**EXPERIMENT 6**

**Objective: Design an IOT system based on ESP32 using I2C Protocol interfaced with LCD**

In this project, we will learn how to interface a HW061 I2C LCD with ESP32 Development Board. We will see how to configure the I2C Pins in ESP32, download necessary libraries for Arduino IDE, understand how ESP32 I2C LCD interface works and display some data on the 16×2 LCD.

### Components Required

* ESP32 DevKit Development Board
* 16×2 Character LCD
* HW061 I2C LCD adapter Module
* Breadboard
* Connecting Wires
* Micro-USB Cable

## Introduction

There are many display devices like OLED Display, 128×64 Graphical Display, Nokia 5110 Display etc. But for displaying simple text and number, all we need is a 16×2 or 20×4 Character Display.

The main disadvantage of 16×2 LCD Display is it requires at least 6 GPIO pins of a Microcontroller to function properly. This is not even the 8-bit mode but reduced 4-bit mode (add another 4 GPIO pins for 8-bit mode).

Using that many pins of a microcontroller just for a simple character LCD Display is not desirable especially if you are designing a project with lot of sensors and devices connected to the microcontroller.

## A Brief Note on HW061 I2C LCD Module

With backlight power control, you can set whether to connect the backlight power through the jumper cap. Plug in the jumper cap to connect the backlight power, unplug the jumper cap to disconnect the backlight power.

The contrast can be adjusted. Rotate the blue potentiometer to increase clockwise and decrease counter clockwise. The potentiometer is designed on the front, which is convenient for customers to adjust freely anytime and anywhere.

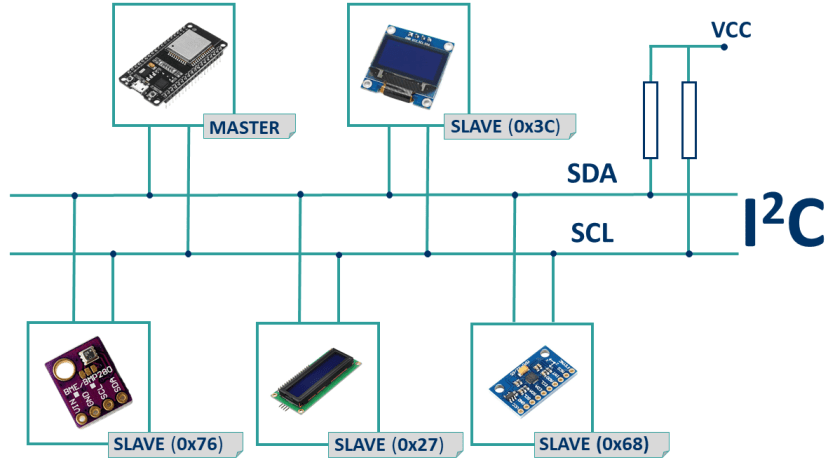
Modules can be cascaded; up to 8 can be cascaded. Modify the device address by shorting A0/A1/A2. The default address is 0x27.

IIC/I2C interface LCD 1602/2004 adapter board, without LCD screen.

Supply voltage: 2.5-6V

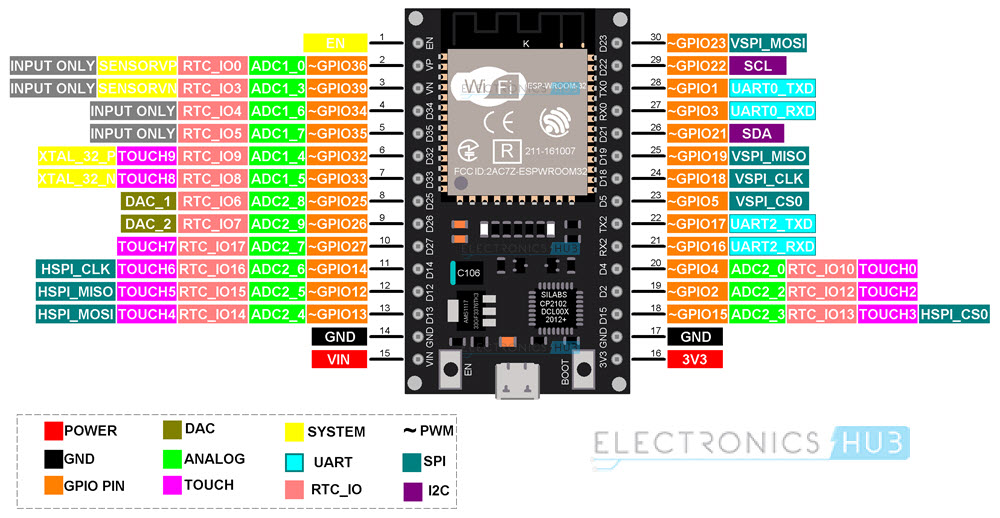
The following diagram showing how I2C interfacing used to connect different slave devices

(here LCD) to master device ESP32

[](https://i0.wp.com/randomnerdtutorials.com/wp-content/uploads/2019/09/I2C-communication-protocol-ESP32.png?quality=100&strip=all&ssl=1)

ESP32 I2C LCD Interface

The first thing we need to figure out for ESP32 I2C LCD Interface to work is the I2C pins of ESP32. If you remember, in the ESP32 Pinout Tutorial, I mentioned that the ESP32 Microcontroller has two I2C interfaces and any GPIO pin can be configured to be used as an I2C Pin.

[](https://www.electronicshub.org/wp-content/uploads/2021/02/ESP32-Pinout-1.jpg)

But after using all the GPIO pins for various purposes like SPI, [UART](https://www.electronicshub.org/basics-uart-communication/), ADC etc., GPIO 22 and GPIO 21 are free (without any important alternative functions). So, in this project, we will be using GPIO 22 as SCL and GPIO 21 as SDA of the I2C interface.

Another important point to note is that the 16×2 LCD runs on standard TTL voltage levels i.e., 4.7V to 5.3V. We cannot apply 3.3V as VCC to the I2C LCD Module even though the HW061 IC works at this voltage.

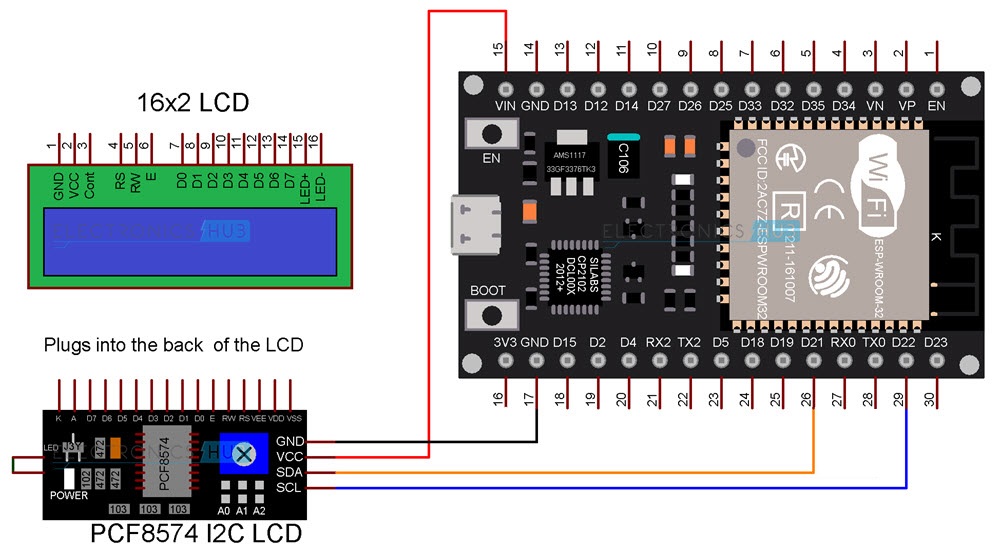
So, keeping this in mind, the following table shows all the connections between I2C LCD Module and ESP32

|  |  |
| --- | --- |
| **HW061 I2C LCD Module** | **ESP32 DevKit** |
| GND | GND |
| VCC | VIN |
| SDA | GPIO 21 (D21) |
| SCL | GPIO 22 (D22) |

As you can see from the table, the VCC of the I2C LCD Module is connected to VIN pin of the ESP32 Development Board. This is connected to USB Supply when ESP32 is powered through USB. So, we will get around 5V on this pin.

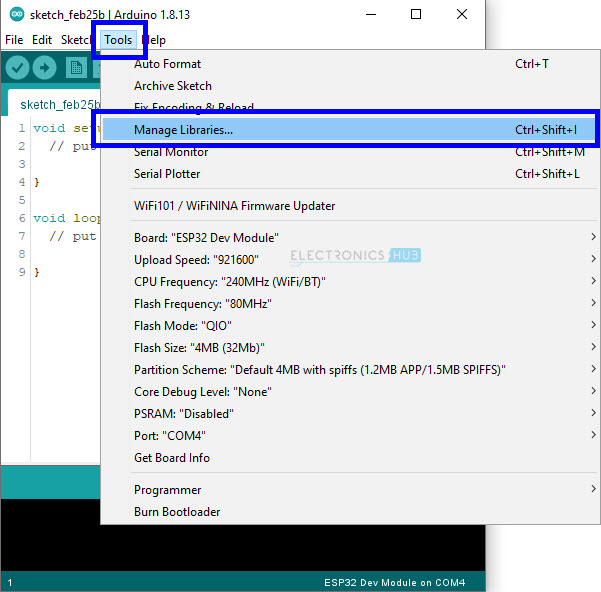
### Circuit Diagram

The following image shows the circuit diagram for demonstrating ESP32 I2C LCD Interface.

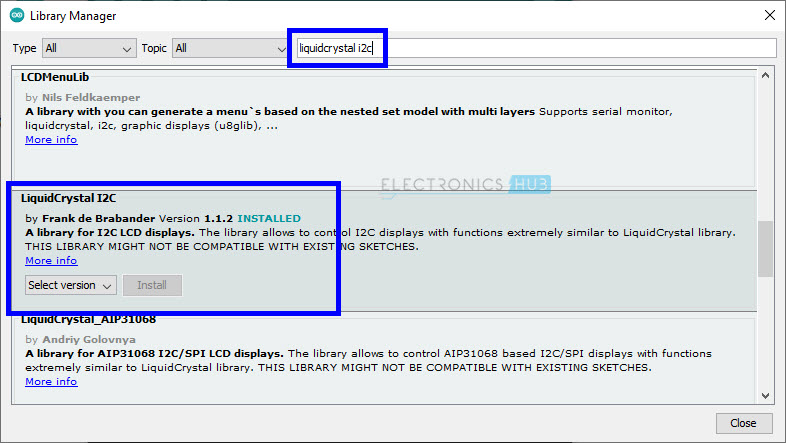
[](https://www.electronicshub.org/wp-content/uploads/2021/02/ESP32-I2C-LCD-Circuit.jpg)

## Preparing Arduino IDE

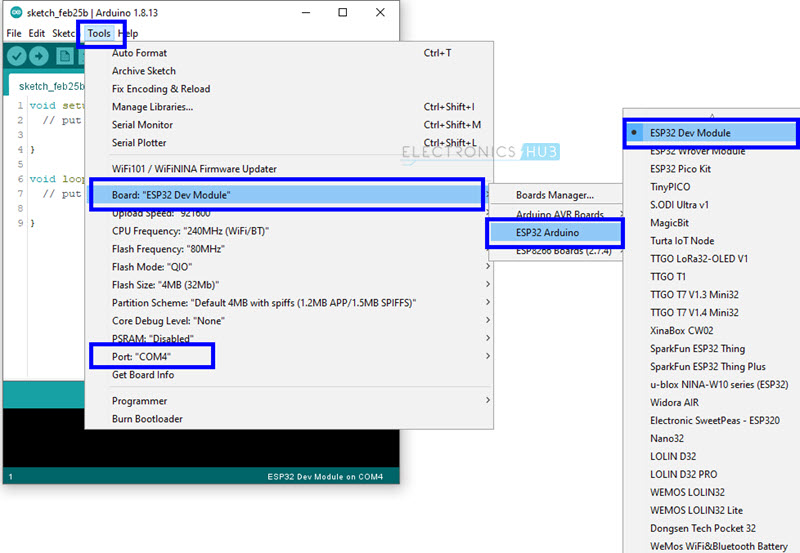
You have to download one library called ‘LiquidCrystal\_I2C’ in order to work with I2C LCD Module. If you already installed this library, you can skip this step. If not, then follow along. Open Arduino IDE and open the library manager by selecting Tools -> Manage Libraries. . .

[](https://www.electronicshub.org/wp-content/uploads/2021/02/I2C-LCD-ESP32-Arduino-IDE-1.jpg)

Once the library manager window opens, search for ‘liquidcrystal i2c’ and install the library ‘LiquidCrystal I2C’ by Frank de Brabander. There are other libraries for I2C LCD. You can try them as well.

[](https://www.electronicshub.org/wp-content/uploads/2021/02/I2C-LCD-ESP32-Arduino-IDE-2.jpg)

Close the library manager. Make all the necessary connections as per the circuit diagram and plug in the ESP32 board to the computer. In the Arduino IDE, select the correct board (ESP32 Dev Module) and also the correct COM Port.

[](https://www.electronicshub.org/wp-content/uploads/2021/02/I2C-LCD-ESP32-Arduino-IDE-3.jpg)

## Finding the Slave Address of I2C LCD Module

Since we are working with I2C interface, the slave address is one of the important things we need to be aware of. Here, the ESP32 Microcontroller is the master and the HW061 I2C LCD Module is the slave.

You can refer to the datasheet of HW061 IC and calculate the slave address but for now we will go with the easy way i.e., a small program which determines the slave address automatically.

After making all the necessary connections between ESP32 and I2C LCD Module, upload the following code and open the serial monitor of Arduino IDE.

//starting sketch to find i2c address--------------------------------------------

#include <Wire.h>

void setup()

{

Wire.begin();

Serial.begin(9600);

while (!Serial); // wait for serial monitor

Serial.println("\nI2C Scanner");

}

void loop()

{

byte error, address;

int nDevices;

Serial.println("Scanning...");

nDevices = 0;

for(address = 1; address < 127; address++ )

{

// The i2c\_scanner uses the return value of

// the Write.endTransmisstion to see if

// a device did acknowledge to the 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(" !");

nDevices++;

}

else if (error==4)

{

Serial.print("Unknown error at address 0x");

if (address<16)

Serial.print("0");

Serial.println(address,HEX);

}

}

if (nDevices == 0)

Serial.println("No I2C devices found\n");

else

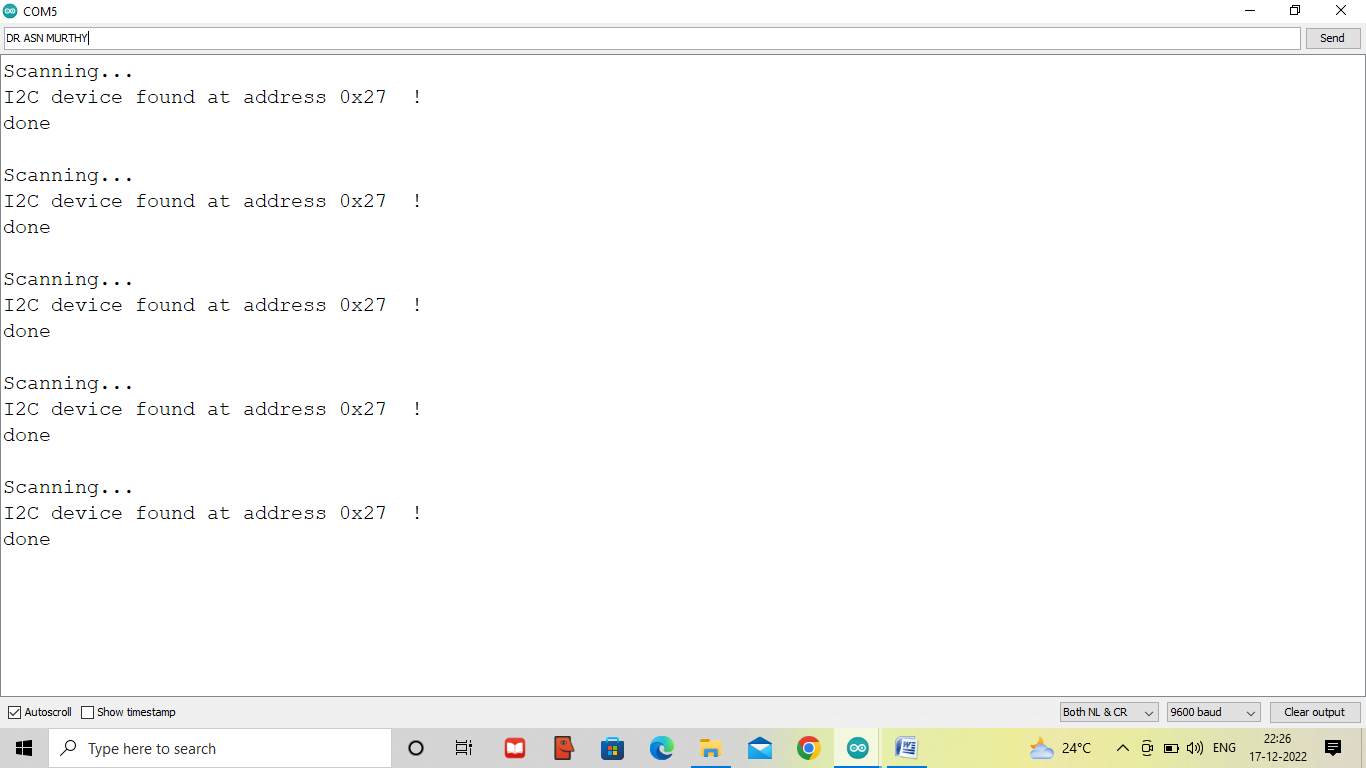
Serial.println("done\n");

delay(5000); // wait 5 seconds for next scan

}

// end of sketch to find i2c address --------------------------------

Result and Discussion: The I2C address observed from the serial monitor and is as shown from the screen shot:



As you can see from the screenshot of the serial monitor, the slave address of HW061 I2C LCD Module is 0x27

## Displaying Simple Text

Let us use this slave address and display some basic text on the 16×2 LCD display using I2C LCD Module. Use the following code and see the output.

//starting sketch----------------------------------------

//#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup()

{

lcd.init(); // initialize the lcd

lcd.backlight();

Serial.begin(9600);

}

void loop()

{

// when characters arrive over the serial port...

if (Serial.available()) {

// wait a bit for the entire message to arrive

delay(100);

// clear the screen

lcd.clear();

// read all the available characters

while (Serial.available() > 0) {

// display each character to the LCD

lcd.write(Serial.read());

}

}

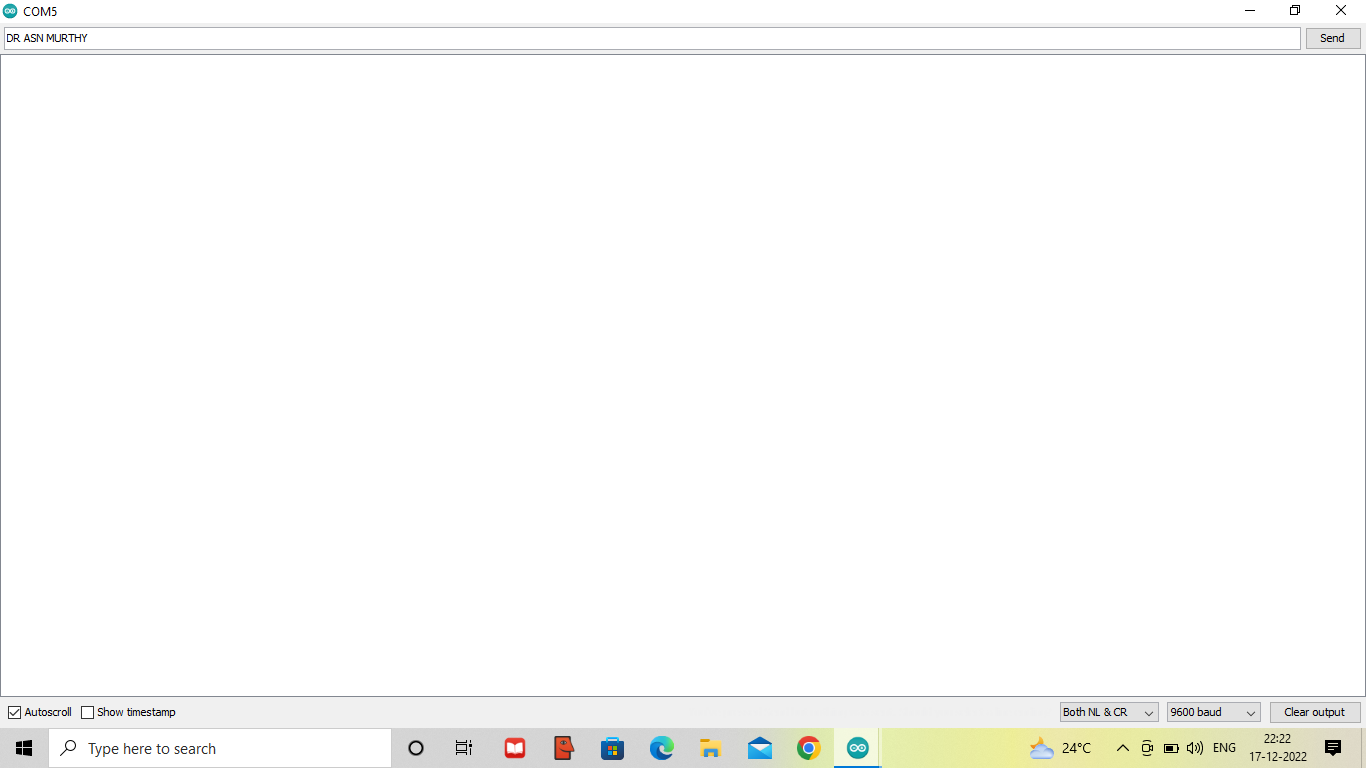
}

//end ------------------------------------------------------

Result and discussion:

The code is verified and uploaded to the module, then open the serial monitor and type the text in the serial MONITOR and press the send button, and then you can observe the typed text on the LCD screen.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



The result as shown in the screen shot as above. If the display is not visible adjust the potentiometer in the clockwise direction on the I2C adapter module to increase the brightness of the text on the LCD screen. (This is the problem faced while doing the experiment and then get corrected by doing so)

-:0:-