

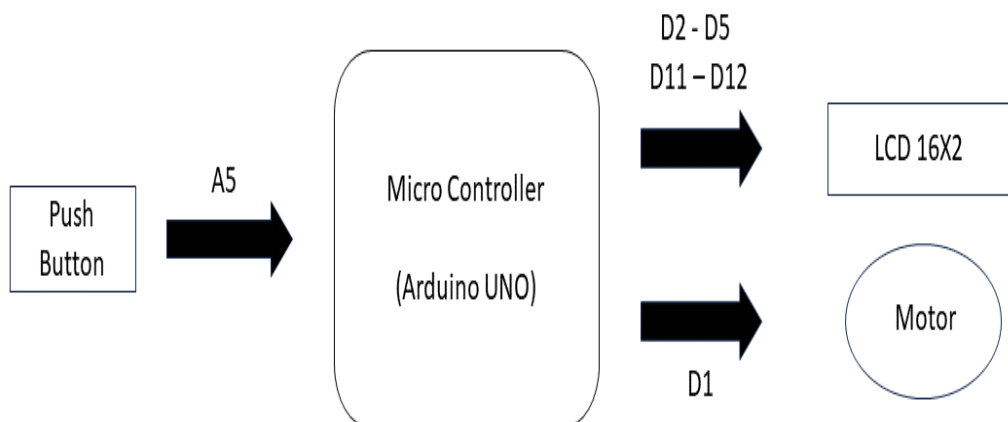
PROJECT 2

DISPLAYING THE MOTOR STATUS ON LCD USING ARDUINO UNO

➤ INTRODUCTION:

Displaying the motor status on an LCD using an Arduino Uno is a common and useful application in various electronics and robotics projects. This integration allows you to monitor and control a motor while providing real-time feedback on its operational status through a liquid crystal display (LCD). This introduction provides an overview of the concept and its practical applications.

➤ BLOCK DIAGRAM:



➤ **MATERIALS REQUIRED:**

1. Arduino Uno board.
2. LCD display (16x2 or 20x4).
3. Motor and motor driver (L298N or similar).
4. Jumper wires.
5. Power supply for the motor (e.g., a separate battery or power source).

➤ **PROJECT OVERVIEW:**

- **Arduino Uno:** The Arduino Uno is a microcontroller board that acts as the brain of the system. It runs the code responsible for controlling the motor and interacting with the LCD.
- **Motor:** The motor is the device you want to control and monitor. It can be a DC motor, stepper motor, or any other type, depending on your project's requirements.
- **Motor Driver:** In many cases, a motor driver is used to control the motor's speed and direction. Motor drivers like the L298N or L293D are commonly used with Arduino projects to interface with motors.
- **LCD (Liquid Crystal Display):** The LCD is used to display information about the motor's status. It can display text, numbers, and symbols to convey information to the user.
- **Jumper Wires:** These are used to connect the various components together, ensuring that electrical signals can flow between them.

➤ **PROGRAM:**

```
#include <LiquidCrystal.h>

// LCD Display setup
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// Pin for the push button
const int buttonPin = A5;

// Pin for analog motor control
const int motorPin = 1;

// Variables
int motorState = 0;
int lastButtonState = HIGH;
int buttonState;

void setup() {
  lcd.begin(16, 2);
  pinMode(buttonPin, INPUT);
  analogWrite(motorPin, motorState); // Set initial motor state
  updateLCD();
}
```

```
void loop() {  
    buttonState = digitalRead(buttonPin);  
  
    if (buttonState != lastButtonState) {  
        if (buttonState == LOW) {  
            motorState = 255 - motorState; // Toggle motor state (0 to 255 and vice versa)  
            analogWrite(motorPin, motorState);  
            updateLCD();  
        }  
    }  
    lastButtonState = buttonState;  
}
```

```
void updateLCD() {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Motor: ");  
    if (motorState == 0) {  
        lcd.print("OFF");  
    } else {  
        lcd.print("ON");  
    }  
}
```

➤ SCHEMATIC DIAGRAM:

