**Practical No. 5**

|  |
| --- |
| **Aim:** Design a sketch to interface LCD with ESP-32 board**.** |
| **Requirement (Hardware/Software):** ESP32 board, Breadboard, A 16×2 I2C LCD display, Jumper Wires. |
| **Theory**:  Displays provide a fantastic way of providing feedback to users of any project and with the 16×2 LCD being one of the most popular displays among makers, and engineers, its probably the right way to start our exploration. For today’s tutorial, we will use an I2C based 16×2 LCD display because of the easy wiring it requires. It uses only four pins unlike the other versions of the display that requires at least 7 pins connected to the microcontroller board.  In this method we are using an I2C adapter which is connected to LCD display with 16 pins and outputs only 4 pins, 2 for data and another 2 for power and ground. The I2C adapter has inbuilt potentiometer where we can adjust the contrast of the display and also has a jumper to turn on/off the LCD backlight.    As you can see from the below circuit diagram the VCC and GND pins are connected to the 5V and GND respectively and the other two data pins SDA and SCL are connected to D21 and D22 respectively. to know which are the SDA and SCL pins of your ESP 32 Board check the pin out diagram.    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 5v 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 - 5v/Vin  SDA - D21(GPIO)  SCL– D22 (GPIO)  To be able to easily write the code to interact with the I2C LCD display, we will use the I2C LCD library. The Library possesses functions and commands that make addressing the LCD easy. Download the I2C LCD library from the link attached and install on the Arduino IDE by simply extracting it into the Arduino’s library folder.  Before writing the code for the project, it’s important for us to know the I2C address of the LCD as we will be unable to talk to the display without it. While some of the LCDs come with the address indicated on it or provided by the seller, in cases where this is not available, you can determine the address by using a simple sketch that sniffs the I2C line to detect what devices are connected alongside their address. This sketch is also a good way to test the correctness of your wiring or to determine if the LCD is working properly.  This sketch basically uses a “for” loop to generate a list of addresses and then sends a begin transmission request to the address. The return value of the write. endTransmission() function shows if a device exists on that particular address.    The address at which a response was received is the address we are a looking for. If we keep getting “no devices found”, it might help to take a look at the connections to be sure you didn’t mix things up and you could also go ahead and try 0x27 as the I2C address. This is a common address for most I2C LCD modules |
| **Code:**  #include <LiquidCrystal\_I2C.h>  // 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, 16, 2);  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  } |
| **Conclusion:** Sketch to interface LCD with ESP-32 board is successfully designed. |