

# Sensors and Motor Control Lab

Individual lab report #1 || October 16, 2015

**RICHA VARMA**

**Team I**

Dorothy Kirlew

Pranav Maheshwari

Shivam Gautam

Mohak Bharadwaj

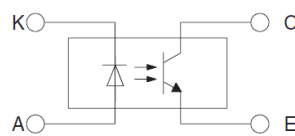
# 1. Individual Progress

The tasks undertaken by me were:

1. Interfacing Omron slot sensor with Arduino
2. Servo Motor control
3. Hardware interfacing of the system

## Interfacing of the Omron Slot Sensor

**Internal Circuit**

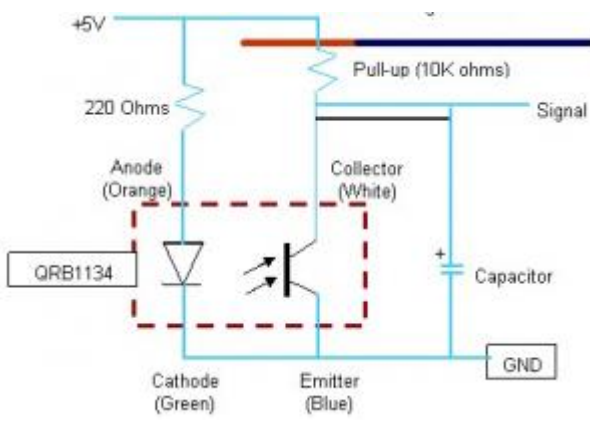


Terminal No.	Name
A	Anode
K	Cathode
C	Collector
E	Emitter

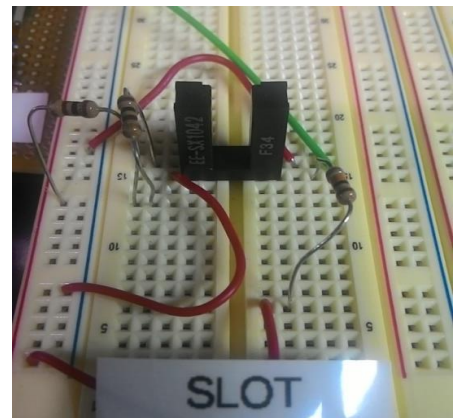
*Figure 1. Internal Circuit of Slot Sensor*

The slot sensor is a transmissive type photo-microsensor. It consists of an infrared emitter-receiver pair in a U-shaped configuration. Slot sensors are sometimes called optical fork sensors because of their "forked" shape. This style of fork sensor is used where a small target needs to be detected or a precise position needs to be consistently detected.

They detect objects that pass between the two arms—one with an infra-red LED, the other with the receiver photo-transistor. The fixed slot width provides reliable sensing of objects as small as 0.30 mm.



(a)



(b)

*Figure 2. (a) Connection Diagram of Slot Sensor (b) Slot sensor circuit on breadboard*

The following code was used to test the functionality of the slot sensor:

```
sketch_slot_sensor | Arduino 1.6.5
File Edit Sketch Tools Help
sketch_slot_sensor $
int slot=A1;
int sensor_val=0;

void setup()
{
  pinMode(slot, INPUT);
  Serial.begin(9600);
}

void loop()
{
  sensor_val= analogRead(slot);
  if(sensor_val>1000)
  {
    sensor_val=1023;
  }
  else
  {
    sensor_val=0;
  }
  Serial.println(sensor_val);
}

1 Arduino/Genuino Uno on COM3
```

## Servo motor control

The servo motor provided is a Hitech HS-225BB.



Figure 3. Hitech HS-225BB Servo moto

It has three leads: VCC, GND and CONTROL INPUT. The control input is connected to a digital pin of the Arduino which controls the angle of rotation of the servo.

The Arduino servo library provides an efficient way to control the servo. Use of the library disables `analogWrite()` (PWM) functionality on pins 9 and 10, whether or not there is a Servo on those pins.

Although servos work well when powered by the Arduino 5V supply, they draw considerable power, it is advisable connect them to a separate supply and not the +5V pin of Arduino.

The code to map sensor values to control the servo motor is:

```
sensor_val = map(sensor_val, 0, 1023, 0, 180);  
myservo.write(sensor_val);  
delay(50);
```

## Hardware integration and testing

I took up the task of hardware integration for the entire setup.

The starting point was soldering connectors for the DC Motor and Stepper Motor. Following that, I made the connections for the stepper motor driver module (Polulu md09b and later Schmalzhaus Easydriver). To complete this task properly, reading the datasheet of the driver ICs proved to be beneficial. Tuning of the potentiometer on the board was required to set the current limit on the driver. It was done to limit the current being supplied to the phases of the motor windings and ensure that the driver chip supplies optimum current to the motor and doesn't burn out itself due to high current.

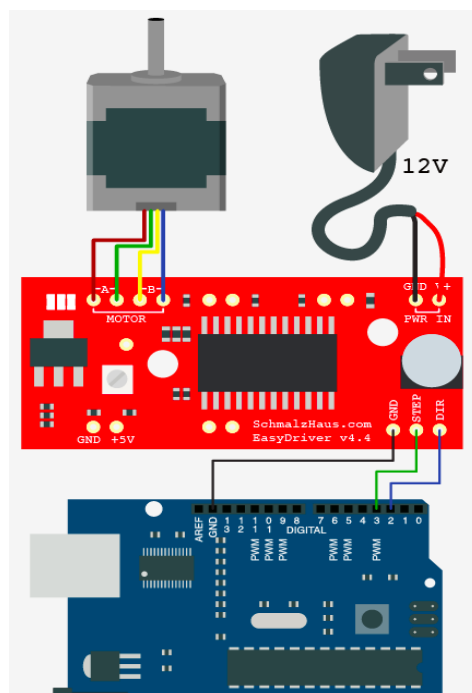


Figure 4. Connection diagram for stepper motor with EasyDriver

The MS0 and MS1 pins on the stepper driver were connected to ground (0V) for full step operation. Also, a 100uF electrolytic by-pass capacitor was connected to the input supply (12V) to the driver.

Thereafter, I interfaced the DC motor and its driver hardware with the Arduino after thoroughly reading the datasheet of the driver.

After ensuring that all the sensors and motors were functioning as they should as separate units, I built the hardware layout of the system, ensuring minimal wiring and reasonable spacing between components.

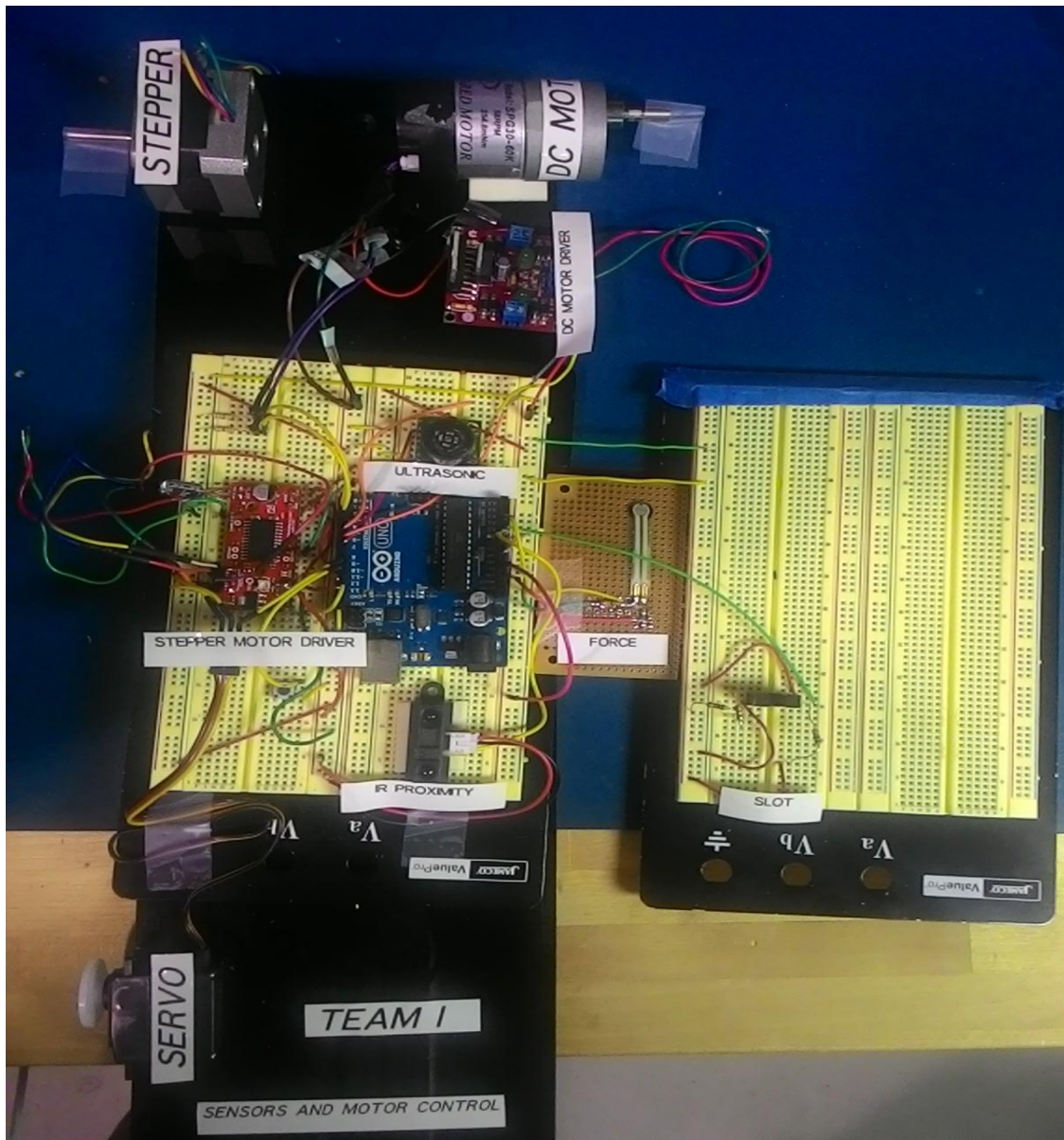


Figure 5. Hardware setup for sensors and motor control demonstration

## Challenges

The main challenges faced during this task were:

1. Making the stepper motor driver work: I started off with the Polulu md09b and got it working. After running the motor for about 10 minutes, the driver burnt out. On reading the datasheet, I found that the current limit needs to be set using the potentiometer on board by monitoring the reference voltage. I assembled the circuit after setting the proper limit as outlined in the datasheet but the second driver failed to work yet again. Due to time constraints, I obtained another driver, the Sparkfun EasyDriver and integrated it with the entire setup. Before doing so, I had set the current limit by changing the potentiometer on board to a reference voltage of 1.5V. The circuit worked as expected without any issues.
2. Another issue was the noise due in the ultrasonic sensor. Due to multipath nature of sound waves, this was expected. The method that finally worked in eliminating false readings was averaging over 10 readings and discarding values which were way out of the expected range.

## Teamwork

Work undertaken by each member:

Member	Tasks
Pranav Maheshwari	Stepper Motor, Sharp IR sensor, Functional layout for Arduino code
Shivam Gautam	DC Motor, Force sensor
Mohak Bhardwaj	GUI
Dorothy Kirlew	Ultrasonic sensor, GUI

The team worked with great coordination during execution of the entire task. Before anyone started working on their respective Arduino codes, Pranav created a functional layout for the entire code where everyone put their code in their respective defined functions. This proved to be a blessing at the time of final integration. Everyone completed their part well before time and integration went smoothly.

## Future Plans

Over the course of next few weeks, the team will work on four main areas:

**Communication/Network:** We aim to conduct a literature survey to select a scalable, reliable and reconfigurable network for vehicle to vehicle communication. Shivam plans to come up with top 3 options. Thereafter, the team will take an informed decision.

**Obstacle Detection:** We shall work on defining the obstacles in our test environment. Thereafter, we shall explore options for sensors and cameras to be used for obstacle detection. Pranav and Mohak will be working on this.

**Mobile Platform:** We have obtained Arduino Mega 2560 from the lab inventory and once the mobile platform arrives, we shall integrate the actuator control board with it. I will be working in this direction. I will also be looking at some of the ROS compatible mobile platforms available in the inventory and will explore their possible use.

**App Development:** We aim to narrow down on the language and tools for creating the Android app for our system. Dorothy and Mohak will be working on this as well as researching means for establishing a Bluetooth connection between the mobile platform and the user's phone.

#### REFERENCES:

Fig.1 : [https://www.components.omron.com/.../EE\\_SX1042\\_1010.pdf](https://www.components.omron.com/.../EE_SX1042_1010.pdf)

Fig.2(a): [robotics.stackexchange.com](http://robotics.stackexchange.com)

Fig.3: [www.modellhobby.de](http://www.modellhobby.de)

Fig.4: [bildr.org](http://bildr.org)