PROJECT 5 & 6

Segment Counter & character Display in I2C LCD

Objective

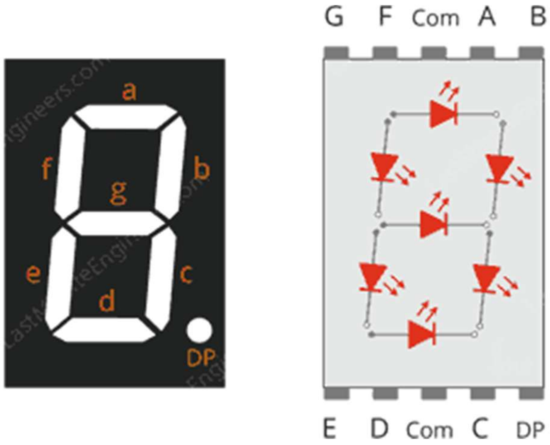
A 0 to 9 counter display using 7 Segment and Arduino UNO.

Required Equipment's

Arduino Uno, 7 Segment Display, 220ohm resistor's, 3 jumper Cables.

What is a 7 Segment display?

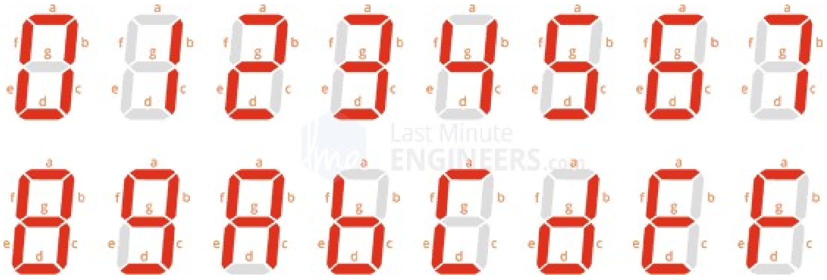
The 7-segment displays are really just seven LEDs lined up in a particular pattern. In this case, each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit (both Decimal and Hex) to be displayed. An additional 8th LED is sometimes used for indication of a decimal point.



How 7 Segment display?

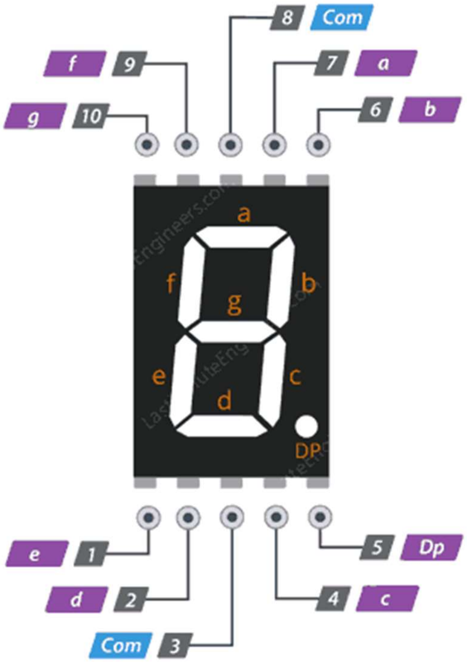
Each one of the seven LEDs in the display is given a positional segment with one of its connection pins being brought straight out of the rectangular plastic package. These individual LED pins are labeled from **A** through to **G** representing each individual LED. The other LED pins are connected together and wired to form a common pin.

To turn on and off a particular part of the display, set the appropriate pin HIGH or LOW just a regular LED work. So that some segments will be light and others will be dark allowing the desired character pattern of the number to be generated on the display. This then allows us to display each of the ten decimal digits 0 through to 9 on the same 7-segment display.



7 Segment display Pinout

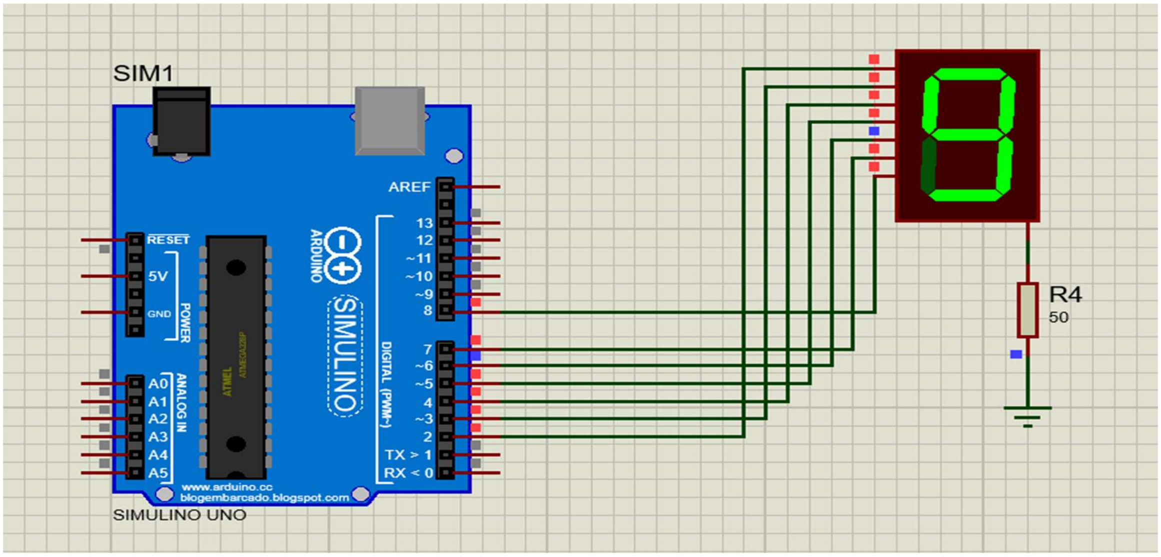
The pinout for the 7-segment display is as follows.



a-g & DP Out of 10, the 8 pins i.e. a, b, c, d, e, f, g and DP segment (decimal point) are connected to digital pins of Arduino. By controlling each LED on the segment connected, numbers can be displayed.
COM The pin 3 and 8 are internally connected to form a common pin. This pin should be connected to **GND (common cathode)** or **5V (common anode)** depending upon the type of the display.

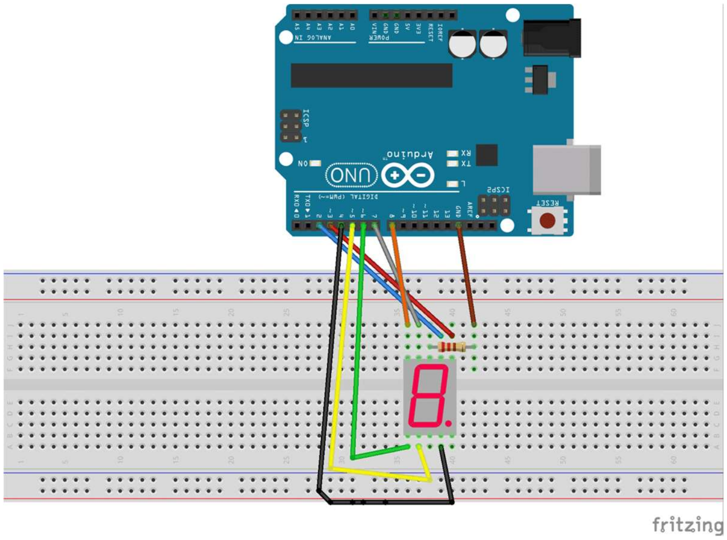
Simulation of 7-segment using Proteus

Navigate to the "**7seg_counter**" folder and open the 7seg_counter.pdsprj project file, then upload the program file **7seg_counter.ino.hex** to the Arduino and start the simulation process.



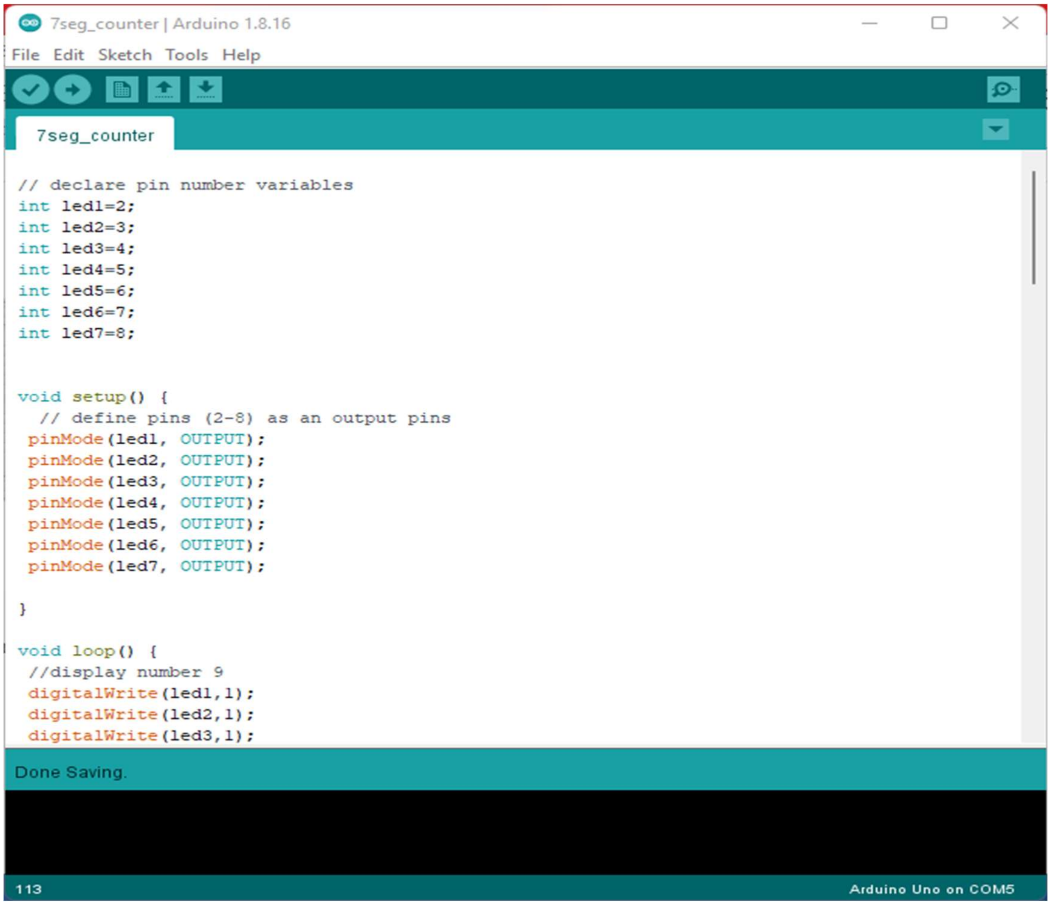
Implementation on Arduino

Use the required component's to achieve the following circuit.



Arduino code

Using the same steps in previous projects upload the Arduino code to the board.



```
// declare pin number variables
int led1=2;
int led2=3;
int led3=4;
int led4=5;
int led5=6;
int led6=7;
int led7=8;


void setup() {
  // define pins (2-8) as an output pins
 pinMode(led1, OUTPUT);
 pinMode(led2, OUTPUT);
 pinMode(led3, OUTPUT);
 pinMode(led4, OUTPUT);
 pinMode(led5, OUTPUT);
 pinMode(led6, OUTPUT);
 pinMode(led7, OUTPUT);

}

void loop() {
 //display number 9
 digitalWrite(led1,1);
 digitalWrite(led2,1);
 digitalWrite(led3,1);
```

Done Saving.

113 Arduino Uno on COM5

PROJECT 6: character Display in I2C LCD

Objective

Display character’s using I2C LCD and Arduino UNO.

Required Equipment’s

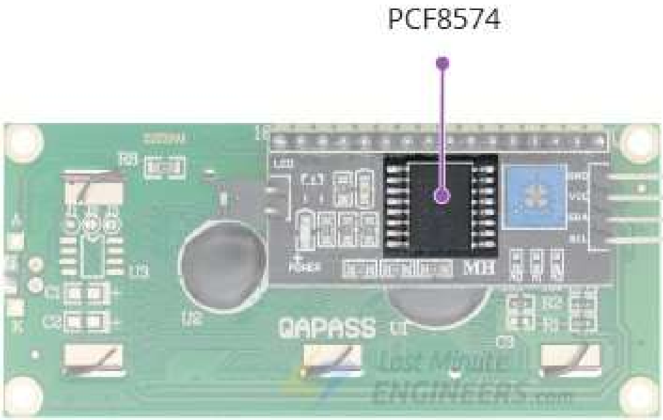
Arduino Uno, I2C LCD, 4 jumper Cables.

What is an I2C LCD display?

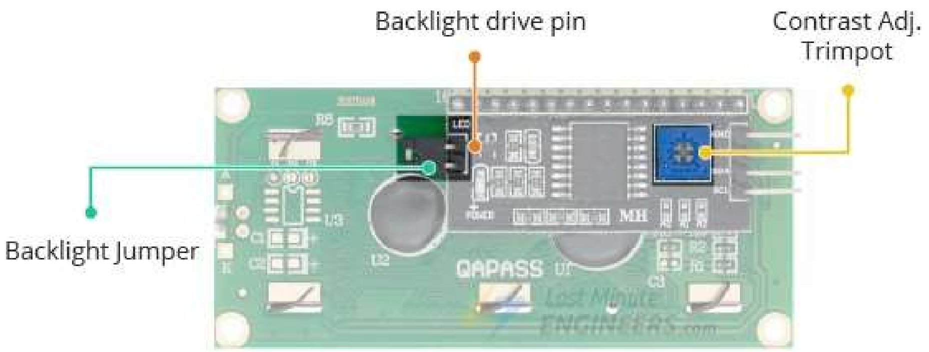
A typical I2C LCD display consists of a HD44780 based character LCD display and an I2C LCD adapter. These LCDs are ideal for displaying text/characters only. A 16×2 character LCD, for example, has an LED backlight and can display 32 ASCII characters in two rows with 16 characters on each row.

I2C LCD Adapter

The main component of the adapter is an 8-Bit I/O Expander chip – PCF8574. This chip converts the I2C data from an Arduino into the parallel data required by the LCD display.

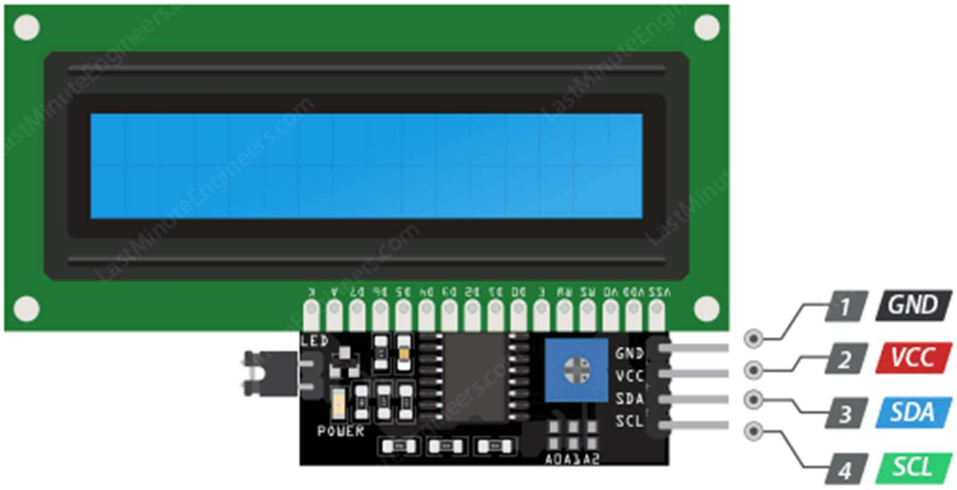


The board also comes with a small trim-pot to make fine adjustments to the contrast of the display. In addition, there is a jumper on the board that supplies power to the backlight. To control the intensity of the backlight, you can remove the jumper and apply an external voltage to the header pin that is marked as ‘LED’.



I2C LCD display Pinout

An I2C LCD has only 4 pins that interface it to the outside world. The connections are as follows:



GND is a ground pin and should be connected to the ground of Arduino.

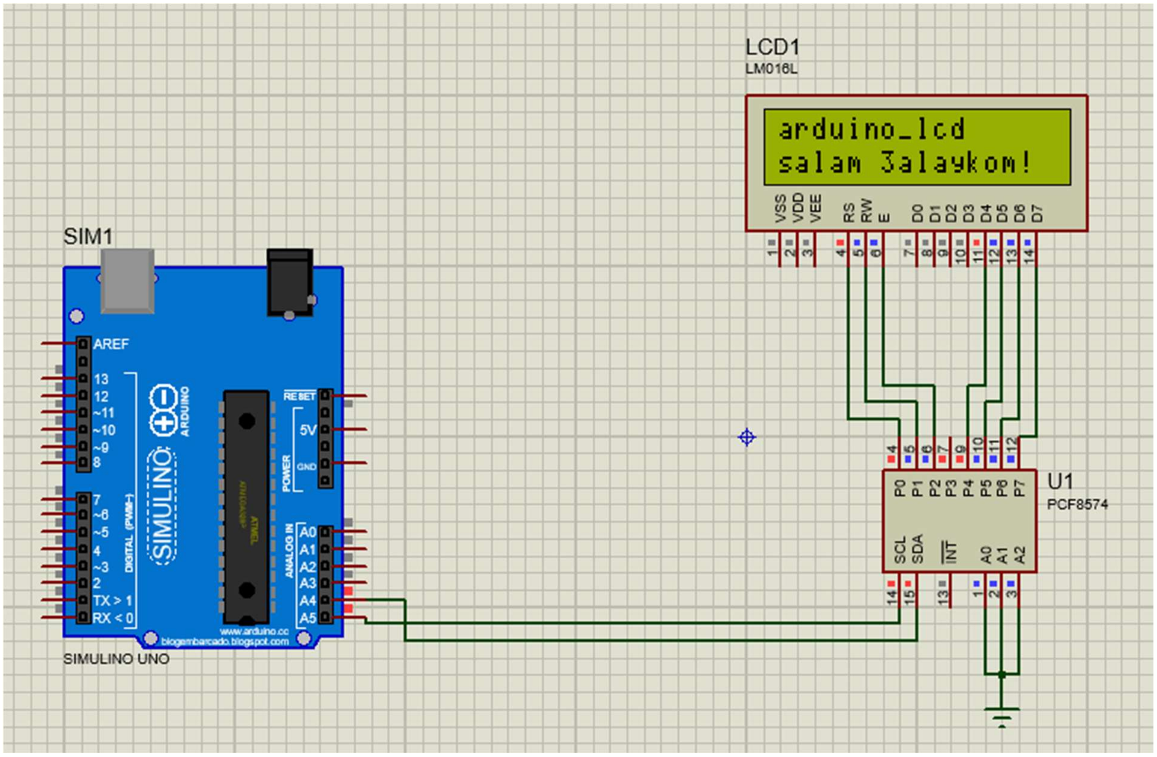
VCC supplies power to the module and the LCD. Connect it to the 5V output of the Arduino or a separate power supply.

SDA is a Serial Data pin. This line is used for both transmit and receive. Connect to the analog pin A4 on the Arduino.

SCL is a Serial Clock pin. This is a timing signal supplied by the Bus Master device. Connect to the analog pin A5 on the Arduino.

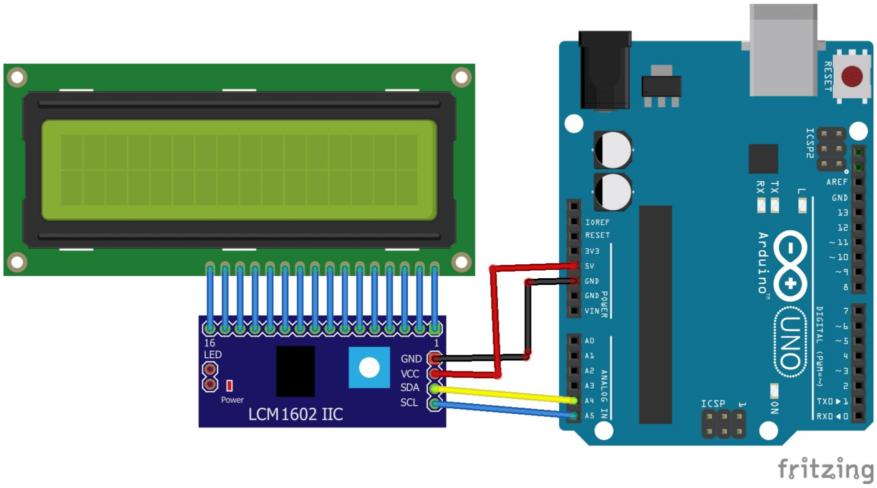
Simulation of I2C LCD Using Proteus

From the LCD folder open the lcd.pdsprj project file, and run the simulation.



Implementation using Arduino

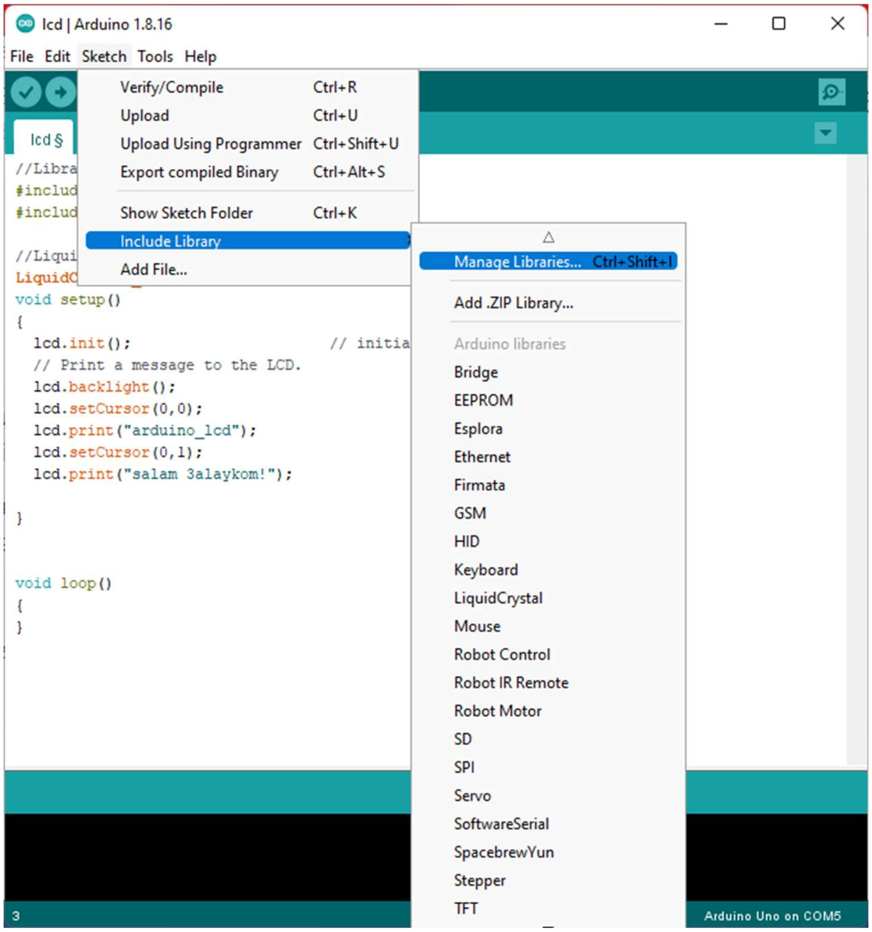
Connect the I2C LCD board to Arduino as follow



Arduino Code

In order to run the subsequent sketches it's necessary to install a library called **LiquidCrystal_I2C**. This library is an improved version of the **LiquidCrystal** library that comes packaged with your Arduino IDE.

To install the library navigate to the **Sketch > Include Library > Manage Libraries**… Wait for Library Manager to download libraries index and update list of installed libraries.



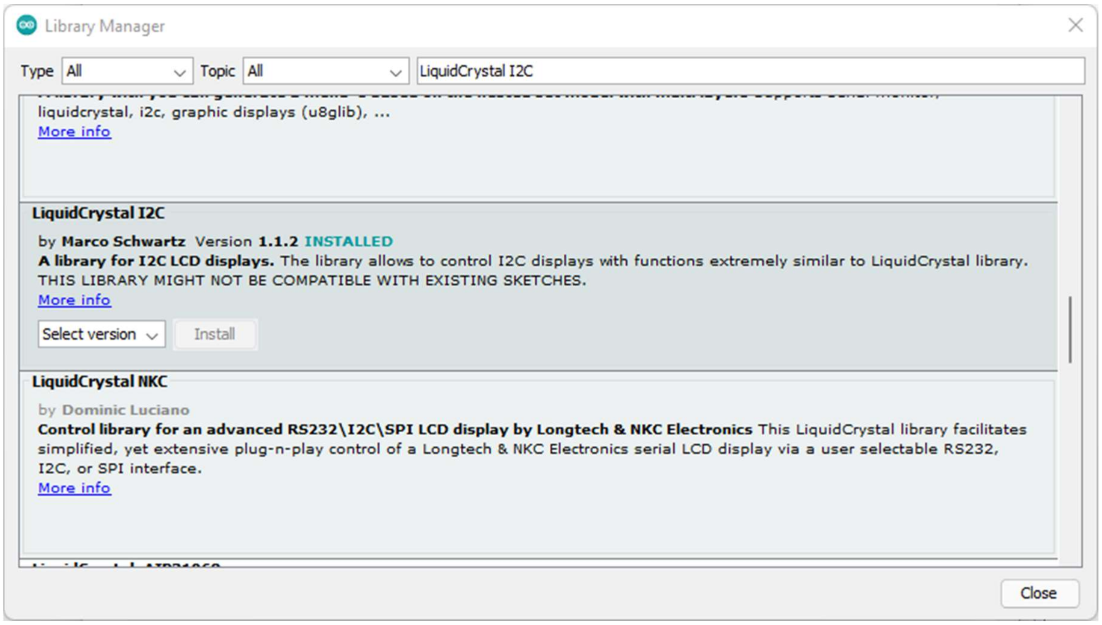
```
//Libra
#includ
#includ

//Liqui
LiquidC
void setup()
{
  lcd.init();                    // initia
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("arduino_lcd");
  lcd.setCursor(0,1);
  lcd.print("salam 3alaykom!");

}


void loop()
{
}
```

Filter your search by typing ‘**liquidcrystal**’. There should be a couple entries. Look for **LiquidCrystal I2C** library by **Marco Schawrts**. Click on that entry, and then select Install.



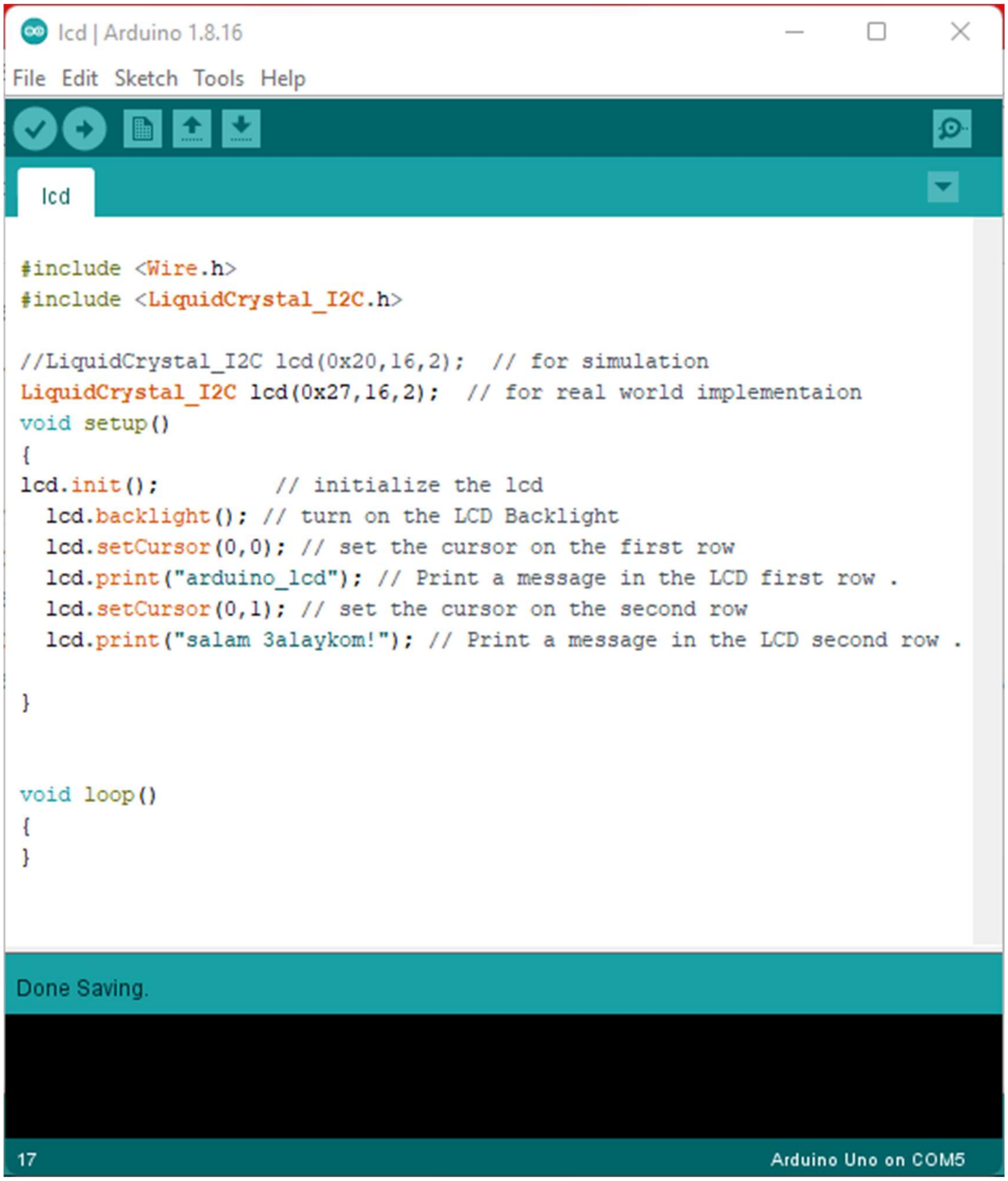
Search: LiquidCrystal I2C

liquidcrystal, i2c, graphic displays (u8glib), ...
More info

LiquidCrystal I2C
by **Marco Schwartz** Version **1.1.2** **INSTALLED**
A library for I2C LCD displays. The library allows to control I2C displays with functions extremely similar to LiquidCrystal library. THIS LIBRARY MIGHT NOT BE COMPATIBLE WITH EXISTING SKETCHES.
More info

LiquidCrystal NKC
by Dominic Luciano
Control library for an advanced RS232\I2C\SPI LCD display by Longtech & NKC Electronics This LiquidCrystal library facilitates simplified, yet extensive plug-n-play control of a Longtech & NKC Electronics serial LCD display via a user selectable RS232, I2C, or SPI interface.
More info

After that, open and compile the **lcd.ino** file then send it to the Arduino Board.



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

//LiquidCrystal_I2C lcd(0x20,16,2);  // for simulation
LiquidCrystal_I2C lcd(0x27,16,2);  // for real world implementaion
void setup()
{
lcd.init();         // initialize the lcd
  lcd.backlight(); // turn on the LCD Backlight
  lcd.setCursor(0,0); // set the cursor on the first row
  lcd.print("arduino_lcd"); // Print a message in the LCD first row .
  lcd.setCursor(0,1); // set the cursor on the second row
  lcd.print("salam 3alaykom!"); // Print a message in the LCD second row .

}


void loop()
{
}
```

Done Saving.