Lab Report

Course number: EE101	**Lab Section: A04**
Student Name: Helen Wang	**Student ID: hwang2**
Lab GTA:	**Lab TA: Hannah, Artur**

TA ONLY

Lab Objectives	OK	NOK	**Points Earned**	0 - Wholly inadequate. 1 - An attempt, but missing many pieces, unprofessional parts, missed the point. 2 – More than one unsatisfactory section. 3- One unsatisfactory section. 4 - Spot on. All sections delivered as required. Professional
Materials	OK	NOK		
Exercise Req/Proc	OK	NOK		
Conclusions	OK	NOK		

Lab Title: using a stepper motor with Arduino controller

Section A: Lab Objectives

In your own words list the objectives of this lab.

A stepper motor is a unique type of motor where the stator windings can be separately excited to precisely control the motion of rotor made of permanent magnets. Since there are several separate windings and they each have their own control, there are several wires connected to a stepper motor. Depending on how one chooses to excite the windings a stepper motor can be used for continuous rotation or precise positioning with digital signals from a microcontroller. Interface your stepper to the Arduino board. Interface your joystick to the Arduino board. Use stick to control motor speed/direction. Use the push button to increment stepper one step.

Section B: Equipment and component materials used

Electronics measurement equipment, power supply, electronics components, etc. Only what you used for this lab.

- Makerspace Kit
  - Datasheet for 28YBJ-48 Stepper Motor
  - Datasheet for ULN2003 Stepper Motor Driver
  - Joystick
- Optional: O-scope
- Optional: Power Supply.

Section C: Reference works and soft materials used

Text book sections, Datasheets, On-line sources, Code snippets, etc. Be thorough and give credit where due.

Dr.Gallagher(.n.d), Arduino digital input/output. Department of Electrical Engineering. Accessed 02/28/2020

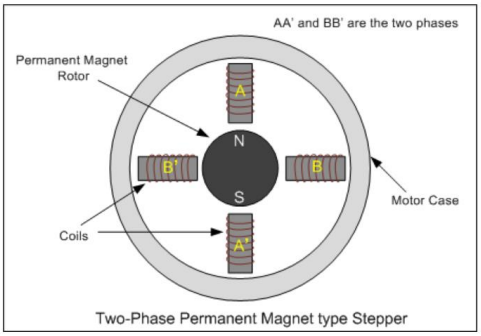
Section D: Exercises

Exercises have unique requirements that are met by your "unique" solutions. Multiple exercises fulfill our lab objectives. List the lab exercise requirements here.

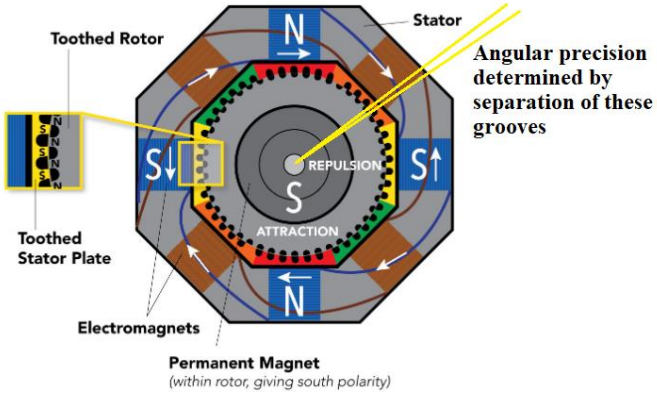
Requirements:

Procedure:

Part1: Establish contact with stepper



Two-Phase Permanent Magnet type Stepper



Part II. Establish contact with the joystick

The goal in this part is to study the function of the joystick.

Build the circuit as in Figure 4 without disconnecting the stepper motor. Make sure to use the same analog inputs and digital input.
Use the serial monitor to watch the value of xPosition and yPosition as you move the joystick.
Note the function of the mapping function. Use this to control the range of the values you're using. MapX can be made to vary between any value.
Modify it so that it returns a -10 < MapX < 10 where MapX = 0 is not adjustment on the joystick

Part III. Integrate the Stepper and joystick.

Create an application that uses the joystick to control the speed and direction of the stepper motor. Your final application should have these characteristics:

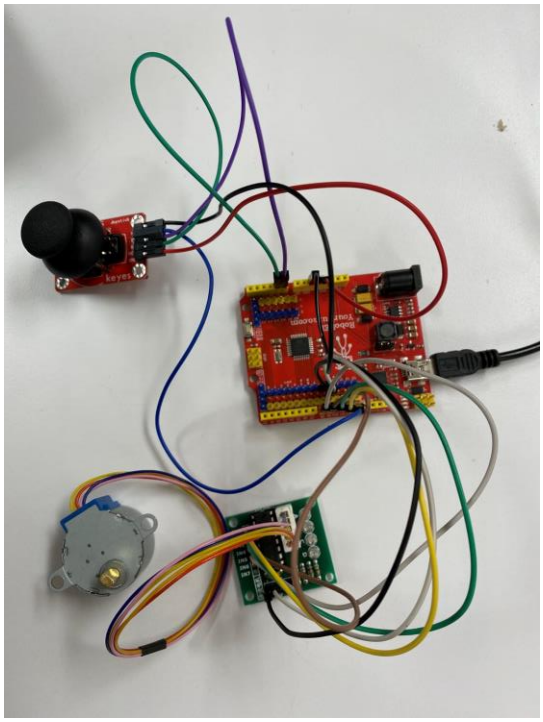
- X position of the joystick controls the stepper motor.
- The stepper should be motionless when the joystick is in its home position.
- It should continuously increase stepper speed as you adjust MapX eventually reaching full speed when the stick is fully pushed to one side.
- It should do this in both directions.
- Pushing the button slows the motion by a factor of 2 but otherwise does not change how the system behaves.

This task is made more difficult because your values for MapX are supposed to be proportional to speed but the motor speed is controlled by a delay, or 1/MapX. In order to do this you might consider making a lookup table or LUT. The table might have these values. You will need to fill in the rest.

MapX	Delay	
0	Turn off motor	Motor should be still
1	10000	
2	5000	
3	3300	
9		
10	1000	full speed

Solution/Procedure:

What did you do to fulfill the exercise requirements?



Code:

```
#define IN1  8
#define IN2  9
#define IN3  10
#define IN4  11
int Steps = 0;
int Direction = 0;
int number_steps=512;//= 2048/4

int VRx = A0;
int VRy = A1;
int SW = 12;

int xPosition = 0;
int yPosition = 0;
int SW_state = 0;
int mapX = 0;
int mapY = 0;

void setup() {
  Serial.begin(9600);

  pinMode(VRx, INPUT);
  pinMode(VRy, INPUT);
  pinMode(SW, INPUT_PULLUP);
```

```
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  // delay(1000);

}

void loop() {
  xPosition = analogRead(VRx);
  yPosition = analogRead(VRy);
  SW_state = digitalRead(SW);
  mapX = map(xPosition, 0, 1023, 0, 100);
  mapY = map(yPosition, 0, 1023, 0, 100);

  Serial.print("X: ");
  Serial.print(mapX);
  Serial.print(" | Y: ");
  Serial.print(mapY);
  Serial.print(" | Button: ");
  Serial.println(SW_state);

  delay(100);


  //1 rotation counter clockwise
  stepper(number_steps);
  delay(1000);
  //1 rotation clockwise
  stepper(-number_steps);
  delay(1000);
  //Keep track of step number

}




void stepper(double nbStep){
  if(nbStep>=0){
    Direction=1;
  }
  else{
    Direction=0;
    nbStep=-nbStep;
  }

  for (int x=0;x<nbStep*8;x++){

      delayMicroseconds(1000);
```

```
    if(Direction==1){ Steps++;}
    if(Direction==0){ Steps--; }
    if(Steps>7){Steps=0;}
    if(Steps<0){Steps=7; }


   switch(Steps){
    case 0:
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, HIGH);
    break;
    case 1:
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, HIGH);
      digitalWrite(IN4, HIGH);
    break;
    case 2:
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, HIGH);
      digitalWrite(IN4, LOW);
    break;
    case 3:
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, HIGH);
      digitalWrite(IN3, HIGH);
      digitalWrite(IN4, LOW);
    break;
    case 4:
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, HIGH);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, LOW);
    break;
    case 5:
      digitalWrite(IN1, HIGH);
      digitalWrite(IN2, HIGH);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, LOW);
    break;
    case 6:
      digitalWrite(IN1, HIGH);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, LOW);
    break;
    case 7:
      digitalWrite(IN1, HIGH);
      digitalWrite(IN2, LOW);
```

```
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, HIGH);
    break;
    default:
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, LOW);
      digitalWrite(IN3, LOW);
      digitalWrite(IN4, LOW);
    break;
  }

  }
}
```

Section E: Lab Conclusion

In this lab, we learned about how to use code to control the stepper and combine stepper and joy stick together. By coding to control the stepper and joy stick, the joy stick can control the stepper motor rotating directions. During the lab, my joy stick didn't work as the table goes but by adding if-else function it can control the range.