JK Lakshmipat University

Institute of Engineering & Technology



Internet of things (IOT) (EE1222)

Submitted To: Mr. Divanshu Jain

**Submitted By:**

Name: Aaditya Gaur

Roll No: 2022Btech119

Section: B

**Acitvity 1: Displaying Message or Potentiometer Value on a 16x2 LCD Using NodeMCU**

**Objective:**

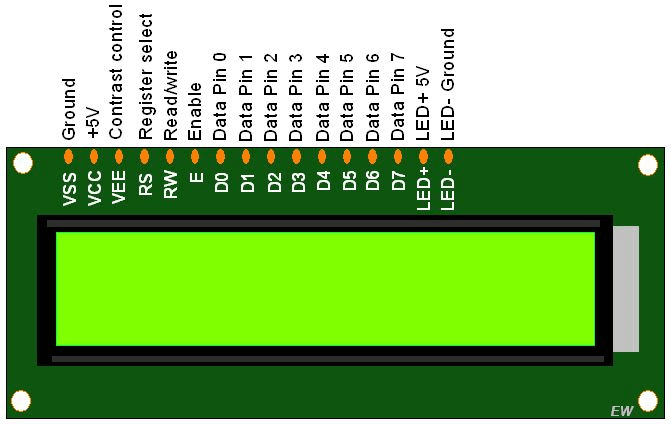
To interface a 16x2 LCD with a NodeMCU to display a custom message or the value of a potentiometer.

**Components Required:**

1. NodeMCU (ESP8266)
2. 16x2 LCD display
3. Potentiometer (10k ohm)
4. Jumper wires
5. Breadboard

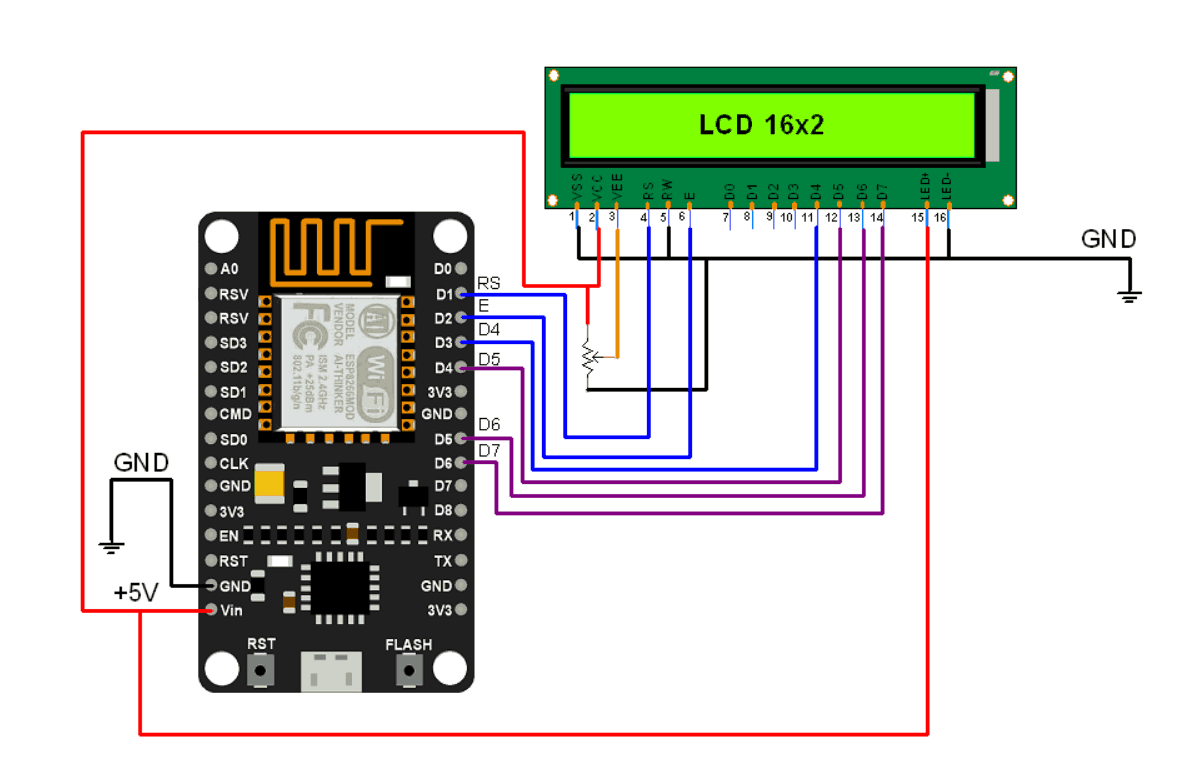
**Theory:**

**LCD (16x2)**

A 16x2 LCD (Liquid Crystal Display) is one of the most common display modules used in embedded systems to display text and numeric information. The "16x2" notation means the display can show 16 characters per row and has 2 rows. This display is widely used due to its simplicity and ease of interfacing with microcontrollers like the NodeMCU.

An LCD works by manipulating liquid crystals that can change the orientation of light when a voltage is applied. The display consists of tiny blocks called pixels that form characters or graphics. Each pixel can be turned on or off by controlling the voltage applied to the liquid crystal.

**Circuit Diagram:**



**Procedure:**

1. LCD Initialization: Initialize the 16x2 LCD in 4-bit mode using the Liquid Crystal library in Arduino IDE.
2. Potentiometer Setup: Connect a potentiometer to the analog input (A0) of the NodeMCU to read varying values based on the rotation of the knob.

**Code:**

#include <LiquidCrystal.h>

const int rs = D1, en = D2, d4 = D3, d5 = D4, d6 = D5, d7 = D6;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int potPin = A0;

void setup() {

lcd.begin(16, 2);

lcd.setCursor(0, 0);

lcd.print("Pot Value: ");

}

void loop() {

int potValue = analogRead(potPin);

lcd.setCursor(0, 1);

lcd.print(potValue); // Display raw analog value (0 to 1023)

delay(500);

}

**Result:**

The 16x2 LCD successfully displayed the potentiometer value based on its rotation.

**Acitvity 2: Controlling a Relay (230V AC Device) Using NodeMCU**

**Objective:**

To control a 230V AC device using a relay connected to NodeMCU.

**Components Required:**

NodeMCU (ESP8266)

5V Relay Module

230V AC Load (e.g., bulb)

Jumper wires

Power supply (5V for relay)

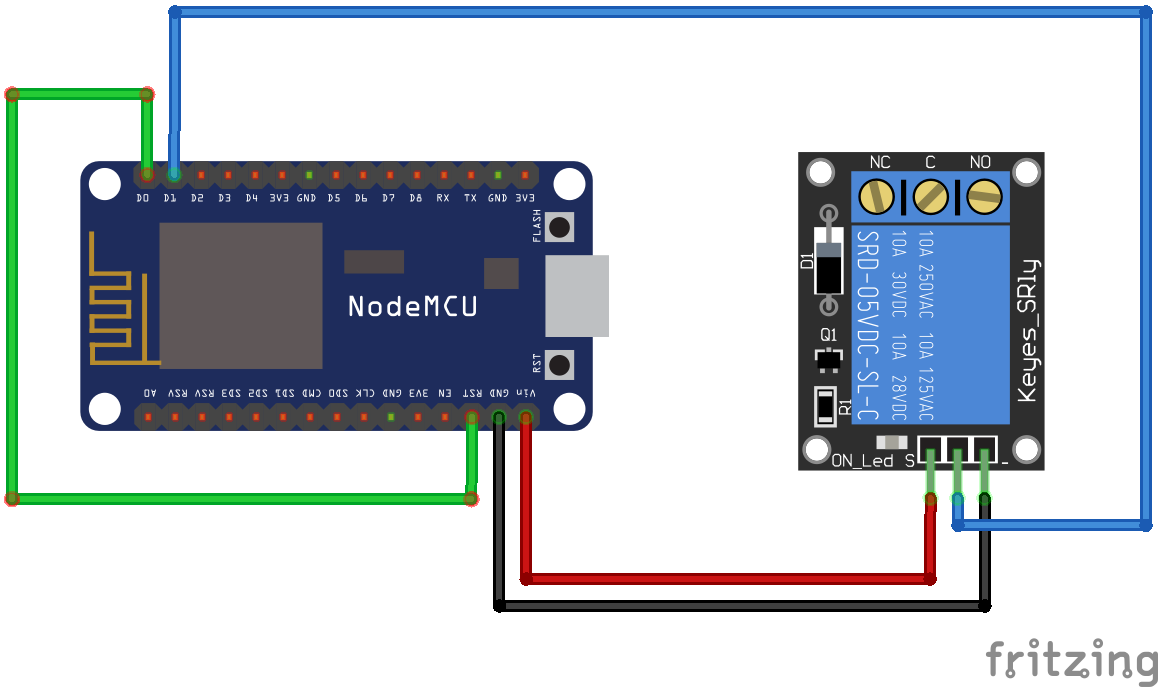
**Theory:**

**Relay**

A relay is an electrically operated switch that allows low voltage signals to control high voltage circuits. In this experiment, a relay is used to control a 230V AC load (such as a light bulb) using the low voltage (3.3V) output from a NodeMCU. Relays are widely used in automation, safety systems, and embedded systems where electrical isolation between the control and power circuits is essential.

A relay consists of an **electromagnet** and a set of **switching contacts**. When a small current flows through the coil of the electromagnet (triggered by a low voltage control signal), it generates a magnetic field that pulls the switch contacts into a different state—either connecting or disconnecting the high voltage circuit.

**Circuit Diagram:**



**Procedure:**

Relay Setup: Connect the relay module to NodeMCU to control the switching of a 230V AC device.

Control Logic: Write a program in Arduino IDE to toggle the relay based on a condition (e.g., potentiometer threshold or a button press).

Relay Safety: Ensure proper isolation between the high-voltage AC side and the low-voltage NodeMCU side.

AC Device Control: Control the connected AC device (light bulb) by switching the relay.

**Code:**

const int relayPin = D7;

void setup() {

pinMode(relayPin, OUTPUT);

}

void loop() {

digitalWrite(relayPin, HIGH);

delay(2000);

digitalWrite(relayPin, LOW);

delay(2000);

}

**Result:**

The 230V AC load was successfully controlled using the relay connected to NodeMCU.A circuit board with wires connected to it

Description automatically generatedA hand holding a circuit board with wires

Description automatically generated