Polytechnic University of Puerto Rico

Electrical & Computer Engineering and Computer Sciences Department

COE 4341 Microcomputer Interfacing Laboratory

Laboratory No 6: Stepper Motor

ID	LAST NAME	FIRST NAME
130541	Sierra Bello	Gabriel
109648	Salgado Rivera	Daniel E.
103733	Meléndez Nieves	Josué
116762	Madera Torres	Elian F.

SUPERVISOR SIGNATURE _____

DATE: **April 25, 2022**

May 5, 2022

Table of Contents

Introduction

In this experiment, we were tasked with powering and controlling a stepper motor. Our goal was to successfully connect the stepper motor, then make it rotate 360 degrees clockwise or counterclockwise. We had to make it so if we input “CCC” on the serial monitor it will be clockwise, if we input “CCW” it will go counterclockwise. Also, we had to make it if we input “STP”, then to serial monitor will ask you how many steps you want the stepper to rotate, the user inputs the steps, and the stepper must rotate that many steps. We had to program the Arduino board to control the stepper motor in this exact fashion and complete the experiment.

Objective

Our objectives throughout this experiment are to learn how to describe the advantages and disadvantages of stepper motors versus DC motors, be able to describe the difference between unipolar and bipolar stepper construction and full-step drive control, learn how to use the provided Arduino Stepper library for basic stepper control and understand how to utilize serial communication to the Arduino from the serial monitor.

Methods

Procedure C Language Program

In this experiment, we established a default rotation for the stepper motor. After this, we utilize serial communication to interact with the user. The user may choose to rotate the motor clockwise or counterclockwise. Once he chooses a direction; the motor will continue to rotate that way until told otherwise. Finally, the user may request the motor to stop which he will do. After this, the user will be asked how many steps he wants the motor to rotate. The motor will oblige and then return to its default rotation cycle.

Source Code for the C Language Program

Stepper2:

```
/* Do a Rotation (4095 steps) Clock Wise
 them stop for 3 seconds
            Roman Lopez Ph. D. 10/24/2014
   BYJ48 Stepper motor code
   Connect :
   IN1 >> Pin 3
   IN2 >> Pin 4
   IN3 >> Pin 5
   IN4 >> Pin 6
   VCC ... 5V Prefer to use external 5V Source
   Gnd
      10/24/2013
  */
#define IN1  22
#define IN2  23
#define IN3  24
#define IN4  25
int Steps = 0;
unsigned long last_time;
unsigned long currentMillis ;
int steps_left=4095;
long time;
boolean Direction;
String esckey, stopSteps;
void setup()
{
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
// delay(1000);
Serial.begin(9600);
Serial.println("Enter CCC for ClockWise or CCW for Counter-ClockWise");
}
void loop()
{
       esckey = Serial.readString();
       if(esckey[0] == 'C')
       if(esckey[1] == 'C')
       if(esckey[2] == 'C')
         Direction = 0;
```

```
       if(esckey[0] == 'C')
       if(esckey[1] == 'C')
       if (esckey[2] == 'W')
         Direction = 1;
         Serial.print("Direction is:");
         Serial.println(Direction);
       if(esckey[0] == 'S')
       if(esckey[1] == 'T')
       if(esckey[2] == 'P'){
         Serial.println("How many steps do you want?");
           while (Serial.available() == 0) {}
           stopSteps = Serial.readString();
           steps_left = stopSteps.toInt();
       }
  while(steps_left>0)
     {
     currentMillis = micros();
     if(currentMillis-last_time>=1000)
       {
       stepper(Steps);
       time=time+micros()-last_time;
       last_time=micros();
       steps_left--;
       }
  }
  delay(5000);
  steps_left=4095;
}
//HALF Stepping switching Mode
void stepper(int xw)
  {
   switch(xw)
    {
     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;
   }
   SetDirection();
}

void SetDirection(){
```

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

Results

https://youtu.be/oN_NpDQPxp8

Conclusion

In this experiment, we learned the differences between DC motors and Stepper motors. We also learned how to utilize serial communication to make the stepper motor perform accordingly. We gave the stepper motor functions to rotate in a certain direction or stop rotating in general. Ultimately, we learned the basic controls of the stepper motor and the Arduino library used to interact with these motors.