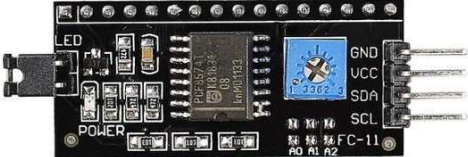


LCD 1602 (serial)

Content

- LCD 1602 (serial)..... 1
- How it works..... 2
 - I2C Communication 2
 - I2C Serial Interface Adapter 3
 - Address of I2C LCD 3
- Step - 1..... 4
 - I2C Scanner Code..... 4
- Step - 2..... 5
 - Connection 6
- Step - 3..... 6
- Step - 4..... 6
- Step - 5..... 6
- Step - 6..... 7
- Step - 7..... 7
- Step - 8..... 7
- Step - 9..... 7
- Source..... 8

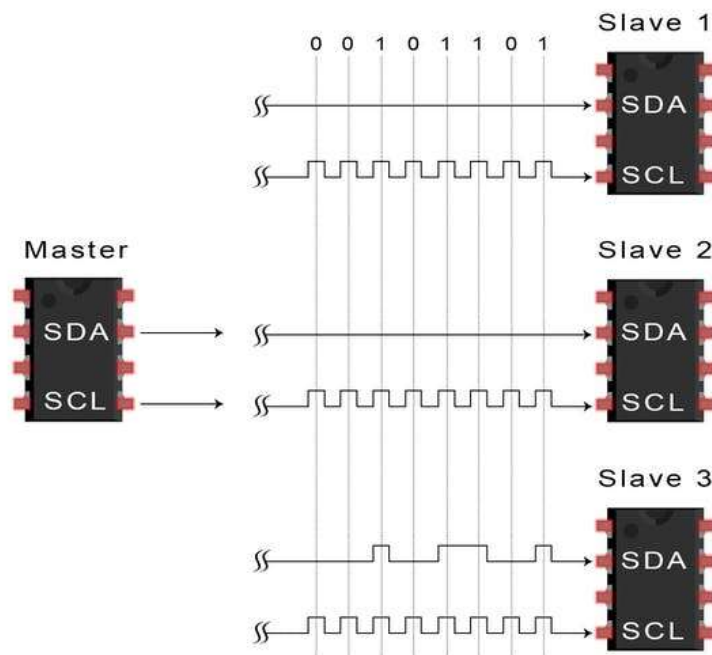


How it works

Here I use the same 16X2 LCD in my previous article. But additionally, attach a I2C Module to the 16x2 LCD. It works as an intermediary between the LCD and MCU (here Arduino).

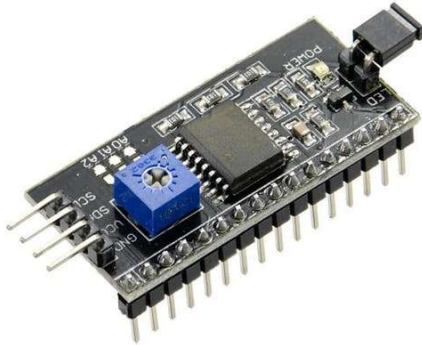
Before starting you must know about I2C Serial Interface Adapter (I2C Module), I2C communication, and Address of I2C LCD

I2C Communication



I2C is short for Inter-IC. And it is a type of BUS. This is designed by Philips semiconductors. I2C is a synchronous, multi slave, multi master packet switched, single-ended serial bus. ie. multiple chips can be connected to the same bus. I2C uses only two bidirectional open collector or open drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL), pulled up with resistors. Typical voltages used are +5 V or +3.3 V, although systems with other voltages are permitted. For more about I2C protocol click [here](#).

I2C Serial Interface Adapter



It is also known as I2C Module. 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 portion and a one lower portion. if, there is a connection between upper portion 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.

I2C Module Address Select

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

0 = Not connected

1 = Connected

Step - 1

Skip this Step - 1 & Step -2 if you already know the address of the LCD!

In some cases, A0, A1, A2 are "Not connected" state, but the address is not 0x27. We can't communicate with this address. So, we need to find the original address of that device. For that we need to run the Arduino with "I2C Scanner" code.

I2C Scanner Code

I2C Scanner code is used for find the number of I2C devices and address of I2C devices. First add the header file for include "Wire.h" library. Then in setup part, begin the "Wire" library by "Wire.begin()". Then begin the serial monitor as the baud rate of 9600 by "Serial.begin()". Next in loop part, define two variables with the datatype "byte" named "error" and "address". Then define another variable with the "Integer (int)" datatype named as "Devices". And set initial value as 0. Next start a for loop with minimum value of 1 and maximum of 127. "address" used as loop variable. Next input the address to wire with the function "Wire.beginTransmission()". The i2c_scanner uses the return value of the "Write.endTransmission()" to see if a device did acknowledge to the address. This return value store the value to the variable "error". The return value become 0, if a device acknowledges to the address. Otherwise, the return value become 4. Next use a if. And the condition is "error==0". Then print the particular address to the serial monitor only if the address<16. Here we print the address in Hexadecimal. The printing instruction is "Serial.print(address, HEX)". And count the Device.

The complete I2C Scanner code is given below:

```
#include <Wire.h>
void setup()
{
```

```

Wire.begin();
Serial.begin(9600);
Serial.println("\nI2C Scanner");
}
void loop()
{
byte error, address;
int Devices;
Serial.println("Scanning...");
Devices = 0;
for(address = 1; address < 127; 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(" !");
Devices++;
}
else if (error==4)
{
Serial.print("Unknown error at address 0x");
if (address<16)
Serial.print("0");
Serial.println(address,HEX);
}
}
if (Devices == 0)
Serial.println("No I2C devices found\n");
else
Serial.println("done\n");
delay(5000);
}

```

Step - 2

If you have a I2C LCD please skip this step. But if you have a 16x2 LCD and a I2C Module see the step to connect this module to LCD.

Connection

First solder the I2C Module. There is no label on the I2C Module for connecting to 16x2 LCD. So solder it with the help of the image given below



After soldering connect the I2C Module to Arduino Uno:

- Analog Pin 4 - SDA
- Analog pin 5 - SCL
- 5V - Vcc
- GND - GND

Connect the Arduino to computer.

Next open Serial monitor from the icon on top right corner of Arduino IDE. And set the baud rate as 9600. Please ensure the correct port. Then you can see the address of LCD in serial monitor like shown below. Finally we find the address of the I2C LCD. That is 0x27

Step - 3

Next, I am going to display a text on our I2C LCD. Before that need to add a library to Arduino IDE. Go to the [link](#) and download the library Arduino-LiquidCrystal-I2C-library. Then open Arduino IDE and go to Sketch>Include Library> Add.ZIP Library. Next select the downloaded ZIP file and click open.

Step - 4

Next create a sketch. First, I include the header "Wire.h". This library helps to communicate with I2C device. Then include "LiquidCrystal_I2C.h" for better communication with display.

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

Step - 5

Next set the address, number of column and number of rows using the function "LiquidCrystal_I2C lcd()". The address is 0x27 (discovered using the I2C Scanner Code). Number of columns is 16 and number of rows is 2. After this, we can call the display using "lcd". You can also use multiple I2C LCDs with Arduino Uno. But set different addresses and variable for each display.

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

Step - 6

Next code the setup part. First initialize the display using the function "lcd.begin()". If you have another display initialize them.

```
lcd.begin();
```

Step - 7

Next turn on the backlight using the function "lcd.backlight()".

```
lcd.backlight();
```

Step - 8

Next clear the lcd using the instruction "lcd.clear()". Then set the cursor to the position (4, 0).

```
lcd.clear();  
lcd.setCursor(4,0);
```

Step - 9

Now the LCD is ready to print. The cursor is at 4th column(count from 0), and 0th row(count from 0). Then print the Message "Hackster" by the function "lcd.print()".

```
lcd.print("Hackster");
```

Then leave the loop part as empty.

The programming is completed. Upload the sketch to Arduino and see the message on LCD:

```
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>  
LiquidCrystal_I2C lcd(0x27, 16, 2);  
void setup()  
{  
  lcd.begin();  
  lcd.backlight();  
  lcd.clear();  
  lcd.setCursor(4,0);  
  lcd.print("Jajajajaja");  
}  
void loop()  
{  
}
```

Source

Van <<https://create.arduino.cc/projecthub/akshayjoseph666/interface-i2c-16x2-lcd-with-arduino-uno-just-4-wires-273b24>>