**EXPERIMENT-5**

**Sketch to interface LCD interfacing with ESP32 in Arduino IDE**

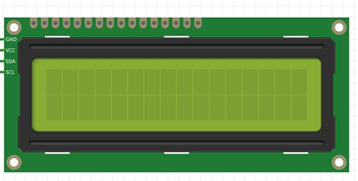
(USING A 16×2 I2C LCD DISPLAY WITH ESP32)

**COMPONENTS REQUIRED :**

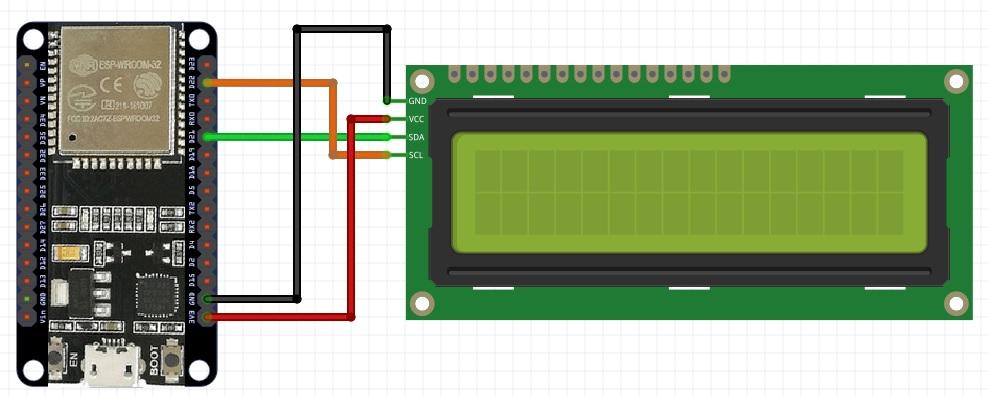
The following components are required,

1. ESP32 DevKit Development Board
2. 16×2 Character LCD module
3. Breadboard
4. Connecting Wires
5. Micro-USB Cable

We will use **16×2 LCD** display. It uses only four pins unlike the other versions of the display that requires at least 7 pins connected to the microcontroller board.



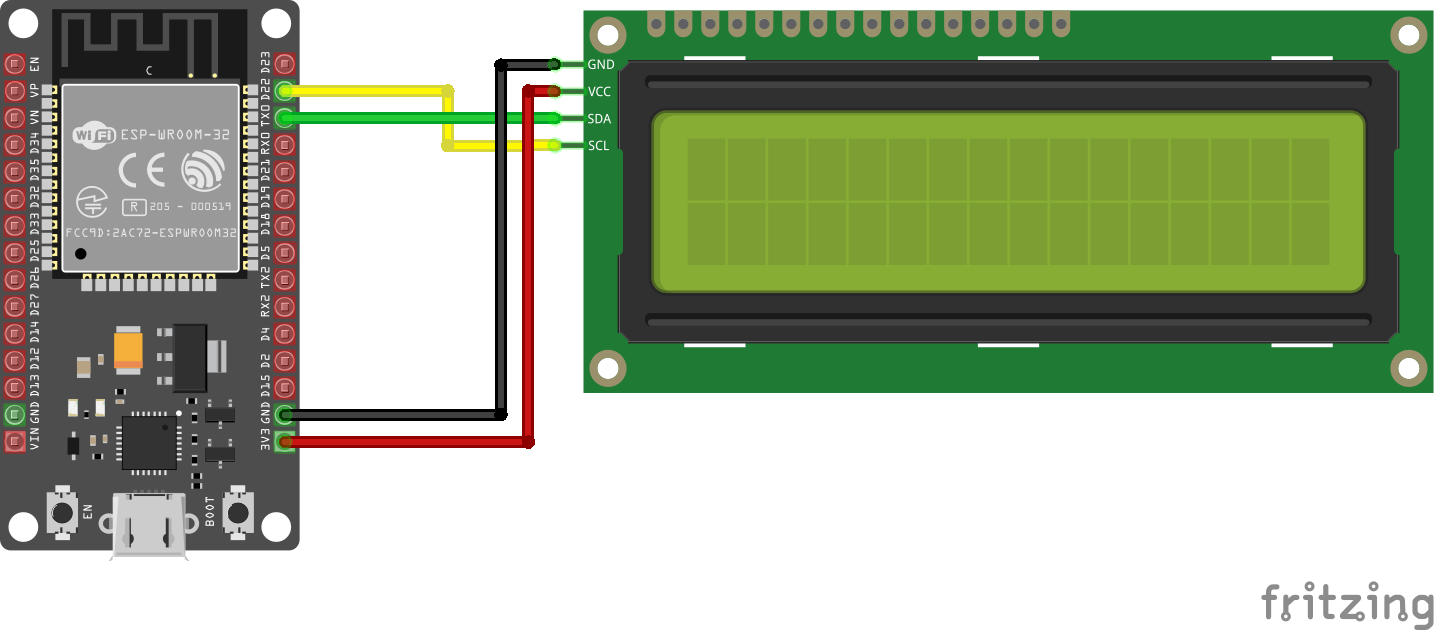
**DOIT ESP32 DevKit V1 16X2 LCD MODULE**



The breadboard requirement is optional as you can choose to connect the LCD directly to the DOIT devkit board using jumper wires.

**SCHEMATICS**

Connect the pins of the LCD module to the I2C pins of the DevKit. Connect the components as shown below. (created in fritzing)



In this circuit,

1. GPIO22 is default SCL pin, and GPIO21 is default SDA pin for I2C communication.
2. Connect GPIO22 with SCL pin of LCD and GPIO21 with SDA pin of liquid crystal display.
3. Connect GND to GND and 3.3v of ESP32 to VCC of LCD module.

A pin map showing how the components are connected is shown below

**LCD Module – ESP32**

**GND - GND**

**VCC - 3.3v/Vin**

**SDA - D21(GPIO)**

**SCL – D22 (GPIO)**



**Display Text on the LCD**

Displaying static text on the LCD. Select the characters to be displayed on the screen, and then send the message to the display.

Here’s a sketch example that displays “**Hello, World!**“.

#include <LiquidCrystal\_I2C.h>

// set the LCD number of columns and rows

int lcdColumns = 16;

int lcdRows = 2;

// set LCD address, number of columns and rows

// if you don't know your display address, run an I2C scanner sketch

LiquidCrystal\_I2C lcd(0x27, lcdColumns, lcdRows);

void setup(){

// initialize LCD

lcd.init();

// turn on LCD backlight

lcd.backlight();

}

void loop(){

// set cursor to first column, first row

lcd.setCursor(0, 0);

// print message

lcd.print("Hello, World!");

delay(1000);

// clears the display to print new message

lcd.clear();

// set cursor to first column, second row

lcd.setCursor(0,1);

lcd.print("Hello, World!");

delay(1000);

lcd.clear();

}

**How the code works**

First, you need to include theLiquidCrystal\_I2C library.

#include <LiquidCrystal\_I2C.h>

The next two lines set the number of columns and rows of your LCD display. If you’re using a display with another size, you should modify those variables.

int lcdColumns = 16;

int lcdRows = 2;

Then, you need to set the display address, the number of columns and number of rows. You should use the display address you’ve found in the previous step.

LiquidCrystal\_I2C lcd(0x27, lcdColumns, lcdRows);

In the setup(), first initialize the display with the init() method.

lcd.init();

Then, turn on the LCD backlight, so that you’re able to read the characters on the display.

lcd.backlight();

To display a message on the screen, first you need to set the cursor to where you want your message to be written. The following line sets the cursor to the first column, first row.

lcd.setCursor(0, 0);

**Note**: 0 corresponds to the first column, 1 to the second column, and so on…

Then, you can finally print your message on the display using the print() method.

lcd.print("Hello, World!");

Wait one second, and then clean the display with the clear() method.

lcd.clear();

After that, set the cursor to a new position: first column, second row.

lcd.setCursor(0,1);

Then, the process is repeated.

So, here’s a summary of the functions to manipulate and write on the display:

* lcd.init(): initializes the display
* lcd.backlight(): turns the LCD backlight on
* lcd.setCursor(int column, int row): sets the cursor to the specified column and row
* lcd.print(String message): displays the message on the display
* lcd.clear(): clears the display

## This example works well to display static text no longer than 16 characters.

## **Conclusion**

Interfacing LCD Module with ESP32 Development Board is practiced in designing software application (fritzing) and learned the importance of LCD Module connection with ESP32, how the ESP32 and LCD Interface works and how to display some text on the LCD is observed by writing sketch in ardunio IDE .