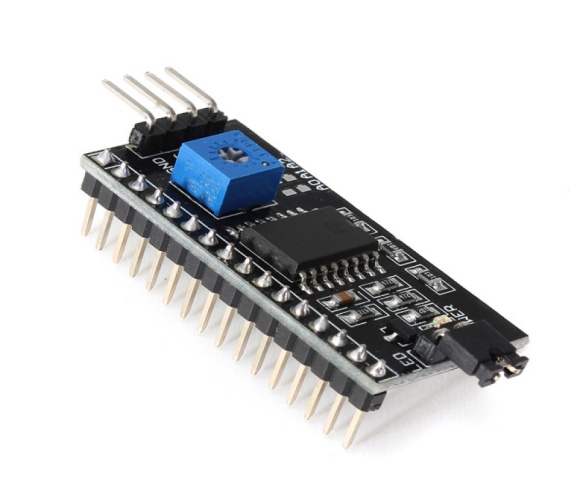
**Interfacing of LCD to Arduino**

A Liquid crystal display is a form of visual display used in electronic devices, in which a layer of a liquid crystal is sandwiched between two transparent electrodes.We are using basic-generic 16x2 liquid crystal display (i2c bus module version) with an arduino.

Each I2C bus consists of two signals: SCL and SDA. SCL is the clock signal, and SDA is the data signal. The clock signal is always generated by the current bus master; some slave devices may force the clock low at times to delay the master sending more data (or to require more time to prepare data before the master attempts to clock it out). This is called “clock stretching”





I2C Module has a inbuilt PCF8574 I2C chip that converts I2C serial data to parallel data for the LCD display.

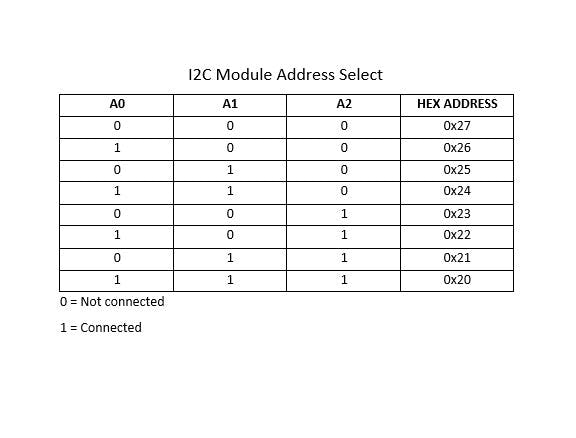
These modules are currently supplied with a default I2C address of either 0x27 or 0x3F. To determine version , check the black I2C adaptor board on the underside of the module. If there a 3 sets of pads labelled A0, A1, & A2 then the default address will be 0x3F. If there are no pads the default address will be 0x27.

The module has a contrast adjustment pot on the underside of the display. This may require adjusting for the screen to display text correctly. It has total of 20 male pins. 16 pins are faced to rear side and 4 pins faced towards front side. The 16 pins for connect to 16x2 LCD and the 2 pins out of 4 pins are SDA and SCL. SDA is the serial data pin and SCL is the clock pin. The rest 2 pins for power supply (Vcc and ground).There is a POT on the I2C Module. We can control the contrast of the LCD display by rotating this POT. And there is a jumper fixed on the module. When we remove the jumper, the backlight of the LCD display will go OFF.

**Address of I2C LCD**

Before starting we need to know about addressing of I2C devices. Every device which can attached to MCU have an address. We need to know this address for communicate with that particular device.

You can see three solder pads on the I2C module. which is labeled as A0, A1 and A2. This is Address selectors. ie, each solder pads have one upper potion and a one lower potion. if, there is a connection between upper potion with lower connection it is called "**Connected**" otherwise it is called "**Not connected**". When A0, A1, A2 are in "Not Connected" condition ( A0 = 0, A1 = 0, A2 = 0) the address would be 0x27. In default the A0, A1, A2 are in "Not connected" condition. And some time default address is 0x3F. There is no need to change the address of the I2C module when we use only one LCD. But when we use more than one LCD, need to change the address. Because two or more different device can't communicate with the same address. For more address see the table given below.



**Features:-**

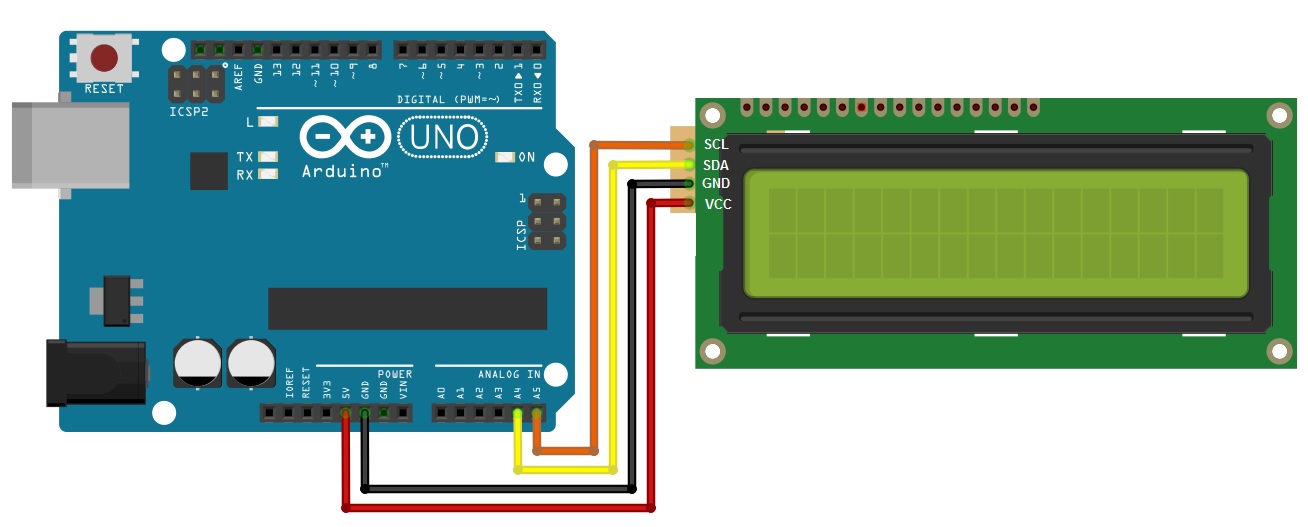
* Operating Voltage: 5V.
* Backlight and Contrast is adjusted by potentiometer.
* Serial I2C control of LCD display using PCF8574 .
* Come with 2 IIC interface, which can be connected by Dupont Line or IIC dedicated cable.
* Compatible for 16x2 LCD.
* This is another great IIC/I2C/TWI/SPI Serial Interface.
* With this I2C interface module, you will be able to realize data display via only 2 wires.



### I2C LCD Connections

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

**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. We are using these lines instead of A4,A5.**



**Installing the LiquidCrystal\_I2C Arduino library**

We will be using the **LiquidCrystal\_I2C** library. This library has many built-in functions that make programming the LCD quite easy.

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.

**Functions used of LiquidCrystal\_I2C library**

## begin()

### Description

Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display. begin() needs to be called before any other LCD library commands.

### Syntax

lcd.begin(cols, rows)

### clear()

#### Description

Clears the LCD screen and positions the cursor in the upper-left corner (first row and first column) of the display.

#### Syntax

lcd.clear()

## setCursor()

#### Description

Position the LCD cursor; that is, set the location at which subsequent text written to the LCD will be displayed.

#### Syntax

lcd.setCursor(col, row)

## print()

### *Description*

Prints text to the LCD.

### *Syntax*

lcd.print(data)

lcd.print(data, BASE)

**LiquidCrystal\_I2c is configured using command:**

**LiquidCrystal\_I2C lcd(addr,en,rw,rs,d4,d5,d6,d7,bl,blpol)**

LiquidCrystal\_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

Pins of I2C module selected are en=2,rw=1,rs=0,data=d4,d5,d6,d7, contrast=3 contrast polarity positive.

**Procedure**

1. Download and Install the Library **LiquidCrystal\_I2c** and include **wire.h library**.
2. Write the code for the Arduino UNO using Arduino IDE.
3. Connect the components as the circuit attached.
4. Edit the code as need to change the text on LCD.
5. Upload the code on Arduino UNO.
6. Now check the text on LCD and make required corrections to fit text on LCD.

**Code**

*/\*-----( Import library )-----\*/*

*#include <Wire.h>*

*#include <LiquidCrystal\_I2C.h> //Install library*

*//i2c pins*

*LiquidCrystal\_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //*

*void setup()*

*{*

*//WE define our LCD 16 columns and 2 rows*

*lcd.begin(16,2);*

*lcd.backlight();//Power on the back light*

*//lcd.backlight(); Power off the back light*

*}*

*void loop()*

*{*

*//College Name*

*lcd.setCursor(0, 0); //we start writing from the first row first column*

*lcd.print("THAKUR COLLEGE");*

*lcd.setCursor(0, 1);//we start writing from the second row first column*

*lcd.print("OF ENGG. & TECH");*

*delay(2000);*

*//Project Name*

*lcd.clear();//Clean the screen*

*lcd.setCursor(0, 0);*

*lcd.print("GSM & GPS BASED ");*

*lcd.setCursor(0, 1);*

*lcd.print("AUTOMATIC VEHICLE");*

*delay(2000);*

*lcd.clear();//Clean the screen*

*lcd.setCursor(0, 0);*

*lcd.print("ACCIDENT DETECTN");*

*lcd.setCursor(0, 1);*

*lcd.print("& ALERTING SYS. ");*

*delay(2000);*

*//STUDENT NAME*

*lcd.clear();//Clean the screen*

*lcd.setCursor(0, 0);*

*lcd.print(" PROJECT BY ");*

*lcd.setCursor(0, 1);*

*lcd.print("AAKASH MOHIKAR ");*

*delay(2000);*

*lcd.clear();//Clean the screen*

*lcd.setCursor(0, 0);*

*lcd.print("AJAYKUMAR MAURYA");*

*lcd.setCursor(0, 1);*

*lcd.print("HARISH PATEL ");*

*delay(2000);*

*}*

*LCD requires 7 lines to display data .So to reduce the number of i/o lines from Arduino we are using i2c protocol-based module which requires only two lines sda and scl. Now in this module address is based on the 3 pins A0,A1,A2.*

