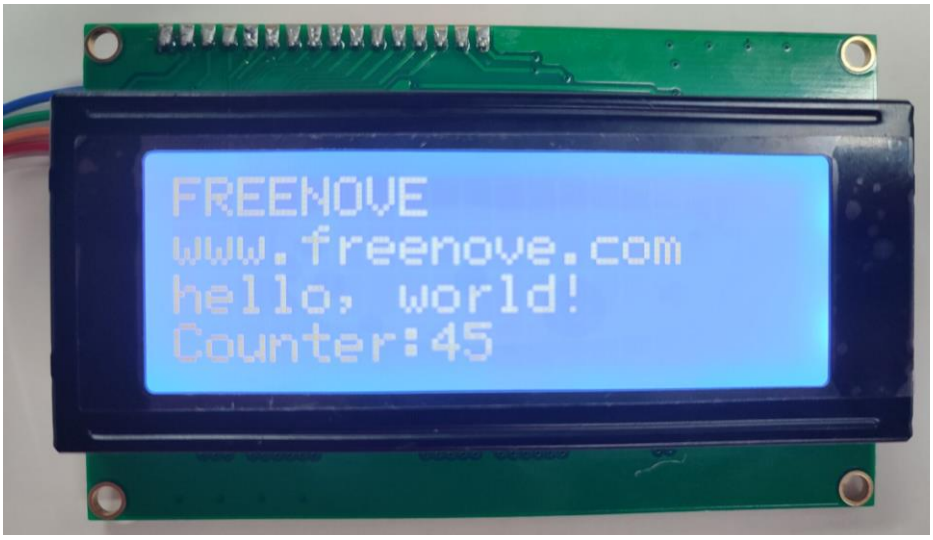

Sketch_2.1_Display_the_string_on_LCD2004 | Arduino 1.8.19
File Edit Sketch Tools Help
Sketch_2.1_Display_the_string_on_LCD2004

```
  lcd.setCursor(0,0);                  // Move the cursor to row 0, column 0
  lcd.print("FREENOVE");               // The print content is displayed on the LCD
  lcd.setCursor(0,1);                  // Move the cursor to row 1, column 0
  lcd.print("www.freenove.com");       // The print content is displayed on the LCD
  lcd.setCursor(0,2);                  // Move the cursor to row 2, column 0
  lcd.print("hello world");            // The print content is displayed on the LCD
}

void loop() {
  lcd.setCursor(0,3);                  // Move the cursor to row 1, column 0
  lcd.print("Counter:");               // The count is displayed every second
  lcd.print(millis() / 1000);
  delay(1000);
}
```

Done uploading.

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

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.

The following is the program code:

```
#include <LiquidCrystal_I2C.h>
/*
* note:If lcd2004 uses PCF8574T, IIC’s address is 0x27,
* or lcd2004 uses PCF8574AT, IIC’s address is 0x3F.
*/
LiquidCrystal_I2C lcd(0x27,20,4);
void setup() {
  lcd.init(); // LCD driver initialization
  lcd.backlight(); // Open the backlight
  lcd.setCursor(0,0);            // Move the cursor to row 0, column 0
  lcd.print("FREENOVE");         // The print content is displayed on the LCD
  lcd.setCursor(0,1);            // Move the cursor to row 1, column 0
  lcd.print("www.freenove.com"); // The print content is displayed on the LCD
  lcd.setCursor(0,2);            // Move the cursor to row 2, column 0
  lcd.print("hello world");      // The print content is displayed on the LCD
}
void loop() {
  lcd.setCursor(0,3); // Move the cursor to row 3, column 0
  lcd.print("Counter:"); // The count is displayed every second
  lcd.print(millis() / 1000);
  delay(1000);
}
```

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

```
#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.

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

Initialize LCD2004 and turn on the backlight of LCD.

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

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

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

Print the number on the fourth line of LCD2004.

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

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.