

Chapter 1 I2C LCD1602

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

Project 1.1 Display the String on I2C LCD1602

In this section we learn how to use lcd1602 to display something.

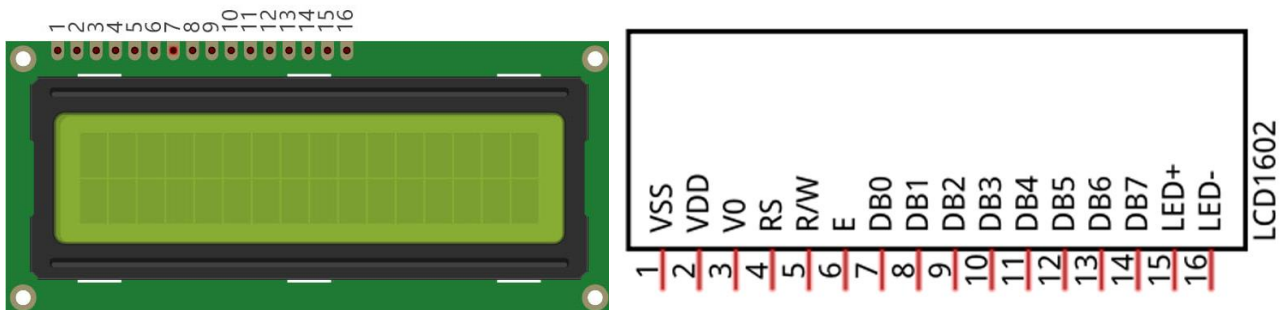
Component Knowledge

I2C communication

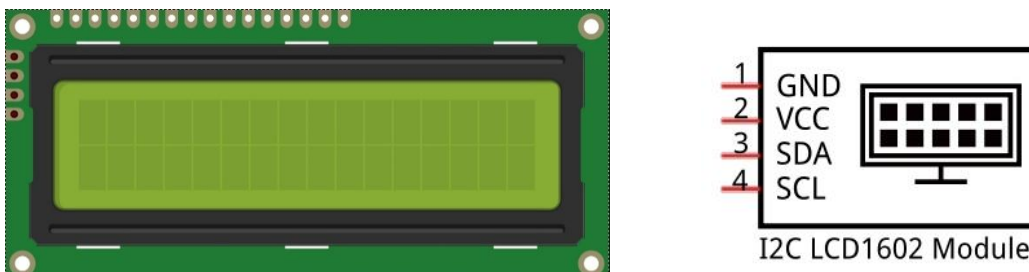
I2C (Inter-Integrated Circuit) is a two-wire serial communication mode, which can be used for the connection of micro controllers and their peripheral equipment. Devices using I2C communication must be connected to the serial data (SDA) line, and serial clock (SCL) line (called I2C bus). Each device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

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.

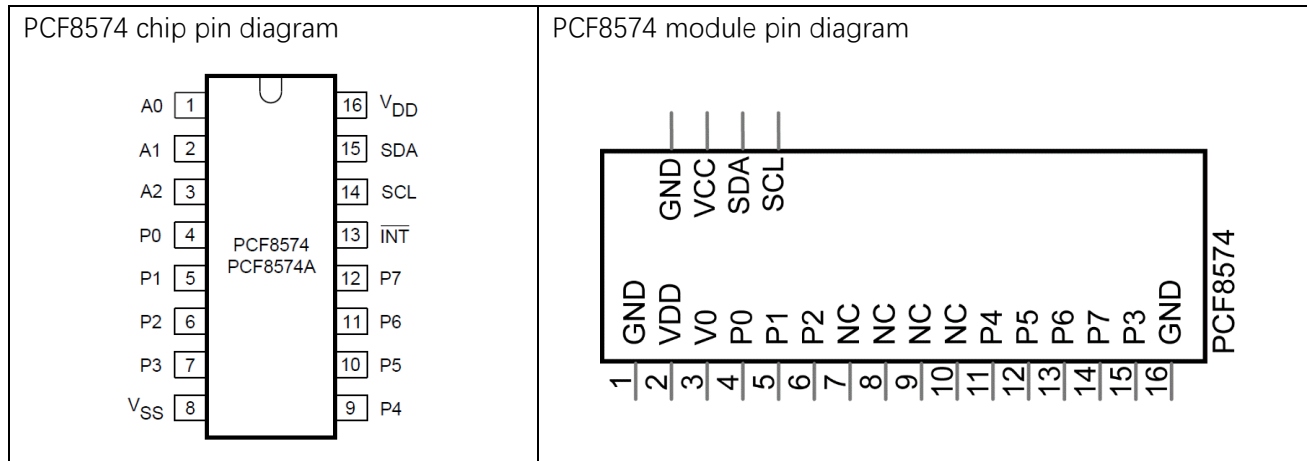


I2C LCD1602 Display Screen integrates an I2C interface, which connects the serial-input & parallel-output module to the LCD1602 Display Screen. This allows us to use only 4 lines to operate the LCD1602.

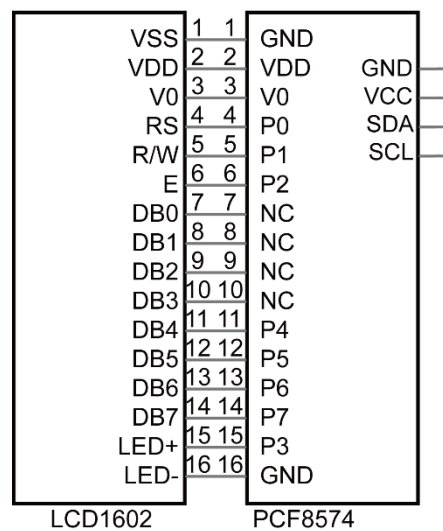


The serial-to-parallel IC chip used in this module is PCF8574T (PCF8574AT), and its default I2C address is 0x27(0x3F).

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



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

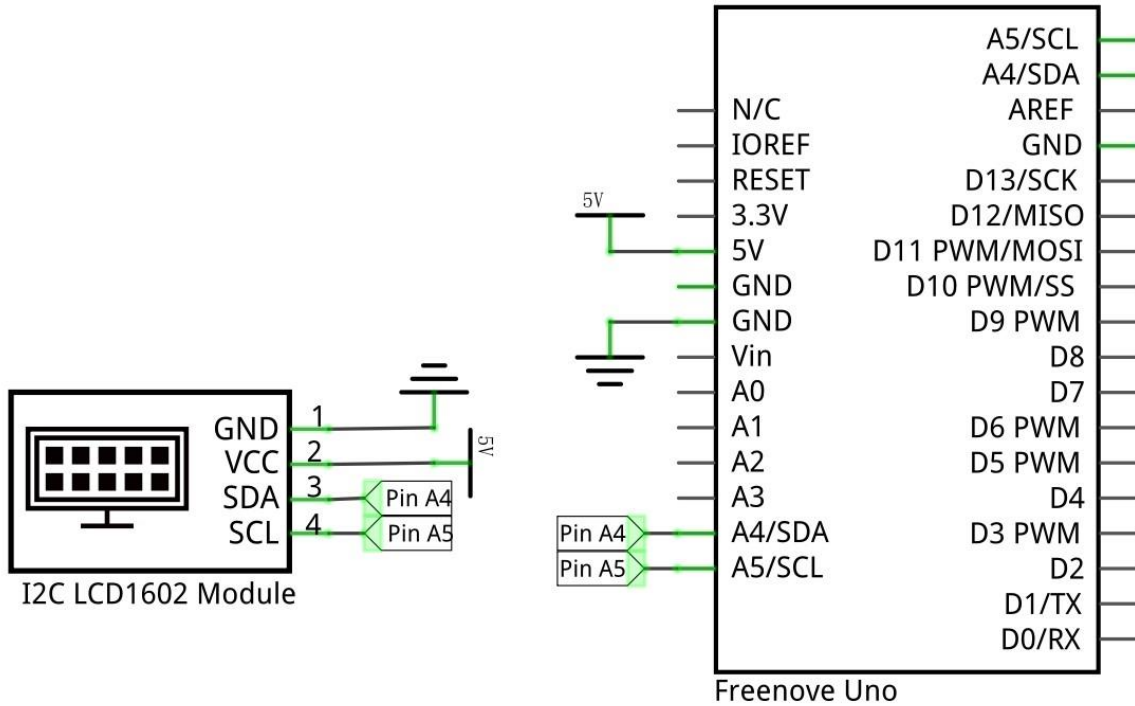


So we only need 4 pins to control the 16 pins of the LCD1602 Display Screen through the I2C interface. In this project, we will use the I2C LCD1602 to display some static characters and dynamic variables.

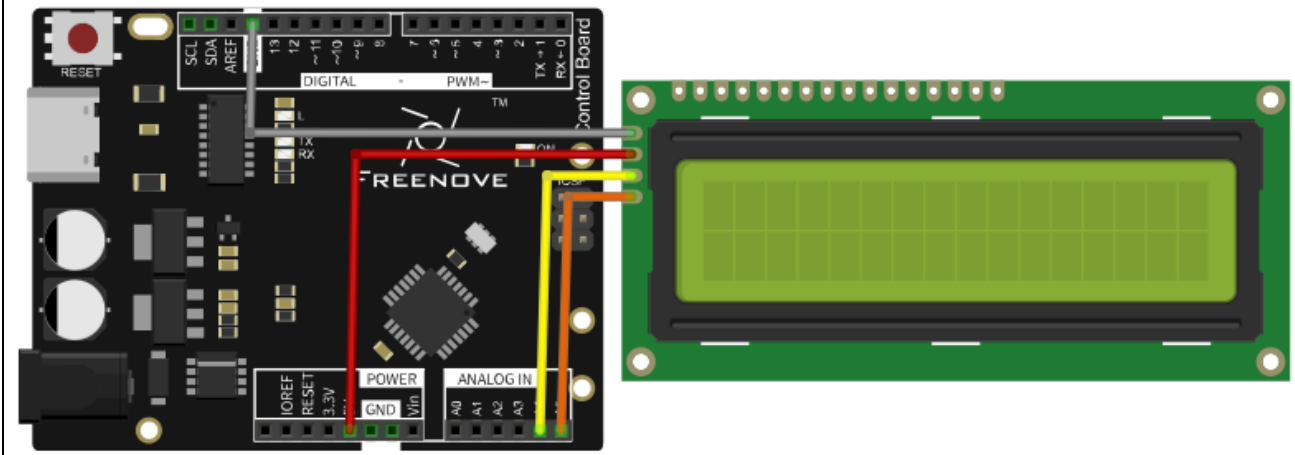
Circuit

The connection of control board and I2C LCD1602 is shown below.

Schematic diagram



Circuit connection

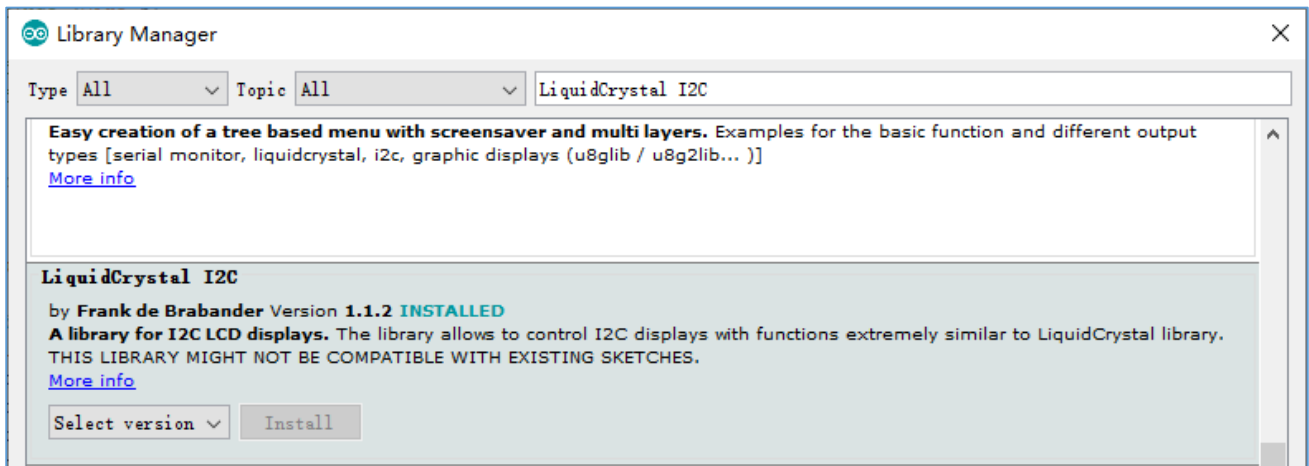


Sketch

Before writing code, we need to import the library needed.

How to install the library

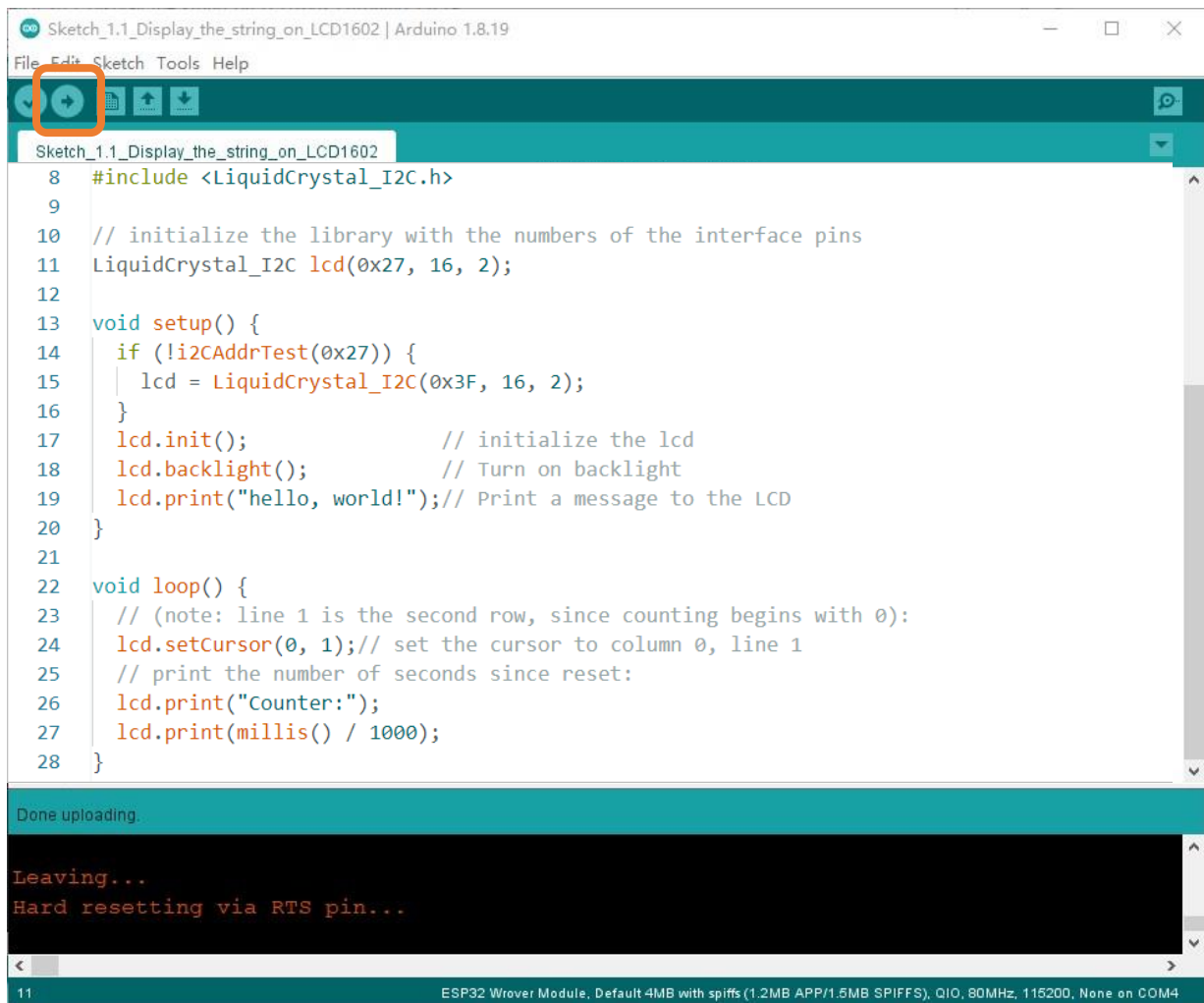
We use the third party library **LiquidCrystal I2C**. If you haven't installed it yet, please do so before learning. The steps to add third-party Libraries are as follows: open arduino->Sketch->Include library-> Manage libraries. Enter " LiquidCrystal I2C" in the search bar and select " LiquidCrystal 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 from the ZIP file we provided). This library can facilitate our operation of I2C LCD1602.

Sketch_1.1_Display_the_string_on_LCD1602

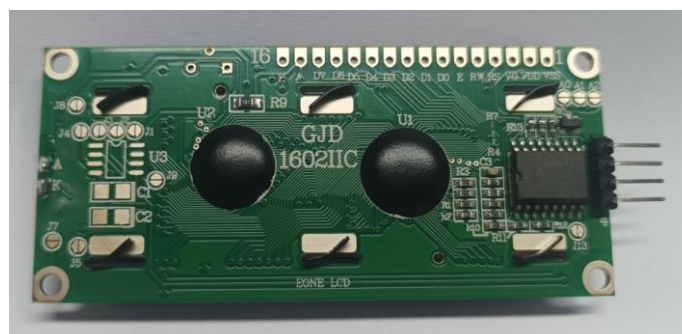


Compile and upload the code to Arduino and the LCD1602 displays characters.



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.



Now let's start to write code to use LCD1602 to display static characters and dynamic variables.

```

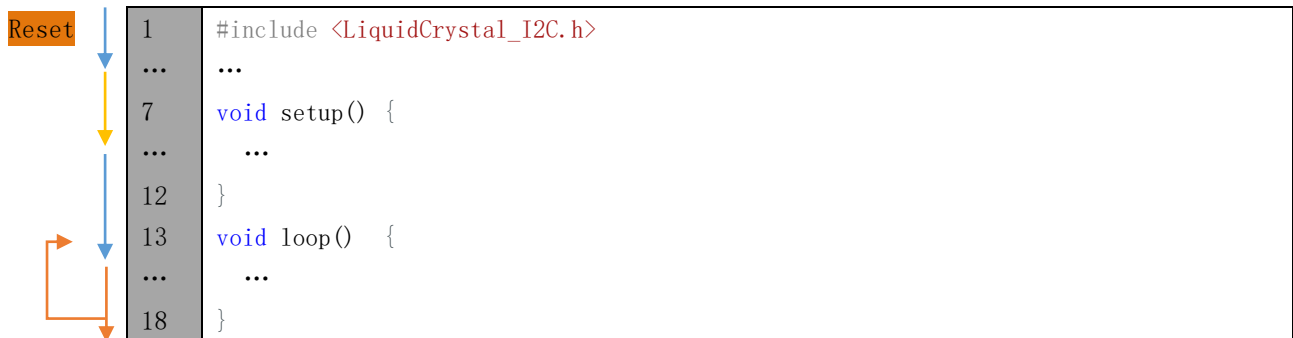
1  #include <LiquidCrystal_I2C.h>
2  /*
3   * note:If lcd1602 uses PCF8574T, IIC's address is 0x27,
4   * or lcd1602 uses PCF8574AT, IIC's address is 0x3F.
5   */
6  LiquidCrystal_I2C lcd(0x27, 16, 2);
7  void setup() {
8      if (!i2CAddrTest(0x27)) {
9          lcd = LiquidCrystal_I2C(0x3F, 16, 2);
10     }
11     lcd.init(); // LCD driver initialization
12     lcd.backlight(); // Open the backlight
13     lcd.setCursor(0,0); // Move the cursor to row 0, column 0
14     lcd.print("hello world"); // The print content is displayed on the LCD
15 }
16 void loop() {
17     lcd.setCursor(0,1); // Move the cursor to row 1, column 0
18     lcd.print("Counter:"); // The count is displayed every second
19     lcd.print(millis() / 1000);
20     delay(1000);
21 }
22 bool i2CAddrTest(uint8_t addr) {
23     Wire.begin();
24     Wire.beginTransmission(addr);
25     if (Wire.endTransmission() == 0) {
26         return true;
27     }
28     return false;
29 }

```

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

After control board is **reset**, the `setup()` function will be executed first, and then the `loop()` function will be executed.

`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.



Following are the `LiquidCrystal_I2C` library used for controlling LCD:

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

`LiquidCrystal_I2C` library provides `LiquidCrystal_I2C` class that controls LCD1602. When we instantiate a `LiquidCrystal_I2C` object, we can input some parameters. 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. And these parameters are the row/column numbers of the I2C addresses and screen that connect to LCD1602:

```
6  LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x27 for a 16 chars and 2 line display
```

First, initialize the LCD and turn on LCD backlight.

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

And then print a string:

```
11  lcd.print("hello world"); // The print content is displayed on the LCD
```

Print a changing number in the `loop()` function:

```
16  void loop() {
17      lcd.setCursor(0, 1); // Move the cursor to row 1, column 0
18      lcd.print("Counter:"); // The count is displayed every second
19      lcd.print(millis() / 1000);
20      delay(1000);
21  }
```

Before printing characters, we need to set the coordinate of the printed character, that is, in which line and which column:

```
17  lcd.setCursor(0, 1); // Move the cursor to row 1, column 0
```

LiquidCrystal_I2C Class

`LiquidCrystal_I2C` class can control common LCD screen. First, we need instantiate an object of `LiquidCrystal_I2C` type, for example:

LiquidCrystal_I2C lcd(0x27, 16, 2);

When an object is instantiated, a constructed function of the class is called a constructor. In the constructor

function, we need to fill in the I2C address of the LCD module, as well as the number of columns and rows of the LCD module. The number of columns and rows can also be set in the `lcd.begin()`.

The functions used in the `LiquidCrystal_I2C` class are as follows:

lcd.setCursor (col, row): set the coordinates of the to-be-printed character. The parameters are the numbers of columns and rows of the characters (start from 0, the number 0 represents first row or first line).

lcd.print (data): print characters. Characters will be printed on the coordinates set before. If you do not set the coordinates, the string will be printed behind the last printed character.

Verify and upload the code, then observe the LCD screen. If the display is not clear or there is no display, adjust the potentiometer on the back of I2C module to adjust the screen contrast until the character is clearly displayed on the LCD.



You can use the I2C LCD1602 to replace the serial port as a mobile screen when you print the data latter.

Chapter 2 I2C LCD2004

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

Project 2.1 Display the String on I2C LCD2004

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

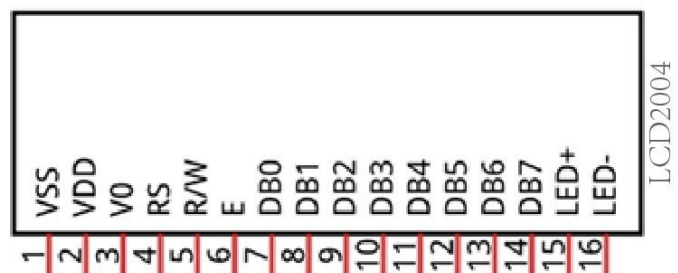
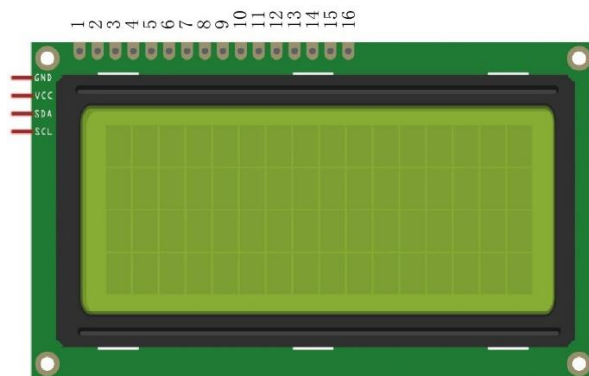
Component Knowledge

I2C communication

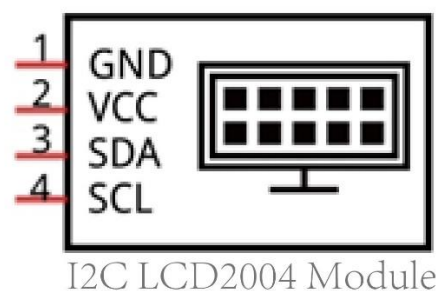
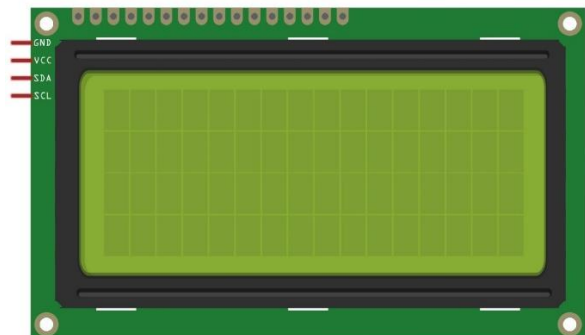
I2C (Inter-Integrated Circuit) is a two-wire serial communication mode, which can be used for the connection of micro controllers and their peripheral equipment. Devices using I2C communication must be connected to the serial data (SDA) line, and serial clock (SCL) line (called I2C bus). Each device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

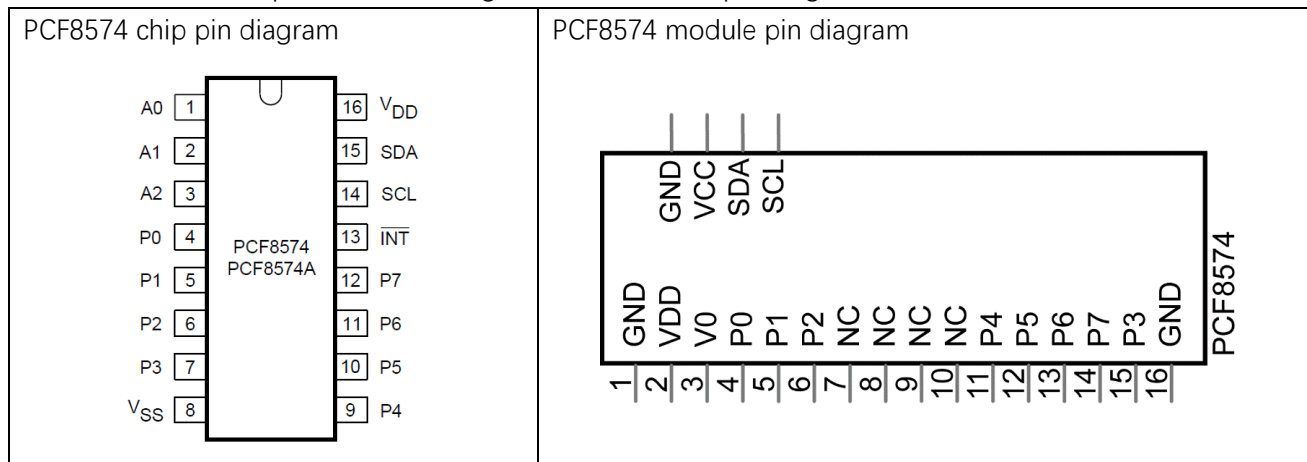


I2C LCD2004 display screen integrates a I2C interface, which connects the serial-input & parallel-output module to the LCD2004 display screen. This allows us to only use 4 lines to the operate the LCD2004.

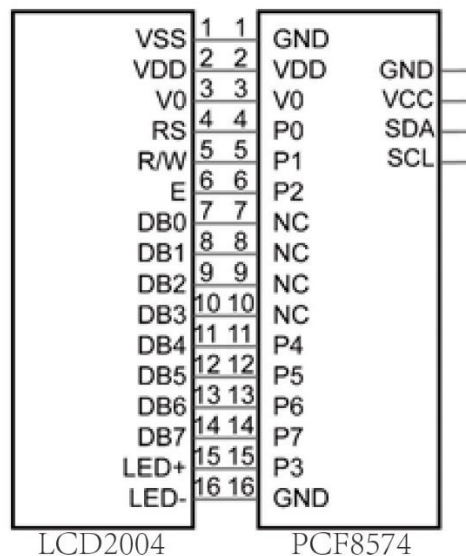


The serial-to-parallel IC chip used in this module is PCF8574T (PCF8574AT), and its default I2C address is 0x27(0x3F).

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



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

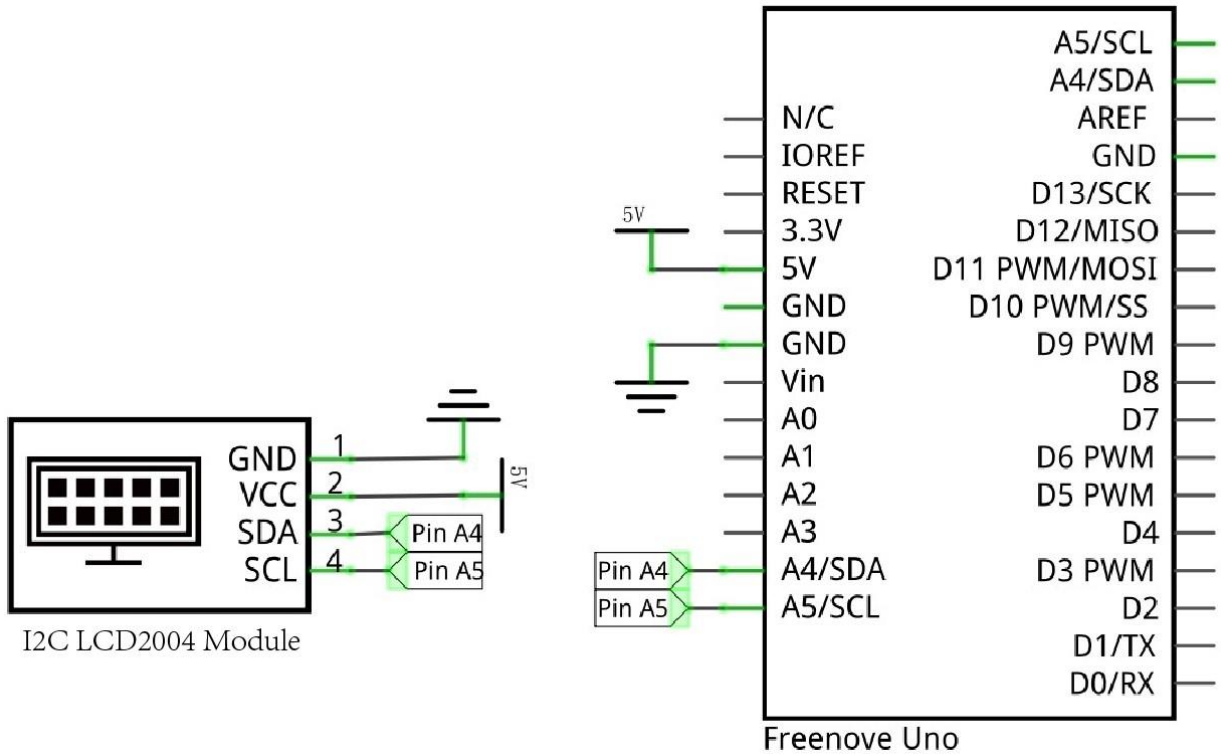


So we only need 4 pins to control the 16 pins of the LCD2004 Display Screen through the I2C interface. In this project, we will use the I2C LCD2004 to display some static characters and dynamic variables.

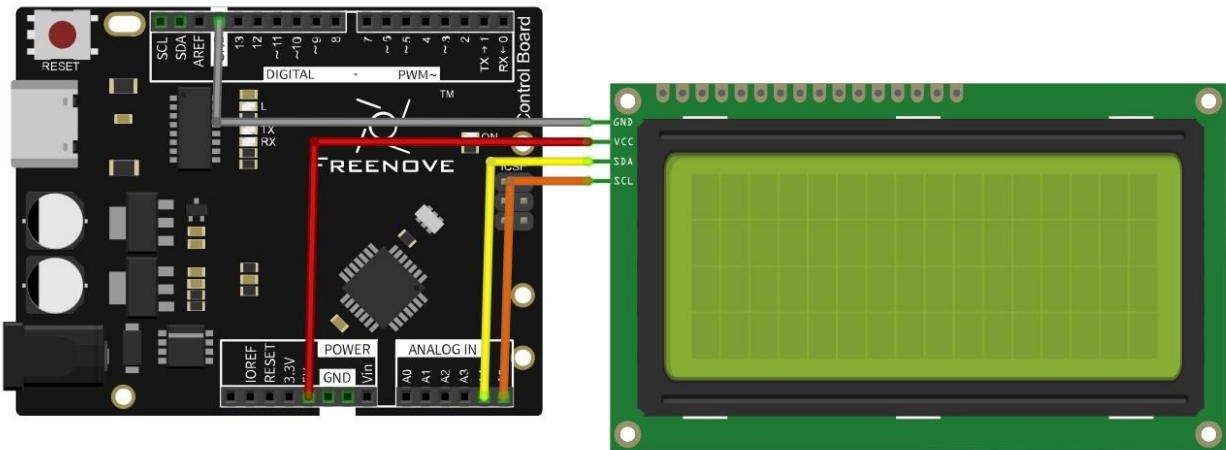
Circuit

The connection of control board and I2C LCD2004 is shown below.

Schematic diagram



Circuit connection

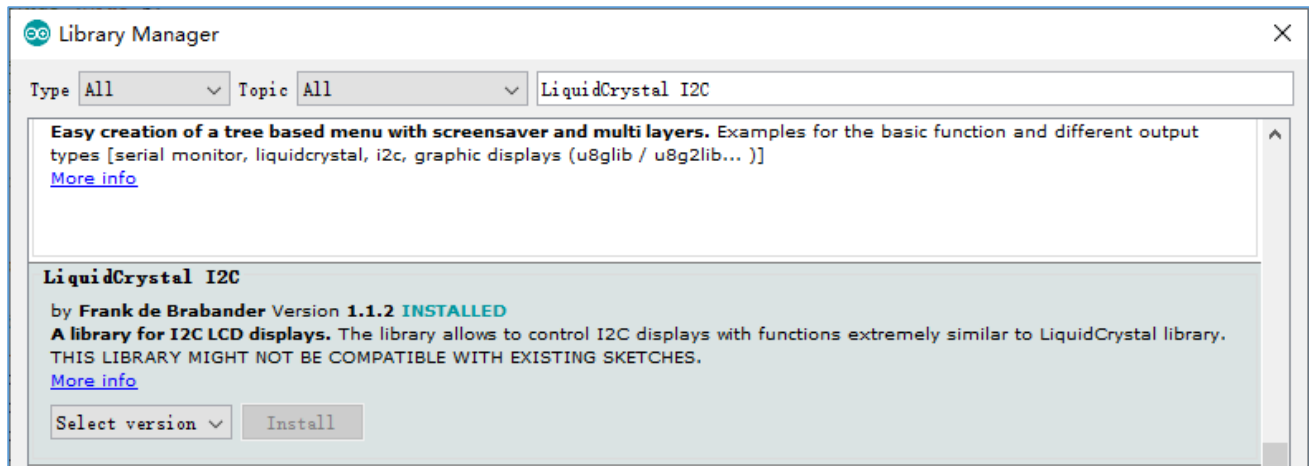


Sketch

Before writing code, we need to import the library needed. Skip this section if you have already installed it, or proceed if you haven't.

How to install the library

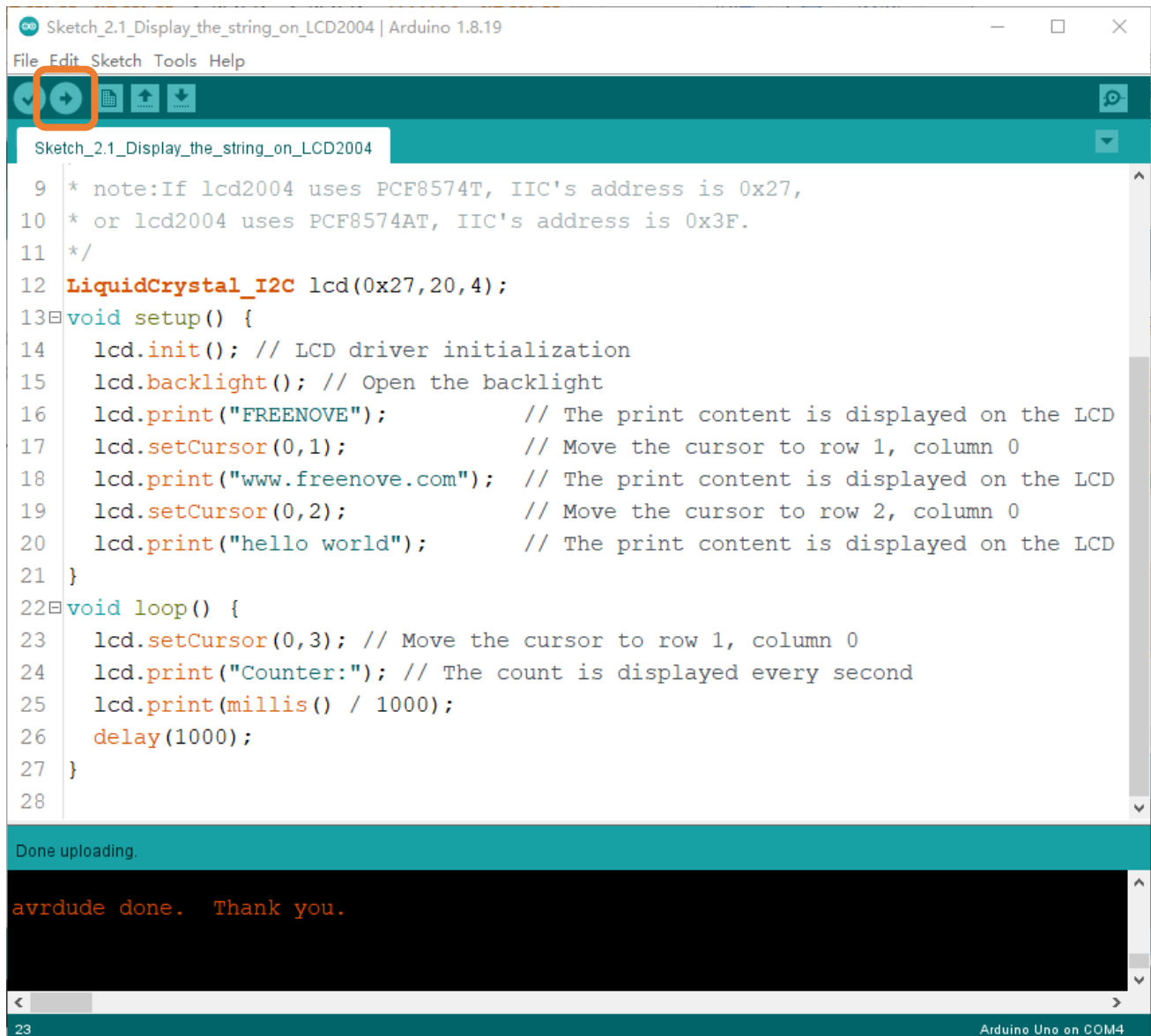
We use the third party library **LiquidCrystal I2C**. If you haven't installed it yet, please do so before learning. The steps to add third-party Libraries are as follows: open arduino->Sketch->Include library-> Manage libraries. Enter " LiquidCrystal I2C" in the search bar and select " LiquidCrystal 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 from the ZIP file we provided). This library can facilitate our operation of I2C LCD2004.

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

9  * note:If lcd2004 uses PCF8574T, IIC's address is 0x27,
10 * or lcd2004 uses PCF8574AT, IIC's address is 0x3F.
11 */
12 LiquidCrystal_I2C lcd(0x27,20,4);
13 void setup() {
14   lcd.init(); // LCD driver initialization
15   lcd.backlight(); // Open the backlight
16   lcd.print("FREENOVE");           // The print content is displayed on the LCD
17   lcd.setCursor(0,1);              // Move the cursor to row 1, column 0
18   lcd.print("www.freenove.com");   // The print content is displayed on the LCD
19   lcd.setCursor(0,2);              // Move the cursor to row 2, column 0
20   lcd.print("hello world");        // The print content is displayed on the LCD
21 }
22 void loop() {
23   lcd.setCursor(0,3); // Move the cursor to row 1, column 0
24   lcd.print("Counter:"); // The count is displayed every second
25   lcd.print(millis() / 1000);
26   delay(1000);
27 }
28

Done uploading.

avrdude done.  Thank you.

23 Arduino Uno on COM4
```

Compile and upload the code to Arduino 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.



Now let's start to write code to use LCD2004 to display static characters and dynamic variables.

```

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 1, column 0
19     lcd.print("Counter:"); // The count is displayed every second
20     lcd.print(millis() / 1000);
21     delay(1000);
22 }

```

Following are the LiquidCrystal_I2C library used for controlling LCD:

```

1  #include <LiquidCrystal_I2C.h>

```

LiquidCrystal_I2C library provides LiquidCrystal_I2C class that controls LCD2004. When we instantiate a LiquidCrystal_I2C object, we can input some parameters. And these parameters are the row/column numbers of the I2C addresses and screen that connect to LCD2004:

```

6  LiquidCrystal_I2C lcd(0x27, 20 4); //set the LCD address to 0x27 for a 16 chars and 2 line display

```

First, initialize the LCD and turn on LCD backlight.

```

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

```

And then print a string:

```
14  lcd.print("hello world");           // The print content is displayed on the LCD
```

Print a changing number in the loop () function:

```
17  void loop() {  
18      lcd.setCursor(0,3); // Move the cursor to row 1, column 0  
19      lcd.print("Counter:"); // The count is displayed every second  
20      lcd.print(millis() / 1000);  
21      delay(1000);  
22  }
```

Before printing characters, we need to set the coordinate of the printed character, that is, in which line and which column:

```
12  lcd.setCursor(0,1);                 // Move the cursor to row 1, column 0
```

LiquidCrystal_I2C Class

LiquidCrystal_I2C class can control common LCD screen. First, we need instantiate an object of LiquidCrystal_I2C type, for example:

LiquidCrystal_I2C lcd(0x27, 20, 4);

When an object is instantiated, a constructed function of the class is called a constructor. In the constructor function, we need to fill in the I2C address of the LCD module, as well as the number of columns and rows of the LCD module. The number of columns and rows can also be set in the lcd.begin ().

The functions used in the LiquidCrystal_I2C class are as follows:

lcd.setCursor (col, row): set the coordinates of the to-be-printed character. The parameters are the numbers of columns and rows of the characters (start from 0, the number 0 represents first row or first line).

lcd.print (data): print characters. Characters will be printed on the coordinates set before. If you do not set the coordinates, the string will be printed behind the last printed character.

Verify and upload the code, then observe the LCD screen. If the display is not clear or there is no display, adjust the potentiometer on the back of I2C module to adjust the screen contrast until the character is clearly displayed on the LCD.



You can use the I2C LCD2004 to replace the serial port as a mobile screen when you print the data latter.