LCD-Blue-I2C (/LCD-Blue-I2C)

Edit | 0 (/LCD-Blue-I2C#discussion) | 107 (/page/history/LCD-Blue-I2C) | … (/page/menu/LCD-Blue-I2C)

LCD Displays (Blue and Yellow) with I2C/TWI Interface

IMPORTANT LCD Software Library Info:

ALL these displays will now use the great Library written by F Malpartida . You should download that Library HERE (Click) (Get the latest file such as LiquidCrystal_V1.2.1.zip)

If you try to Verify and you see this:
error: 'LiquidCrystal_I2C' does not name a type
...this means you do not have the library installed correctly. See:
How-To-Install-Libraries
NOTE!! Move any other LCD libraries to another folder,or rename or delete them.

NOTE! THERE ARE THREE (or More??) VERSIONS OF THE 2 and 4 LINE DISPLAYS. Check the small "backpack" interface board on the back. These small circuit boards on the back interface a 2-wire I2C "I Squared C" bus (plus Ground and +5V power) to the many pins on the LCD display itself. This allows running the display from only 2 signal pins on Arduino. The YourDuino RoboRED (UNO Compatible) has a 4-pin connector that works well for this type display.

LCD Version 1 - Marked "YwRobot Arduino LCM1602 IIC V1" ALSO version marked "A0 A1 A2" on lower right.
LCD Version 2 - Marked "Arduino-IIC-LCD GY-LCD-V1"
LCD Version 3 - Marked "LCM1602 IIC A0 A1 A2"

These displays are Available HERE:

NOTE! These displays are very clear bright white on dark blue (or Black on Yellow) background: it is difficult to get a good digital photo due to polarization effects.

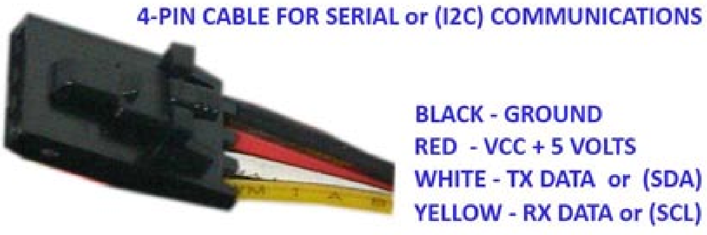
NOTE! I2C ADDRESSES:

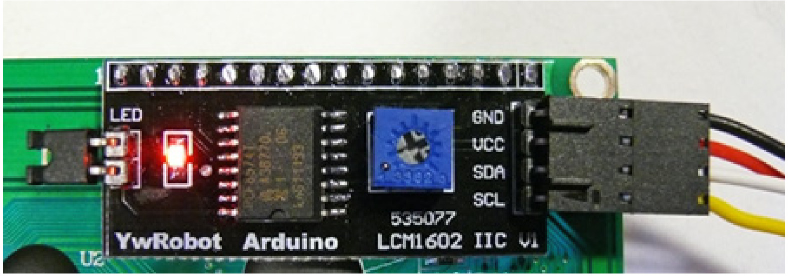
These I2C interface displays have a built-in ADDRESS. If you try to run a display and there is no response or nothing is displayed you may have the wrong address in the code you are using. You can run the I2C ADDRESS SCANNER (Bottom of this page) on your Arduino to check the address on your display if necessary. For LOTS of detailed technical information about I2C Interfaces see THIS from Nick Gammon.

WHY THE I2C (IIC) TYPE LCD DISPLAY?

To use this type LCD directly with Arduino, you would need 6 pins: RS, EN, D7, D6, D5, and D4 to talk to the LCD. If you are doing more than a simple project, you may be out of pins using a normal LCD shield. With this I2C interface LCD module, you only need 2 lines (I2C) to display information. If you already have I2C devices in your project, this LCD module actually uses no more pins at all. More information about I2C/TWI (Wikipedia)
This unit connects with 4 wires including Vcc and Gnd. It is easiest with a 4-wire cable that plugs into the Sensor Shield communications connector or directly to a YourDuino RoboRED (UNO Compatible)..

But you can wire it directly yourself if needed: There are 4 pins on the display.. (see photo below)


4-PIN CABLE FOR SERIAL or (I2C) COMMUNICATIONS
BLACK - GROUND
RED - VCC + 5 VOLTS
WHITE - TX DATA or (SDA)
YELLOW - RX DATA or (SCL)



Top to bottom:
GND - GND
VCC - 5V
SDA - ANALOG Pin 4
SCL - ANALOG pin 5
On most Arduino boards,
SDA (data line) is on analog input pin 4, and SCL (clock line) is on analog input pin 5. On the Arduino Mega, SDA is digital pin 20 and SCL is 21. On the newer

Arduino UNO (The "V3" pinout), the SCL and SDA pins are also on two new leftmost top pins. The YourDuino RoboRED has those pins and also a nice 4-pin connector arranged exactly like the LCD display pins.
NOTE: The Blue Potentiometer (Photo) adjusts ***Contrast***. If you don't see any characters, adjust it. Start clockwise and back down to where the characters are bright and the background does not have boxes behind the characters.

See a LCD used in the Temperature-Humidity project.

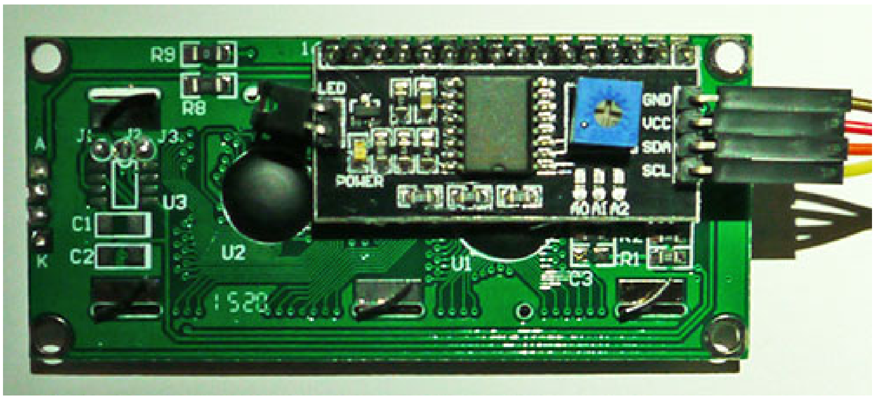
LiquidCrystal_I2C Library Support Functions

See the "Docs" folder within the Library folder. There are many advanced functions.

Below are Example Software Sketches for different displays. They will display characters you type on the Serial Monitor screen on the LCD. NOTE: Line 1 only is correct when writing a long sequence of characters. The characters fill the first line and continue on the third, due to the way the LCD internal addressing works. So this is "normal" and has to do with the LCD hardware. Usually you will set the cursor position before writing characters. (Details in the document linked at the end of this page, if you want them.)
(Cut and paste these examples into a blank page on the Arduino IDE).

I2C LCD DISPLAY VERSION 1:

Example Software Sketch for 2 line 16 character Displays:



(NOTE: for displays with backpack interface labelled "YwRobot Arduino LCM1602 IIC V1")
(NOTE: Also for displays with backpack interface labelled "A0 A1 A2" photo: right).
(NOTE: Most displays use I2C Adress 0x27 but a FEW use 0x3F (Change in code below)

```
/* YourDuino.com Example Software Sketch
```

```
 16 character 2 line I2C Display
 Backpack Interface labelled "A0 A1 A2" at lower right.
 ..and
 Backpack Interface labelled "YwRobot Arduino LCM1602 IIC V1"
 MOST use address 0x27, a FEW use 0x3F
 terry@yourduino.com */

/*-----( Import needed libraries )-----*/
#include <Wire.h>  // Comes with Arduino IDE
// Get the LCD I2C Library here:
// https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
// Move any other LCD libraries to another folder or delete them
// See Library "Docs" folder for possible commands etc.
#include <LiquidCrystal_I2C.h>

/*-----( Declare Constants )-----*/
/*-----( Declare objects )-----*/
// set the LCD address to 0x27 for a 16 chars 2 line display
// A FEW use address 0x3F
// Set the pins on the I2C chip used for LCD connections:
//                    addr, en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  // Set the LCD I2C address

/*-----( Declare Variables )-----*/
//NONE

void setup()   /*----( SETUP: RUNS ONCE )----*/
{
  Serial.begin(9600);  // Used to type in characters

  lcd.begin(16,2);   // initialize the lcd for 16 chars 2 lines, turn on backlight

// ------- Quick 3 blinks of backlight  -------------
  for(int i = 0; i< 3; i++)
  {
    lcd.backlight();
    delay(250);
    lcd.noBacklight();
    delay(250);
  }
  lcd.backlight(); // finish with backlight on

//-------- Write characters on the display ------------------
// NOTE: Cursor Position: (CHAR, LINE) start at 0
  lcd.setCursor(0,0); //Start at character 4 on line 0
  lcd.print("Hello, world!");
  delay(1000);
  lcd.setCursor(0,1);
  lcd.print("HI!YourDuino.com");
  delay(8000);

// Wait and then tell user they can start the Serial Monitor and type in characters to
// Display. (Set Serial Monitor option to "No Line Ending")
  lcd.clear();
  lcd.setCursor(0,0); //Start at character 0 on line 0
  lcd.print("Use Serial Mon");
  lcd.setCursor(0,1);
  lcd.print("Type to display");


}/*--(end setup )---*/
```

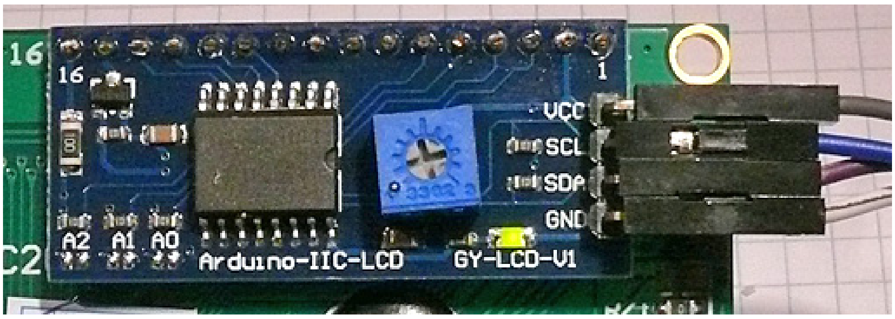
```
void loop()   /*----( LOOP: RUNS CONSTANTLY )----*/
{
  {
    // when characters arrive over the serial port...
    if (Serial.available()) {
      // wait a bit for the entire message to arrive
      delay(100);
      // clear the screen
      lcd.clear();
      // read all the available characters
      while (Serial.available() > 0) {
        // display each character to the LCD
        lcd.write(Serial.read());
      }
    }
  }

}/* --(end main loop )-- */


/* ( THE END ) */
```

Example Software Sketch for 4 line 20 character Displays:

(NOTE: for displays with backpack interface labelled "YwRobot Arduino LCM1602 IIC V1")

```
/* YourDuino.com Example Software Sketch
 20 character 4 line I2C Display
 Backpack Interface labelled "YwRobot Arduino LCM1602 IIC V1"
 Connect Vcc and Ground, SDA to A4, SCL to A5 on Arduino
 terry@yourduino.com */

/*-----( Import needed libraries )-----*/
#include <Wire.h>  // Comes with Arduino IDE
// Get the LCD I2C Library here:
// https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
// Move any other LCD libraries to another folder or delete them
// See Library "Docs" folder for possible commands etc.
#include <LiquidCrystal_I2C.h>

/*-----( Declare Constants )-----*/
/*-----( Declare objects )-----*/
// set the LCD address to 0x27 for a 20 chars 4 line display
// Set the pins on the I2C chip used for LCD connections:
//                    addr, en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  // Set the LCD I2C address


/*-----( Declare Variables )-----*/


void setup()   /*----( SETUP: RUNS ONCE )----*/
{
  Serial.begin(9600);  // Used to type in characters
  lcd.begin(20,4);         // initialize the lcd for 20 chars 4 lines, turn on backlight

// ------- Quick 3 blinks of backlight  -------------
  for(int i = 0; i< 3; i++)
  {
    lcd.backlight();
    delay(250);
    lcd.noBacklight();
    delay(250);
  }
  lcd.backlight(); // finish with backlight on

//-------- Write characters on the display ------------------
  // NOTE: Cursor Position: Lines and Characters start at 0
  lcd.setCursor(3,0); //Start at character 4 on line 0
  lcd.print("Hello, world!");
  delay(1000);
  lcd.setCursor(2,1);
  lcd.print("From YourDuino");
  delay(1000);
  lcd.setCursor(0,2);
  lcd.print("20 by 4 Line Display");
  lcd.setCursor(0,3);
  delay(2000);
  lcd.print("http://YourDuino.com");
  delay(8000);
  // Wait and then tell user they can start the Serial Monitor and type in characters to
  // Display. (Set Serial Monitor option to "No Line Ending")
  lcd.setCursor(0,0); //Start at character 0 on line 0
  lcd.print("Start Serial Monitor");
  lcd.setCursor(0,1);
  lcd.print("Type chars 2 display");


}/*--(end setup )---*/


void loop()   /*----( LOOP: RUNS CONSTANTLY )----*/
{
  {
    // when characters arrive over the serial port...
    if (Serial.available()) {
      // wait a bit for the entire message to arrive
      delay(100);
      // clear the screen
      lcd.clear();
      // read all the available characters
      while (Serial.available() > 0) {
        // display each character to the LCD
        lcd.write(Serial.read());
      }
    }
  }

}/* --(end main loop )-- */


/* ( THE END ) */
```

I2C LCD DISPLAY VERSION 2:



Marked "Arduino-IIC-LCD GY-LCD-V1" NOTE: The wire connections are in a different order! See the labels on the PC Board.

NEW TYPE 4 line 20 character Displays: Example Software Sketch

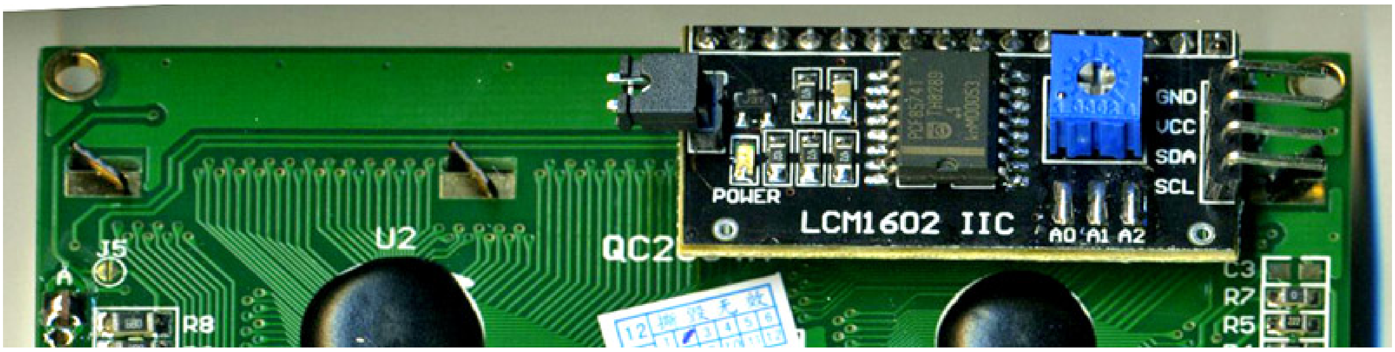
```
/* YourDuino.com Example Software Sketch
 20 character 4 line I2C Display
 Backpack Interface labelled "LCM1602 IIC  A0 A1 A2"
 terry@yourduino.com */

/*-----( Import needed libraries )-----*/
#include <Wire.h>  // Comes with Arduino IDE
// Get the LCD I2C Library here:
// https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
// Move any other LCD libraries to another folder or delete them
// See Library "Docs" folder for possible commands etc.
#include <LiquidCrystal_I2C.h>

/*-----( Declare Constants )-----*/
//none
/*-----( Declare objects )-----*/
// set the LCD address to 0x27 for a 20 chars 4 line display
// Set the pins on the I2C chip used for LCD connections:
//                    addr, en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x20, 4, 5, 6, 0, 1, 2, 3, 7, NEGATIVE);  // Set the LCD I2C address



/*-----( Declare Variables )-----*/
//none

void setup()   /*----( SETUP: RUNS ONCE )----*/
{
  Serial.begin(9600);  // Used to type in characters

  lcd.begin(20,4);         // initialize the lcd for 20 chars 4 lines

// NOTE: Cursor Position: CHAR, LINE) start at 0
  lcd.setCursor(3,0); //Start at character 4 on line 0
  lcd.print("Hello, world!");
  delay(1000);
  lcd.setCursor(2,1);
  lcd.print("From YourDuino");
  delay(1000);
  lcd.setCursor(0,2);
  lcd.print("20 by 4 Line Display");
  lcd.setCursor(0,3);
  delay(2000);
  lcd.print("http://YourDuino.com");
  delay(8000);
// Wait and then tell user they can start the Serial Monitor and type in characters to
// Display. (Set Serial Monitor option to "No Line Ending")
  lcd.setCursor(0,0); //Start at character 0 on line 0
  lcd.print("Start Serial Monitor");
  lcd.setCursor(0,1);
  lcd.print("Type chars 2 display");


}/*--(end setup )---*/


void loop()   /*----( LOOP: RUNS CONSTANTLY )----*/
{
  {
    // when characters arrive over the serial port...
    if (Serial.available()) {
      // wait a bit for the entire message to arrive
      delay(100);
      // clear the screen
      lcd.clear();
      // read all the available characters
      while (Serial.available() > 0) {
        // display each character to the LCD
        lcd.write(Serial.read());
      }
    }
  }

}/* --(end main loop )-- */


/* ( THE END ) */
```

I2C LCD DISPLAY VERSION 3:

This has Yet Another I2C board type marked "LCM1602 IIC A0 A1 A2" (Photo)

NOTE: Some of this type board come with the A0 A1 A2 connections (See photo) that are NOT bridged with solder as in the photo. Those will have address 0x27 not address 0x20. For that version you need to change the sketch example below.

```
LiquidCrystal_I2C lcd(0x20, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
MUST BE CHANGED TO:
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
```



You need the same new library in the section above.

Here's an example Software Sketch that works with this display. Copy this and paste into blank Arduino IDE window. Verify (Need the library above installed), then upload.

```
/* YourDuino.com Example Software Sketch
 20 character 4 line I2C Display
 Backpack Interface labelled "LCM1602 IIC  A0 A1 A2"
 terry@yourduino.com */

/*-----( Import needed libraries )-----*/
#include <Wire.h>  // Comes with Arduino IDE
// Get the LCD I2C Library here:
// https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
// Move any other LCD libraries to another folder or delete them
// See Library "Docs" folder for possible commands etc.
#include <LiquidCrystal_I2C.h>

/*-----( Declare Constants )-----*/
//none
/*-----( Declare objects )-----*/
// set the LCD address to 0x20 for a 20 chars 4 line display
// Set the pins on the I2C chip used for LCD connections:
//                    addr, en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x20, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  // Set the LCD I2C address


/*-----( Declare Variables )-----*/
//none

void setup()   /*----( SETUP: RUNS ONCE )----*/
{
  Serial.begin(9600);  // Used to type in characters

  lcd.begin(20,4);         // initialize the lcd for 20 chars 4 lines and turn on backlight

// ------- Quick 3 blinks of backlight  -------------
  for(int i = 0; i< 3; i++)
  {
    lcd.backlight();
    delay(250);
    lcd.noBacklight();
    delay(250);
  }
  lcd.backlight(); // finish with backlight on

//-------- Write characters on the display ----------------
// NOTE: Cursor Position: CHAR, LINE) start at 0
  lcd.setCursor(3,0); //Start at character 4 on line 0
  lcd.print("Hello, world!");
  delay(1000);
  lcd.setCursor(2,1);
  lcd.print("From YourDuino");
  delay(1000);
  lcd.setCursor(0,2);
  lcd.print("20 by 4 Line Display");
  lcd.setCursor(0,3);
  delay(2000);
  lcd.print("http://YourDuino.com");
  delay(8000);
// Wait and then tell user they can start the Serial Monitor and type in characters to
// Display. (Set Serial Monitor option to "No Line Ending")
  lcd.setCursor(0,0); //Start at character 0 on line 0
  lcd.print("Start Serial Monitor");
  lcd.setCursor(0,1);
  lcd.print("Type chars 2 display");


}/*--(end setup )---*/


void loop()   /*----( LOOP: RUNS CONSTANTLY )----*/
{
  {
    // when characters arrive over the serial port...
    if (Serial.available()) {
      // wait a bit for the entire message to arrive
      delay(100);
      // clear the screen
      lcd.clear();
      // read all the available characters
      while (Serial.available() > 0) {
        // display each character to the LCD
        lcd.write(Serial.read());
      }
    }
  }

}/* --(end main loop )-- */


/* ( THE END ) */
```

I2C ADDRESS SCANNER FOR ARDUINO/YOURDUINO:

You can scan for the I2C address used by your display. Connect your display (See connection info at top of this page). Then upload the following sketch to your Arduino. Click on "Serial Monitor at upper right. Set the Speed (lower right pulldown) to 115200. Push the Reset button on your Arduino and see the results. The results should look like this (address changes):

```
I2C scanner. Scanning ...
Found address: 39 (0x27)
Done.
Found 1 device(s).
```

Copy and Paste this sketch (Thanks Nick Gammon!) into a blank Arduino IDE window:

```
// I2C Scanner
// Written by Nick Gammon
// Date: 20th April 2011

#include <Wire.h>

void setup() {
  Serial.begin (115200);

  // Leonardo: wait for serial port to connect
  while (!Serial)
    {
    }

  Serial.println ();
  Serial.println ("I2C scanner. Scanning ...");
  byte count = 0;

  Wire.begin();
  for (byte i = 8; i < 120; i++)
  {
    Wire.beginTransmission (i);
    if (Wire.endTransmission () == 0)
      {
      Serial.print ("Found address: ");
      Serial.print (i, DEC);
      Serial.print (" (0x");
      Serial.print (i, HEX);
      Serial.println (")");
      count++;
      delay (1);  // maybe unneeded?
      } // end of good response
  } // end of for loop
  Serial.println ("Done.");
  Serial.print ("Found ");
  Serial.print (count, DEC);
  Serial.println (" device(s).");
}  // end of setup

void loop() {}
```

CHANGING I2C ADDRESSES:

SOME I2C interfaces have pins (or solder pads) that can be changed to change the address. They are usually labelled A0-A1-A2 . Here's the way addresses change from a default 0x27 with if you connect address pads together. (1 = Not Connected. 0 = Connected):

A0	A1	A2	HEX Address
1	1	1	0x27
0	1	1	0x26
1	0	1	0x25
0	0	1	0x24
1	1	0	0x23
0	1	0	0x22
1	0	0	0x21
0	0	0	0x20

If you make a change use the I2C Address Scanner to confirm it...

Detailed information about LCD displays from Donald Weiman (weimandn@alfredstate.edu)

LCD Addressing.pdf
Details Download 251 KB

z