Makerguides.com

How to control a character I2C LCD with Arduino

Written by Benne de Bakker (https://www.makerguides.com/author/benne-de-bakker/)

This article includes everything you need to know about using a character I2C LCD (https://amzn.to/2TffbbL) with Arduino. I have included a wiring diagram and many example codes to help you get started.

The first part of this article covers the basics of displaying text and numbers. In the second half, I will go into more detail on how to display custom characters and how you can use the other functions of the LiquidCrystal_I2C library.

Once you know how to display text and numbers on the LCD, I suggest you take a look at the articles below. In these tutorials, you will learn how to measure and display sensor data on the LCD.

Recommended articles

- How to use an HC-SR04 Ultrasonic Distance Sensor with Arduino (https://www.makerguides.com/hc-sr04-arduino-tutorial/)
- How to use DHT11 and DHT22 Sensors with Arduino (https://www.makerguides.com/dht11-dht22-arduino-tutorial/)
- LM35 analog temperature sensor with Arduino tutorial (https://www.makerguides.com/lm35-arduino-tutorial/)
- TMP36 analog temperature sensor with Arduino tutorial (https://www.makerguides.com/tmp36-arduino-tutorial/)

If you have any questions, please leave a comment below.

Supplies

Hardware components

(https://amzn.to/2TffbbL)	16×2 character I2C LCD (https://s.click.aliexpress.com/e/_d82Tvi7)	× 1
(https://amzn.to/2uDtKf3)	20×4 character I2C LCD (https://s.click.aliexpress.com/e/_d6oF24L) (alternative)	× 1
	Arduino Uno Rev3 (https://amzn.to/374aIiX)	×

(https://amzn.to/374aJjX)	~~Arduino Uno Rev3 (https://amzn.to/374aJjX)~~	1
	Jumper wires (https://amzn.to/2EG9wDc) (male to female)	× 4
	USB cable type A/B (https://amzn.to/34SBuXf)	× 1

Tools

	Small screwdriver (https://amzn.to/2SkE0ms)	Amazon (https://amzn.to/2SkE0ms)

Software

Arduino IDE (https://www.arduino.cc/en/Main/Software)

I2C LCD Basics

This type of LCD is ideal for displaying text and numbers, hence the name ‘character LCD’. The I2C LCD that we are using in this tutorial comes with a small add-on circuit mounted on the back of the module. This module features a PCF8574 chip (for I2C

communication) and a potentiometer to adjust the LED backlight. The advantage of an I2C LCD is that the wiring is very simple. You only need two data pins to control the LCD.

Standard LCDs typically require around 12 connections, which can be a problem if you do not have many GPIO pins available. Luckily, you can also buy the I2C add-on circuit separately on Amazon (https://amzn.to/2ZKMSpv), so you can easily upgrade a standard LCD as well.

For a tutorial and wiring diagram for standard character LCDs, please see the following article:

- How to use a 16×2 character LCD with Arduino (https://www.makerguides.com/character-lcd-arduino-tutorial/)

If you look closely at the LCD, you can see the small rectangles that form the individual characters of the LCD. Each rectangle is made up of a grid of 5×8 pixels. Later in this tutorial, I will show you how you can control the individual pixels to display custom characters on the LCD.

Specifications

The specifications of the 16×2, 20×4, and other sized LCDs are mostly the same. They all use the same HD44780 Hitachi LCD controller (https://en.wikipedia.org/wiki/Hitachi_HD44780_LCD_controller), so you can easily swap them. You will only need to change the size specifications in your Arduino code.

The specifications of a typical 16×2 I2C display can be found in the table below.

16×2 I2C LCD Specifications

Operating voltage	5 V
Controller	Hitachi HD44780 LCD controller
Default address	0x27
Screen resolution	2-lines × 16 characters
Character resolution	5 × 8 pixels
Module dimensions	80 × 36 × 12 mm
Viewing area dimensions	64.5 × 16.4 mm
Cost	Check price (https://amzn.to/2ZIeegT)

For more information, you can check out the datasheets below. The 16×2 and 20×4 datasheets include the dimensions of the LCD and you can find more information about the Hitachi LCD driver in the HD44780 datasheet. The PCF8574 chip is used in the I2C module on the back of the LCD.

16×2 Character LCD Datasheet

20×4 Character LCD Datasheet

HD44780 Datasheet

PCF8574 Datasheet

How to connect the I2C LCD to Arduino UNO

The wiring diagram below shows you how to connect the I2C LCD to the Arduino. Wiring an I2C LCD is a lot easier than connecting a standard LCD. You only need to connect 4 pins instead of 12.

(https://www.makerguides.com/wp-content/uploads/2019/02/I2C-LCD-with-Arduino-Wiring-Diagram-Schematic-Pinout.jpg)

I2C LCD with Arduino wiring diagram

The connections are also given in the table below.

I2C LCD Connections

I2C Character LCD	Arduino
GND	GND
VCC	5 V
SDA	A4
SCL	A5

If you are not using an Arduino Uno, the SDA and SCL pins can be at a different location. Note that an Arduino Uno with the R3 layout (1.0 pinout) also has the SDA (data line) and SCL (clock line) pin headers close to the AREF pin. Check the table below for more details.

Board	SDA	SCL
Arduino Uno (https://amzn.to/2N80aGy)	A4	A5
Arduino Nano (https://amzn.to/2N7I9bg)	A4	A5
Arduino Micro (https://amzn.to/32zK3oQ)	2	3
Arduino Mega 2560 (https://amzn.to/2N6aheY)	20	21
Arduino Leonardo (https://amzn.to/2HZdDMD)	2	3
Arduino Due (https://amzn.to/2NTFtOn)	20	21

SDA and SCL pin locations on different Arduino boards.

Adjusting the contrast of the LCD

After you have wired up the LCD, you will need to adjust the contrast of the display. On the I2C module, you will find a potentiometer that you can turn with a small screwdriver.

Plug in the USB connector of the Arduino to power the LCD. You should see the backlight light up. Now rotate the potentiometer until one (16×2 LCD) or 2 rows (20×4 LCD) of rectangles appear.

You can tweak the contrast later if needed.

Once that is done, we can start programming the LCD.

Installing the LiquidCrystal_I2C Arduino library

In this tutorial, I will be using the **LiquidCrystal_I2C** library. This library has many built-in functions that make programming the LCD quite easy. The latest version of this library can be found here on GitHub (https://github.com/johnrickman/LiquidCrystal_I2C) or click the download button below.

LiquidCrystal_I2C-master.zip

Make sure that you have this exact library installed and delete any other libraries that have the same name (LiquidCrystal_I2C). Other libraries will probably work as well but might use slightly different names for the different functions.

The LiquidCrystal_I2C library works in combination with the **Wire.h** library which allows you to communicate with I2C devices. This library comes pre-installed with the Arduino IDE.

To install this library, go to Tools > Manage Libraries (Ctrl + Shift + I on Windows) in the Arduino IDE (https://www.arduino.cc/en/main/software). The Library Manager will open and update the list of installed libraries.

Now search for 'liquidcrystal_i2c' and look for the library by **Frank de Brabander**. Select the latest version and then click Install.

Installing the LiquidCrystal_I2C Arduino library

The library does include some examples that you can use, but you will have to modify them to match your hardware setup. I have included many example codes below that you can use with the wiring setup I have shown earlier.

First I will show you some basic example code and then I will explain the functions in more detail.

How to find the I2C address of my LCD?

Most I2C LCDs ship with the default address '0x27', but it can be different depending on the batch/manufacturer. If this is the case, you will need to find the actual address of the LCD before you can start using it.

On the Arduino website, you can find a simple example sketch that scans the I2C-bus for devices. If a device is found, it will display the address in the serial monitor.

You can copy the code by clicking on the button in the top right corner of the code field.

```
/*I2C_scanner
  This sketch tests standard 7-bit addresses.
  Devices with higher bit address might not be seen properly.*/

#include <Wire.h>

void setup() {
  Wire.begin();

  Serial.begin(9600);
  while (!Serial);
  Serial.println("\nI2C Scanner");
}

void loop() {
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for (address = 1; address < 127; address++ ) {
    Wire.beginTransmission(address);
    error = Wire.endTransmission();
```

```
25.
26.       if (error == 0) {
27.         Serial.print("I2C device found at address 0x");
28.         if (address < 16)
29.           Serial.print("0");
30.         Serial.print(address, HEX);
31.         Serial.println("  !");
32.
33.         nDevices++;
34.       }
35.       else if (error == 4) {
36.         Serial.print("Unknown error at address 0x");
37.         if (address < 16)
38.           Serial.print("0");
39.         Serial.println(address, HEX);
40.       }
41.     }
42.     if (nDevices == 0)
43.       Serial.println("No I2C devices found\n");
44.     else
45.       Serial.println("done\n");
46.
47.     delay(5000);
48.   }
```

If you upload this sketch to the Arduino and run it, you should see the following output in the Serial Monitor (Ctrl + Shift + M).

(https://www.makerguides.com/wp-content/uploads/2019/02/I2C-Address-finder-serial-

monitor-output.jpg)

I2C address scanner Serial Monitor output

Write down the address you find, you will need it later when programming the LCD.

Basic Arduino example code for I2C LCD

You can upload the following example code to the Arduino using the Arduino IDE.

For this tutorial, I used this 16×2 I2C character LCD display (https://amzn.to/36pTc6S), but you can use other I2C LCDs of different sizes as well.

This example sketch will display the classic 'Hello World!' on the first line of the LCD and 'LCD tutorial' on the second line. Next, I will explain how the code works.

```
1.  /* I2C LCD with Arduino example code. More info: https://www.makerguides.com */
2.
3.  // Include the libraries:
4.  // LiquidCrystal_I2C.h: https://github.com/johnrickman/LiquidCrystal_I2C
5.  #include <Wire.h> // Library for I2C communication
6.  #include <LiquidCrystal_I2C.h> // Library for LCD
7.
8.  // Wiring: SDA pin is connected to A4 and SCL pin to A5.
9.  // Connect to LCD via I2C, default address 0x27 (A0-A2 not jumpered)
10. LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2); // Change to (0x27,20,4) for 20x4
    LCD.
```

```
11.
12.  void setup() {
13.    // Initiate the LCD:
14.    lcd.init();
15.    lcd.backlight();
16.  }
17.
18.  void loop() {
19.    // Print 'Hello World!' on the first line of the LCD:
20.    lcd.setCursor(2, 0); // Set the cursor on the third column and first row.
21.    lcd.print("Hello World!"); // Print the string "Hello World!"
22.    lcd.setCursor(2, 1); //Set the cursor on the third column and the second row (counting
    starts at 0!).
23.    lcd.print("LCD tutorial");
24.  }
```

You should see the following output on the LCD:

How the code works

First, the required libraries are included. As mentioned earlier we need both the wire.h* and the LiquidCrystal_I2C library. In the rest of this tutorial, I will cover more of the built-in functions of this library.

**When using the latest version of the LiquidCrystal_I2C library it is no longer needed to include the wire.h library in your sketch. The other library imports wire.h automatically.*

```
3.  // Include the libraries:
```

```
// LiquidCrystal_I2C.h: https://github.com/johnrickman/LiquidCrystal_I2C
#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // Library for LCD
```

The next step is to create an LCD object with the LiquidCrystal_I2C class and specify the address and dimensions. For this, we use the function `LiquidCrystal_I2C(address, columns, rows)`. This is where you will need to change the default address to the address you found earlier if it happens to be different.

When using a 20×4 LCD, change this line to `LiquidCrystal_I2C(0x27,20,4);`

Note that we have called the display 'lcd'. You can give it a different name if you want like 'menu_display'. You will need to change 'lcd' to the new name in the rest of the sketch.

```
// Connect to LCD via I2C, default address 0x27 (A0-A2 not jumpered)
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2); // Change to (0x27,20,4) for 20x4 LCD.
```

In the setup, the LCD is initiated with `lcd.init()` and the backlight is turned on with `lcd.backlight()`.

```
void setup() {
  // Initiate the LCD:
  lcd.init();
  lcd.backlight();
}
```

In the loop section of the code, the cursor is set to the third column and the first row of the LCD with `lcd.setCursor(2,0)`. Note that counting starts at 0 and the first argument specifies the column. So `lcd.setCursor(2,1)` sets the cursor on the third column and the second row.

Next the string 'Hello World!' is printed with `lcd.print("Hello World!")`. Note that you need to place quotation marks (" ") around the text since we are printing a text string (https://www.arduino.cc/reference/en/language/variables/data-types/string/). When you want to print numbers, no quotation marks are necessary. For example `lcd.print(12345)`.

```
18.  void loop() {
19.    lcd.setCursor(2, 0); // Set the cursor on the third column and first row.
20.    lcd.print("Hello World!"); // Print the string "Hello World!".
21.    lcd.setCursor(2, 1); //Set the cursor on the third column and the second row.
22.    lcd.print("LCD tutorial"); // Print the string "LCD tutorial".
23.  }
```

If you want to see an example for displaying (changing) variables on the LCD, check out my tutorial for the HC-SR04 ultrasonic distance sensor:

- How to use a HC-SR04 Ultrasonic Distance Sensor with Arduino (https://www.makerguides.com/hc-sr04-arduino-tutorial/#example-code-hc-sr04-with-dht11-and-i2c-lcd)

Other useful functions of the LiquidCrystal_I2C library

The example sketch above shows you the basics of displaying text on the LCD. Now we will take a look at the other functions of the LiquidCrystal_I2C library.

clear()

Clears the LCD screen and positions the cursor in the upper-left corner (first row and first column) of the display. You can use this function to display different words in a loop.

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.init();
  lcd.backlight();
}

void loop() {
  lcd.clear();
  lcd.print("Monday");
  delay(2000);
  lcd.clear();
  lcd.print("13:45");
  delay(2000);
}
```

home()

Positions the cursor in the top-left corner of the LCD. Use `clear()` if you also want to clear the display.

cursor()

Displays the LCD cursor: an underscore (line) at the position of the next character to be printed.

noCursor()

Hides the LCD cursor. The following example creates a blinking cursor at the end of “Hello World!”.

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.print("Hello World!");
}

void loop() {
  lcd.cursor();
  delay(500);
  lcd.noCursor();
```

```
    delay(500);
  }
```

blink()

Creates a blinking block style LCD cursor: a blinking rectangle at the position of the next character to be printed.

noBlink()

Disables the block style LCD cursor. The following example displays the blinking cursor for 5 seconds and then disables it for 2 seconds.

```
#include <LiquidCrystal_I2C.h>

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

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.print("blink() example");
}

void loop() {
  lcd.blink();
  delay(5000);
  lcd.noBlink();
  delay(2000);
}
```

display()

This function turns on the LCD screen and displays any text or cursors that have been printed to the display.

noDisplay()

This function turns off any text or cursors printed to the LCD. The text/data is not cleared from the LCD memory. This means it will be shown again when the function `display()` is called.

The following example creates a blinking text effect.

```
#include <LiquidCrystal_I2C.h>

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

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.print("Blinking text");
}

void loop() {
  lcd.display();
  delay(2000);
  lcd.noDisplay();
  delay(2000);
}
```

write()

This function can be used to write a character to the LCD. See the section about creating and displaying custom characters below for more info.

scrollDisplayLeft()

Scrolls the contents of the display (text and cursor) one space to the left. You can use this function in the loop section of the code in combination with `delay(500)` , to create a scrolling text animation.

```
#include <LiquidCrystal_I2C.h>

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

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.print("Hello World!");
}

void loop() {
  lcd.scrollDisplayLeft();
  delay(500);
}
```

scrollDisplayRight()

Scrolls the contents of the display (text and cursor) one space to the right.

autoscroll()

This function turns on automatic scrolling of the LCD. This causes each character output to the display to push previous characters over by one space. If the current text direction is left-to-right (the default), the display scrolls to the left, if the current direction is right-to-left, the display scrolls to the right. This has the effect of outputting each new character to the same location on the LCD.

The following example sketch enables automatic scrolling and prints the character 0 to 9 at the position (16,0) of the LCD. Change this to (20,0) for a 20×4 LCD.

```
#include <LiquidCrystal_I2C.h>

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

void setup() {
  lcd.init();
  lcd.backlight();
}

void loop() {
  lcd.autoscroll();
  lcd.setCursor(16, 0);
  for (int x = 0; x < 10; x++) {
    lcd.print(x);
    delay(500);
  }
  lcd.clear();
}
```

noAutoscroll()

Turns off automatic scrolling of the LCD.

leftToRight()

This function causes text to flow to the right from the cursor, as if the display is left-justified (default).

rightToLeft()

This function causes text to flow to the left from the cursor, as if the display is right-justified.

How to create and display custom characters?

With the function `createChar()` it is possible to create and display custom characters on the LCD. This is especially useful if you want to display a character that is not part of the standard ASCII character set (https://en.wikipedia.org/wiki/ASCII).

CGROM and CGRAM

LCDs that are based on the Hitachi HD44780 LCD controller have two types of memory: CGROM and CGRAM (Character Generator ROM and RAM). CGROM generates all the 5 x 8 dot character patterns from the standard 8-bit character codes. CGRAM can generate user-defined character patterns.

For 5 x 8 dot displays, CGRAM can write **up to 8 custom characters** and for 5 x 10 dot displays 4. For more info see the datasheet.

Custom character example code

The following example sketch creates and displays eight custom characters (numbered 0 – 7).

You can copy the code by clicking on the button in the top right corner of the code field.

```
 /* Arduino example code to display custom characters on I2C character LCD. More info: www.makerguides.com */

 // Include the library:
 #include <LiquidCrystal_I2C.h>

 // Create lcd object of class LiquidCrystal_I2C:
 LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2); // Change to (0x27,20,4) for 20x4 LCD.

 // Make custom characters:
```

```
byte Heart[] = {
  B00000,
  B01010,
  B11111,
  B11111,
  B01110,
  B00100,
  B00000,
  B00000
};

byte Bell[] = {
  B00100,
  B01110,
  B01110,
  B01110,
  B11111,
  B00000,
  B00100,
  B00000
};

byte Alien[] = {
  B11111,
  B10101,
  B11111,
  B11111,
  B01110,
  B01010,
  B11011,
  B00000
};

byte Check[] = {
  B00000,
  B00001,
  B00011,
  B10110,
  B11100,
  B01000,
  B00000,
  B00000
};

byte Speaker[] = {
  B00001,
  B00011,
  B01111,
  B01111,
  B01111,
  B00011,
  B00001,
  B00000
};

byte Sound[] = {
```

```
    B00001,
    B00011,
    B00101,
    B01001,
    B01001,
    B01011,
    B11011,
    B11000
  };

  byte Skull[] = {
    B00000,
    B01110,
    B10101,
    B11011,
    B01110,
    B01110,
    B00000,
    B00000
  };

  byte Lock[] = {
    B01110,
    B10001,
    B10001,
    B11111,
    B11011,
    B11011,
    B11111,
    B00000
  };

  void setup() {
    // Initialize LCD and turn on the backlight:
    lcd.init();
    lcd.backlight();

    // Create new characters:
    lcd.createChar(0, Heart);
    lcd.createChar(1, Bell);
    lcd.createChar(2, Alien);
    lcd.createChar(3, Check);
    lcd.createChar(4, Speaker);
    lcd.createChar(5, Sound);
    lcd.createChar(6, Skull);
    lcd.createChar(7, Lock);

    // Clear the LCD screen:
    lcd.clear();

    // Print a message to the lcd:
    lcd.print("Custom Character");
  }

  // Print all the custom characters:
  void loop() {
```

```
122.    lcd.setCursor(0, 1);
123.    lcd.write(0);
124.
125.    lcd.setCursor(2, 1);
126.    lcd.write(1);
127.
128.    lcd.setCursor(4, 1);
129.    lcd.write(2);
130.
131.    lcd.setCursor(6, 1);
132.    lcd.write(3);
133.
134.    lcd.setCursor(8, 1);
135.    lcd.write(4);
136.
137.    lcd.setCursor(10, 1);
138.    lcd.write(5);
139.
140.    lcd.setCursor(12, 1);
141.    lcd.write(6);
142.
143.    lcd.setCursor(14, 1);
144.    lcd.write(7);
145.  }
```

You should see the following output on the LCD:

How the code works

After including the library and creating the LCD object, the custom character arrays are defined. Each array consists of 8 bytes (only 5 bits are considered). There is 1 byte for each row of the 5 x 8 led matrix. In this example, 8 custom characters are created.

```
// Make custom characters:
byte Heart[] = {
  B00000,
  B01010,
  B11111,
  B11111,
  B01110,
  B00100,
  B00000,
  B00000
};
```

When looking closely at the array, you will see the following. Each row consists of 5 numbers corresponding to the 5 pixels in a 5 x 8 dot character. A 0 means pixel off and a 1 means pixel on. The prefix ‘B’ is the Arduino specific binary formatter.

It is possible to edit each row by hand, but I recommend using this visual tool on GitHub (https://maxpromer.github.io/LCD-Character-Creator/). This application automatically creates the character array and you can click on the pixels to turn them on or off.

In the setup, the custom characters are created with `lcd.createChar(num, data)` .

The first argument in this function is the number of the custom character (0-7) and the second argument is the character array that we created.

```
103.    // Create new characters:
104.    lcd.createChar(0, Heart);
105.    lcd.createChar(1, Bell);
106.    lcd.createChar(2, Alien);
107.    lcd.createChar(3, Check);
108.    lcd.createChar(4, Speaker);
109.    lcd.createChar(5, Sound);
110.    lcd.createChar(6, Skull);
111.    lcd.createChar(7, Lock);
```

In the loop, all the characters are displayed with lcd.write(). As the argument, we use the number of the custom character that we want to display.

```
122.    lcd.setCursor(0, 1);
123.    lcd.write(0);
```

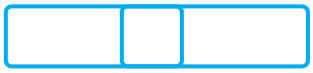
Conclusion

In this article, I have shown you how to use a character I2C LCD with Arduino. I hope you found it useful and informative. If you did, please **share it with a friend** that also likes electronics and making things!

I would love to know what projects you plan on building (or have already built) with these LCDs. If you have any questions, suggestions or if you think that things are missing in this tutorial, **please leave a comment down below**.

Note that comments are held for moderation to prevent spam.

Beginner


What to read next?

LM35 analog temperature sensor with Arduino tutorial (https://www.makerguides.com/lm35-arduino-tutorial/)

TMP36 analog temperature sensor with Arduino tutorial (https://www.makerguides.com/tmp36-arduino-tutorial/)

Arduino Nano Board Guide (Pinout, Specifications, Comparison) (https://www.makerguides.com/arduino-nano/)

The complete guide for DS18B20 digital temperature sensors with Arduino (https://www.makerguides.com/ds18b20-arduino-tutorial/)

How to use an IR receiver and remote with Arduino (https://www.makerguides.com/ir-receiver-remote-arduino-tutorial/)

Comments

Eddie says
February 27, 2021 at 11:19 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-7255)

Thank you for these tutorials. I can't tell you how much this has help me as I am trying to get back into programming after a 50 year break. Your examples and explanations has made it much easier for me to understand what's going on with in the sketches. I have a lot to relearn but your efforts here have given a good base from which to build on.

Reply

Friedhelm Böhl says
February 16, 2021 at 6:15 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-6943)

A very good explanation! Thanks a lot!
I have experience with LCD Moduls in combination with “Conrad C-Conrol Unit) several year ago Heating control, Home-Lighting aso.
Now my Question:
I have bought a OLED-LCD 1602 type “W 162-X3LG.
In the datasheet is the information: similar to HD44780, So generally it works but there is one big issue:
After start with power on (Arduino UNO R3) most times I dedect a normal operation. But afte new start by pressing the Arudino reset button the display shows confuse characters, dots, lines or darkness.
Whats the reason and what can we do?
Is there any instruction for a “module reset”?
Friedhelm

Reply

Benne de Bakker says
March 2, 2021 at 9:00 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-7296)

You can use the lcd.clear() command in the setup section of your code.

Reply

Neill Davison (http://United%20Kingdom) says

February 16, 2021 at 10:03 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-6922)

Thanks for this.
Note that this code also works on a Seeeduino XIAO but you do get a compiler message that the library requires AVR architecture and may not run on the SAMD architecture of the Seeeduino. But it does!

Reply

frothingham says
January 13, 2021 at 5:16 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-5977)

Just what I needed. Easy to use and understand. Thank you for this tutorial!

Reply

wouter says
January 4, 2021 at 11:24 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-5791)

very easy to read explaination, together with good examples to make stuff clear.

makes me wonder, do you have such an easy tutorial on using port expanders like MCP23016/mcp23017.

Reply

Fergus Thomson says
November 26, 2020 at 6:42 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-5000)

Excellent tutorial thanks. The only issue I was having was the SDA and SDL pin wiring. I am using an ELEGOO UNO R3 and the pins are on the other side of the board above the AREF pin on this board. Once connected the code ran correctly and my messages are appearing on the display.

Reply

David Reuveni says
September 18, 2020 at 8:37 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-3908)

Very good work! Congratulations. I tried quite a large number of libraries and “tutorials” but yours was the first that worked.
I was so please that I reprogrammed the top line to “Finally it works”.

Reply

Denton says
August 25, 2020 at 9:19 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-3613)

Does the strength of the connection matter when it comes to printing on the LCD? I have a pro micro without the M-M header pins soldered on and the I2C LCD and some F-F jumper cable. I was holding the header pins to the pro micro and the LCD was lit up, but nothing was printing on it. Obviously the loose/unstable connection is

not ideal but I'm waiting on a soldering kit in the mail and wanted to play around with it as the LCD just arrived in the mail today. Even when I held it still enough that the LCD was continuously lit up for 20-30s it still didn't print.

Thanks!

Reply

Benne de Bakker says
August 26, 2020 at 8:21 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-3622)

Hi Denton,

An unstable connection can definitely be an issue. Have you tried adjusting the contrast of the display by turning the potentiometer?

Benne

Reply

Julian Hernandez says
December 26, 2020 at 10:27 pm
(https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-5633)

Very useful Tutorial, just what i need. Thank You so much. I am working in an oxymeter to help an ong that help the animals in my country.

Reply

Simondalt says
June 30, 2020 at 9:39 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-3077)

Excellent tutorial with all the small bits in place, this made it work at the first attempt.
Thank you!

Reply

Mark Saunders says
June 27, 2020 at 5:30 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-3039)

Hi Thanks for the article, is their a command to vary the brightness? I have used the backlight pin with a pwm signal on other project and was wondering if something has been implemented in this library

Reply

Benne de Bakker says
June 29, 2020 at 11:04 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-3071)

Hi Mark,

With this library you can only turn the backlight on or off.

Benne

Reply

tin says
August 8, 2020 at 1:13 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-3440)

Try using analogWrite(https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/ (https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/)) pin is pwm pin that backlight is connected to with at least 1k resistor in series(if that's not built in the screen).

Reply

JC Smith says
June 6, 2020 at 9:59 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2873)

Excellent work! Well done!

Reply

Benne de Bakker says
June 7, 2020 at 9:37 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2879)

Thanks!

Reply

snk says
May 23, 2020 at 8:10 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2731)

Hoi Benne,
Een mooi duidelijke oefening.
Ik merk op dat bij lcd.print(2345) de nummers juist worden weergegeven behalve als de string met een 0 begint. Enig idee hoe dat kan?

Reply

Benne de Bakker says
May 23, 2020 at 8:24 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2732)

Hi, good question! When you add a leading ‘0’ to the number, it is interpreted as an octal (base 8). You will see that you can’t print 0198 for example. You can find more info about this here on the Arduino website:

https://www.arduino.cc/reference/en/language/variables/constants/integerconstants/
(https://www.arduino.cc/reference/en/language/variables/constants/integerconstants/)

Reply

snk says
May 23, 2020 at 12:23 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2733)

Thank you!

Reply

Fred44 says
May 22, 2020 at 2:19 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2728)

Hi Benne,

Thanks for the tutorial! It made setting up my I2C LCD a lot simpler than I expected.
I also tried the “Other useful functions” with the additional commands. I got stuck because I inserted the command clear() just like that and got a compilation error.
Looking into your codes I noticed that the correct syntax is lcd.clear(), something I had not realised myself. A mistake like that is of course only made once, but:
Maybe you could explain that more explicitly in the text of your tutorial?
Or am I the first one being so ignorant?

Reply

Eduardo says
May 18, 2020 at 3:24 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2698)

Really great tutorial!!! Clear and direct to the point and the right and complete code.
Thanks a lot for sharing.

Reply

Bill Cragie says
April 17, 2020 at 1:48 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2455)

What a brilliant tutorial. I was trying various sites for advice, a code to find the address and a library.
I came upon this tutorial which dealt with all of my problems.
Many thanks and keep up the splendid work

Reply

Marc says
May 25, 2020 at 6:20 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2745)

Me too!

Reply

Luis says
April 11, 2020 at 1:46 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2349)

Beautiful tutorial! Thank you very much!

Reply

wehibe says

April 4, 2020 at 2:47 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2267)

Benne

I really appreciate for your time to laid out the details and providing a good example. I used the I2C bus address identifier code. I was using Ardafruit 292 and somehow Adafruit guide has as 0x70 when no jumper.

see the second page on this link
https://learn.adafruit.com/i2c-spi-lcd-backpack/arduino-i2c-use
(https://learn.adafruit.com/i2c-spi-lcd-backpack/arduino-i2c-use)

Spent hours and hours using what Adafruit suggested address when no jumper. However I came across your site and I run the code you have to identify the address of the i2c bus driver 292 and it comes out as 0x20 which completely unrelated what the vendor stated.

At least one down and I am still struggling not showing the hellow world using the CrystalDisplay 499. I connected 1 to 16 I2C 292 and 499, and somehow I don’t see any char being displayed.

I was using your the code you provided, I changed the 0x27 to 0x20,

Any thought you may have please let me know

Thank you again

Reply

Benne de Bakker says
April 4, 2020 at 9:21 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2269)

Hi, it looks like the Adafruit 292 uses a different I2C I/O expander (MCP23008) than the one you commonly find on I2C LCDs (PCF8574). The LiquidCrystal_I2C library is, therefore, probably not compatible with your setup. I recommend using the Adadruit library instead (see their website).
Benne

Reply

Shane Claussen says
March 17, 2020 at 4:44 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2067)

Terrific article, detailed, worked perfectly, many thanks for putting in the effort.

Reply

James Thomsen says
March 8, 2020 at 9:23 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2009)

Thank you so much. This is all new to me and your description was perfect!

Reply

sugi says
March 7, 2020 at 2:19 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-1999)

Nice, clear& very helpful.
Thanks a lot!

Reply

raan says
February 23, 2020 at 3:43 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-1911)

i m able to view only black boxes instead of text please help me

Reply

Benne de Bakker says
May 14, 2020 at 7:39 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2676)

Have you tried adjusting the contrast potentiometer on the back of the module?

Reply

Oswald Braeuer says
February 14, 2020 at 1:32 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-1844)

If have tried manny Libraries but this is the first Library that works.
But I have a question: is there a command to switch off the Backligt?

Reply

Benne de Bakker says
May 14, 2020 at 7:41 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-2677)

Yes, you can use lcd.noBacklight()

Reply

Jeyaraman P says
January 15, 2020 at 12:40 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-1592)

Thanks for sharing this trove.

liked the way you have explained. i am trying out the most of the functions.

Reply

Joe Simone says
January 6, 2020 at 12:42 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-1516)

Good job. Everything is much clearer to me now. Thank you.
I plan to build a PID Temperature Kiln controller .

Reply

Mostafa abdulaziz says
September 28, 2019 at 11:04 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-666)

Nice topic thanks

Reply

Rahul Padman says
August 13, 2019 at 3:40 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-374)

Can’t thank you enough for such an detailed explanation… Great help… 👍 👍

Reply

Richard says
June 29, 2019 at 1:19 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-166)

Thank you very much for taking the time.

Reply

Dr Bryan Roe says
May 13, 2019 at 8:53 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-46)

Thanks very much for your reply Benne. Prior to picking up your reply I'd had a look at the various libraries which I had installed and suspected that was they were the issue. So I decided to download the version from your tutorial and it worked first time!
I am using a number of different LCD modules as part of my STEM Ambassador role to demonstrate what can be done with Arduinos.

Reply

Dr Bryan Roe says
May 10, 2019 at 1:36 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-39)

I have tried to load/compile this file into my Arduino Uno, but repeatedly get an error message
Liquid\Crystal_I2C – no such file or file directory.

I have a Liquid Crystal I2C library loaded into my IDE.

Help!

Reply

Benne de Bakker says
May 10, 2019 at 2:03 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-42)

Hi Bryan,

Thank you for your comment. I suspect you have a different library installed than the one I used in this tutorial. There are several Liquid Crystal I2C libraries available on the web and some have the same name. If you download the library by clicking on the download button in this tutorial or via the GitHub link in the code it should work. I am not sure if you need to remove the previously installed library or not.

Benne

Reply

Trackbacks

Automatic Plant Watering with Arduino IoT Cloud (with Pictures) (https://www.makerguides.com/automatic-plant-watering-system-with-arduino-iot-cloud/) says:
June 23, 2020 at 1:09 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-3003)
[...] How to control a character I2C LCD with Arduino [...]

TM1637 4-Digit 7-Segment Arduino Tutorial (3 Examples) (https://www.makerguides.com/tm1637-arduino-tutorial/) says:
September 6, 2019 at 3:07 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-503)
[...] How to control a character I2C LCD with Arduino [...]

TM1637 4-Digit 7-Segment Display Arduino Tutorial (3 Examples) (https://www.makerguides.com/tm1637-4-digit-7-segment-display-arduino-tutorial/) says:
September 3, 2019 at 11:23 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-481)

[...] How to control a character I2C LCD with Arduino [...]

DHT11/DHT22 Sensors with Arduino Tutorial (2 Examples) (https://www.makerguides.com/dht11-dht22-arduino-tutorial/) says:
August 20, 2019 at 8:52 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-417)
[...] How to control a character I2C LCD with Arduino [...]

16x2 Character LCD Arduino Tutorial (wiring diagram + examples) (https://www.makerguides.com/character-lcd-arduino-tutorial/) says:
July 8, 2019 at 9:37 am (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-204)
[...] How to control a character I2C LCD with Arduino [...]

How to use HC-SR04 Ultrasonic Sensor with Arduino (5 examples) (https://www.makerguides.com/hc-sr04-arduino-tutorial/) says:
June 16, 2019 at 12:38 pm (https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/#comment-112)
[...] How to control a character I2C LCD with Arduino [...]

(https://www.makerguides.com/servo-arduino-tutorial/)

How to control servo motors with Arduino (https://www.makerguides.com/servo-arduino-tutorial/)

(https://www.makerguides.com/lm35-arduino-tutorial/)

LM35 analog temperature sensor with Arduino tutorial (https://www.makerguides.com/lm35-arduino-tutorial/)

(https://www.makerguides.com/tb6560-stepper-motor-driver-arduino-tutorial/)

TB6560 Stepper Motor Driver with Arduino Tutorial (https://www.makerguides.com/tb6560-stepper-motor-driver-arduino-tutorial/)