

Welcome

Thank you for choosing Freenove products!

How to Start

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

Attention

Pay attention to safety when using and storing this product:

- This product is not suitable for children under 12 years of age because of small parts and sharp parts.
- Minors should use this product under the supervision and guidance of adults.
- This product contains small and sharp parts. Do not swallow, prick and scratch to avoid injury.
- This product contains conductive parts. Do not hold them to touch power supply and other circuits.
- To avoid personal injury, do not touch parts rotating or moving while working.
- The wrong operation may cause overheat. Do not touch and disconnect the power supply immediately.
- Operate in accordance with the requirements of the tutorial. Fail to do so may damage the parts.
- Store this product in a dry and dark environment. Keep away from children.
- Turn off the power of the circuit before leaving.

About

Freenove provides open source electronic products and services.

Freenove is committed to helping customers learn programming and electronic knowledge, quickly implement product prototypes, realize their creativity and launch innovative products. Our services include:

- Kits for learning programming and electronics
- Kits compatible with Arduino®, Raspberry Pi®, micro:bit®, etc.
- Kits for robots, smart cars, drones, etc.
- Components, modules and tools
- Design and customization

To learn more about us or get our latest information, please visit our website:

<http://www.freenove.com>

Copyright

All the files provided in the ZIP file are released under [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/). You can find a copy of the license in the ZIP file.



It means you can use these files on your own derived works, in part or completely. But not for commercial use.

Freenove® brand and logo are trademarks of Freenove Creative Technology Co., Ltd. Must not be used without 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/>).

micro:bit® is a trademark of Micro:bit Educational Foundation (<https://www.microbit.org/>).

Contents

Welcome	i
Contents	1
Preface	2
Control Board	2
Programming Software	4
First Use	7
Chapter 1 I2C LCD1602	11
Project 16.1 Display the String on I2C LCD1602	11
Chapter 2 I2C LCD2004	20
Project 2.1 Display the String on I2C LCD2004	20
What's next?	28
What's next?(others)	28
End of the Tutorial	28

Preface

If you want to make some interesting projects or want to learn electronics and programming, this document will greatly help you.

Projects in this document usually contains two parts: the circuit and the code. No experience at all? Don't worry, this document will show you how to start from scratch.

If you encounter any problems, please feel free to send us an email, we will try our best to help you.

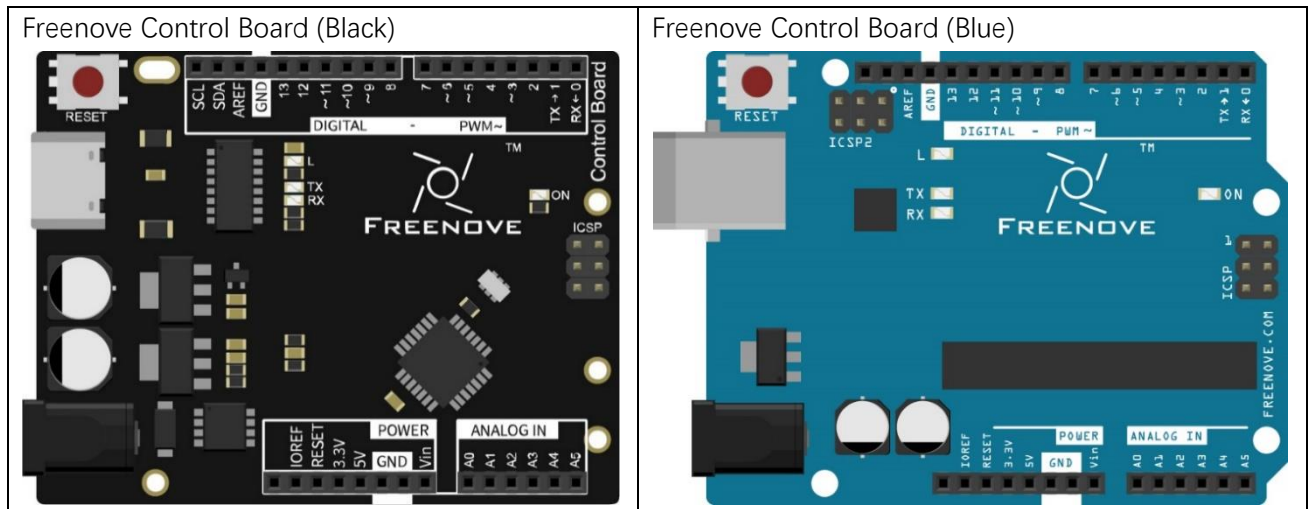
Support email: support@freenove.com

To complete these projects, you need to use a control board and software to program it, as well as some commonly used components.

Control Board

The control board is the core of a circuit. After programming, it can be used to control other components in the circuit to achieve intended functions.

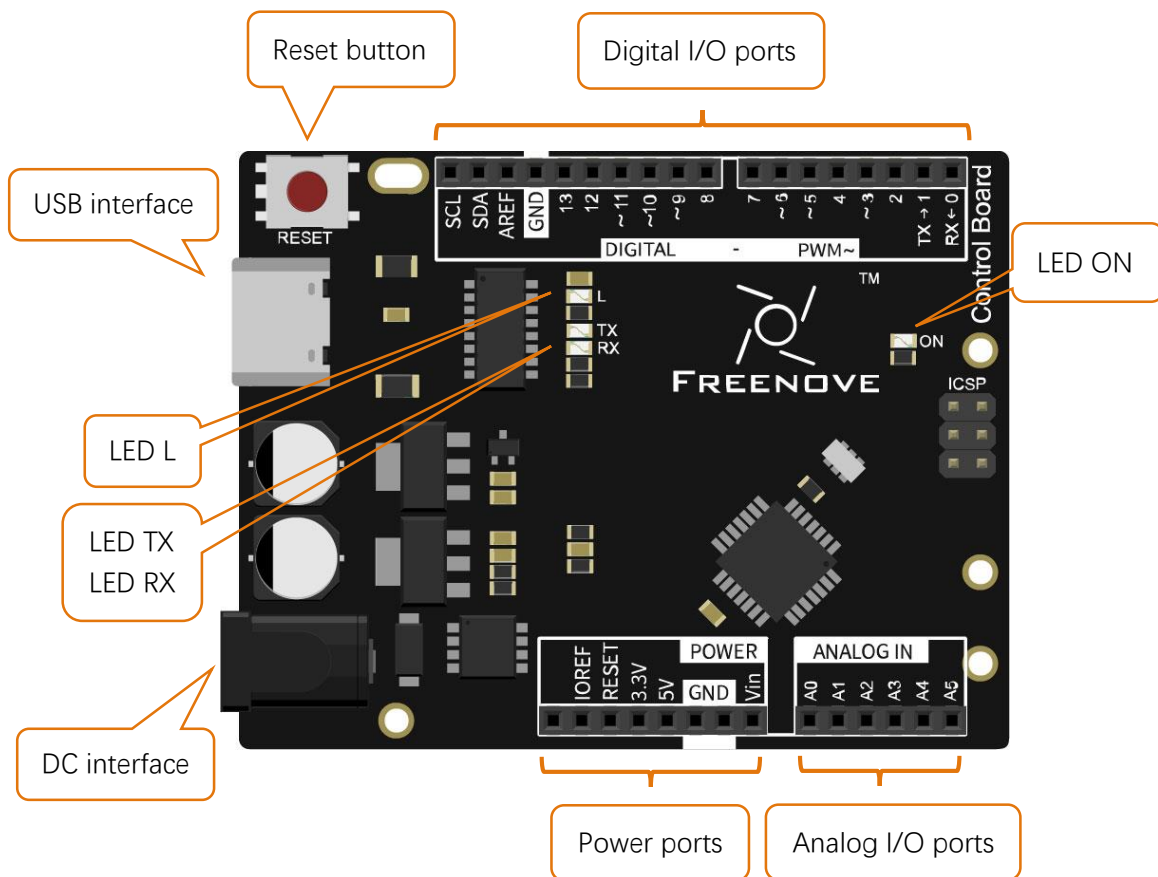
There are multiple versions of Freenove control board. Your purchase may be one of the following:



Note: Although the colors and shapes of these control boards are somewhat different, their ports and functions are the same. They can be replaced with each other, and there is no difference in their usages.

Note: Only the black control board is used to display the hardware connection in this document. The hardware connection of the blue control board is the same.

Diagram of the Freenove control board is shown below:



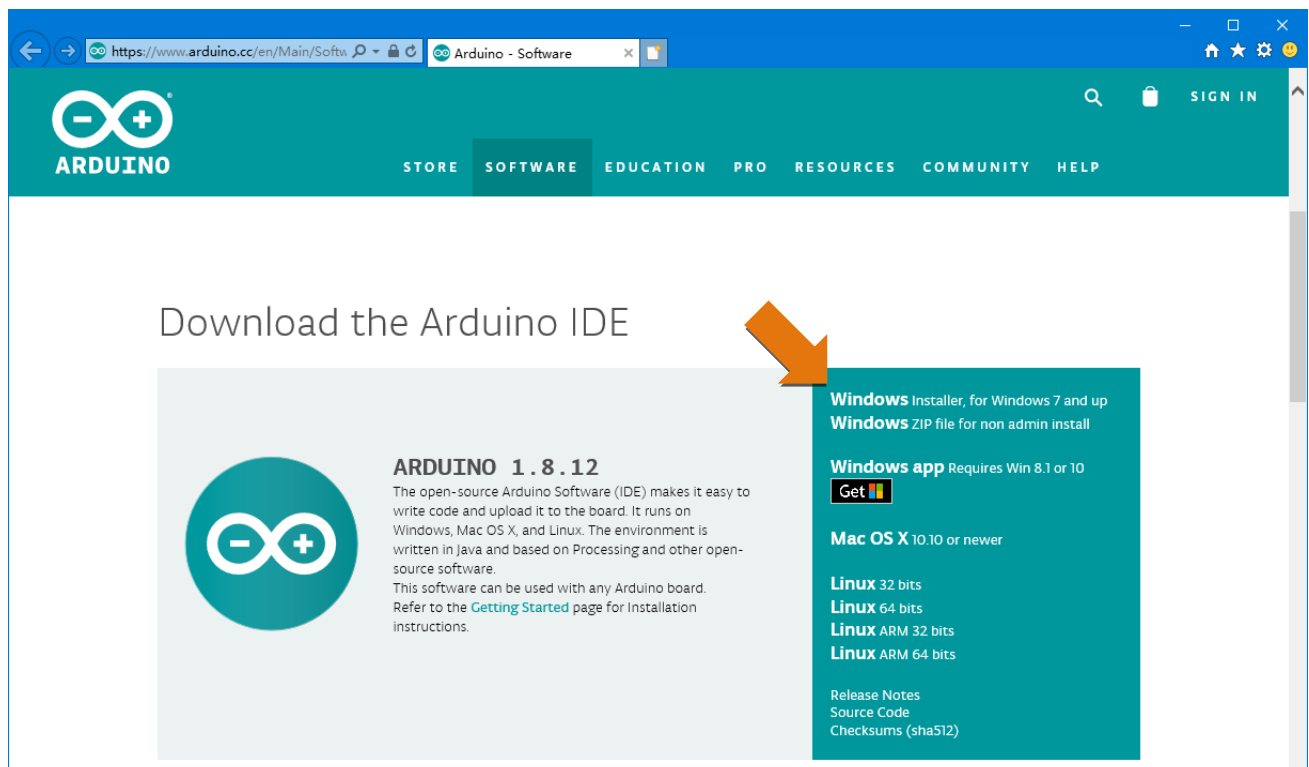
- Digital I/O ports is used to connect to other components or modules, to receive an input signal, or to send a control signal. Usually, we name it by adding a "D" in front of the number, such as D13 (pin 13).
- USB interface is used to provide power, upload code or communicate with PC.
- LED L is connected to digital I/O port 13 (pin 13).
- LED TX, RX is used to indicate the state of the serial communication.
- DC interface is connected DC power to provide power for the board.
- Power ports can provide power for electronic components and modules.
- Analog I/O ports can be used to measure analog signals.
- LED ON is used to indicate the power state.

Programming Software

We use Arduino® IDE to write and upload code for the control board, which is a free and open source. (Arduino® is a trademark of Arduino LLC.)

Arduino IDE uses C/C++ programming language. Don't worry if you have never used it, because this document contains programming knowledge and detailed explanation of the code.

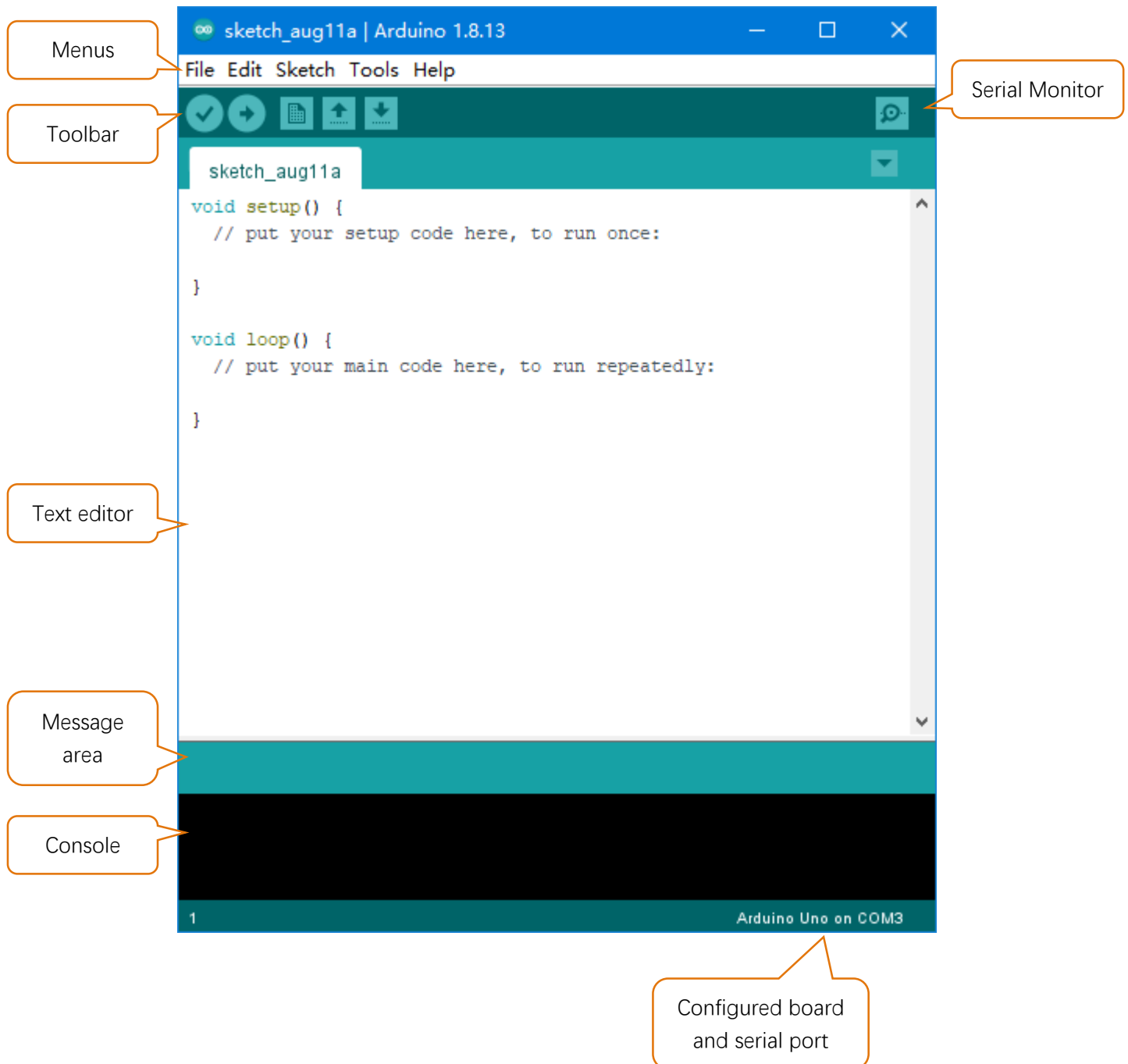
First, install Arduino IDE. Visit <https://www.arduino.cc/en/Main/Software>. Select and download a corresponding installer according to your operating system. If you are a windows user, please select the "Windows Installer".



After the downloading completes, run the installer. For Windows users, there may pop up an installation dialog box of driver during the installation process. When it is popped up, please allow the installation. After installation is completed, an shortcut will be generated in the desktop.



Run it. The interface of the software is as follows:



Programs written with Arduino IDE are called **sketches**. These sketches are written in a text editor and are saved with the file extension **.ino**. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino 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

Checks your code for errors compiling it.



Upload

Compiles your code and uploads it to the configured board.



New

Creates a new sketch.



Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.



Save

Saves your sketch.



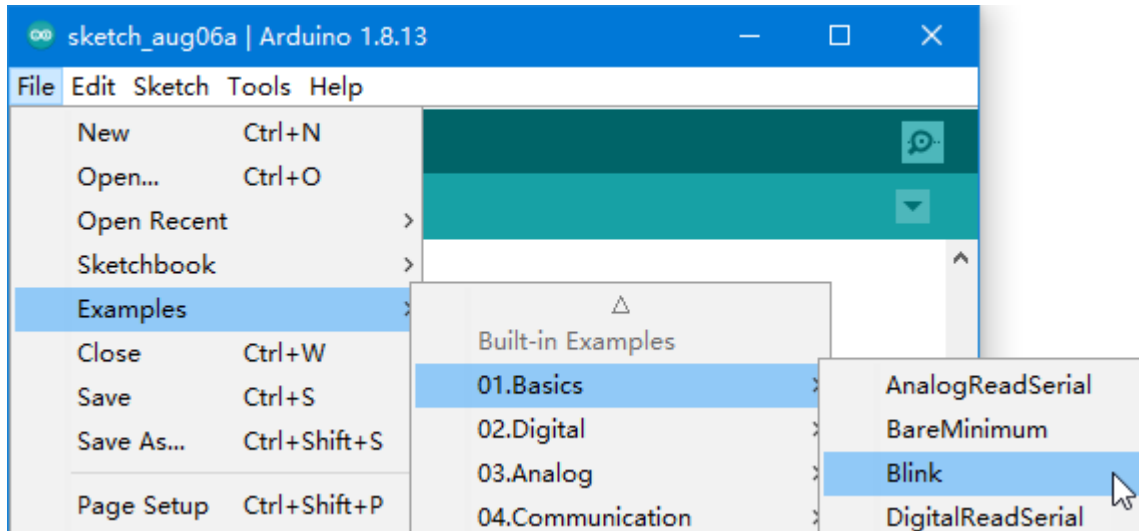
Serial Monitor

Opens the serial monitor.

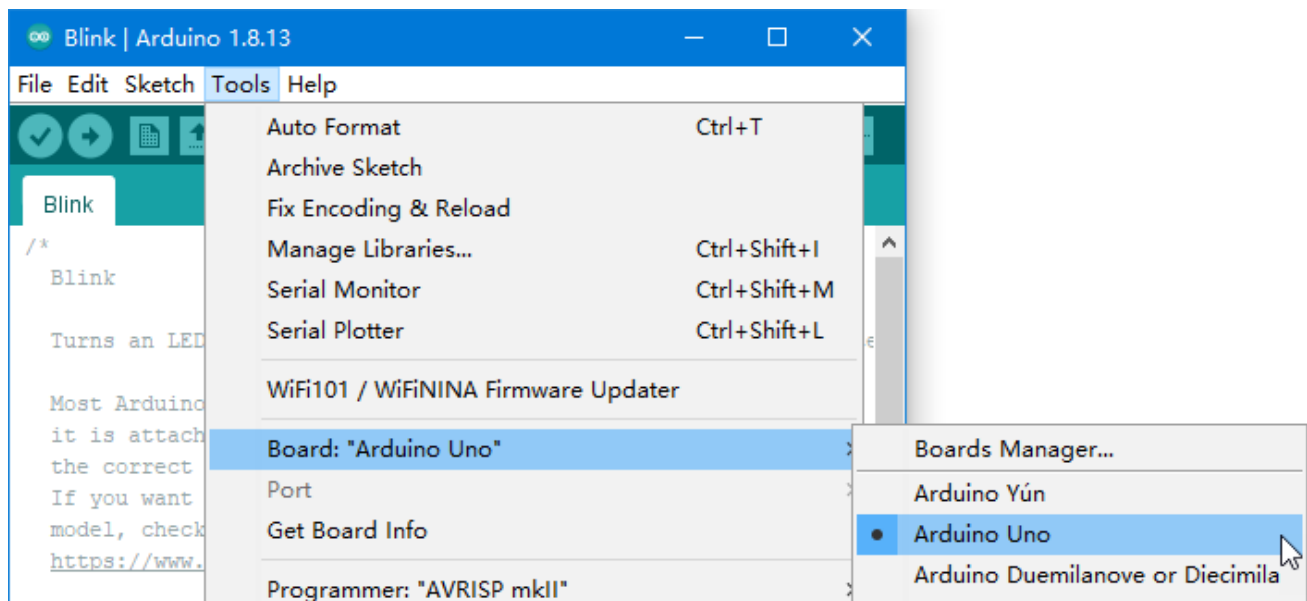
Additional commands are found within five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

First Use

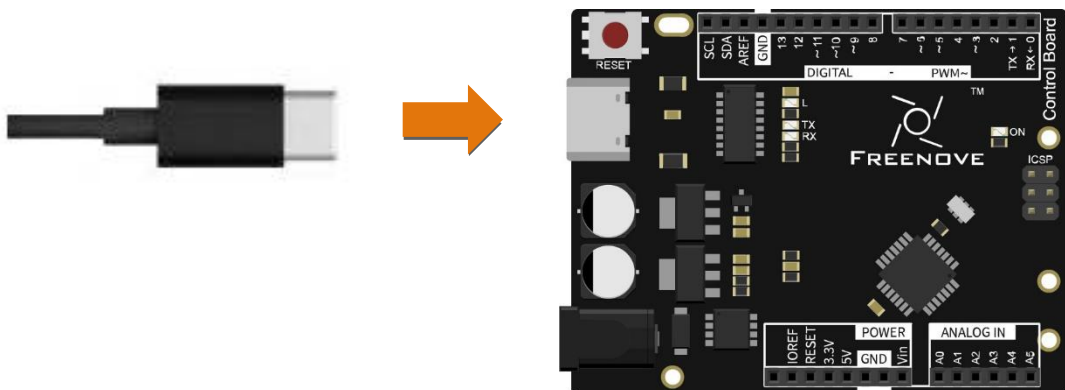
Open the example sketch "Blink".



Select board "Arduino Uno". (Freenove control board is compatible with this board.)



Connect control board to your computer with USB cable.



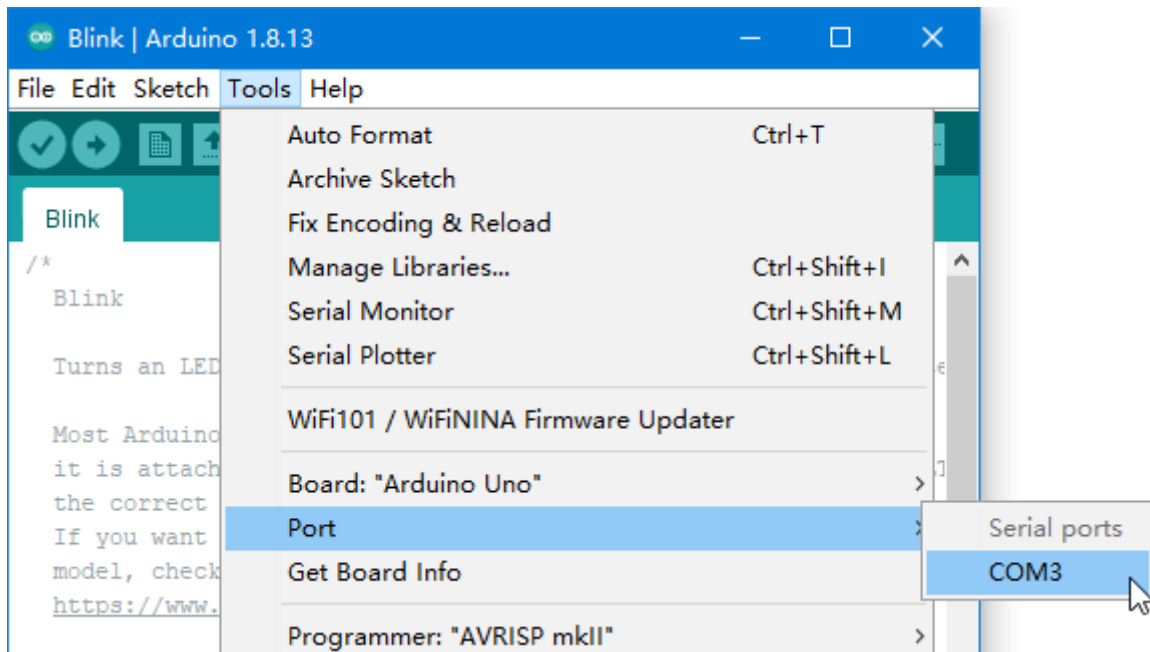
Select the port.

Note: Your port may be different from the following figure.

On Windows: It may be COM4, COM5 (Arduino Uno) or something like that.

On Mac: It may be /dev/cu.usbserial-710, /dev/cu.usbmodem7101 (Arduino Uno) or something like that.

On Linux: It may be /dev/ttyUSB0, /dev/ttyACM0 or something like that.



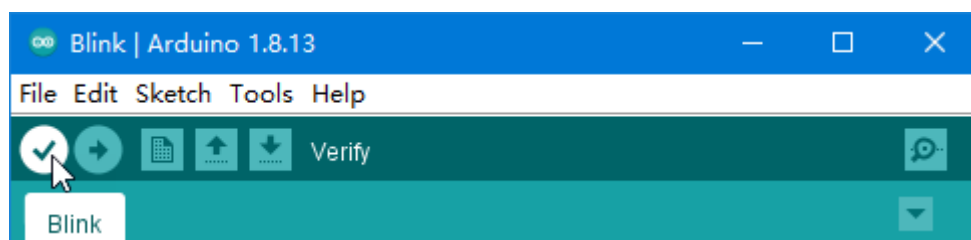
Note: If there is more than one port and you cannot decide which one to choose, disconnect the USB cable and check the port. Then connect the USB cable and check the port again. The new one is the correct port.

If there is no COM port for control board, you may need to install a driver to your computer.

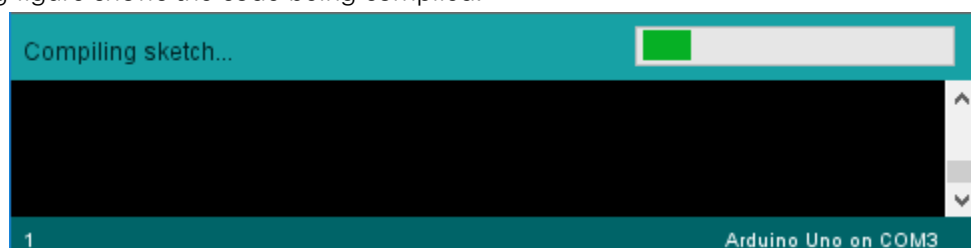
- For blue board, reinstall the latest version of Arduino IDE. During installation, agree to install the driver.
- For black board, see "InstallDriver.pdf" in "Drivers" folder (in the folder contains this Tutorial.pdf).

Having problems? Contact us for help! Send mail to: support@freenove.com

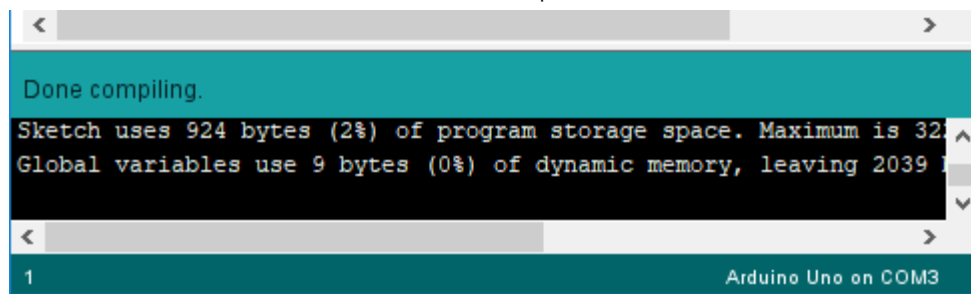
Click "Verify" button.



The following figure shows the code being compiled.



Wait a moment for the compiling to be completed. Figure below shows the code size and percentage of space occupation. If there is an error in the code, the compilation will fail and the details are shown here.



Click "Upload" button.

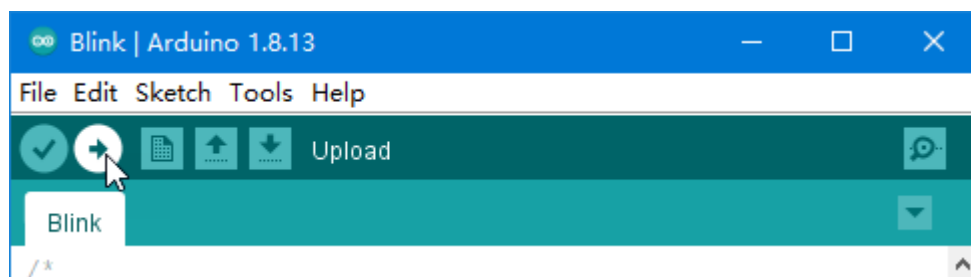
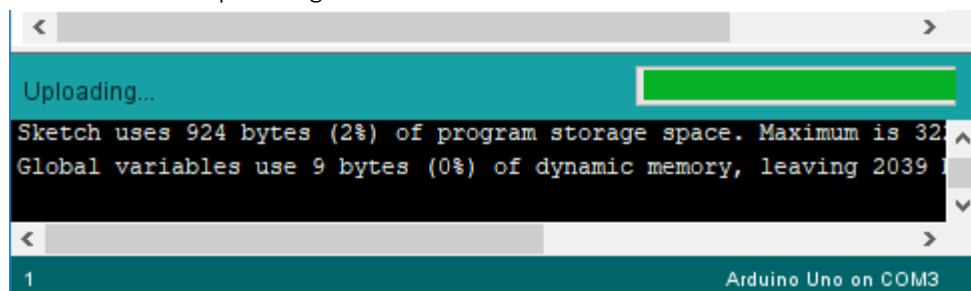
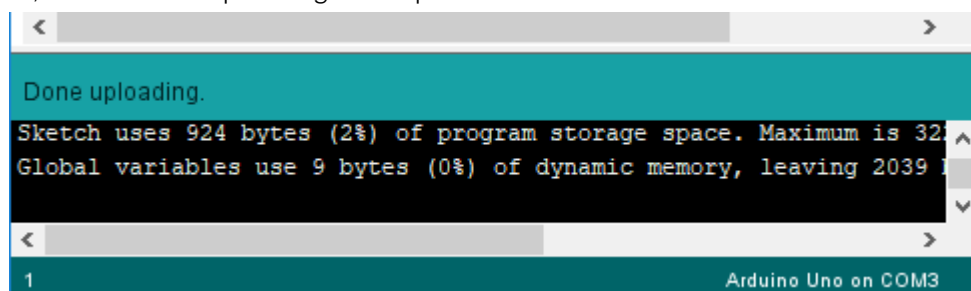


Figure below shows code are uploading.

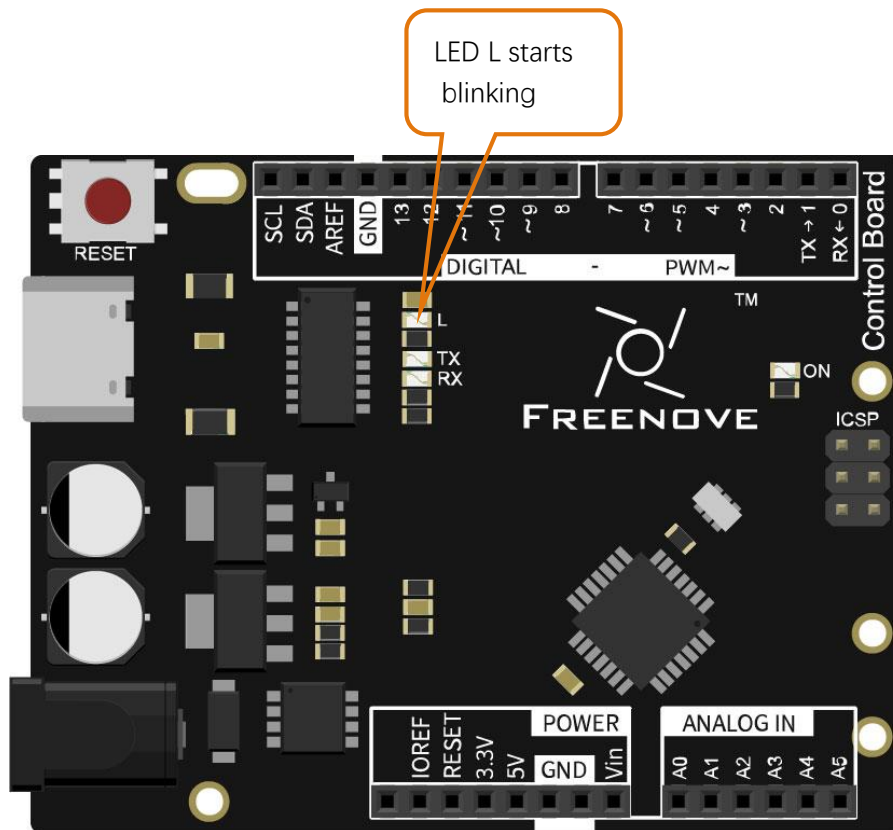


Wait a moment, and then the uploading is completed.



Having problems? Contact us for help! Send mail to: support@freenove.com

After that, we will see the LED marked with "L" on the control board start blinking. It indicates that the code is running now!



So far, we have completed the first use. I believe you have felt the joy of it. Next, we will carry out a series of projects, from easy to difficult, taking you to learn programming and the building of electronic circuit.

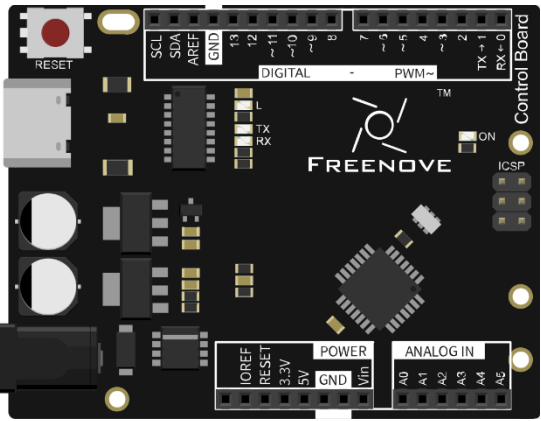
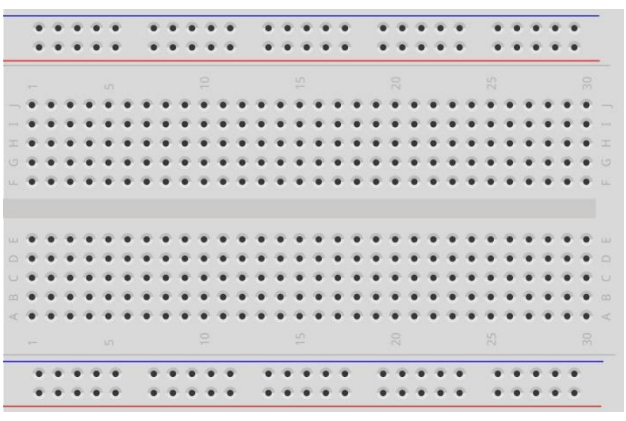

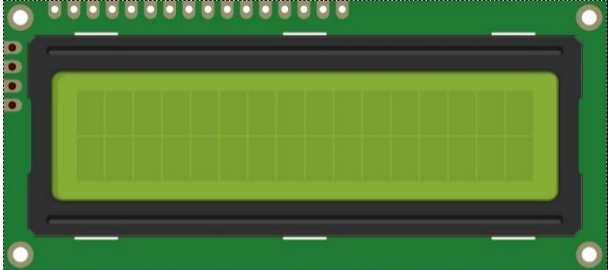

Chapter 1 I2C LCD1602

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

Project 16.1 Display the String on I2C LCD1602

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

Component List

<p>Control board x1</p>  A black PCB control board with various components including a USB port, reset button, LEDs, and pin headers for digital, power, and analog connections. The Freenove logo is visible in the center.	<p>Breadboard x1</p>  A standard white breadboard with a grid of holes and colored power rails (red and blue) along the top and bottom edges.
<p>USB cable x1</p>  A black USB cable with a standard USB-A connector on one end and a micro-USB connector on the other.	<p>I2C LCD1602 Module x1</p>  A green PCB module featuring a 16x2 character LCD display with a black frame and a single-pin I2C connector on the left side.
<p>Jumper M/F x4</p>  A single green jumper wire with a black plastic housing and metal pins at both ends.	

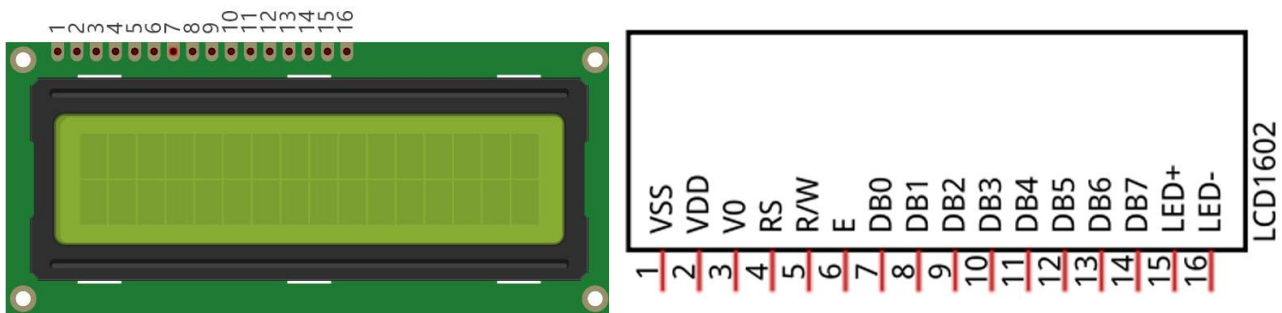
Component Knowledge

I2C communication

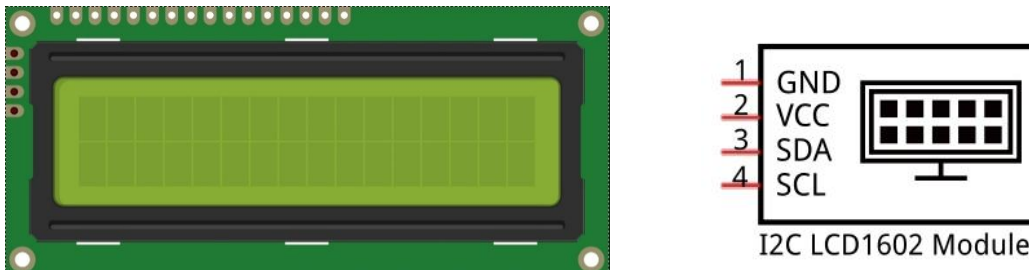
I2C (Inter-Integrated Circuit) is a two-wire serial communication mode, which can be used for the connection of micro controllers and their peripheral equipment. Devices using I2C communication must be connected to the serial data (SDA) line, and serial clock (SCL) line (called I2C bus). Each device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

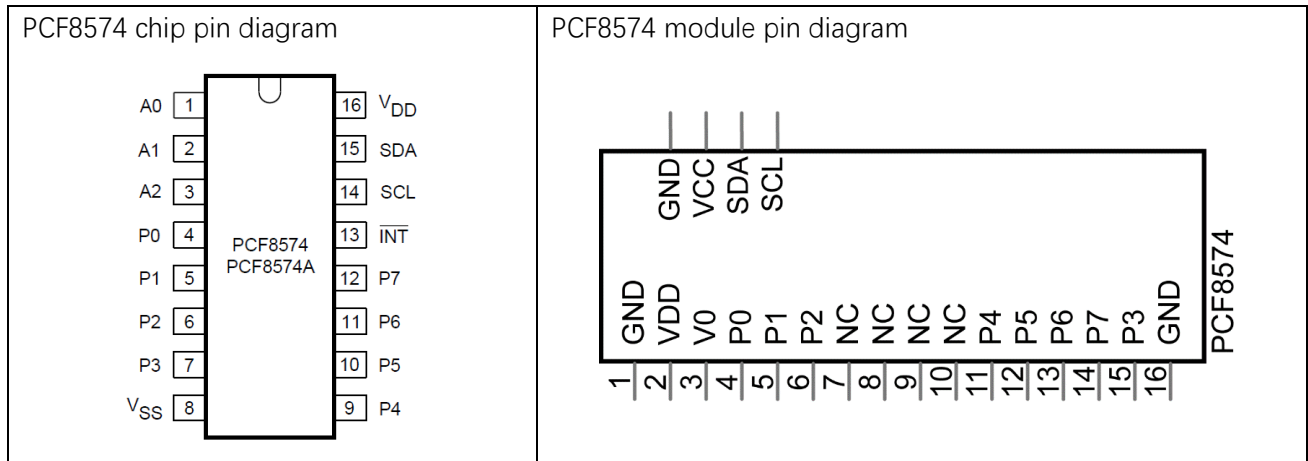


I2C LCD1602 Display Screen integrates an I2C interface, which connects the serial-input & parallel-output module to the LCD1602 Display Screen. This allows us to use only 4 lines to operate the LCD1602.

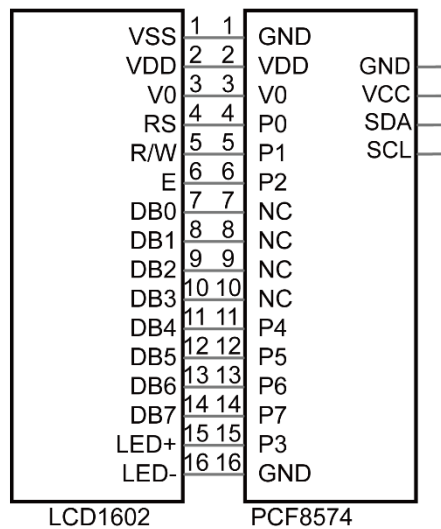


The serial-to-parallel IC chip used in this module is PCF8574T (PCF8574AT), and its default I2C address is 0x27(0x3F).

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:

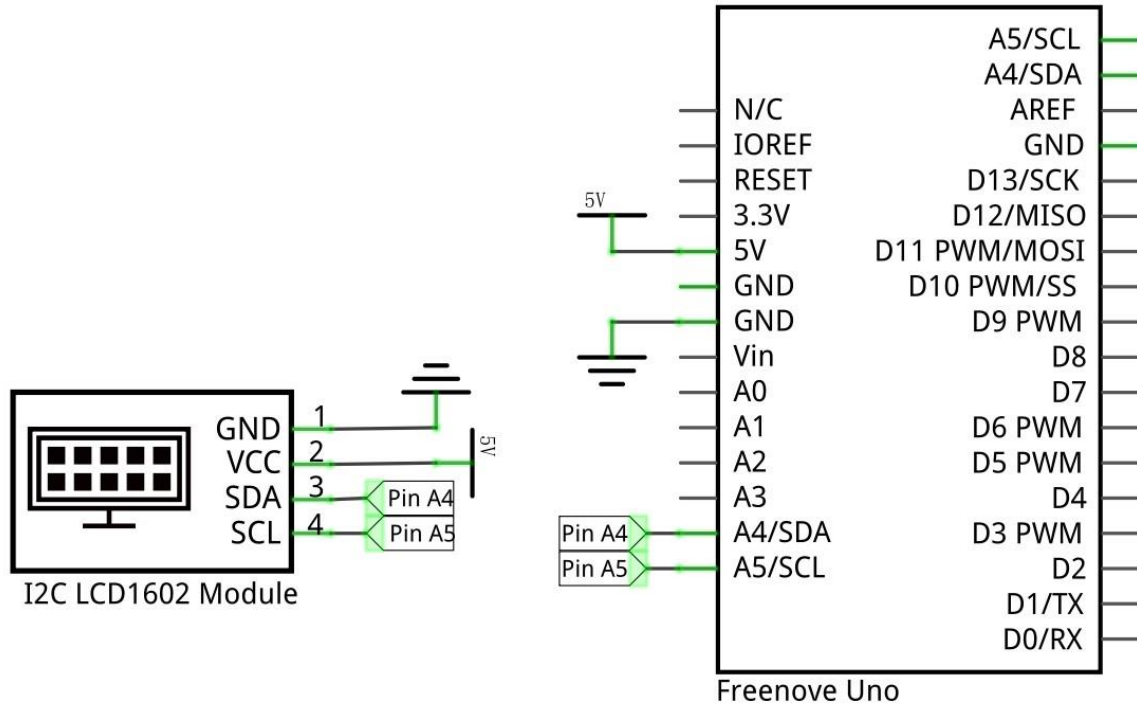


So we only need 4 pins to control the 16 pins of the LCD1602 Display Screen through the I2C interface. In this project, we will use the I2C LCD1602 to display some static characters and dynamic variables.

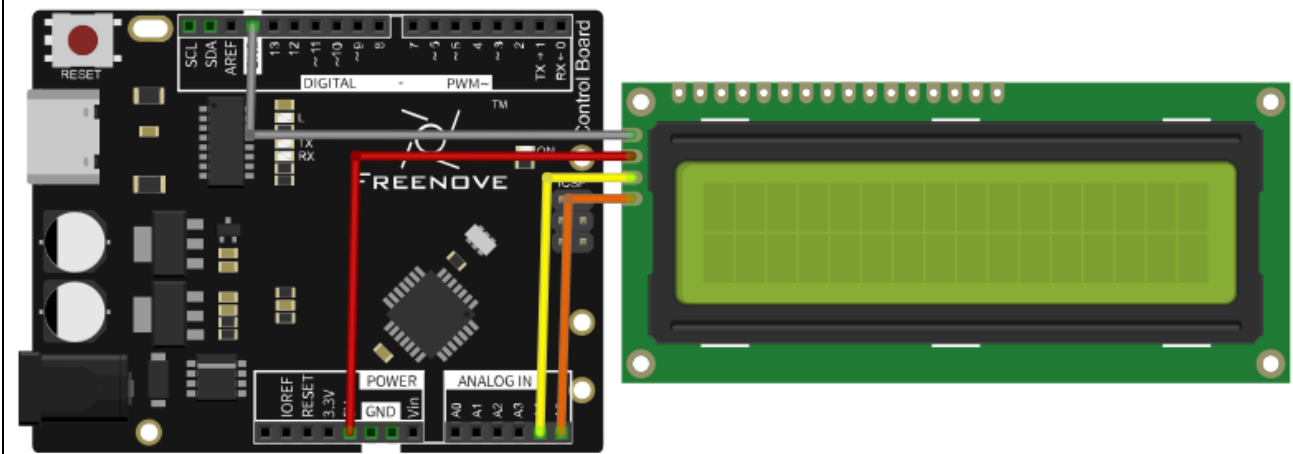
Circuit

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

Schematic diagram



Circuit connection

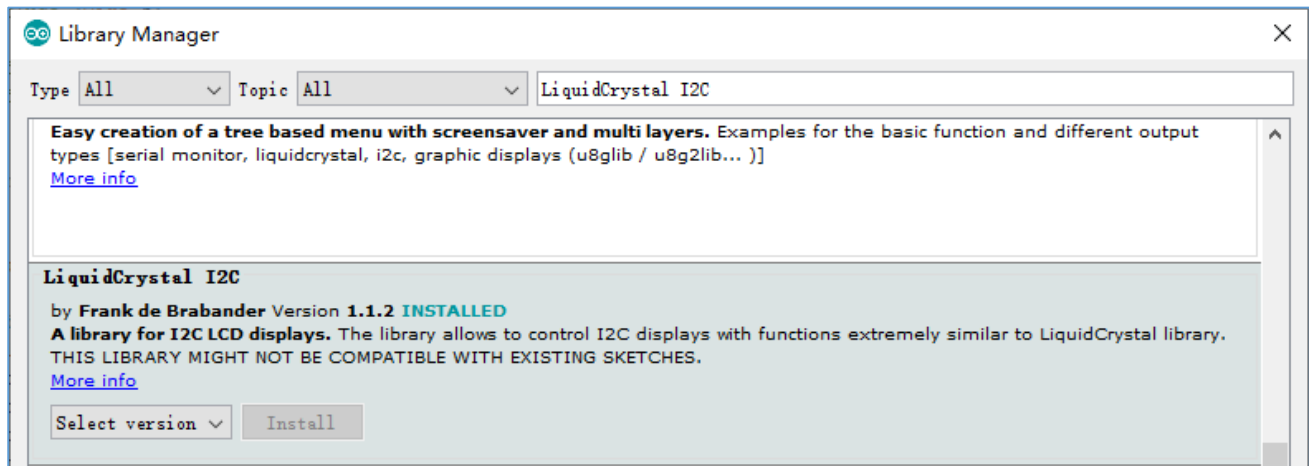


Sketch

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

How to install the library

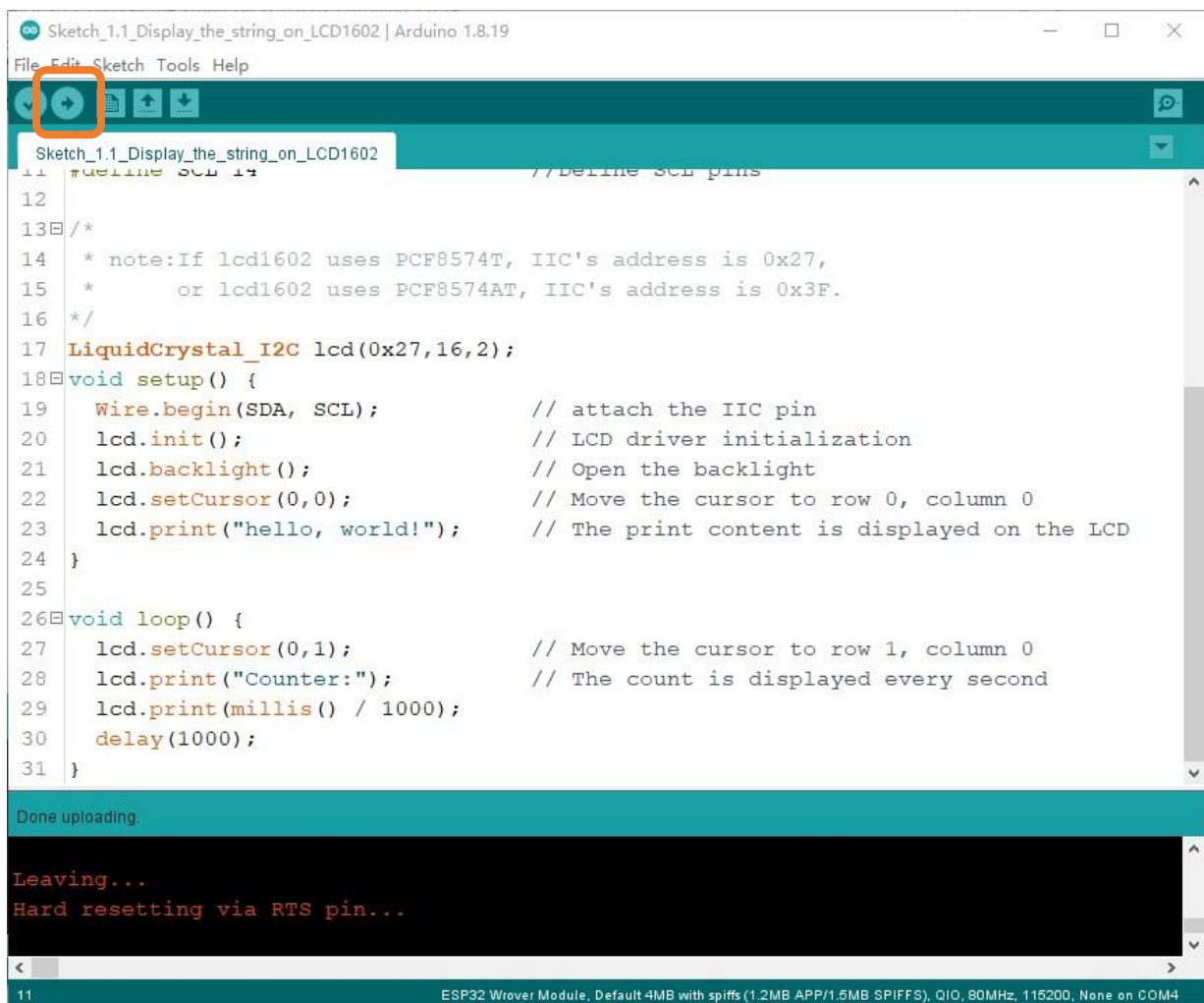
We use the third party library **LiquidCrystal I2C**. If you haven't installed it yet, please do so before learning. The steps to add third-party Libraries are as follows: open arduino->Sketch->Include library-> Manage libraries. Enter " LiquidCrystal I2C" in the search bar and select " LiquidCrystal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



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.



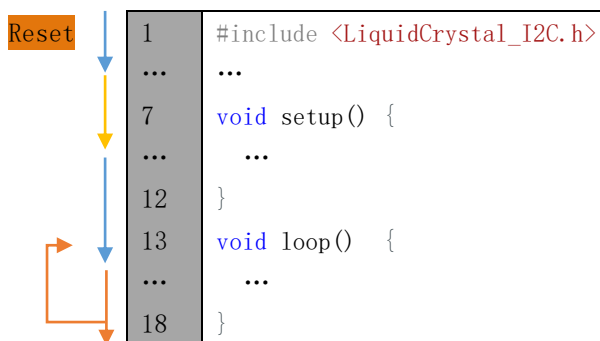
Now let's start to write code to use LCD1602 to display static characters and dynamic variables.

```
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      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 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.

```
8 lcd.init(); // LCD driver initialization
```

```
9 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:

```
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 }
```

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

```
14 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.

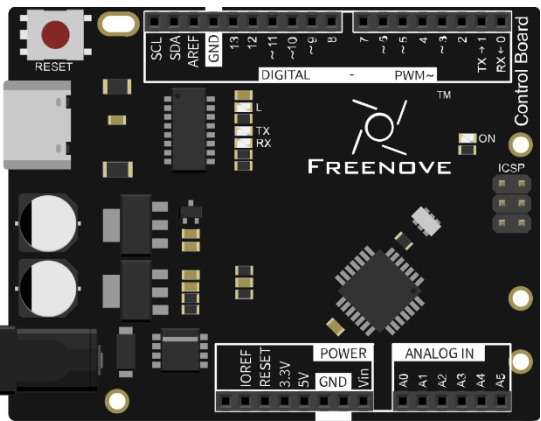
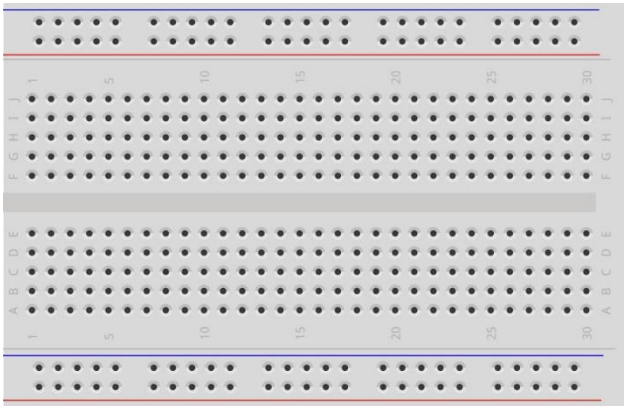

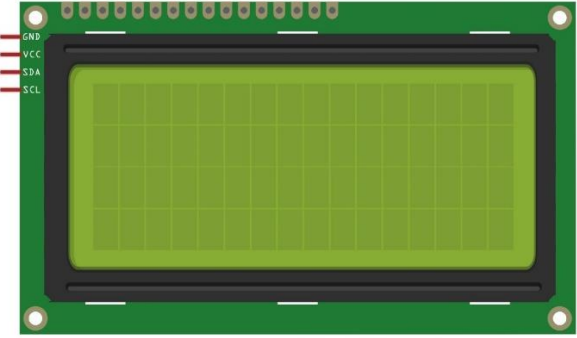

Chapter 2 I2C LCD2004

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 List

<p>Control board x1</p>  <p>The image shows a black PCB control board with various components. It features a red reset button, a USB Type-C port, two circular potentiometers, and several pin headers labeled SCL, SDA, GND, DIGITAL, TX, RX, and ANALOG IN. The Freenove logo is prominently displayed in the center.</p>	<p>Breadboard x1</p>  <p>The image shows a standard white breadboard with a grid of holes. It has power rails on the top and bottom edges, labeled with letters A through J. The central area is divided into two main sections by a vertical line.</p>
<p>USB cable x1</p>  <p>The image shows two black USB cables. One has a standard USB-A connector, and the other has a USB-C connector. Both cables are shown from a top-down perspective.</p>	<p>I2C LCD2004 Module x1</p>  <p>The image shows a green PCB module with a 2004 LCD display. The display is green and has a grid pattern. The module has a header on the left side with pins labeled GND, VCC, SDA, and SCL.</p>
<p>Jumper M/F x4</p>  <p>The image shows a single green jumper wire with a male header on one end and a female header on the other.</p>	

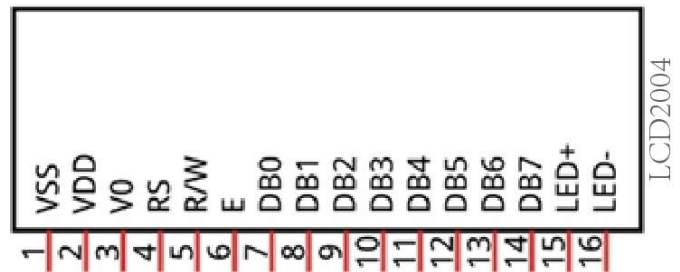
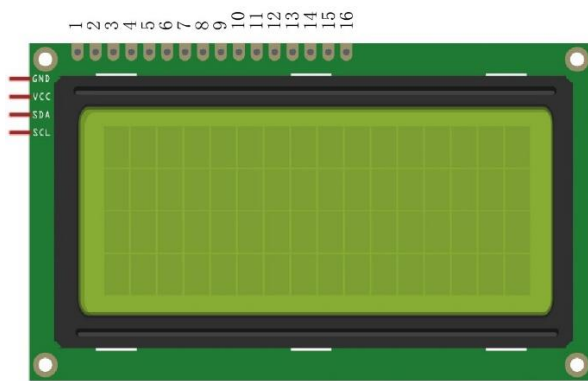
Component Knowledge

I2C communication

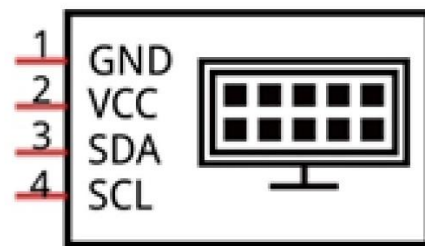
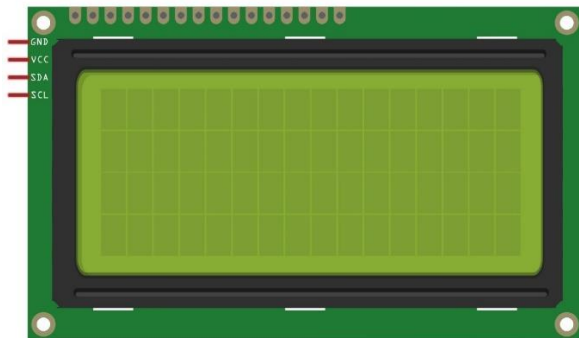
I2C (Inter-Integrated Circuit) is a two-wire serial communication mode, which can be used for the connection of micro controllers and their peripheral equipment. Devices using I2C communication must be connected to the serial data (SDA) line, and serial clock (SCL) line (called I2C bus). Each device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

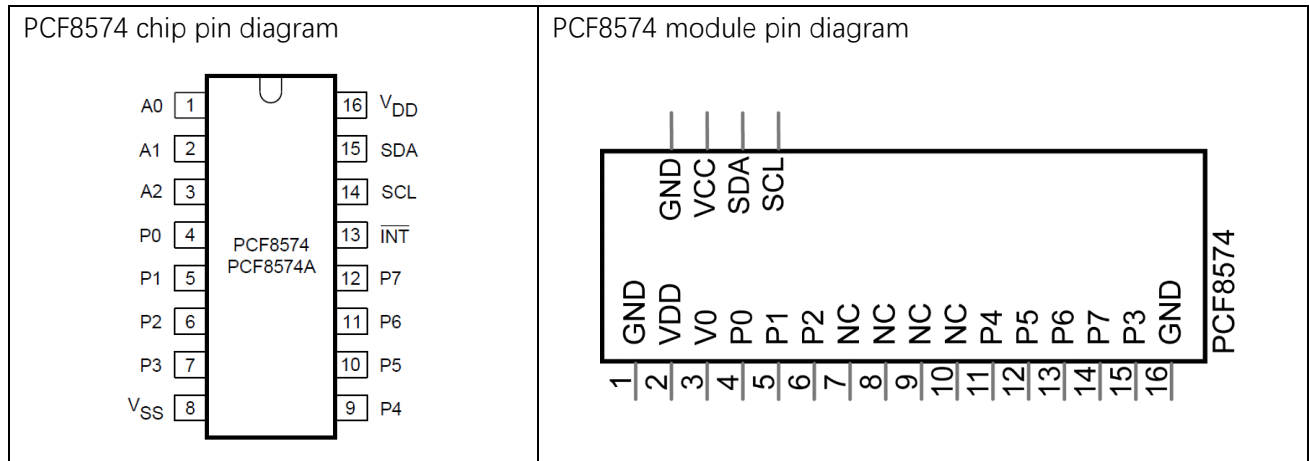


I2C LCD2004 display screen integrates a I2C interface, which connects the serial-input & parallel-output module to the LCD2004 display screen. This allows us to only use 4 lines to the operate the LCD2004.

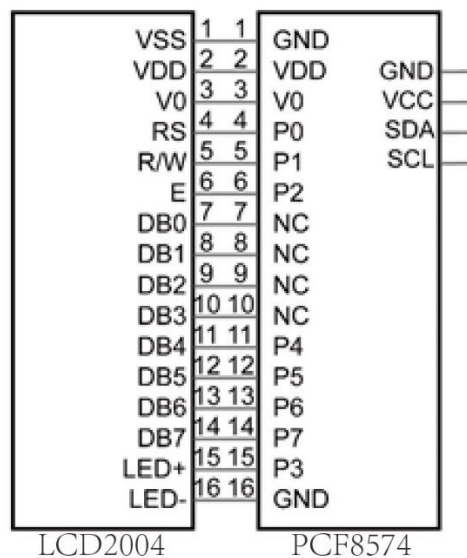


The serial-to-parallel IC chip used in this module is PCF8574T (PCF8574AT), and its default I2C address is 0x27(0x3F).

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:

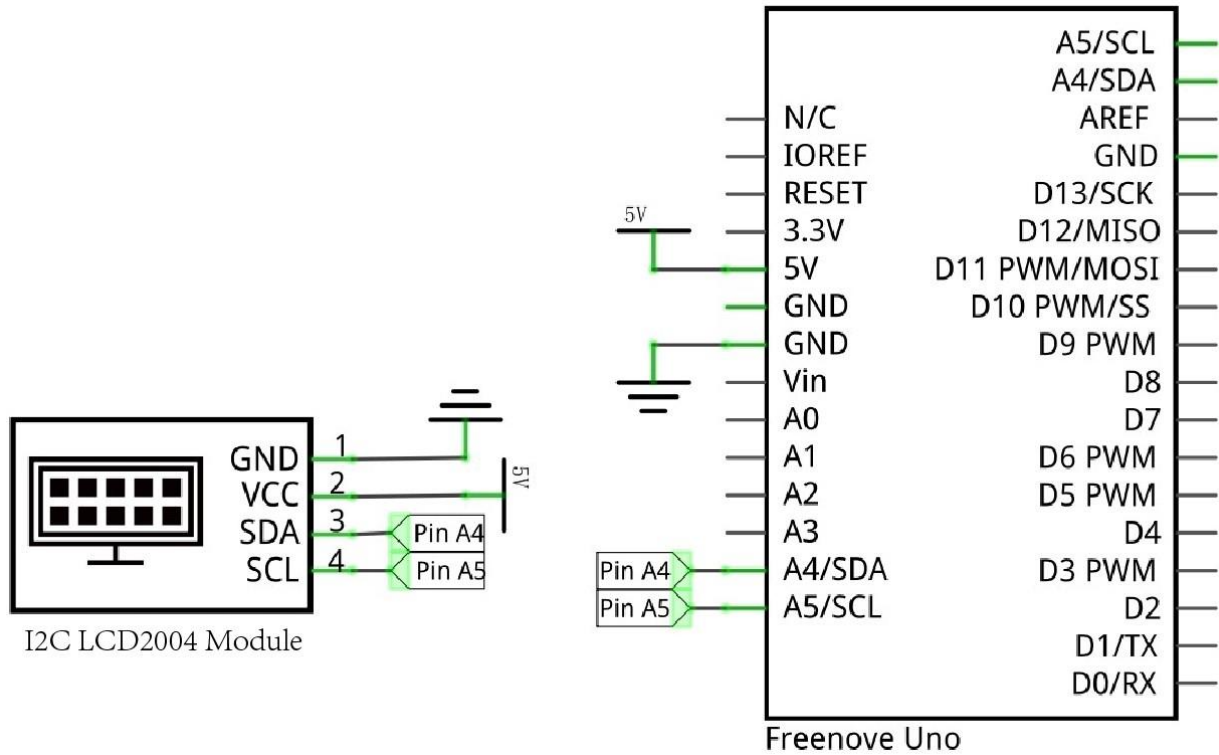


So we only need 4 pins to control the 16 pins of the LCD2004 Display Screen through the I2C interface. In this project, we will use the I2C LCD2004 to display some static characters and dynamic variables.

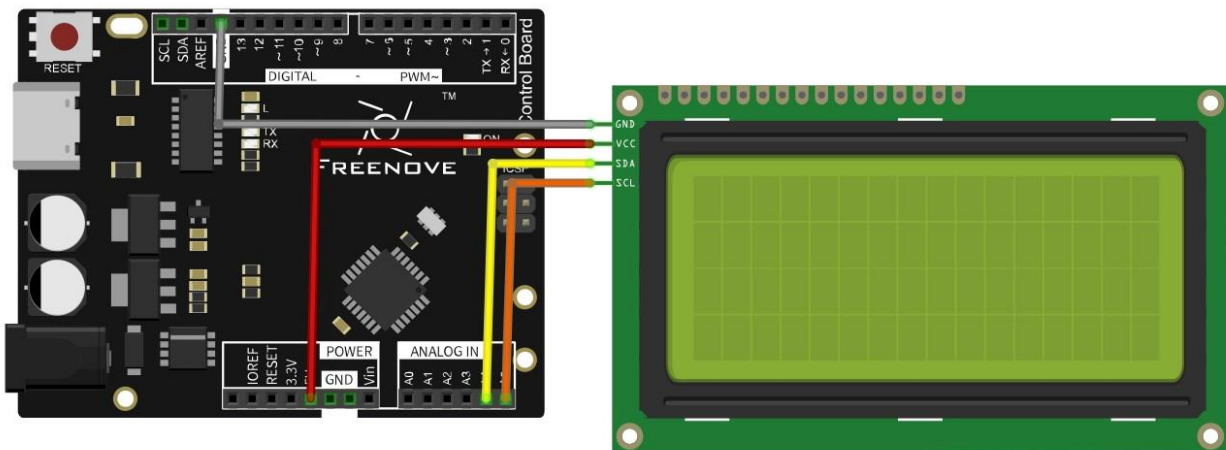
Circuit

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

Schematic diagram



Circuit connection

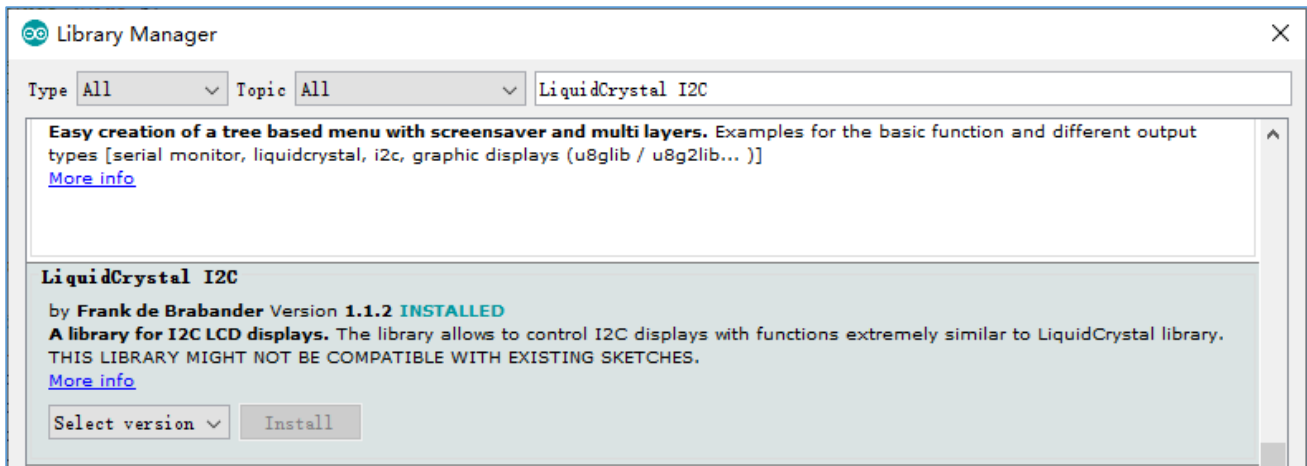


Sketch

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

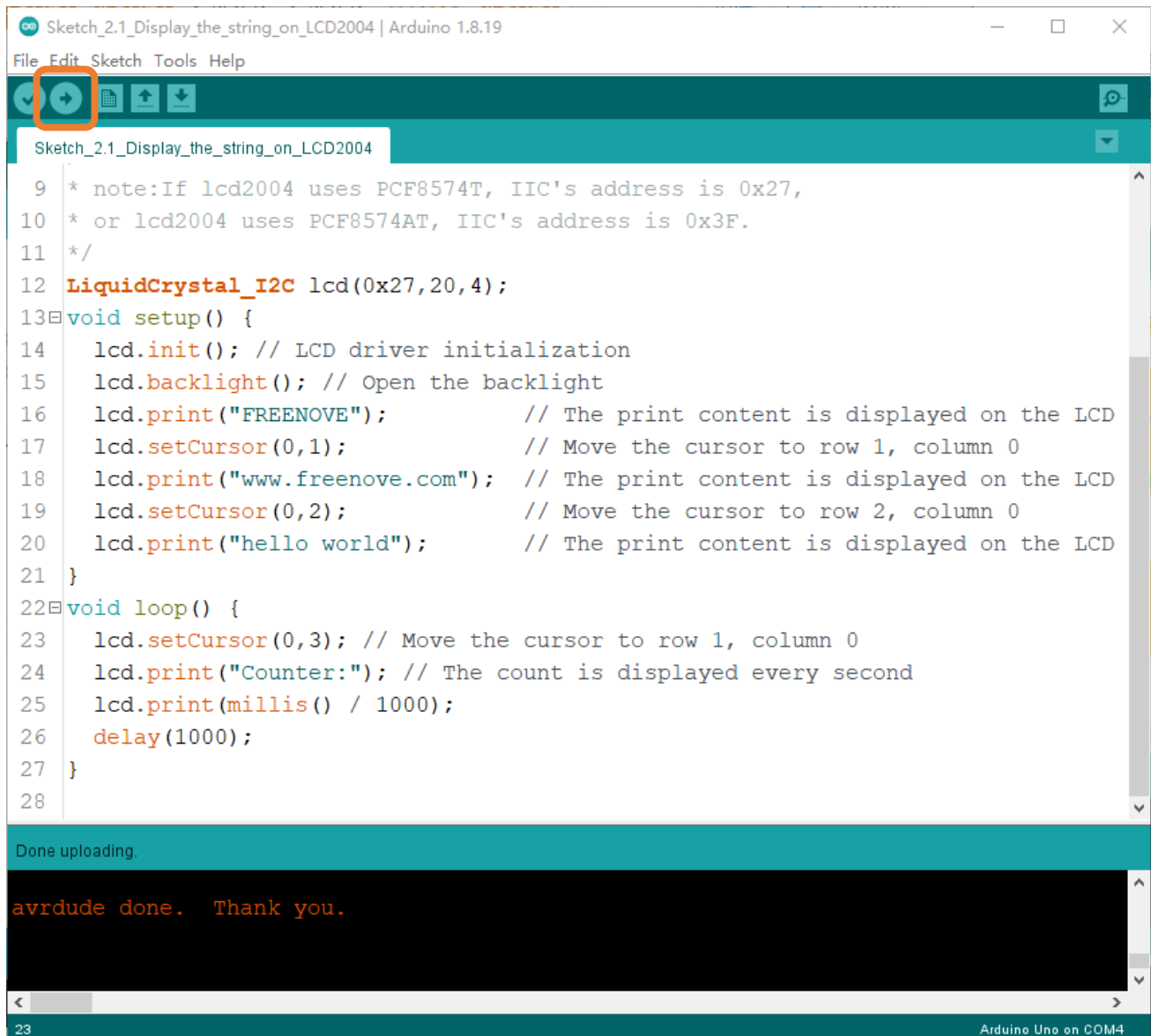
We use the third party library **LiquidCrystal I2C**. If you haven't installed it yet, please do so before learning. The steps to add third-party Libraries are as follows: open arduino->Sketch->Include library-> Manage libraries. Enter " LiquidCrystal I2C" in the search bar and select " LiquidCrystal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

What's next?

Thanks for your reading. This tutorial is all over here. If you find any mistakes, omissions or you have other ideas and questions about contents of this tutorial or the kit and etc., please feel free to contact us:

support@freenove.com

We will check and correct it as soon as possible.

If you want to learn more about Arduino, Raspberry Pi, smart cars, robots and other interesting products in science and technology, please continue to focus on our website. We will continue to launch cost-effective, innovative and exciting products.

<http://www.freenove.com/>

What's next?(others)

Thanks for your reading. This tutorial is all over here. If you find any mistakes, omissions or you have other ideas and questions about contents of this tutorial or the kit and etc., please feel free to contact us:

support@freenove.com

We will check and correct it as soon as possible.

If you want to learn more about Arduino, Raspberry Pi, smart cars, robots and other interesting products in science and technology, please continue to focus on our website. We will continue to launch cost-effective, innovative and exciting products.

<http://www.freenove.com/>

End of the Tutorial

Thank you again for choosing Freenove products.