

# Two types of LCD displays

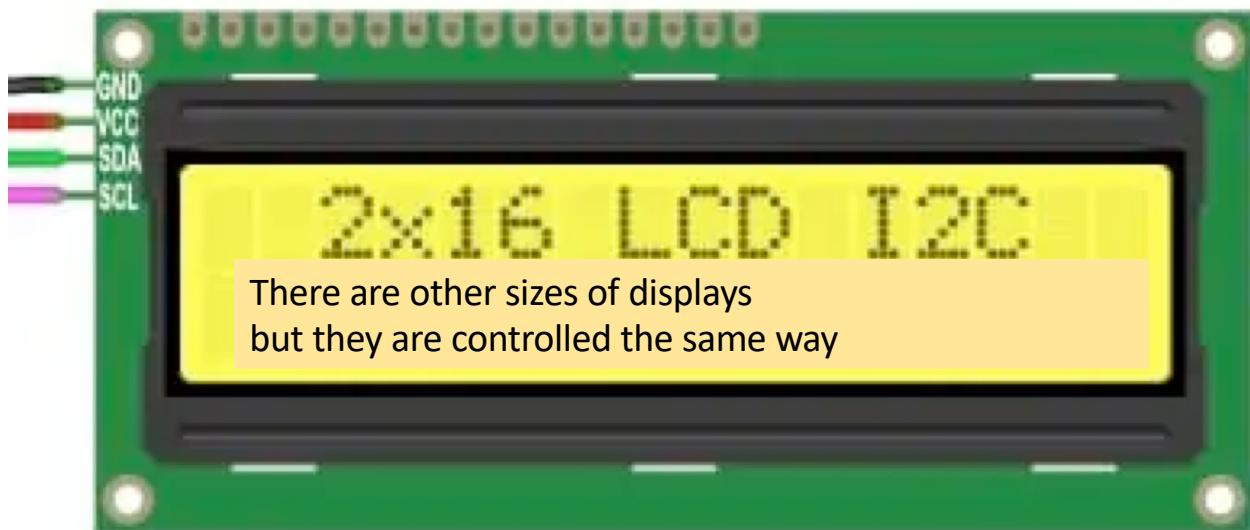
This is for I2C displays

# LCD Display

- LCD displays are popular
- We will show how to use a simple LCD display and
- Use the LCD library
- There are two types of LCD displays that are in common use
  - Ones with multiple wires
  - Ones with 4 wires that support I2C protocol
- Depending on which kit you have, you may have one or the other
- These slides give additional information on the I2C display

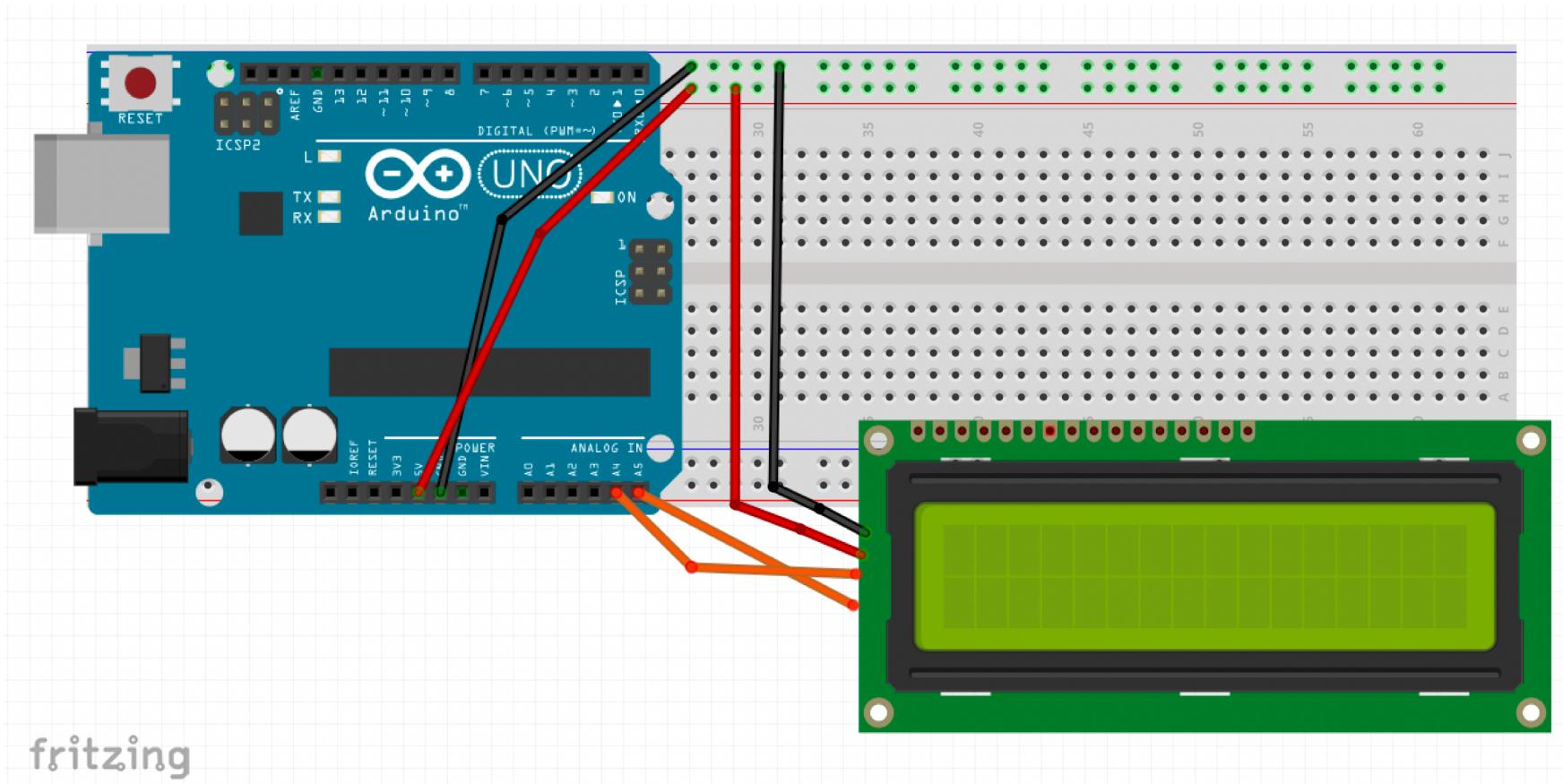
# Connections between I2C LCD and Arduino

LCD1602	Arduino
GND	-> GND
VCC	-> +5V
SDA	-> A4
SCL	-> A5



16 character by 2 line display

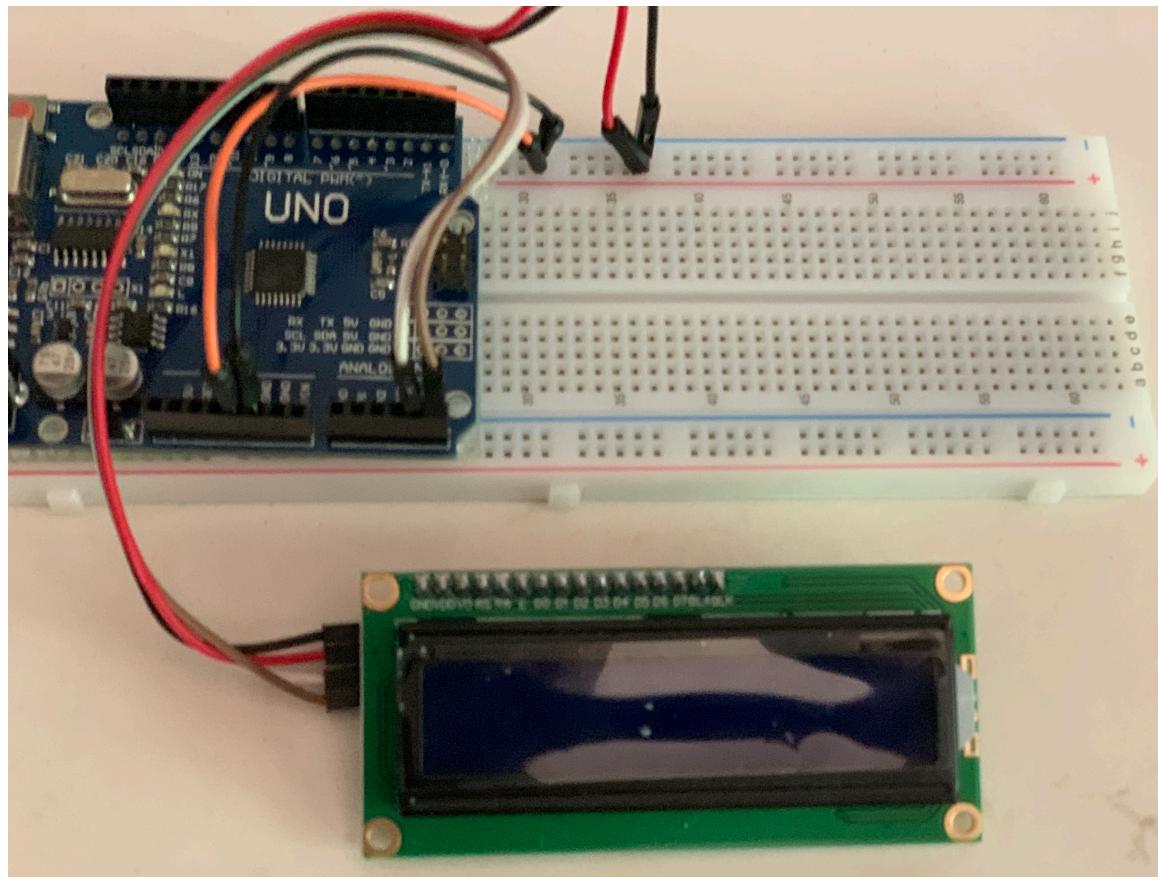
# One View of Wiring -I2C



fritzing

Copyright 2023 Joseph E. Wilkes

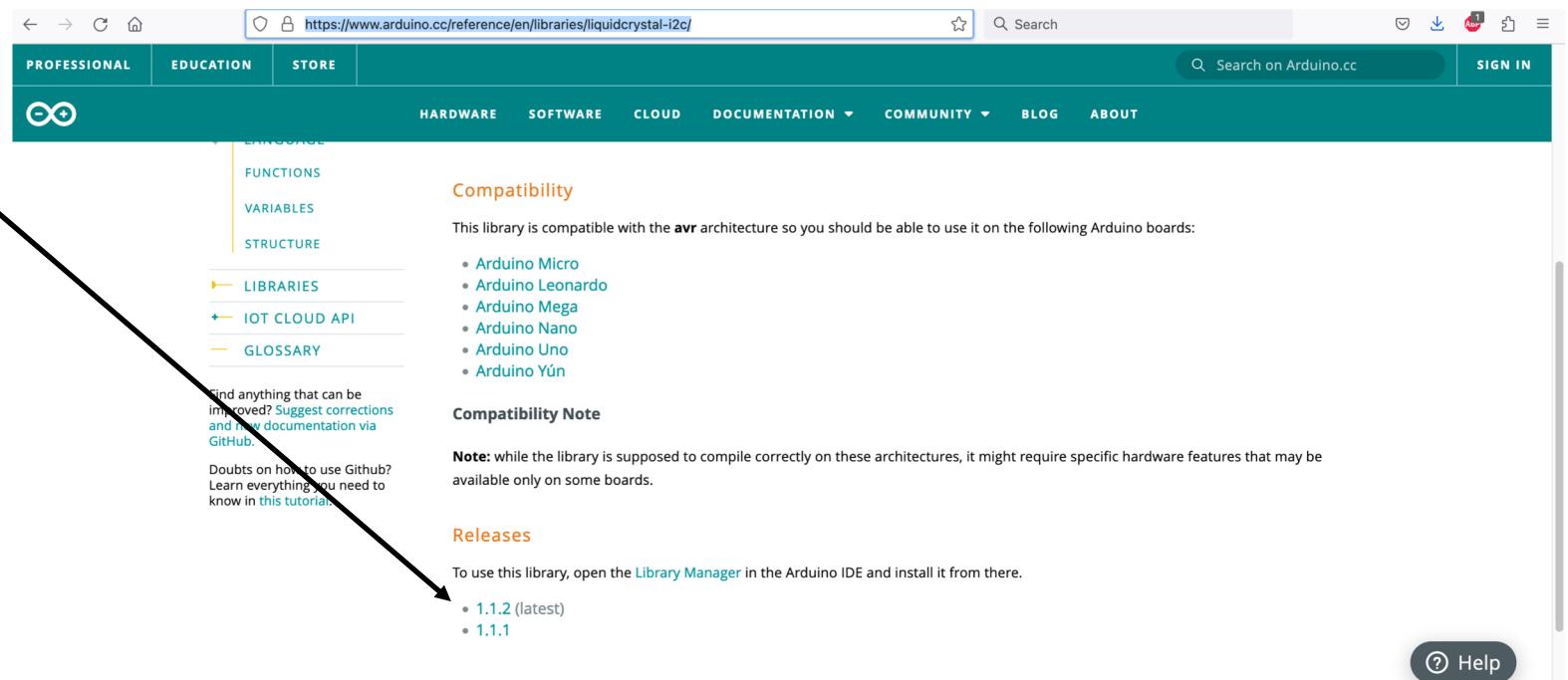
# LCD display experimental setup



Copyright 2023 Joseph E. Wilkes

# Liquid Crystal Library

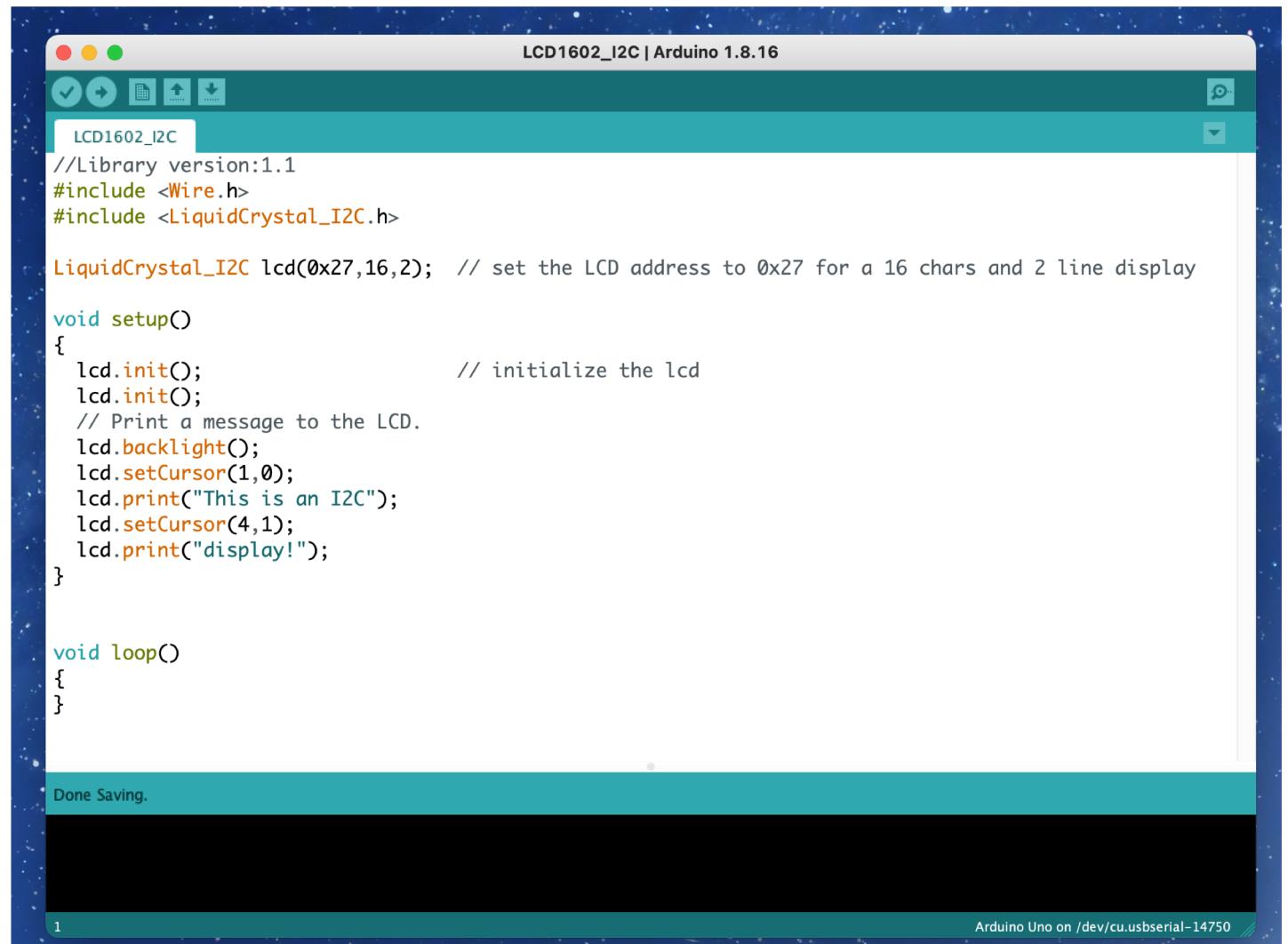
- Go to <https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>
- click here to down load the library



- Install library

# LCD I2C Display

- We import some libraries



The screenshot shows the Arduino IDE interface with a sketch titled "LCD1602\_I2C". The code uses the Wire library and the LiquidCrystal\_I2C library to initialize and control a 16x2 LCD connected via I2C. The setup() function initializes the LCD, sets the cursor at position (1,0), prints "This is an I2C", sets the cursor at (4,1), and prints "display!". The loop() function contains an empty brace {}, indicating no code is run in the loop. A status bar at the bottom indicates "Done Saving." and "Arduino Uno on /dev/cu.usbserial-14750".

```
	LCD1602_I2C
//Library version:1.1
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display

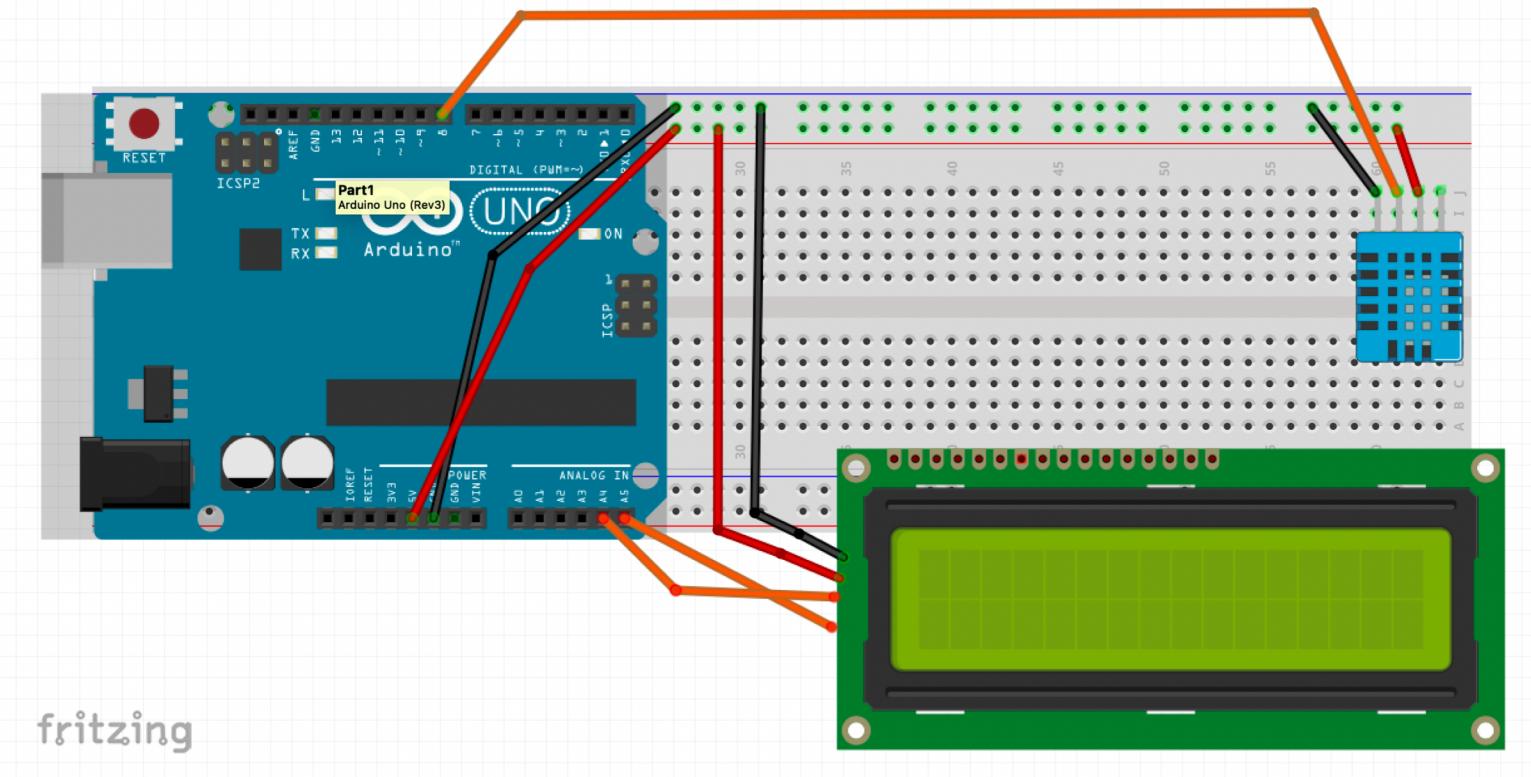
void setup()
{
    lcd.init(); // initialize the lcd
    lcd.init();
    // Print a message to the LCD.
    lcd.backlight();
    lcd.setCursor(1,0);
    lcd.print("This is an I2C");
    lcd.setCursor(4,1);
    lcd.print("display!");
}

void loop()
{}
```

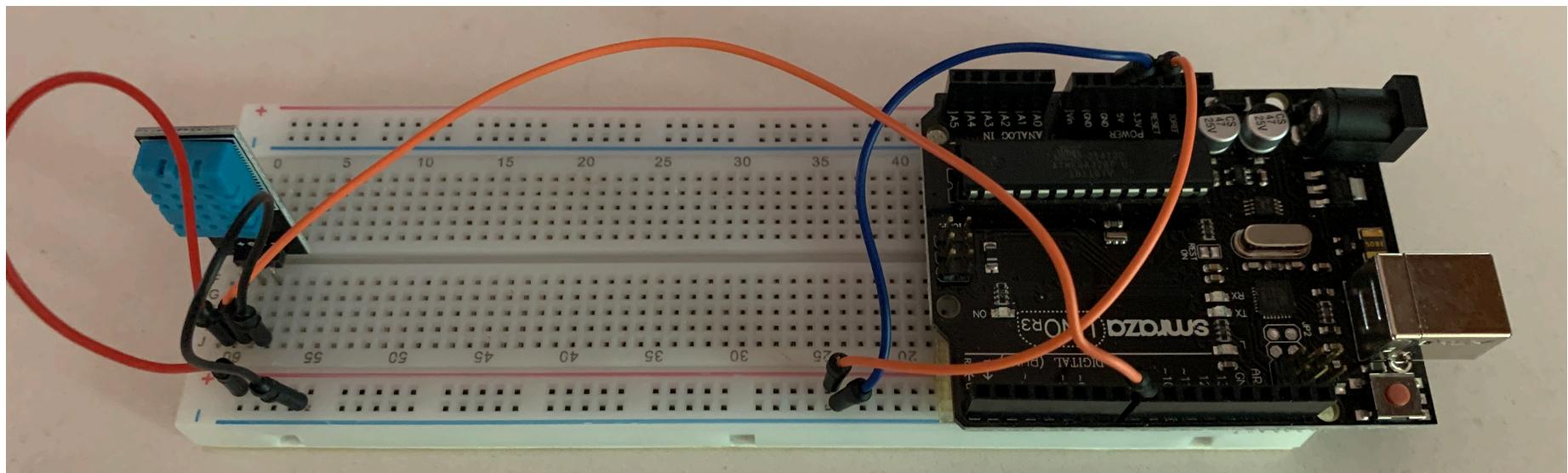
Done Saving.

Arduino Uno on /dev/cu.usbserial-14750

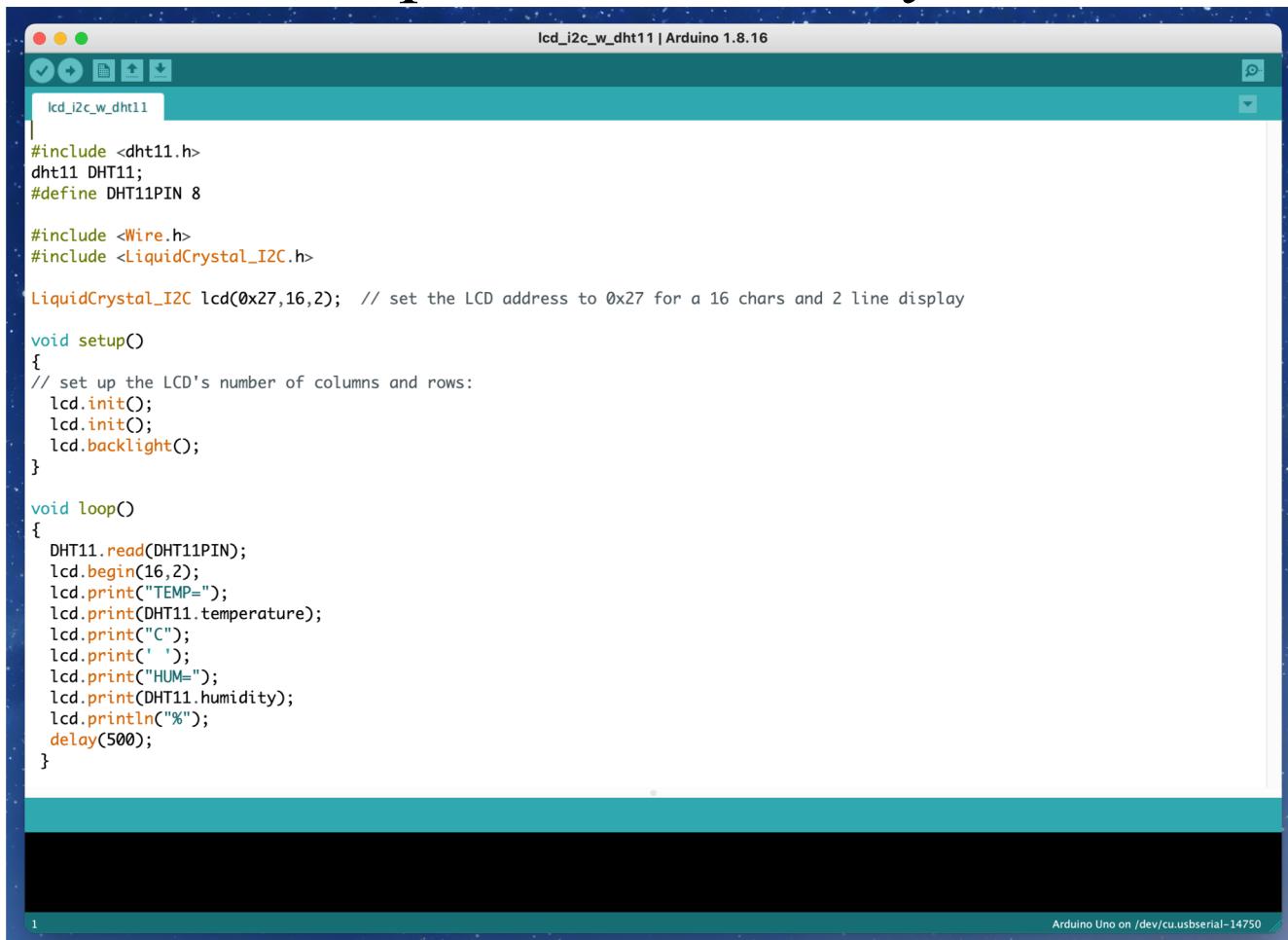
# LCD I2C with temperature humidity sensor



# LCD I2C with temperature humidity sensor



# LCD I2C with temperature humidity sensor - Code



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** lcd\_i2c\_w\_dht11 | Arduino 1.8.16
- Code Area:** The code is written in C++ and includes libraries for DHT11, Wire, and LiquidCrystal\_I2C. It initializes the LCD at address 0x27 with 16 columns and 2 rows. It then sets up the LCD's number of columns and rows, initializes it, and turns on the backlight. In the loop, it reads the DHT11 sensor, prints the temperature (in Celsius) and humidity (in percent) to the LCD, and delays for 500ms.

```
#include <dht11.h>
dht11 DHT11;
#define DHT11PIN 8

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

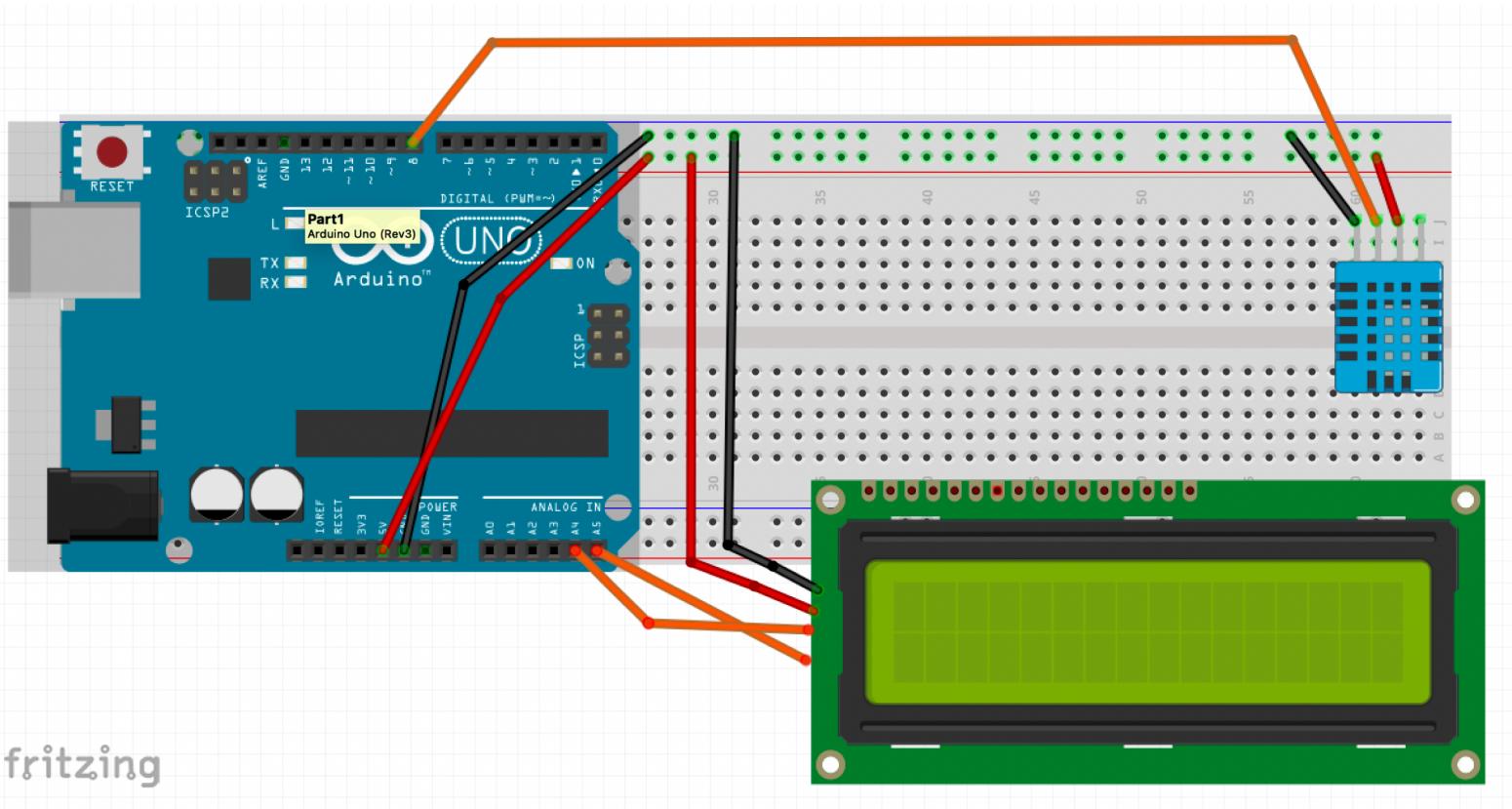
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup()
{
// set up the LCD's number of columns and rows:
lcd.init();
lcd.init();
lcd.backlight();
}

void loop()
{
DHT11.read(DHT11PIN);
lcd.begin(16,2);
lcd.print("TEMP=");
lcd.print(DHT11.temperature);
lcd.print("C");
lcd.print(" ");
lcd.print("HUM=");
lcd.print(DHT11.humidity);
lcd.println("%");
delay(500);
}
```

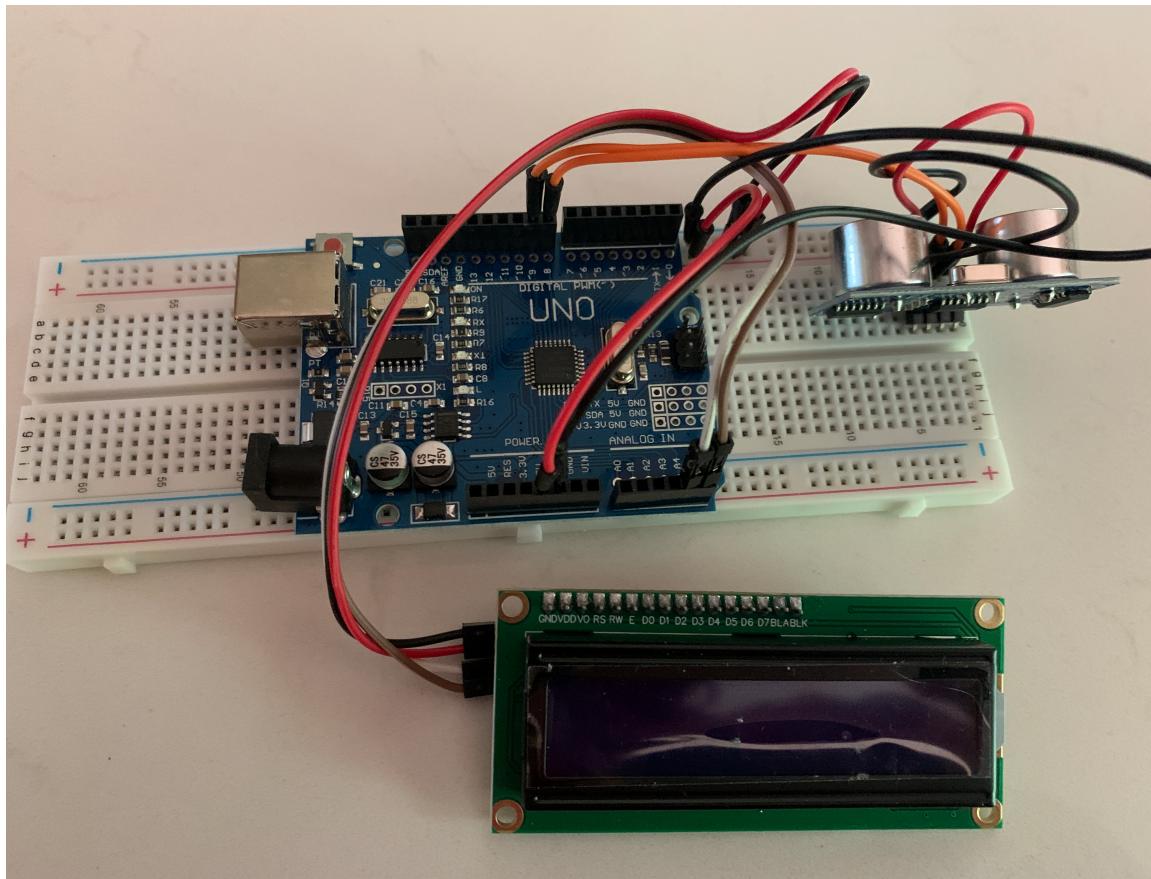
- Bottom Status Bar:** Shows "1" on the left and "Arduino Uno on /dev/cu.usbserial-14750" on the right.

# LCD I2C with ultrasonic sensor



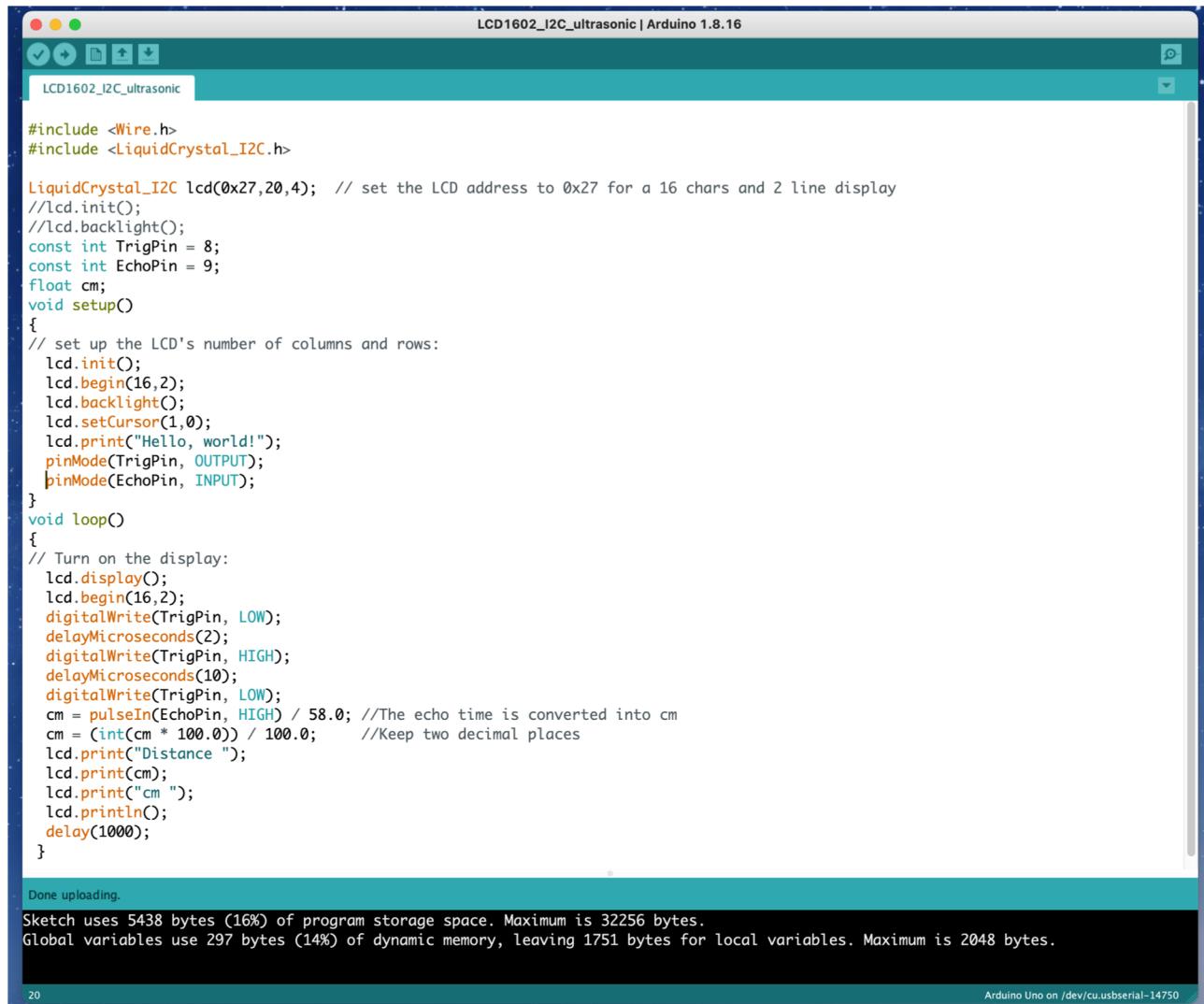
fritzing

# LCD I2C with ultrasonic sensor



Copyright 2023 Joseph E. Wilkes

# LCD I2C with ultrasonic sensor Code



The screenshot shows the Arduino IDE interface with the title bar "LCD1602\_I2C\_ultrasonic | Arduino 1.8.16". The code editor contains the following C++ code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16 chars and 2 line display
//lcd.init();
//lcd.backlight();
const int TrigPin = 8;
const int EchoPin = 9;
float cm;
void setup()
{
    // set up the LCD's number of columns and rows:
    lcd.init();
    lcd.begin(16,2);
    lcd.backlight();
    lcd.setCursor(1,0);
    lcd.print("Hello, world!");
    pinMode(TrigPin, OUTPUT);
    pinMode(EchoPin, INPUT);
}
void loop()
{
    // Turn on the display:
    lcd.display();
    lcd.begin(16,2);
    digitalWrite(TrigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(TrigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(TrigPin, LOW);
    cm = pulseIn(EchoPin, HIGH) / 58.0; //The echo time is converted into cm
    cm = (int(cm * 100.0)) / 100.0;    //Keep two decimal places
    lcd.print("Distance ");
    lcd.print(cm);
    lcd.print("cm ");
    lcd.println();
    delay(1000);
}

Done uploading.
```

At the bottom of the IDE window, status messages indicate:

- Sketch uses 5438 bytes (16%) of program storage space. Maximum is 32256 bytes.
- Global variables use 297 bytes (14%) of dynamic memory, leaving 1751 bytes for local variables. Maximum is 2048 bytes.

At the very bottom, it says "20" on the left and "Arduino Uno on /dev/cu.usbserial-14750" on the right.

Copyright 2023 Joseph E. Wilkes