

Sensors and Motors Lab

Gauri Gandhi

Team G Robographers

Teammates:

Rohit Dashrathi

Jimit Gandhi

Tiffany May

Sida Wang

ILR #1

October 16, 2015

a. Individual Progress

For the Sensors and Motors Lab, I was responsible for interfacing a Force Resistive Sensor and a Stepper Motor with the Arduino Uno. I also worked collaboratively with Tiffany May to integrate the hardware for the complete task.

I. **Force Resistive Sensor**

Circuit: To measure the force applied, I used FSR400. This analog sensor results in a decrease in resistance on increasing the force. I connected one of its pins to the analog pin 1 of the Arduino. It was also connected in parallel to a pulldown resistor of 10kohm as shown in Figure 1. The other pin was grounded.

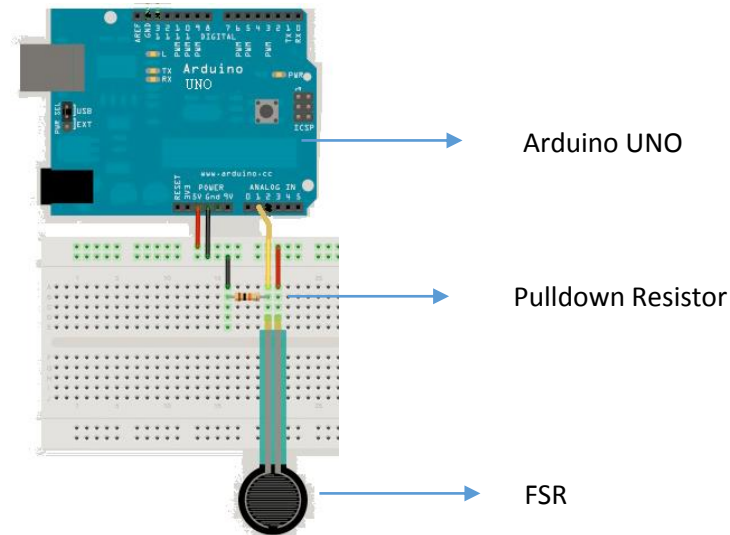


Figure 1 FSR Interfacing

Calculations and working: The encoded values of the force were converted from a 0-1023 scale to a 0-255 scale.

$$\text{fsrReading} = (\text{analog input}) / 4$$

II. **Stepper Motor**

Circuit: I used a Sparkfun EasyDriver (Stepper Motor Driver) to drive a Mercury SM-42BYG011-25 stepper motor. This is a bipolar motor with a step size of 1.8°. I first soldered the pins on the driver board. The connections were done as shown in figure 2. I set the MS1 to HIGH and MS2 pin to LOW to adjust the resolution to half step. The Enable was set to LOW.

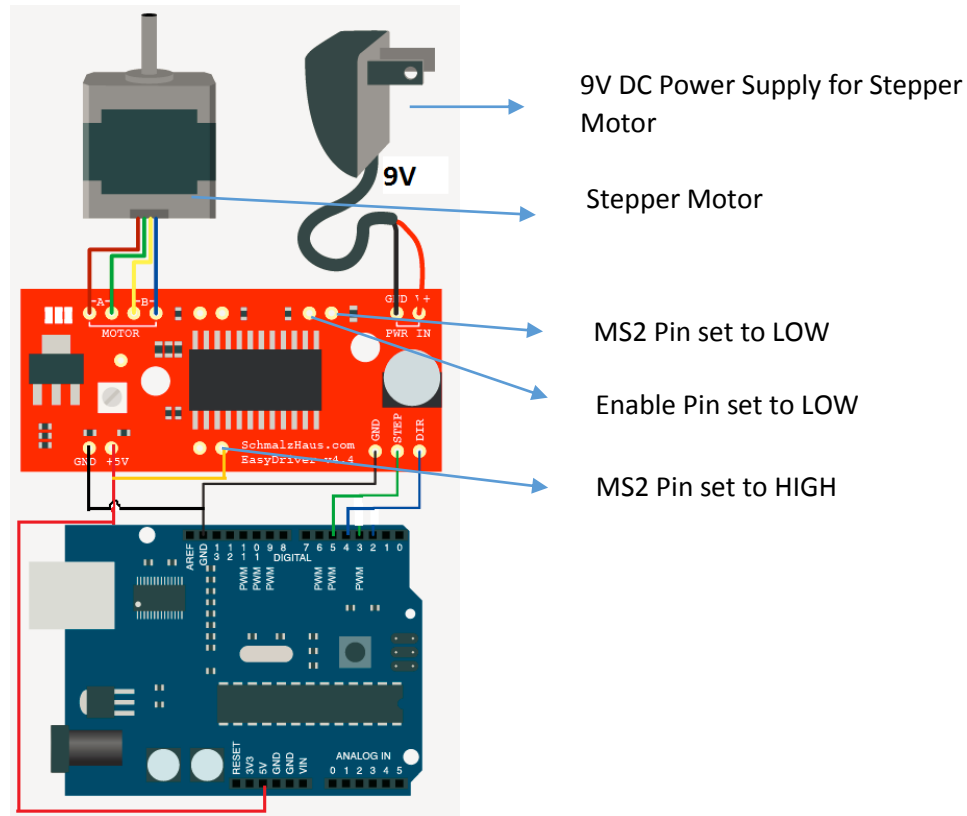


Figure 2 Stepper Motor Interfacing

Power Supply: I supplied 9V of DC Power to the driver for the motor. To avoid any noise or glitches in the input voltage, I used a 100uF capacitor in parallel to the supply. The input current was maintained at 0.25A so that the motor could run smoothly. To set the value of this current, I adjusted the potentiometer on the driver until the motor started moving.

Also a 5V supply for the driver circuit was provided from the Arduino board.

Calculations and Working: I first received the input for this interface on a scale of 0-360. This input value corresponded to the number of degrees of rotation required. Then, this user value was converted to number of steps using the given formula:

$$\text{\#Steps} = (\text{angle reading between 0-360}) / 1.8$$

The motor was made to run through a cycle of multiple ons and offs. The number of iterations of these cycles were decided by the number of steps calculated above. The speed of the motor was controlled by the delay (500us) after each on and each off. Same process was repeated for the motor to move in opposite direction. Only the value of Direction bit was changed from LOW to HIGH.

III. Interfacing the sensor and the motor

I merged the hardware of both the parts to a single Arduino board.

Calculations and Working: As the force sensor was pressed with varying force, the encoded values were printed on the Serial Port. These values were then mapped onto a scale of 0-360 using the following function:

```
Angle_read= map (fsrReading,0,255,0,360)
```

This value was then used to set the number of steps for the stepper motor as shown above.

The complete code for this interfacing has been attached in the Appendix 1.

IV. Progress in MRSD Project

In our Robographers project, I am collaboratively responsible for the Planning and Navigation part with Rohit Dashrathi. We are working on the breakdown of tasks required to be done under this module. We are also preparing an inventory of items required to be ordered to accomplish them. As of now, I am studying the previous year's team algorithms used for localization, planning and navigation. As most of the work for planning has to be done on ROS, I am going through the tutorials to learn the ROS basics.

b. Challenges

- I. At first I used the Pololu A4988 stepper motor driver. It worked for one day. Then maybe due to high current inflow it stopped working. I had to change both the motor and the driver as I was not sure where the problem was. Therefore for the second set of hardware, I first adjusted the current to a lower value without putting the motor. Then after connecting the motor, I set the current to 0.25.
- II. Shifting to the Sparkfun driver was a challenge as I had to solder all the pins again and change all the wired connections.
- III. At first, I was changing the pin status of MS1 and MS2 using programming to move varying step sizes. But that was not fulfilling the requirements of the task. So I permanently set those pins to HIGH and LOW respectively and varied the number of steps instead of the total step size. Rohan Thakker explained this concept to me.
- IV. When I first uploaded the merged code, the motor was running absurdly and gave noisy outputs. So I had to calibrate it according to the readings from the force sensor so that it ran smoothly in accordance with the inputs.
- V. It took us a lot of time to assemble all the hardware together as we wanted it to be neat and understandable. Also the wires had to be kept at some distance from each other. This way, they did not interfere with each other. Short circuit conditions were also avoided.

c. Team Work

To complete this task, we divided the work amongst the five of us in an efficient way so that each one of us got something new to learn. Sida took the GUI part. Rohit took the Ultrasonic sensor and the servo motor. Tiffany took the temperature sensor. She worked collaboratively with Rohit on the servo motor. They developed a switching operation to run their motor with two sensors. Jimit took the slot sensor and interfaced it with the DC motor. He implemented the PID controller for the motor. As mentioned above, I took the force sensor and the stepper motor. Tiffany and I did the hardware integration. For the software integration, all of us worked collaboratively as that was one of the most difficult parts of this lab task. This task was a great learning experience for us both individually and as a team.

d. Future Plans

We will be using our experiences from this task in our main MRSD project.

Technically,

- We are planning to implement a similar angle control of the servo and stepper motor for our pan tilt camera units in the project.
- For the Spring Validation Experiment, we might develop a similar GUI interface to control the Turtlebots.

Other than these,

- We will set strict deadlines for each one of us to complete our individual allotted tasks. This thing helped us in completing the PCB task much before the deadline.
- We will effectively strategize plans to manage the work load on each member of the team.

Appendix A

Code for FSR Sensor-Stepper Motor Interfacing

```
int fsrAnalogPin = A1; // FSR is connected to analog 1
int fsrReading; // the analog reading from the FSR resistor divider
int steps;
void setup() {
  Serial.begin(9600); //to send information via serial port
  pinMode(6,OUTPUT); // Enable
  pinMode(5,OUTPUT); // Step
  pinMode(4,OUTPUT); // Dir
  digitalWrite(6,LOW); // Set Enable low
}
void loop() {
  fsrReading = analogRead(fsrAnalogPin)/4; //take analog reading from analog 1
  Serial.print("Analog reading = ");
  Serial.println(fsrReading);
  steps=map(fsrReading,0,255,0,360)/1.8; //calculate no. of steps
  Serial.print("No of steps = ");
  Serial.println(steps);
  digitalWrite(4,HIGH); // Set Dir high
  for(int x = 0; x < steps; x++)
  {
    digitalWrite(5,HIGH); // Output high
    delayMicroseconds(500); // Wait 1/2 a ms
    digitalWrite(5,LOW); // Output low
    delayMicroseconds(500); // Wait 1/2 a ms
  }
  delay(150); // pause 150 ms
  digitalWrite(4,LOW); // Set Dir low
  for(int x = 0; x < steps; x++)
  {
    digitalWrite(5,HIGH); // Output high
    delayMicroseconds(500); // Wait 1/2 a ms
    digitalWrite(5,LOW); // Output low
    delayMicroseconds(500); // Wait 1/2 a ms
  }
  delay(150); // pause 150 ms
}
```