Page No.
Date

Devanshu Surana
PC-23, 1032210755
Batch C1

MAIOT Lab Assignment 3

Problem statement:
To interface simple actuators such as DC|servo|stepper motor, relays etc. with Raspberry Pi | ESP8266 boards | Reaglebone board | Tinker CAD Arduino Uno.

Objectives:
1. To understand actuators interfacing with development boards.
2. DC Motor or stepper motor control using LN298 motor driver and Arduino UNO.

Theory:

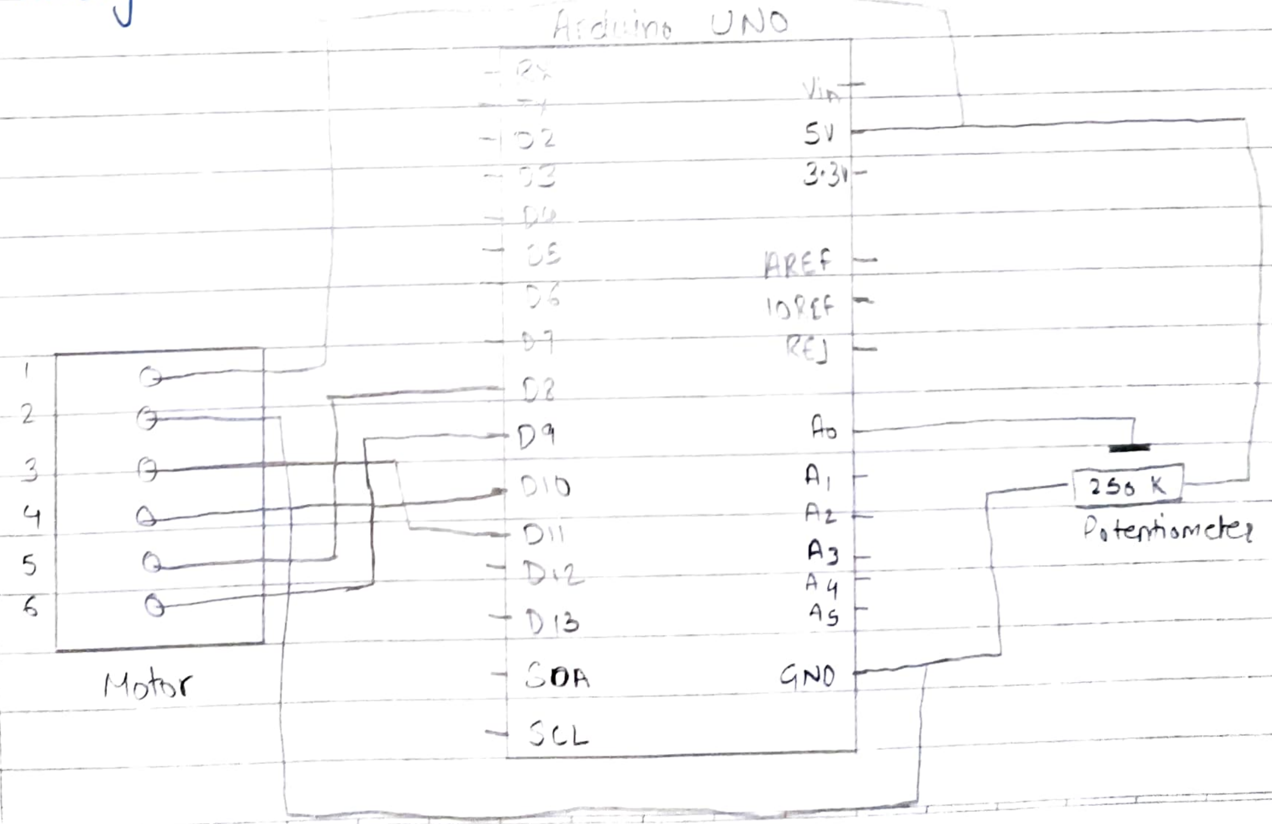


Fig: stepper motor setup.

Teacher's Sign:

- DC motors are the simplest and most common type of electric motor. They operate using a direct current (DC) power source and have a rotating armature that is powered by a commutator. DC motors are known for their high starting torque and easy speed control.
- Stepper motors are slow, easy setup, precise rotation and control - Advantages like feedback mechanism and backing circuitry to drive locating, this motor has positional control through its nature of rotation by fractional additions.
- Servo motors are high torque, fast, accurate rotation in a limited angle. Generally, a high-performance alternative to stepper motors, but more complicated setup with PNM turning. suited for robotic arm/legs or rudder control, etc.

- LN298

The speed of a DC Motor can be controlled by changing its input voltage. A widely used technique to accomplish this, a Pulse Width Modulation (PMW). The spinning direction of a DC motor can be controlled by changing the polarity of its input voltage. A widkely used technique to accomplish this is to use an ~~voltage~~ bridge. The L298 chip contains two standard H-bridges capable of driving a pair of DC motors, making it deal for building of two-wheeler robotic platform.

- Component List
- Arduino UNO R3
- 165 DC motor with Encoder
- 250 KΩ potentiometer

Code:

```
# include <stepper.h>
  const int  steps Per Revolution = 200;
  Stepper  myStepper (Steps Per Revolution ,8,9,10,11);

  void setup () {
     my stepper  setspeed (20);
  }
  void loop() {
    int  sensorread = analogRead (A0);
    int  speed  = map (sensorread, 0, 1023, 0, 100);
    mystepper.setspeed (speed);
    mystepper.step (steps Per Revolution / 100);
    delay (1000);
  }
```

In order to drive the stepper motor we will be using a technique called "Half stepping". The motor used in this project has 200 step count with one phase stepper excitation. i.e energising only one phase at a time, we can achieve the normal 200 step revolution with least power consumption.

Applications:

- Numeric control of machine tools.
- Used in floopy disc, printer, electric watches.
- It uses in x - y plotter and robotics.
- Wristwatches.

Conclusion: Thus implemented stepper motor using arduino uno R3 and understood all applications of the same.

Page No.

Date

FAQ's

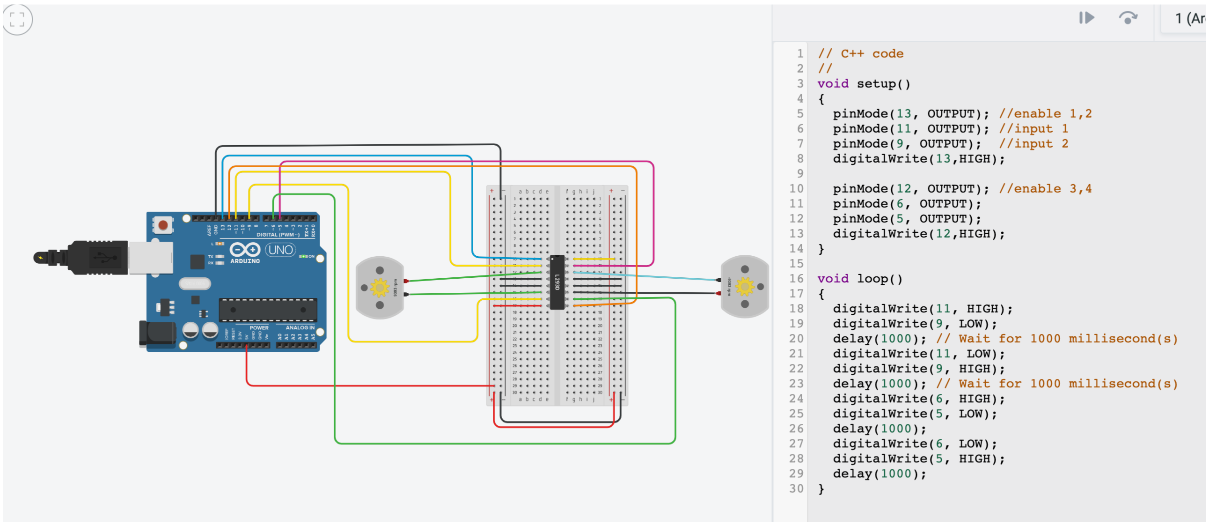
Ans 1) We use motor drivers to give high power to the motor by using a small voltage signal from a microcontroller or a control system. If the microprocessor transmits a high ~~power~~ input to the motor driver. The driver will rotate the motor, in one direction keeping the one pin as HIGH and one pin as LOW.

Ans 2) A simple way to choose a stepper drive is to look for four things - voltage, current, microstepping and maximum step pulse rate. Ensure that the drive can handle a wide range of currents so that you can test the system at different voltage levels to fit your application.

Ans 3) Arduino, Rospberry Pi, Node MCU | ESP, STM & NUVOTON, PIC and ATmel, FFGA and Programmers.

Ans 4) Linear Actuators - solenoid
Rotatory Actuators - DC motor, servo motor
LED, Buzzer, etc are some examples of actuators.

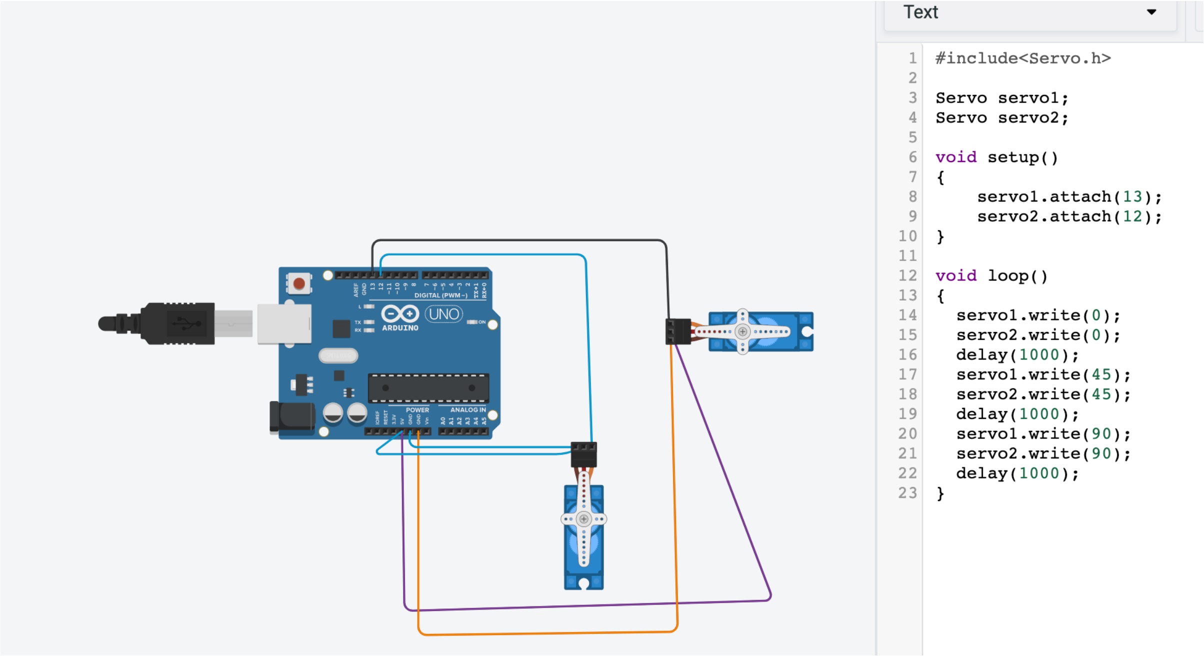
Ans 5) The relay permits a small amount of electrical current to control high current loads. When voltage is supplied to the coil, small current passes through the coil, resulting in a larger amount of current passing through the contacts to control the electrical load.



```
// C++ code
//
void setup()
{
  pinMode(13, OUTPUT); //enable 1,2
  pinMode(11, OUTPUT); //input 1
  pinMode(9, OUTPUT);  //input 2
  digitalWrite(13,HIGH);

  pinMode(12, OUTPUT); //enable 3,4
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
  digitalWrite(12,HIGH);
}

void loop()
{
  digitalWrite(11, HIGH);
  digitalWrite(9, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(11, LOW);
  digitalWrite(9, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(6, HIGH);
  digitalWrite(5, LOW);
  delay(1000);
  digitalWrite(6, LOW);
  digitalWrite(5, HIGH);
  delay(1000);
}
```



```
#include<Servo.h>

Servo servo1;
Servo servo2;

void setup()
{
    servo1.attach(13);
    servo2.attach(12);
}

void loop()
{
  servo1.write(0);
  servo2.write(0);
  delay(1000);
  servo1.write(45);
  servo2.write(45);
  delay(1000);
  servo1.write(90);
  servo2.write(90);
  delay(1000);
}
```