SHOP

TUTORIALS FORUMS BLOG My Account 🛒

TUTORIAL

Using a Character LCD the easy way with the help of the I2C bus.

VIDEO

Use the I2

OVERVIEW

Using a character LCD with an Arduino is nothing new, but let’s make it very simple by using the I2C bus.

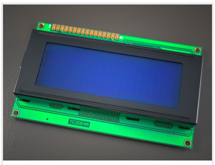
By using this little I2C LCD board, we can control the LCD using only 2 wires, and not worry about resistors to adjust the contrast since it’s all included.

To make this tutorial more interesting, we will use the HC-SR04 ultrasonic range finder that we used before and display the results on the LCD.

SCHEMATIC

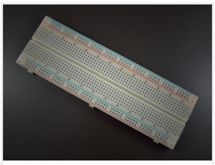
Use the I2...

PARTS USED



Character LCD Display 20×4 Module

$11.95

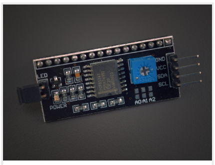


Full size Breadboard

$6.95

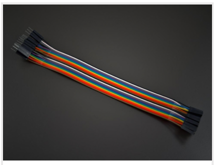
HC-SR04 Ultrasonic Sensor module

~~$3.50~~
$2.95

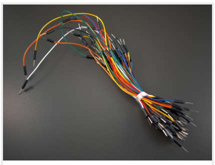


i2c LCD backpack

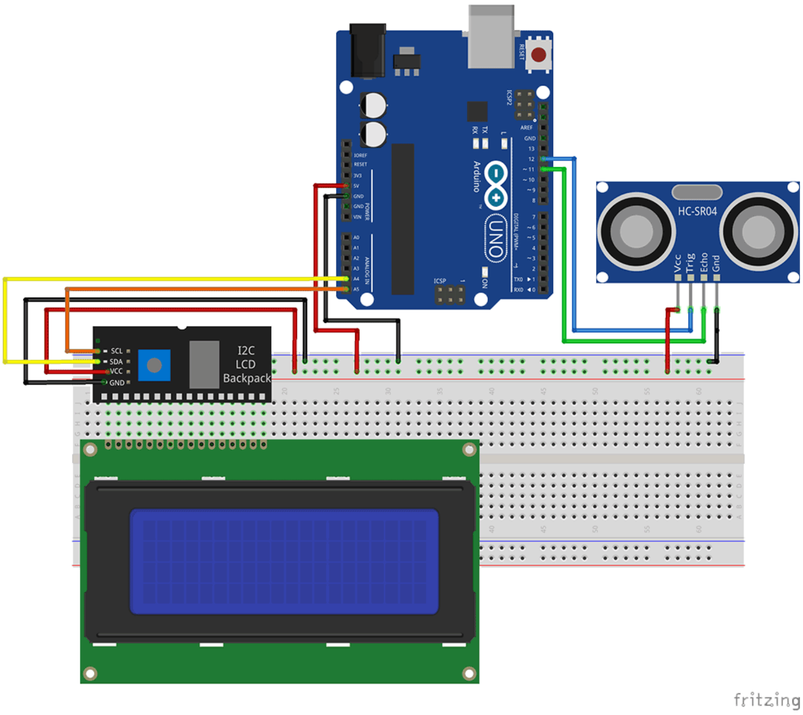
$2.95



Jumper Wires male to female 20pcs



Jumper Wires male to male – 65pcs



We are using a 20×4 Character LCD so we have 4 lines of 20 characters each available.
The I2C LCD module is connected to pin A4-SDA and A5-SCL.
The HC-SR04 is connected to pin 11-echo and 12-trigger.
VCC and Ground from the Arduino are connected to the breadboard rails .
We then connect the VCC and Ground to both the I2C module and the HC-SR04.

THE CODE

Before we can use the I2C module, we need to find out it’s HEX address so we can communicate with it.

We find the I2C module address by running this “I2C Scanner” sketch on our Arduino :

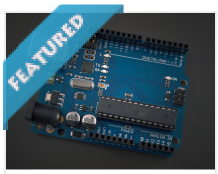
**note:* *Watch the tutorial video to see the code in action and the results.*

```
#include <Wire.h>


void setup()
{
  Wire.begin();

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


```



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

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

  nDevices = 0;
  for(address = 1; address < 127; address++ )
  {
    // The i2c_scanner uses the return value of
    // the Write.endTransmisstion to see if
    // a device did acknowledge to the address.
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x")
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println("  !");

      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknow error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(5000);           // wait 5 seconds for next
}
```

As you can see in the tutorial video, the address for our I2C module is 0X27, so we will use that value in our Sketch to communicate with it.

We will use the “NewPing” library to communicate with the HC-SR04 ultrasonic range sensor.

We are also using the “NewLiquidCrystal” library in our Sketch, this library unlike the one included with the Arduino IDE is more recent, faster and can use the I2C bus, so might as well use it.

You can download the libraries at the bottom of this tutorial page.

**Rename the original LiquidCrystal folder in the Arduino/Libraries to something like LiquidCrystal_Old before extracting the*

NewLiquidCrystal library, since the folders have the same name, and the original library will conflict with this new one.

As always, please watch the tutorial video for more

Here's is the Sketch to display the HC-Sr04 result on the LCD using I2C :

HC-SR04 results on LCD using I2C bus

```
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#include <NewPing.h>

#define I2C_ADDR 0x27 // <<- Add your address here.
#define Rs_pin 0
#define Rw_pin 1
#define En_pin 2
#define BACKLIGHT_PIN 3
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7

#define ECHO_PIN 11 // Arduino pin tied to echo pin
#define TRIGGER_PIN 12 // Arduino pin tied to trigge
#define MAX_DISTANCE 500 // Maximum distance we want

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

LiquidCrystal_I2C lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,

void setup()
{
 lcd.begin (20,4); // <<-- our LCD is a 20x4, change

// LCD Backlight ON
lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
lcd.setBacklight(HIGH);

lcd.home (); // go home on LCD
lcd.print("Range Finder HC-SR04");
}

void loop()
{
 unsigned int uS = sonar.ping(); // Send ping, get p
 unsigned int cm = sonar.convert_cm(uS); // Convert

 lcd.setCursor (0,1); // go to start of 2nd line
 lcd.print("Current Distance:");
 lcd.setCursor (0,3); // go to start of 4th line
 lcd.print("Ping: ");
 lcd.print(cm);
 lcd.print(" cm ");

 delay(500);
}
```

Translate »

DOWNLOAD

Download the NewLiquidCrystal library here: New_LiquidCrystal_Library.zip

Download the HC-SR04 library here: HC-SR04_Library.zip

Download the I2C_Scanner here: I2C_Scanner.zip

Download the Tutorial Sketch here: I2C_LCD_Tutorial.zip

Share This Tutorial, Choose Your Platform!

If you have questions or would like to discuss this tutorial, click here to go to the Brainy-Bits Forum.

ABOUT US

CONTACT / SUPPORT

SHIPPING & RETURNS

PRODUCT SEARCH :

Search...

“ The real problem is not whether machines think but whether men do. ”

— B.F. Skinner