



Andhra Pradesh State Skill Development Corporation



INTERNET OF THINGS (IoT)

**MOTOR WITH L293D AND RELAY
INTERFACING WITH ARDUINO**

Motor interfacing with Arduino with l293d

AIM: To control the motor with the help of the L293d Motor driver

Software: Arduino IDE

Components Required:

1. System -1
2. Arduino Uno Board -1
3. Arduino dumping cable -1
- 4.7 Segment display
5. Potentiometer 10k Ω -1
6. Resistor 220 Ω -1
7. Breadboard-1
8. Connecting Wires -Required
- 9.SN7446AN

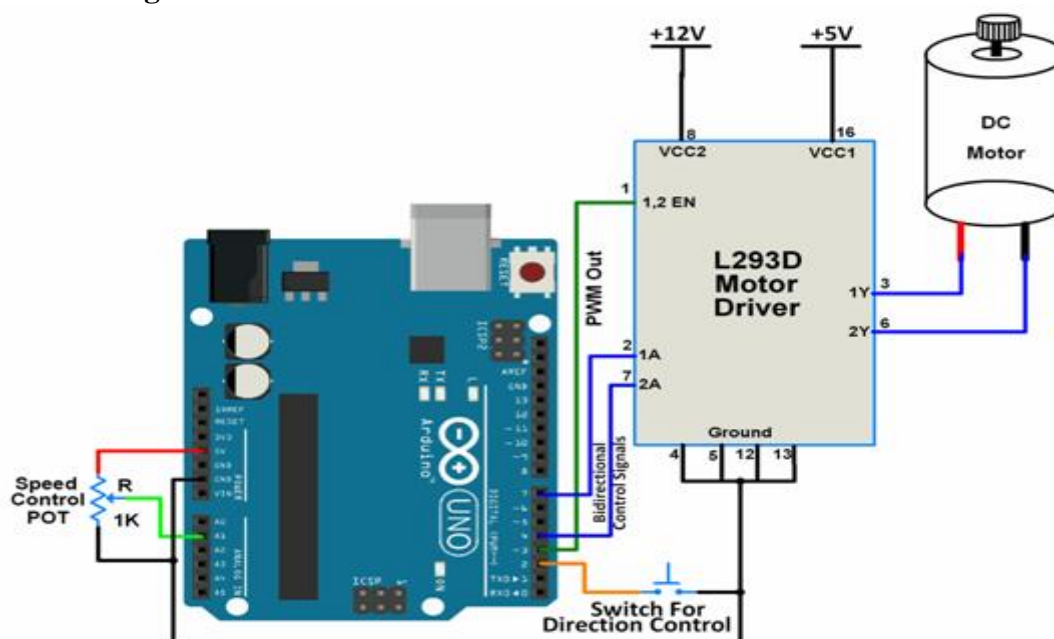
Theory:-A DC motor converts electrical energy in the form of Direct Current into mechanical energy in the form of rotational motion of the motor shaft. The DC motor speed can be controlled by applying varying DC voltage; whereas the direction of rotation of the motor can be changed by reversing the direction of current through it. For applying varying voltage, we can make use of PWM technique. For reversing the current, we can make use of H-Bridge circuit or motor driver ICs that employ the H-Bridge technique. For more information about DC motors and how to use them, H-Bridge circuit configurations, and PWM technique, refer to the topic DC Motors in the sensors and modules section.

Here, we are going to control the speed and rotational direction of the DC motor using Arduino Uno.

Here, a potentiometer is used as a means for speed control and an input from a tactile switch is used to change the direction of the motor.

L293D motor driver IC is used for controlling the direction of the motor. PWM wave generated on the Arduino UNO is used to provide a variable voltage to the motor through L293D. In Arduino, analogWrite function is used to generate PWM waves.

Circuit Diagram:-





Functions Used:-

1. digitalPinToInterrupt(pin)

- This function is used to declare the digital pin as an interrupt pin.
- Example, digitalPinToInterrupt(2) is used to declare digital pin 2 as an interrupt pin.
- On UNO board, only pins 2 and 3 can be configured as interrupt pins. Hence, argument to this function can only be pin 2 or pin 3.

2. attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)

- This function is used to configure the mode of interrupt event and declare the ISR for that interrupt. The interrupt event and ISR is for the interrupt pin declared by the function digitalPinToInterrupt(pin).
 - ISR in this function is the name of the ISR that will be used for this interrupt.
 - mode defines when the interrupt will be triggered. There are four modes available to choose from :
 - LOW : trigger the interrupt whenever the pin is low.
 - CHANGE : trigger the interrupt whenever the pin changes value.
 - RISING : trigger when the pin goes from low to high.
 - FALLING : trigger when the pin goes from high to low.
 - Example, attachInterrupt(digitalPinToInterrupt(2), motor, FALLING) configures digital pin 2 as an interrupt pin with ISR named motor and which generates interrupt for every falling edge event on pin 2.
- ### 3. analogWrite(pin,value)
- This function is used for generating PWM on PWM digital pins(pins 3,5,6,9,10,11 for Arduino UNO).
 - value can be any number between 0 to 255. 0 being 0% duty cycle and 255 being 100% duty cycle

Code:-

```
const int POT_input = A1; /* assign ADC Channel */
bool d1 = HIGH;
bool d2 = LOW;
void setup() {
    pinMode(4, OUTPUT); /* Motor control pin 1 */
    pinMode(7, OUTPUT); /* Motor control pin 2 */
    pinMode(3, OUTPUT); /* PWM pin for Speed Control */
    pinMode(2, INPUT_PULLUP); /* Interrupt pin for direction control */
    attachInterrupt(digitalPinToInterrupt(2), motor, FALLING); /* Interrupt on falling edge on pin 2 */
}
void loop() {
    int pwm_adc;
    pwm_adc = analogRead(POT_input); /* Input from Potentiometer for speed control */
    digitalWrite(4,d1);
    digitalWrite(7,d2);
    analogWrite(3, pwm_adc / 4);
}
void motor(){
    d1 = !d1;
    d2 = !d2;
```



```
_delay_ms(200);  
}
```

Relay Module interfacing with arduino

AIM: To control high voltage with the help of the 5V

Software: Arduino IDE

Components Required for relay module

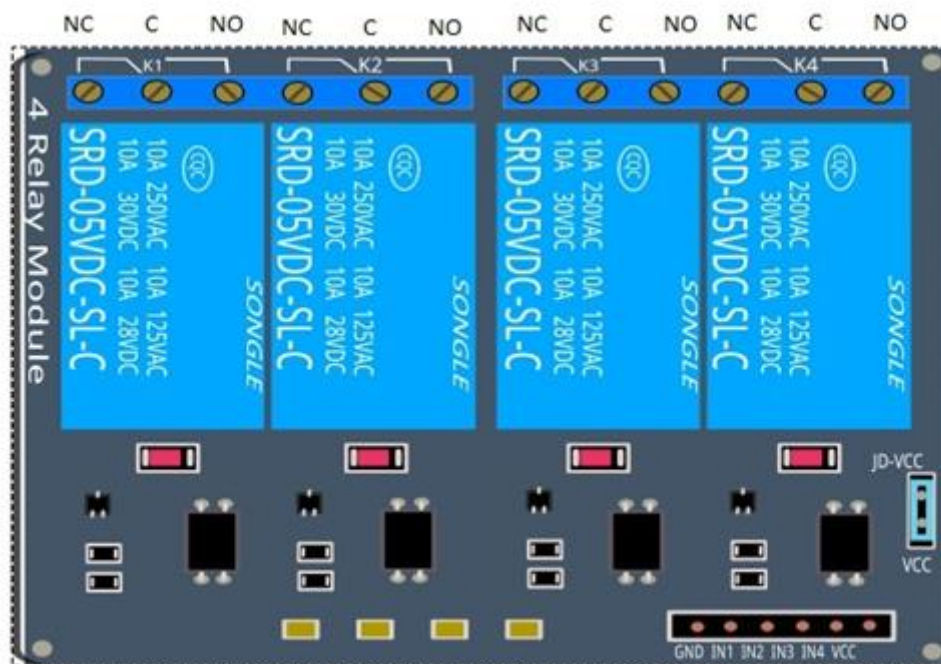
1. Arduino Uno (You can use any other type)
2. 4 Relay module
3. 3v Battery
4. Led's
5. 220 ohm resistors
6. Breadboard

Theory:-Relay module for Arduino is one of the most powerful applications for Arduino as it can be used to control both A.C and D.C devices by simply controlling the relay by giving 5V. A relay is basically a switch which is operated electrically by electromagnet. A relay can be used to control high voltage electronic devices such as motors and as well as low voltage electronic devices such as a light bulb or a fan.

Relays work on the principle of electromagnetism. When the electricity is provided to the relay coil then it acts like a magnet and changes the state of the switch. The part which powers the relay module is completely isolated from the part which turns ON or OFF. This is why we can control a 220V appliance by simply controlling it using the 5V Arduino. you should also read getting started projects of arduino.

Relay Module

There are many types of relay modules available like 1 relay, 2 relay, 5 relay and 8 relay but we are using the 4 relay Arduino module. Once you will understand this then you can use all other types.



Relay Module Pinout

Input Connections of relay module

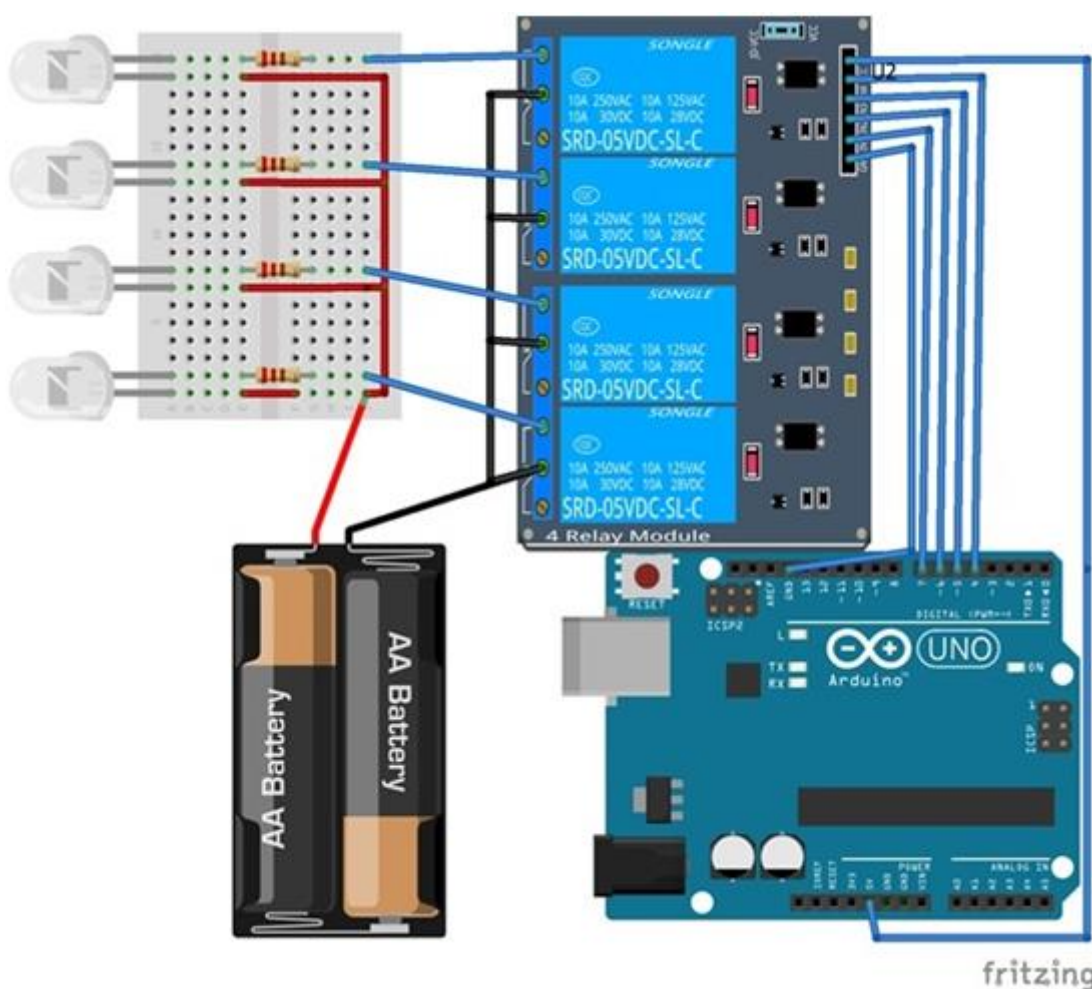
- I recommend you to learn how to use the digital output of an Arduino microcontroller. Before reading this article further.
- Logic GND: This will be connected to GND on your Arduino.
- Input 1 (IN 1): This will be connected to digital pin on your Arduino, or leave it unconnected if you do not want to use this channel.
- Input 2 (IN 2): This will be connected to the digital pin on your Arduino, or leave it unconnected if you do not want to use this channel.
- Input 3 (IN 3): This will be connected to the digital pin on your Arduino, or leave it unconnected if you do not want to use this channel.
- Input 4 (IN 4): This will be connected to the digital pin on your Arduino, or leave it unconnected if you do not want to use this channel.
- Logic VCC: This will be connected to the 5v pin of the Arduino to power the 4 relay module.
- You can also power the 4 relay module using the external power by giving voltage from 5 to 24V DC.

Output Connections of relay module

- you may also like to read **Xbee interfacing with arduino**
Before getting into the output connections of the relay, we have to understand the NO, COM and NC connections.
- COM (Common connection): The COM is the center terminal of the relay and it is used in both (Normally open and normally closed) connections.
- NO (Normally open): This acts like a switch. In a normally open connection, there will be no contact between COM and NO, since it is normally open. But when we will activate the relay module, then it will get connected to the COM and will supply power to the load, which will power up the light. Thus the circuit will initially be in open state until we trigger the state.

- NC (Normally closed): its behavior is opposite to the normally open connection. It always remains in contact with COM, even when the relay module is not powered. When we will trigger the relay module then it will open the circuit, so the connection is lost.
- We will use the normally open connection. When you will use the relays in NO (Normally Open) connection and you set the corresponding IN pin to LOW, then the power will flow from the COM connector and out of the normally open connector powering your device.
- Relay 1 (K1): Connect one end of the load to the COM and the other end to the NO.
- Connections for the K2, K3 and K4 are the same.

relay module interfacing with arduino



The connection of 4 relay modules to an Arduino is very easy and allows you to control many devices through Arduino (both A.C and D.C). In this example we will connect a simple load such as a led at the output of the relay and will control it by using the Arduino. First connect the 5v of the Arduino to the VCC of the 4 relay module and the ground of the Arduino to the ground of the 4 relay module. Then we will have to connect the communication pins IN1, IN2, IN3, and IN4 to the Arduino data pins 7, 6, 5, and 4.

We will use a 3v battery to power the Led's. So we have connected the positive of the battery to the positive side of the led's and then we have connected the negative side to the com of each relay and the NO to the negative side of the led's.



Code of relay module interfacing with microcontroller

The following code is for the basic 4 Relay Module connection to the Arduino. Each relay will turn on for 5 seconds and then will turn off. You will hear the click sound as there state changes from OFF to ON or from ON to OFF. You will also see the Red LED on the 4 Relay board light up when the relay is on. Necessary comments are made in the code for better understanding. For 4 input relay modules, we will turn on each relay for 5 seconds and then turn it off.

```
#define RELAY1 7 //Defining the pin 7 of the Arduino for the 4 relay module
#define RELAY2 6 //Defining the pin 6 of the Arduino for the 4 relay module
#define RELAY3 5 //Defining the pin 5 of the Arduino for the 4 relay module
#define RELAY4 4 //Defining the pin 4 of the Arduino for the 4 relay module

void setup()
{
  pinMode(RELAY1, OUTPUT); //Defining the pin 7 of the Arduino as output
  pinMode(RELAY2, OUTPUT); //Defining the pin 6 of the Arduino as output
  pinMode(RELAY3, OUTPUT); //Defining the pin 5 of the Arduino as output
  pinMode(RELAY4, OUTPUT); //Defining the pin 4 of the Arduino as output
}

void loop()
{
  digitalWrite(RELAY1,LOW); // This will Turn ON the relay 1
  delay(5000); // Wait for 5 seconds
  digitalWrite(RELAY1,HIGH); // This will Turn the Relay Off

  digitalWrite(RELAY2,LOW); // This will Turn ON the relay
  delay(5000); // Wait for 5 seconds
  digitalWrite(RELAY2,HIGH); // This will Turn the Relay Off

  digitalWrite(RELAY3,LOW); // This will Turn ON the relay
  delay(5000); // Wait for 5 seconds
  digitalWrite(RELAY3,HIGH); // This will Turn the Relay Off

  digitalWrite(RELAY4,LOW); // This will Turn ON the relay
  delay(5000); // Wait for 5 seconds
  digitalWrite(RELAY4,HIGH); // This will Turn the Relay Off
}
```