

Arduino lesson - I2C LCD1602 Display

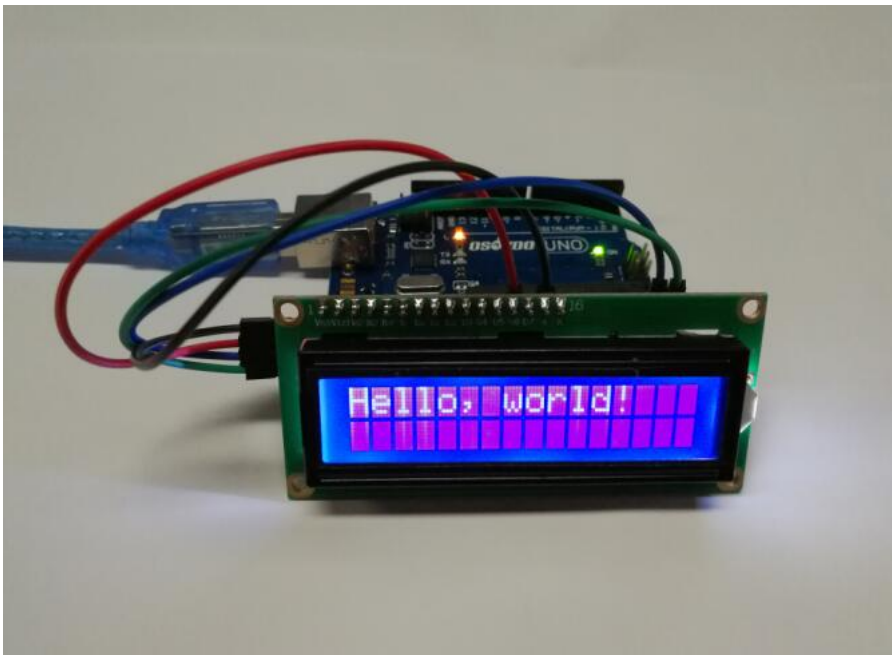
Post Time: 2017-07-09 19:07:29 Category: [Arduino Advanced Kit](#) [Arduino Basic Kit](#) [Arduino Ultimate Kit](#)

CONTENT

1. Introduction
2. Preparations
 - Hardware
 - Software
3. About the I2C LCD 1602 Display
4. Connection
5. Find the I2C address
6. Upload Sketch
 - Install Library
 - Library Support Functions
 - Code Program
 - Compile and upload
7. Program Running Result
8. The Expansion Example

Introduction

During the building of your projects for Arduino, you'll often need to read the output data directly from a LCD display. In this lesson we will show how to mount a LCD display on your Arduino using the I2C communication. Finally you will see how to program a simple example showing how to display text on the display.



Preparations

HARDWARE

- Osoyoo UNO Board (Fully compatible with Arduino UNO rev.3) x 1
- I2C LCD 1602 Display x 1
- F/M jumpers
- USB Cable x 1
- PC x 1

SOFTWARE

- Arduino IDE (version 1.6.4+)
- [LiquidCrystal_I2C.h](#)

About I2C LCD 1602 Display

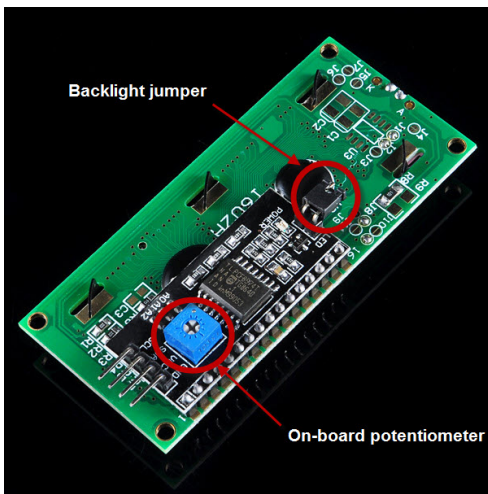
The integration of an LCD display greatly facilitates the interactivity of the project you are developing, allowing the user to directly read some output parameters. These values can be either a simple text or numerical values read by the sensors, such as temperature or pressure, or even the number of cycles that the Arduino is performing.



However, these displays have a small problem. When they are connected to a microcontroller (such as Arduino for example), displays require virtually many connection PINs occupying practically almost all available IO and leaving the multiprocessor few outputs for any other devices and sensors. This problem has been solved thanks to the communication on the I2C bus.

The LCD1602 display has an integrated microchip that manages this type of communication, and then all of the input and output information are limited to only two PINs (excluding power supply). I2C is a type of serial bus developed by Philips, which uses bidirectional lines, called SDA (Serial Data Line) and SCL (Serial Clock Line). Both must be connected via pulled-up resistors. Usage voltages are standard as 5V and 3.3V.

The blue potentiometer on the I2C LCD1602 (see the figure below) is used to adjust the backlight for better display. And there is a jumper on the board, if you take away this jumper, the backlight will always be off.



For the usage of other functions, please refer to the “LiquidCrystal_I2C.h” in the sourcecode of **I2C LCD library**.

Connection

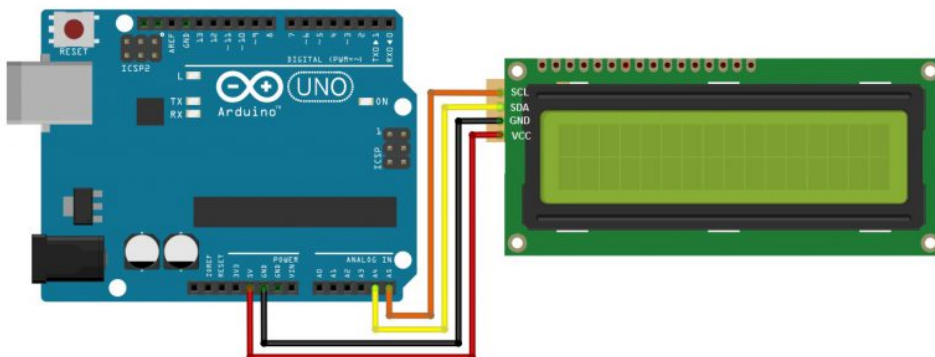
Before you write the code you have to build the circuit. To do this, connect the pins as follows:

Osoyoo UNO	LCD1602
GND	GND
5V	VCC
A4	SDA
A5	SCL

Note:

- For Mega2560: the I2C connections are on SDA=20 and SCL=21. So go ahead and wire these up, along with the two pc leads to the 5V and GND terminals.
- *For Arduino Leonardo: connect SDA to digital pin 2 and SCL to digital pin 3 on your Arduino.*

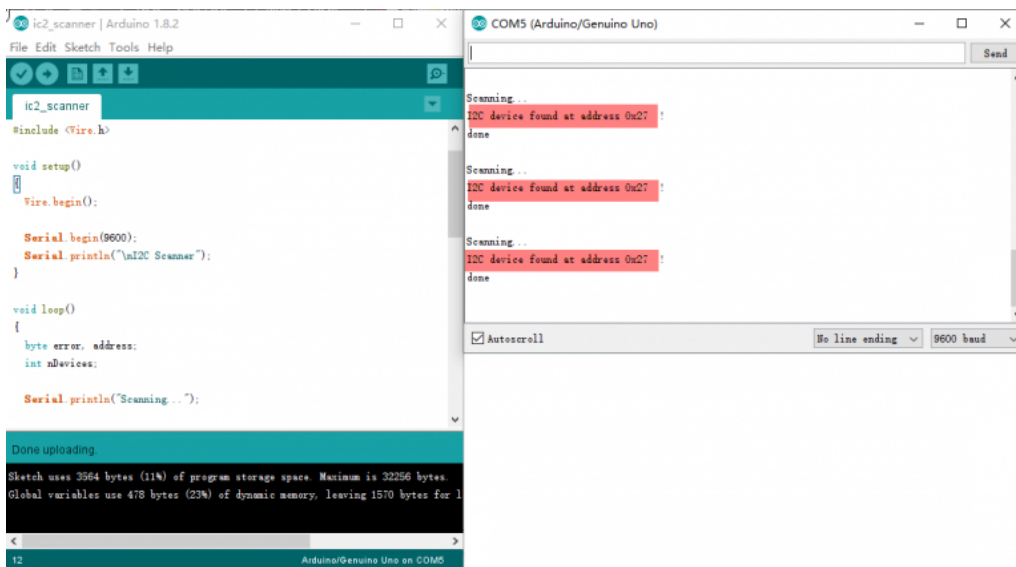
From sketches of Fritzing, you can look at the connections to be done in a simpler way:



Find the I2C address

Each device has an I2C address that it uses to accept commands or send messages. For Uno board, this address usually is But sometimes the address might be changed 0x37,0x24, So let's go and look for the one on your device.

Download [ic2_scanner sketch zip file](#) , then unzip and load it into Arduino IDE. By opening up the serial monitor in the upright Arduino will scan the address range looking for a reply. Most Arduino board will show 0x27, however it be other number.



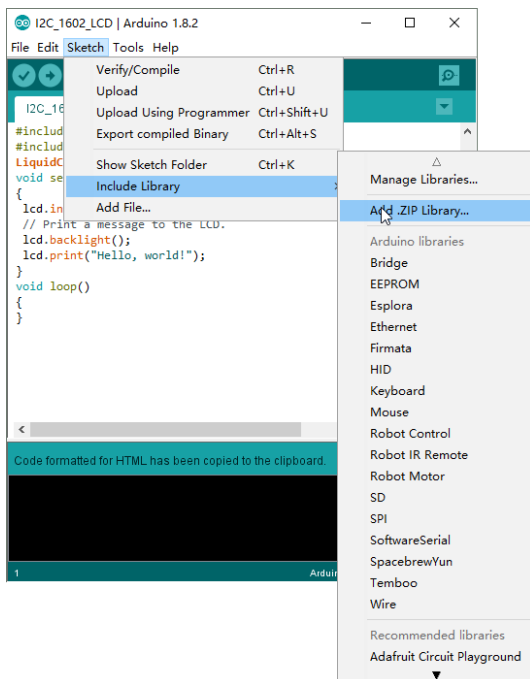
Write down the Address that you have found, you'll need it in the next step.

Upload Sketch

After above operations are completed, connect the Arduino board to your computer using the USB cable. The green power L (labelled **PWR**) should go on.

INSTALL LIBRARY

To use the I2C protocol with an LCD display and Arduino, there is a special library to be downloaded and included in the code. The name of this library is **LiquidCrystal I2C**. Download the library, a zip file (you can download it from [here](#)) and extract the code in the libraries folder of the Arduino IDE. You can do directly from the Arduino IDE, select **Sketch > include Library > Add library** from the menu.



LIBRARY SUPPORT FUNCTIONS

- **LiquidCrystal_I2C()** //set the LCD address for a 16 chars and 2 line display
- **init()** //Initialization for the LCD

- **clear()** //clear display, set cursor position to zero
- **home()** //set cursor position to zero
- **createChar()** //Fill the first 8 CGRAM locations with custom characters
- **setCursor()** //set the position of the cursor
- **cursor()** //Turns the underline cursor on
- **noCursor()** //Turns the underline cursor off
- **blink()** //Turn on the blinking cursor
- **noBlink()** //Turn off the blinking cursor
- **display()** //Turn the display on(quickly)
- **noDisplay()** //Turn the display Off(quickly)
- **backlight()** //Turn the backlight on
- **noBacklight()** //Turn the backlight off
- **scrollDisplayLeft()** //Make the display scroll left without changing the RAM
- **scrollDisplayRight()** //Make the display scroll right without changing the RAM
- **autoscroll()** //This will 'right justify' text from the cursor
- **noAutoscroll()** //This will 'left justify' text from the cursor
- **leftToRight()** //This is for text that flows Left to Right
- **rightToLeft()** //This is for text that flows Right to Left

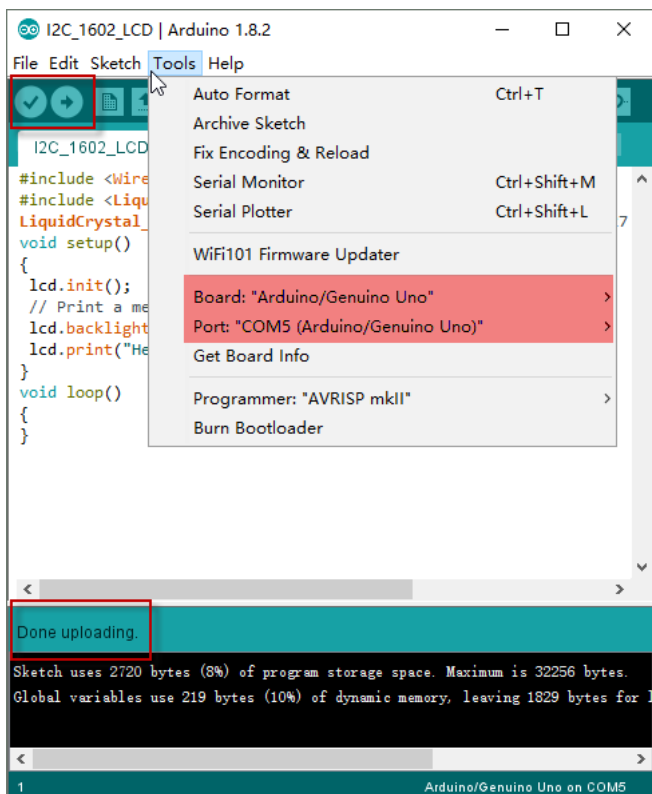
CODE PROGRAM

You can [download the sketch from this link](#) or copy below code to your Arduino IDE window:

```
#include <Wire.h> #include <LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd(0x27,16,2); // set th
```

COMPILE AND UPLOAD

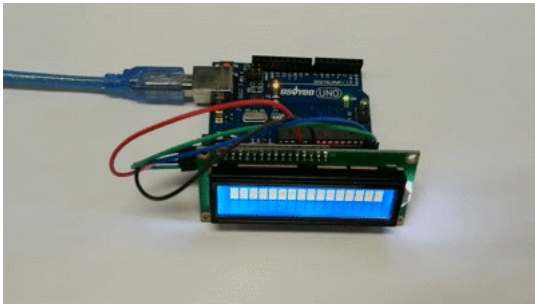
Open the Arduino IDE and select corresponding board type and port type for your Arduino board.



After compile this sketch, simply click the “Upload” button in the environment. Wait a few seconds – you should see the RX a leds on the board flashing. If the upload is successful, the message “Done uploading.” will appear in the status bar.

Running Result

A few seconds after the upload finishes, you should now see your I2C LCD1602 display the static characters: “Hello, World!”.

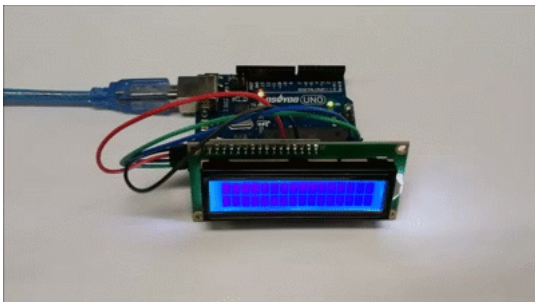


Extended experiment

In this experiment, the sketch will make a connection between Arduino and I2C LCD display and then print a text on two lines first line will display “Hello all !” and second the “Welcome to www.osoyoo.com !” message. The circuit and the board / port type settings are same as above example.

```
#include <Wire.h> #include <LiquidCrystal_I2C.h> char array1[] = " Hello all !
```

Compile and upload this sketch to the Osoyoo UNO board, you should now see your I2C LCD1602 display the flowing characters “Hello all !” and “Welcome to www.osoyoo.com”.



Download Url : osoyoo.com

3 Comments



rluxon says:

November 8, 2017 at 10:35 pm

loving these lessons. well designed, but not as informative in the sketches as I had hoped.

When I try to change the array1 to a longer message (about 40 characters), it only does the first 24 on the first line, the rest is on the second. Is this something I am doing wrong or a limitation to the I2C protocol? (ie. 16 characters on screen, and another 8 in memory?).

[Log in to Reply](#)



csylvain says:

October 27, 2018 at 12:16 pm

adjusted both code samples to include a banner visible via the serial monitor and to blink the backlight in a loop. both to make the examples a bit more dynamic for the case where the LCD darkness adjusted well and text cannot be seen — blinking and banner proves code is running even if text is invisible.

```
--cut here--  
/* _____  
 * /_\\_/)_\\/_\\/_\\/_\\/_\\/_\\ \\ \\  
 * |_| |_| |_| |_| |_| |_| |_| |_| |_|  
 * _\\/(/_\\/_\\/_\\/_\\/_\\( )_)__/_|_|_  
 * (_/)  
  
* In this sketch, we will show how to use your I2C LCD1602 display the  
static characters: "Hello, World!"  
* Tutorial URL http://osoyoo.com/2017/07/09/arduino-lesson-i2c-lcd1602-display/  
* Copyright http://www.osoyoo.com  
*/  
  
#include  
#include  
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display  
void setup()  
{  
Serial.begin(9600);  
Serial.println("I2C LCD");  
  
lcd.init(); // initialize the lcd  
// Print a message to the LCD.  
lcd.backlight();  
lcd.print("Hello, world!");  
}  
  
void loop()  
{  
delay(1000);  
lcd.noBacklight();  
delay(1000);  
lcd.backlight();  
}  
  
--cut here--  
/* _____  
 * /_\\_/)_\\/_\\/_\\/_\\/_\\/_\\ \\ \\  
 * |_| |_| |_| |_| |_| |_| |_| |_| |_|  
 * _\\/(/_\\/_\\/_\\/_\\/_\\( )_)__/_|_|_  
 * (_/)  
  
* In this experiment, the sketch will make a connection between  
Arduino and I2C LCD display and then print a text on two lines.  
The first line will display "Hello all !" and second the  
message "http://www.osoyoo.com !"  
Tutorial URL http://osoyoo.com/2017/07/09/arduino-lesson-i2c-lcd1602-display/  
Copyright (c) %YEARDIGITS% http://www.osoyoo.com  
*/  
  
#include  
#include  
  
// 000000000011111111112222222222  
// 012345678901234567890123456789  
char array1[] = " Hello all ! "; //the string to print on the LCD  
char array2[] = "Welcome to http://www.osoyoo.com"; //the string to print on the LCD  
int slen = 30; //max length of strings  
int tim = 300; //the value of delay time  
// initialize the library with the numbers of the interface pins  
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display  
  
void setup()  
{  
Serial.begin(9600);  
Serial.println("I2C Welcome");  
  
lcd.init(); //initialize the lcd  
lcd.backlight(); //open the backlight  
}  
  
void loop()  
{  
lcd.setCursor(15,0); // set the cursor to column 15, line 0  
for (int positionCounter1 = 0; positionCounter1 < slen; positionCounter1++)  
{  
lcd.scrollDisplayLeft(); //Scrolls the contents of the display one space to the left.  
lcd.print(array1[positionCounter1]); // Print a message to the LCD.  
delay(tim); //wait {tim} microseconds  
}  
  
lcd.clear(); //Clears the LCD screen and positions the cursor in the upper-left corner.  
lcd.setCursor(15,1); // set the cursor to column 15, line 1  
for (int positionCounter = 0; positionCounter < slen; positionCounter++)  
{  
lcd.scrollDisplayLeft(); //Scrolls the contents of the display one space to the left.  
lcd.print(array2[positionCounter]); // Print a message to the LCD.  
delay(tim); //wait {tim} microseconds  
}
```

[Log in to Reply.](#)



[Log in to Reply.](#)

You must be [logged in](#) to post a comment.

[Privacy Policy](#) © 2009-2017 Osoyoo.com