



Hey there! Thanks for dropping by alselectro! Take a look around and grab the [RSS feed](#) to stay updated. See you around!

8051 ARDUINO BlueTooth/WIFI CCTV Electronics ESP32 ESP8266 Friendly ARM Mini2440 GPS GSM GSM,GPS
 IOT LPC 2148 MicroControllers MICROPYTHON NEXTION HMI PIC MicroController platform io Port Forwarding
 QUADCOPTER Raspberry PI Robotics STM32 ARM CORTEX M3 Ubuntu Linux Uncategorized Windows XBEE Radios

ESP8266 WEMOS D1 With I2C Serial LCD

Filed under: [ESP8266](#) — 1 Comment

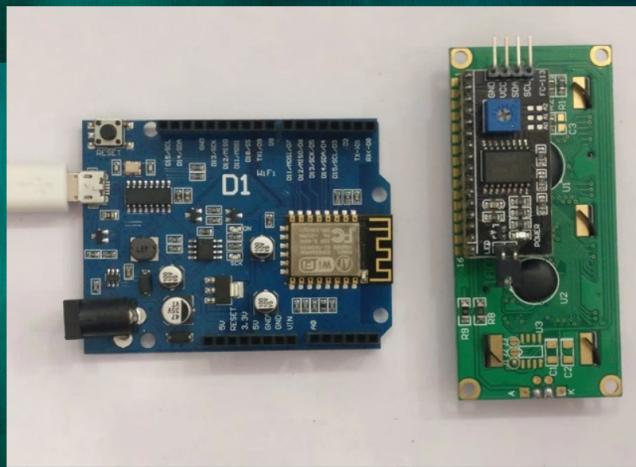
April 16, 2018

WemosD1 R1 board is now the most preferred IOT WIFI Board .As the hardware resembles Arduino , it is most sought out board for any IOT project , than NODEMCU.

To add a display I prefer to use the SERIAL LCD WITH I2C BACKPACK .

The serial LCD is built on PCF8574 & works on I2C Bus ,thus requires only 2 wires SCL & SDA apart from the power pins.

The LCD works on 5V & it can be sourced from D1 board itself.



For more detailed information on this Serial LCD you can read my previous blog here :

<https://alselectro.wordpress.com/2016/05/12/serial-lcd-i2c-module-pcf8574/>

The connection between D1 board & LCD is simple .

Wemos D1 LCD

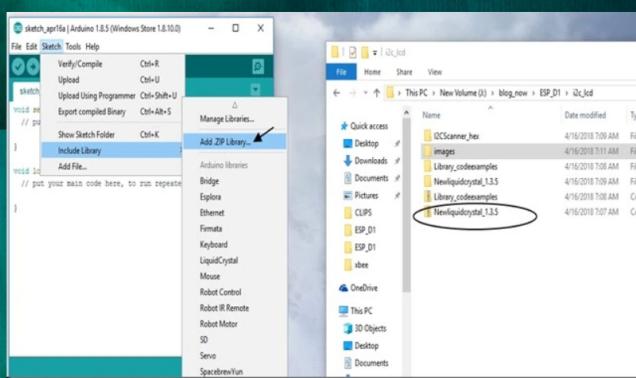
5v	— Vcc
Gnd	— Gnd
SCL D15 (GPIO5)	— SCL
SDA D14 (GPIO4)	— SDA

To start with download the latest Library from this link :

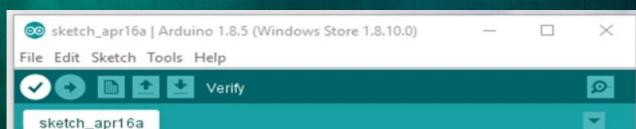
<https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads>

From Arduino IDE Sketch -> Include library -> Add ZIP Library

Browse to the location of the downloaded ZIP file & select it



Now the Library for Serial LCD is installed.



Blog Stats

2,663,267 hits

Follow Blog via Email

Enter your email address to follow this blog and receive notifications of new posts by email.

Enter your email address

Follow

Recent Posts

Contactless Thermometer project using MLX 90614 + Nano + Ultrasonic +OLED
 7" HDMI-TOUCH 1024 X 600 Display for Raspberry Pi
 Getting Started with Programming PIC 12F675
 ATTINY 85-AVR PROGRAMMING With WINAVR
 Automatic Sanitizer Dispenser & Tap Water Control
 Sensor Data on CAN BUS-Arduino with CAN2515
 Home Automation with EASYESP & DOMOTICZ
 ESP EASY with DHT11 & Domoticz
 ESP-Easy LED On Off from DOMOTICZ
 ESP EASY IOT OS ON NODEMCU
 Nextion display with DHT11
 Nextion Analog Read value & Slider Tool
 Nextion Display- LED On Off with Dual State Button & Image
 ESP32 MicroPython ULTRASONIC Sensor
 ESP32 MicroPython DHT11 Sensor
 ESP32 MicroPython Interfacing OLED Display
 ESP32 MICROPYTHON ADC with Analog Sensors
 ESP32 MICROPYTHON PWM & SERVO CONTROL
 MicroPython ESP32 Station -AutoConnect to WiFi AccessPoint
 MICROPYTHON ON ESP32

Archives

Select Month

Address



Saravana Electronics
 Visualizar mapa ampliado
 N DAPUR
 10, Edayar Street,
 Coimbatore,
 India

```

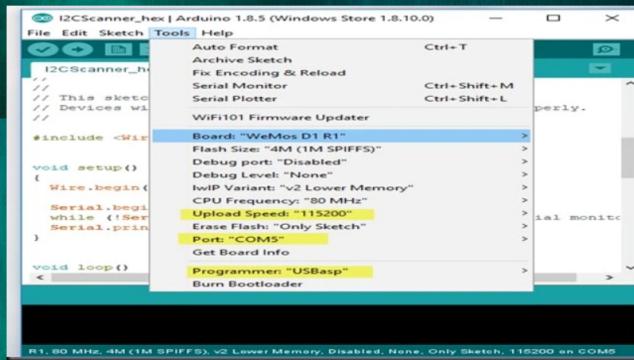
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

Library added to your libraries. Check "Include library" menu

```

Under Tools select the board as WEMOS D1 R1 , upload speed as 115200, port is the one allotted in Device manager & programmer USBASP.



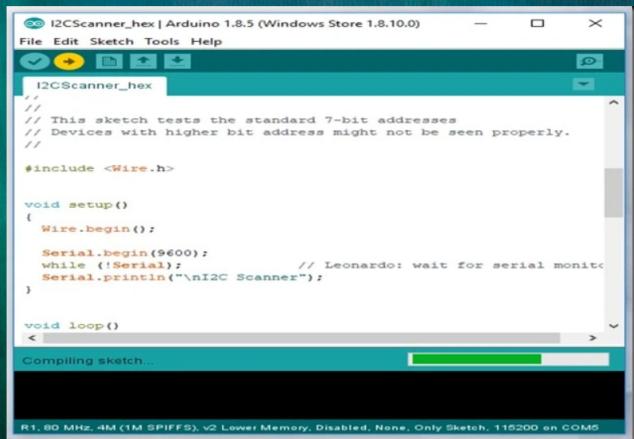
Read this previous blog post of getting started with D1 board, on how to install Arduino core using Boards Manager

<https://alselectro.wordpress.com/2018/04/14/wifi-esp8266-development-board-wemos-d1/>

Now we need to know the I2C Address of the LCD board connected to D1.

For this upload the following Scanner sketch.

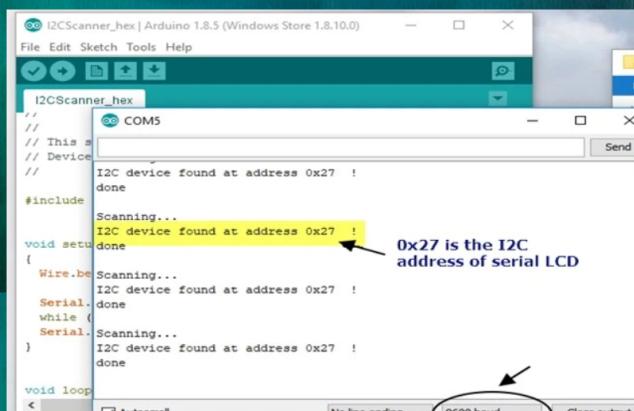
http://www.alselectro.com/files/Library_codeexamples.zip



Once uploaded, Open the serial monitor with 9600 baud setting.

The I2C HEX Address of the device is displayed .Here it is 0x27 .

Note that this address varies according to the manufacturer and the jumper solder on A0 A1 A2 pins on the backpack pcb.





Now open the first example lcd_test.

In this example 2 header files are used

One is the **Wire.h**, which is the I2C library that is built in already with IDE.

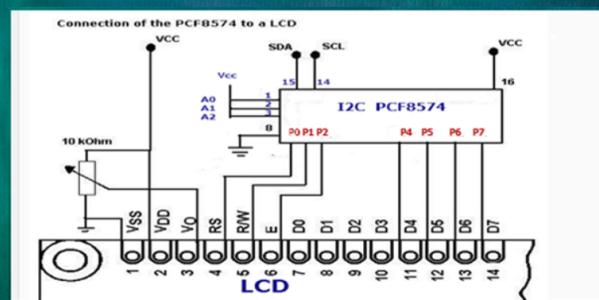
Second is the **LiquidCrystal_I2C.h** which we installed initially.

Then we create an instance `lcd` with parameters I2C Address , EN , RW , RS , D4,D5,D6,D7

I2C address is the one we got from the scanner code -0x27

EN,RW,RS are the pins of I2C board connected to LCD Enable ,Read/Write & Register Select respectively.

D4 to D7 are Data pins



`lcd.begin(16,2)` initializes the 16 x 2 LCD

Then we set the backlight ON for the LCD

`lcd.home()` moves the cursor to Home position

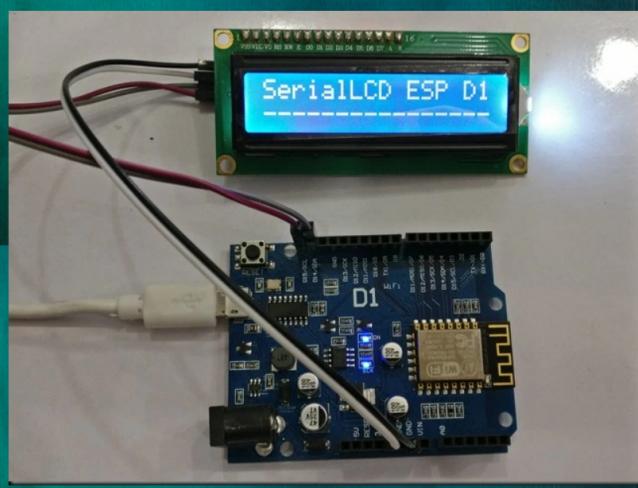
`lcd.print("...")` prints the message on 1st row.

`lcd.setCursor(0,1)` moves the cursor to second line.

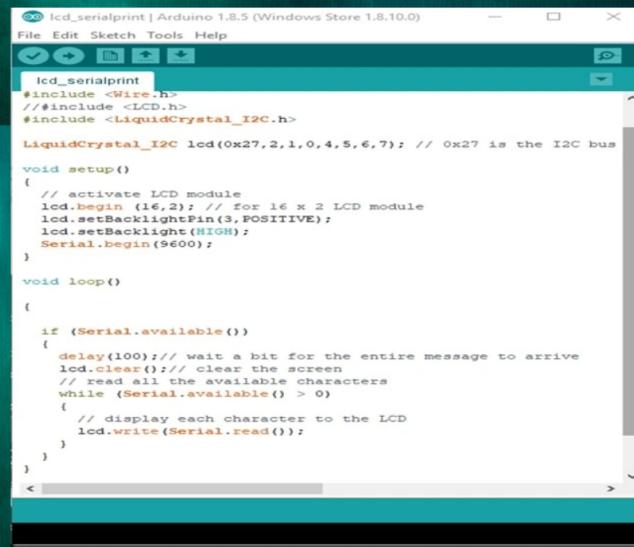
A screenshot of the Arduino IDE showing the code for "lcd_test". The code includes #include <Wire.h> and #include <LiquidCrystal_I2C.h>. It defines an I2C address of 0x27 and creates an instance of the LiquidCrystal_I2C library named "lcd". The setup() function initializes the LCD and sets the backlight to HIGH. The loop() function prints "SerialLCD ESP D1" to both lines of the LCD. At the bottom, it shows the upload progress bar and the message "Uploading...".

Upload the code to see the result on LCD.

The Contrast to LCD can be adjusted by the preset on the back.



Here is the second example which prints the message typed on the Serial monitor.



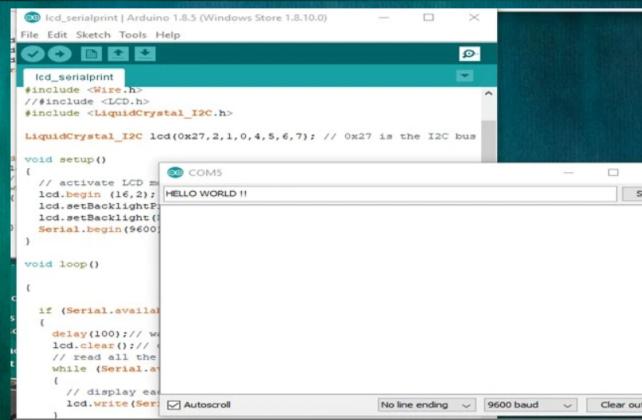
```
lcd_serialprint
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 2, 0, 4, 5, 6, 7); // 0x27 is the I2C bus

void setup()
{
    // activate LCD module
    lcd.begin (16,2); // for 16 x 2 LCD module
    lcd.setBacklightPin(3, POSITIVE);
    lcd.setBacklight(HIGH);
    Serial.begin(9600);
}

void loop()
{

    if (Serial.available())
    {
        delay(100); // wait a bit for the entire message to arrive
        lcd.clear(); // clear the screen
        // read all the available characters
        while (Serial.available() > 0)
        {
            // display each character to the LCD
            lcd.write(Serial.read());
        }
    }
}
```



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

LiquidCrystal_I2C lcd(0x27, 2, 0, 4, 5, 6, 7); // 0x27 is the I2C bus

void setup()
{
    // activate LCD module
    lcd.begin (16,2);
    lcd.setBacklightPin(3, POSITIVE);
    lcd.setBacklight(HIGH);
    Serial.begin(9600);
}

void loop()
{

    if (Serial.available())
    {
        delay(100); // wait a bit for the entire message to arrive
        lcd.clear(); // clear the screen
        // read all the available characters
        while (Serial.available() > 0)
        {
            // display each character to the LCD
            lcd.write(Serial.read());
        }
    }
}
```

The Serial Monitor window shows the text "HELLO WORLD!!" displayed on the screen.

VIDEO :



Share this:



Be the first to like this.

Related

Serial LCD I2C Module-
PCF8574
In "ARDUINO"

WiFi ESP8266 Development
Board WEMOS D1
In "ESP8266"

Sensor Data on CAN BUS-
Arduino with CAN2515
In "ARDUINO"

1 Comment:**Ben Wolodko**
July 18, 2018 at 8:13 pm

How is it possible to connect a 3.3v I2C with pullups to 3.3v (the ESP8266) to a 5 volt device (the LCD back pack)with 4.7k pullups to 5.0 volt? should you be using a level shifter?

[Reply](#)**Leave a Reply**

Enter your comment here...

[« WiFi ESP8266 Development Board WEMOS D1](#)[GPS Tutorial-Sending GPS Coordinates over GSM »](#)**Friends & links**

- [Discuss](#)
- [Get Inspired](#)
- [Get Polling](#)
- [Get Support](#)
- [Learn WordPress.com](#)
- [WordPress Planet](#)
- [WordPress.com News](#)

Pages

- [About](#)

Blogroll

- [Discuss](#)
- [Get Inspired](#)
- [Get Polling](#)
- [Get Support](#)
- [Learn WordPress.com](#)
- [WordPress Planet](#)
- [WordPress.com News](#)

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.
To find out more, including how to control cookies, see here: [Cookie Policy](#)

[Close and accept](#)