

## **ECE 110 honor project**

### **Automatically photoreceptive curtain**

**Team member: Ruhao Xia, Kaichen Le**

**Date: 05/08/16**

## **Introduction**

### Problem statement

People use curtain to adjust the light density in the room but sometimes they do not like changing the curtain frequently to get the best light density, even if it is a simple manual manipulation. For example, if someone is enjoying sleeping in the Sunday morning, close the curtain will be really annoying. Also, when someone is driving the car, he may need to adjust the light shield frequently, which also influences the driving experience. What's more is that, the disabled or the people who cannot walk need it when there are no other people staying with them. So we really need an 'intelligent' curtain that can just by itself, so that we will not longer be troubling with the changing natural light.

### Overview of solution

The basic idea is to use light sensor to measure the light density and use the motor to control the curtain open or close. The light sensor is a photoreceptive resistor which can change resistance with light. We combine it in series with another normal resistor and use Arduino board to read

voltage on the light sensor. After measurement, we determine a threshold value and program to control the motor with respect to certain range of value.

## Design

### Block Diagram

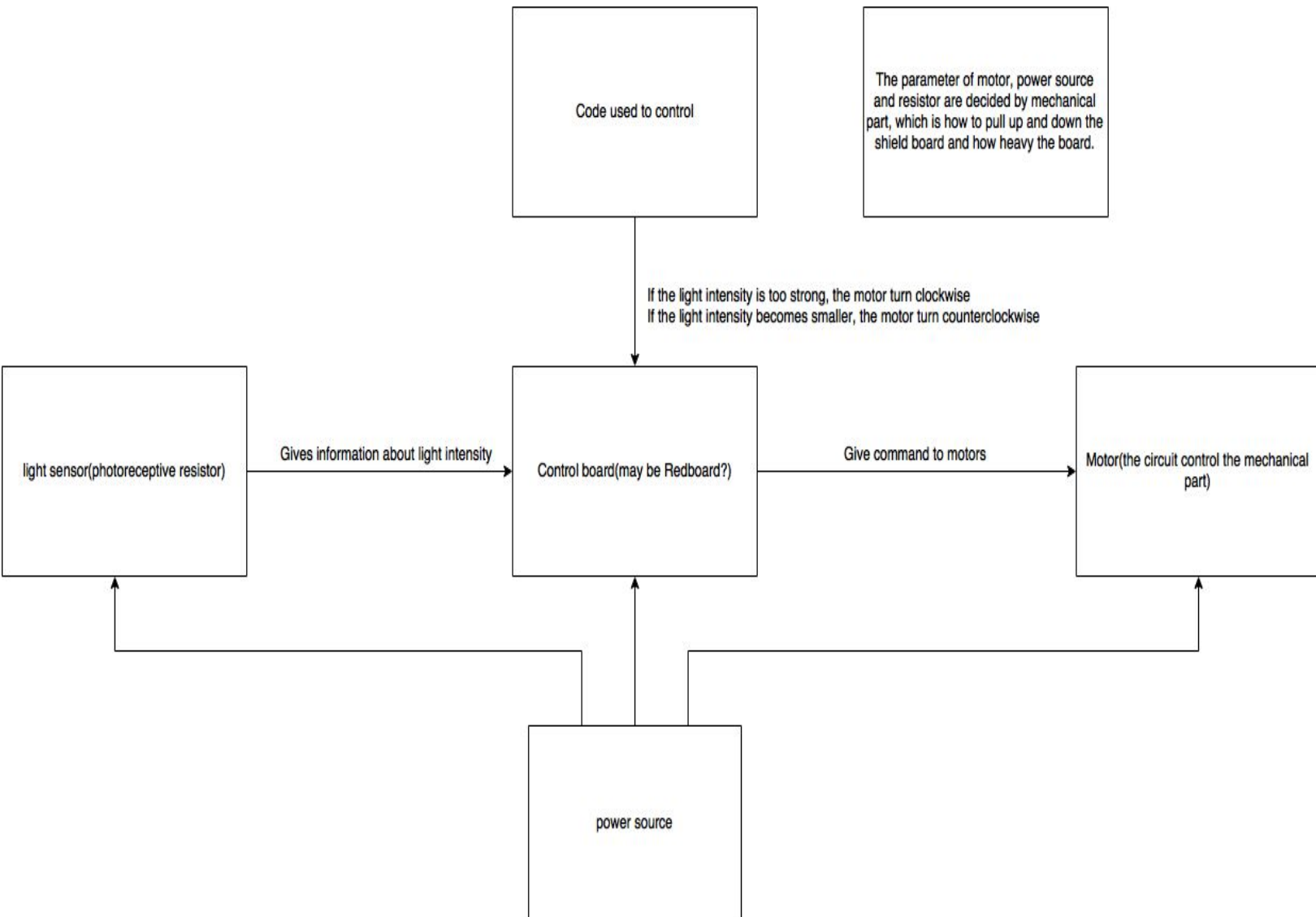


Figure 1: High level block diagram

Description: Our system includes five parts: sensor, control board, motor, power source, and code. The sensor we use to give feedback to control board is the light sensor and the voltage of the light sensor will change as long as the resistance changes with light density. We use Arduino board to read the voltage change and control the servo motor to rotate at certain speed and direction. When the light is strong outside, the motor will rotate to close the curtain and when the light is weak, the motor will open the curtain. We use arduino program to control the board, which is easy to learn. To operate the model, we use our computer USB port as power source which can supply 5 Volts.

#### Picture of device

Ski from the mechanical shop helped us to build the metal structure. Because the friction between the axis and structure was too large for servo to turn, we asked him to help us add a wheel inside the hole to reduce the friction. Moreover, in order to let the magnet swipe the hall effect sensor more easily we attached a breadboard which holds the sensor on the left inside the structure.

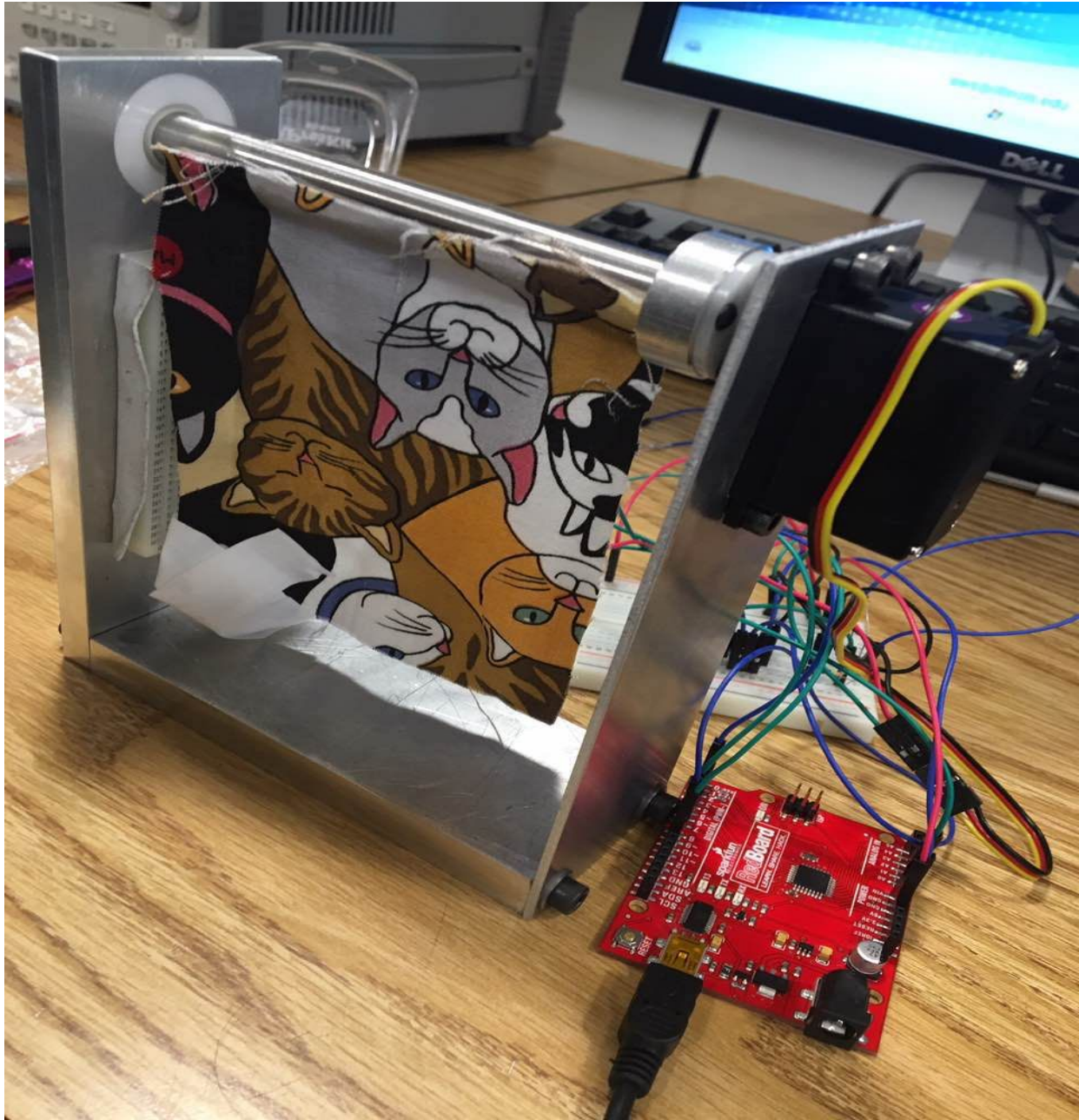


Figure 2: Picture of photoreceptive curtain's model

## Flow chart of software

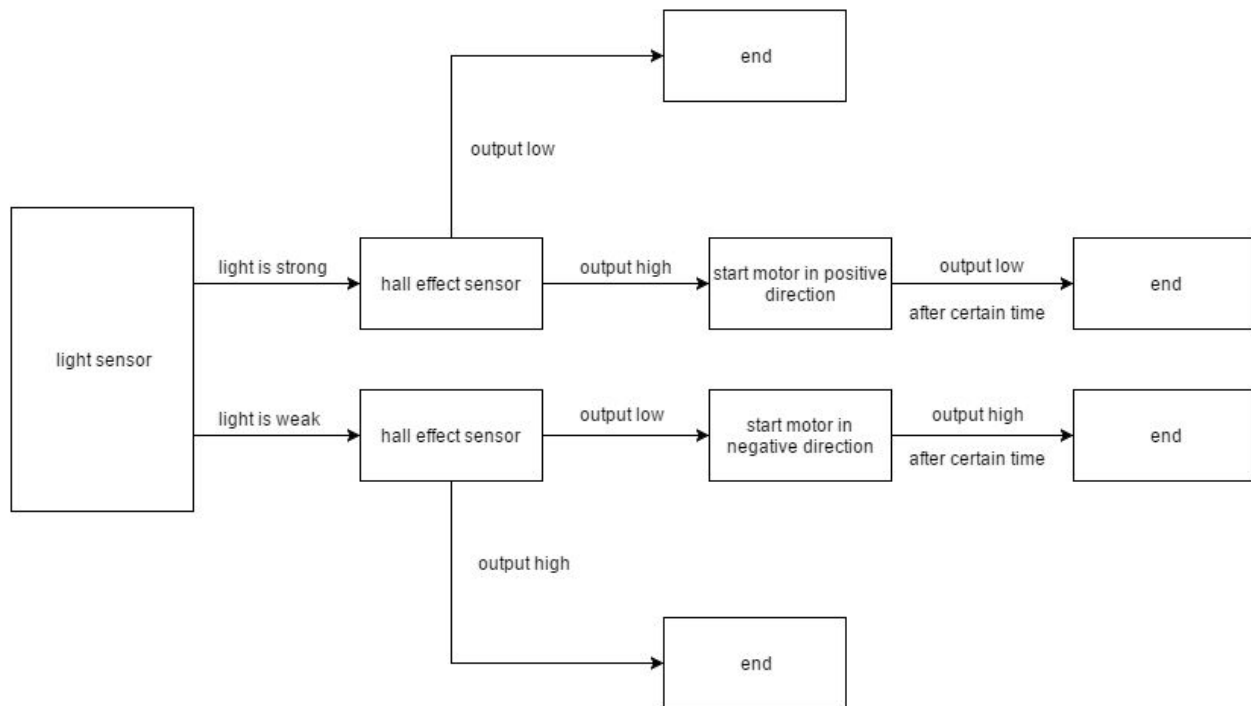


Figure 3: Flow chart of software

### Code Used

```
#include <Servo.h>
Servo servo1;
```

```
#include library
```

```
void setup() {
  Serial.begin(9600);
  servo1.attach(10);
  pinMode(2,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,INPUT);
  digitalWrite(7,HIGH);
  digitalWrite(2,LOW);
```

```
#Reset the D flip-flop
```

```

delay(200);
digitalWrite(2,HIGH);
digitalWrite(7,HIGH);

}

void loop() {
  Serial.println                #print out the value of voltage applied on light sensor
  (analogRead(0));
  if(digitalRead(8) == LOW && analogRead(0) < 250)
  {
    servo1.write(135);          #servo rotate in one direction with certain speed
    delay(2000);                #rotate with 2 seconds
  }
  while(digitalRead(8) == HIGH && analogRead(0) < 250)
  {
    servo1.write(95);           #servo stops
  }
  if(analogRead(0) > 350 && digitalRead(8) == HIGH)
  {
    servo1.write(45);           #rotate in opposite direction
    delay(2000);
  }
  while(digitalRead(8) == LOW && analogRead(0) > 350)
  {
    servo1.write(95);
  }

}

```

Block diagram of circuit

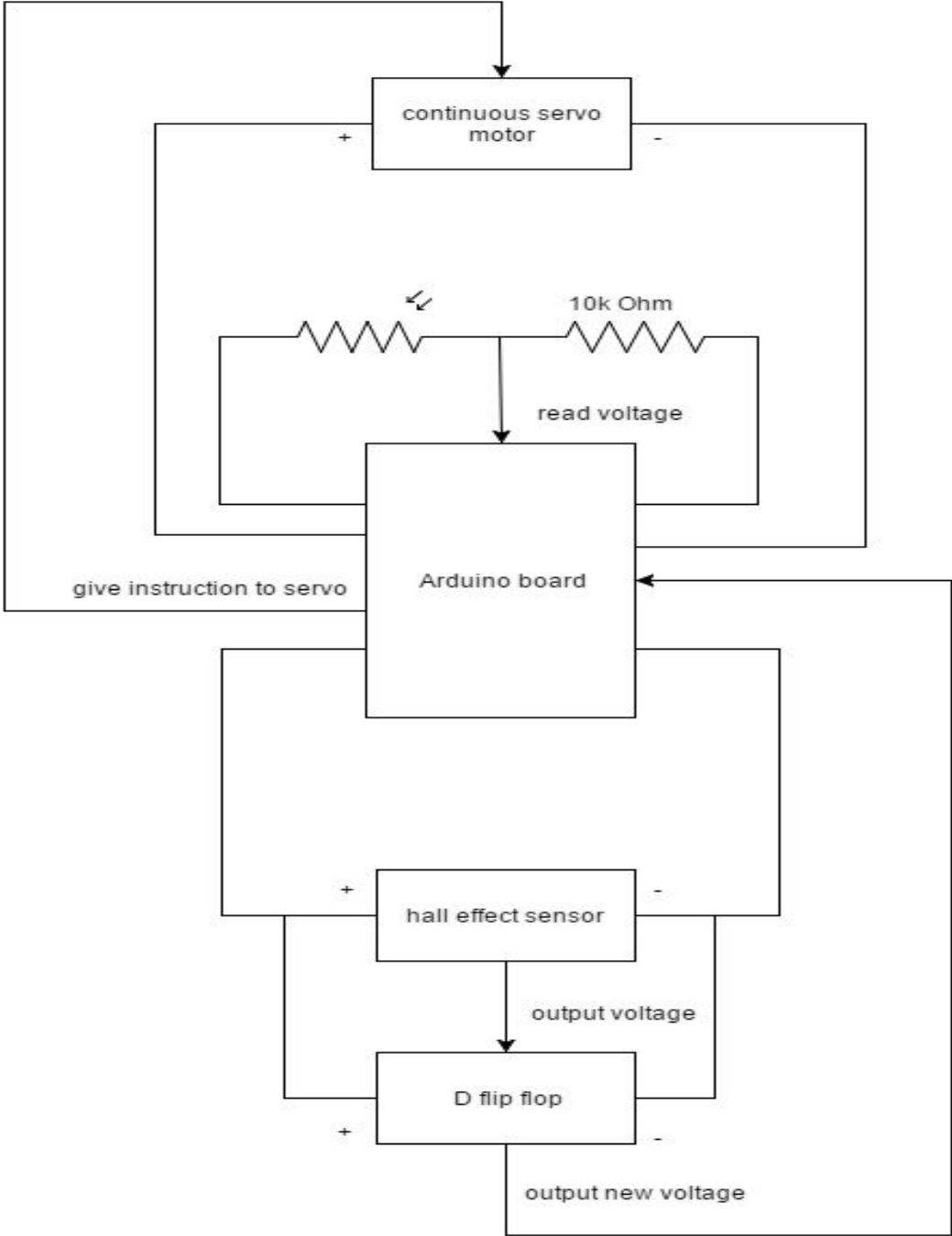


Figure 4: block diagram of the circuit

## Components

### 1. Light sensor

The resistance range for this light sensor is from 1k ohm (light) to 10k ohm (dark) so we combine a 10k ohm resistor to keep the voltage of the light sensor range from 0.5V to 2.5V.

According to datasheet, the sensitivity reaches maximum under the wavelength about 530 nm, and for our purpose, this value is good enough since we are using it to measure sunlight which includes large range of wavelength.

### 2. Servo - Generic High Torque Continuous Rotation (Standard Size)

We use servo motor because the speed and direction can be controlled with PWM signal. If we use DC motor, we may also install a H-bridge to change the direction and the speed may be unstable. In most case, the speed and precision cannot be achieved at the same time. Also, we choose continuous servo motor because it can turn over 180 degrees which is the limitation of a standard servo motor. To turn the curtain on and off, we need several rotations, but the standard servo motor cannot do that without a complex gear system.

### 3. Hall effect sensor (US1881)

We use hall effect sensor to make the servo stops at the position we want. Because the angle that continuous servo turns cannot be controlled, we have to find another way to make the servo stop. As a result we attached a magnet at the bottom of the shield so that every time shield get to a certain position the status of the hall effect sensor will change.



#### 4.D flip-flop (SN7474N)

Because of the characteristic of magnetic field, we have to use D flip-flop to work as an adjustment of the output signal of the hall effect sensor. Because the output voltage of hall effect sensor will change twice when the magnet swipes it, we use D flip-flop to catch the rising edge of the signal, and make a new output which will change only once when the magnet swipes the hall effect sensor.

Inputs		Outputs	
CLK	D	Q	Q
RISING	H	H	L
RISING	L	L	H

Table 1: truth table of D flip-flop

#### Results

When we power the device, Arduino first runs the code the reset D flip-flop once and make decisions based on two input voltage. One input voltage is the voltage of light sensor and another one is the output voltage of D flip-flop. If we shield light from the light sensor, and the curtain is at bottom, then the motor will rotate the curtain to the top. And at the process of rising, the magnet on the curtain will swipe the hall effect sensor so that the D flip-flop's output voltage will change. Once the curtain reaches top, it will stop. At this time, if we put a strong light on the light sensor, the voltage of light sensor will change from high to low and under this condition, the curtain will go down. Also, during the process of going down, the output of D flip-flop will

change so that it will stop after finish one loop of program. By this way, our curtain can automatically operate to open and close according to external light.

### **Future work**

We are only using one hall effect sensor right now, so the light that get into the room is not controlled very concisely. As a result, we should use more hall effect sensors which can make shield stops at more different positions so that the light that come inside will be more concise.

What's more is that, we are only using one light sensor. In the future we can connect more light sensors, put them to many different places inside the room and develop a function to determine the light condition in the room so that the whole system can actually create an environment with suitable light density.

The magnet is an unstable factor in our design because the magnetic field is very complex. We consider to use an electromagnet to work as the same function as a magnet and that should be more stable and easy to control.

### **Conclusion**

Fortunately, everything worked well when we presented. We learned a lot as we doing the project. We burned a servo right before the presentation because the power we plug in was too big and the friction that stops servo from turning was to big. We learned we should use a power supply with reading before we plug in a battery pack and knew that we should check the

datasheet next time before powering the circuit. For next project, we should include a fuse wire in our design. And the most interesting thing we discover is that we can hack the standard servo motor to continuous one, though the programming can be a big problem.

## **Reference**

<https://www.arduino.cc/>