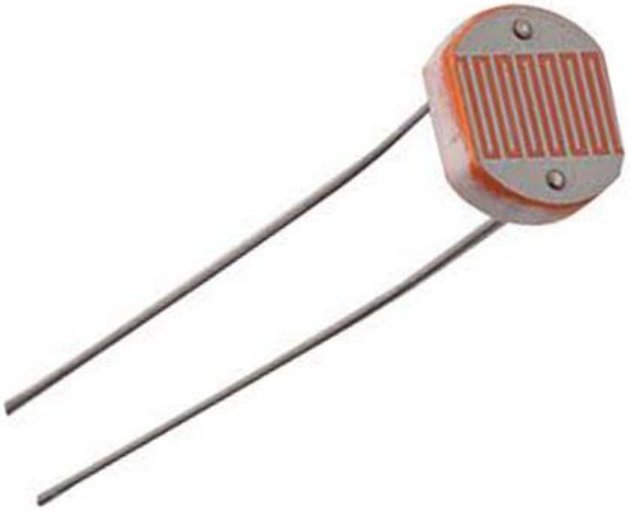
LECTURE 6
Light Sensor and Application: Solar Tracker

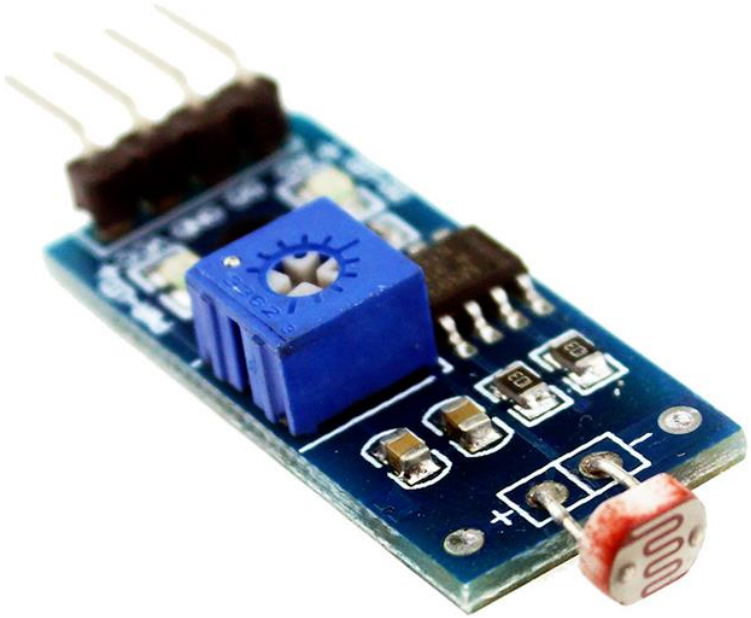
Instructor: Osman Gul, Ph.D. Candidate
Department of Mechanical Engineering
Korea Advanced Institute of Science and Technology (KAIST)

Syllabus:

Week	Date& Time*	Content
1		No Class (Course Add/Drop Period)
2	March 6 (Mon) 20-22	Orientation and Course Overview
3	March 13 (Mon) 20-22	Sensor Fundamentals
4	March 20 (Mon) 20-22	Arduino Fundamentals
5	March 27 (Mon) 20-22	Arduino Programming and Applications
6	April 3 (Mon) 20-22	Force Sensor and Application: Control Multiple LEDs
7	April 10 (Mon) 20-22	Light Sensor and Application: Solar Tracker
8	April 17	No Class (Midterm)
9	April 24 (Mon) 20-22	Humidity and Temperature Sensor and Application: Temperature Controlled Fan
10	May 1 (Mon) 20-22	Gas Sensor and Application: Smoke Detector
11	May 8 (Mon) 20-22	Sound Sensor and Application: Control LED by Clapping
12	May 15 (Mon) 20-22	Accelerometer Sensor and Application: Ping Pong Game
13	May 22 (Mon) 20-22	Ultrasonic Sensor and Application: Flappy Bird Game
14	May 29 (Mon) 20-22	Course Wrap-up
15	June 5	No Class (Final)
16	June 12	No Class (Final)

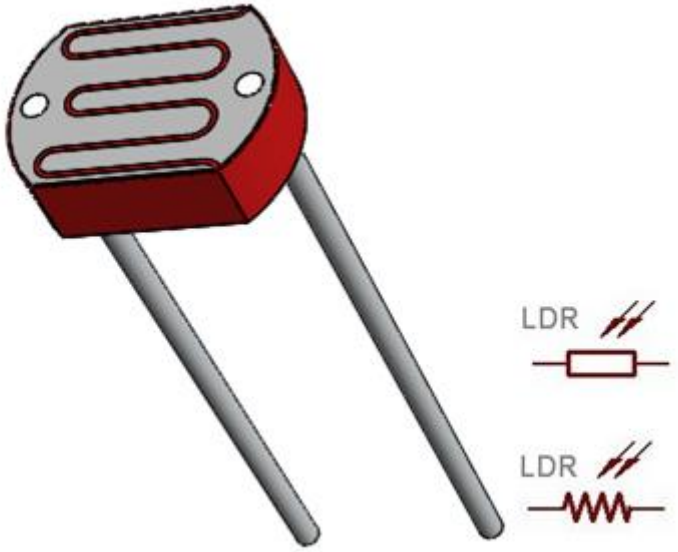
Light sensor:



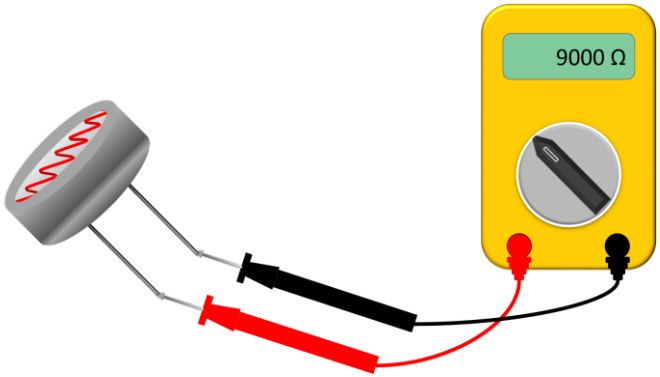


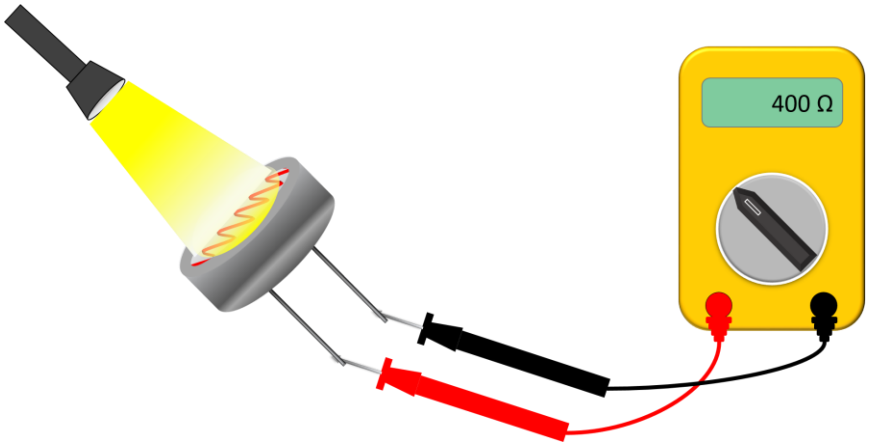
Photoresistor (or Photocell)

- ✓ A photoresistor or photocell is a light-controlled variable resistor.
- ✓ The resistance of a photoresistor decreases with increasing incident light intensity.
- ✓ A photoresistor can be applied in light-sensitive detector circuits, and light- and dark-activated switching circuits. It's also called light-dependent resistor (LDR).

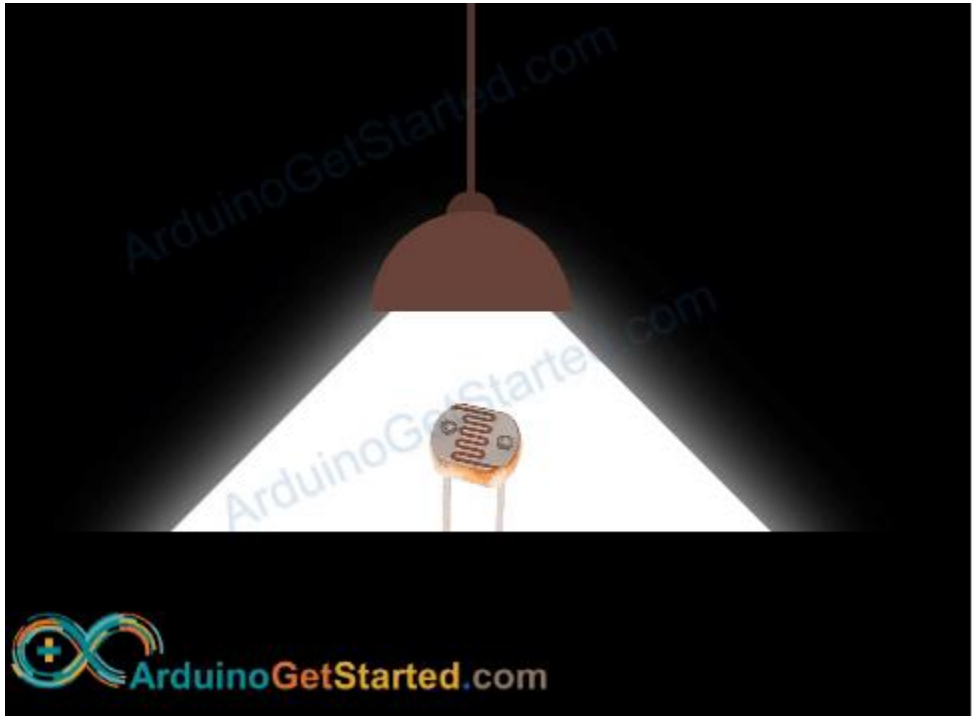


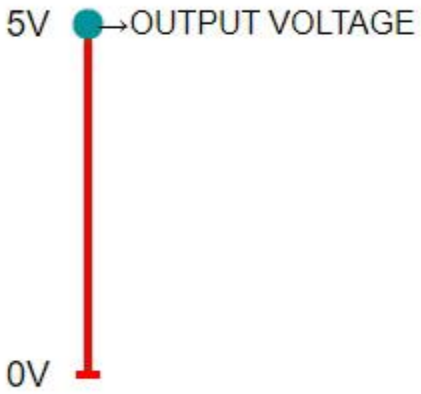
- ✓ When the resistor is placed in a dark room it will have a resistance of few Mega ohms and as we gradually impose light over the sensor its resistance will start to decrease from Mega Ohms to few Ohms.

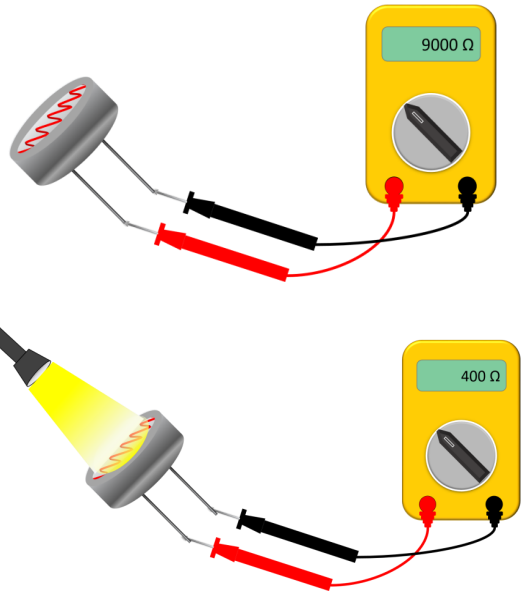


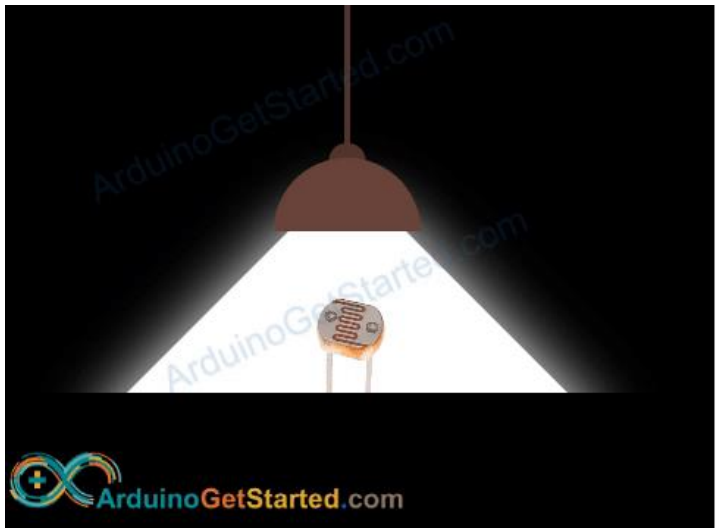


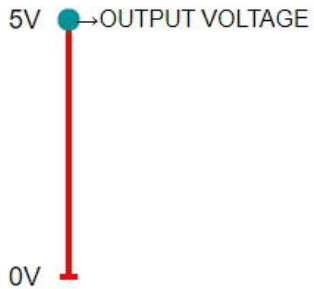
https://engineershub.co.in/


ArduinoGetStarted.com

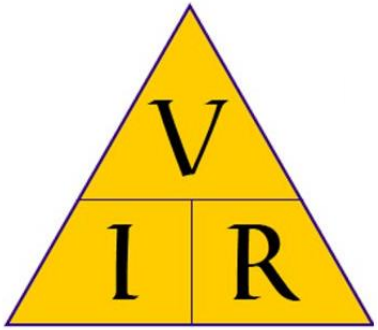

5V
OUTPUT VOLTAGE
0V







V = I R



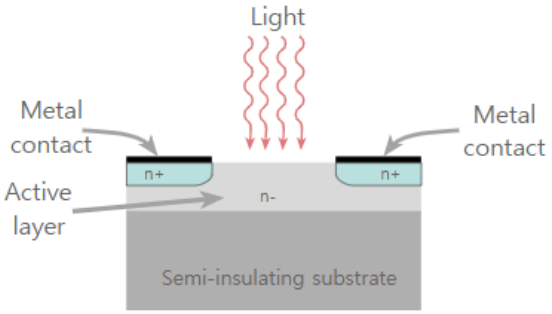
V	R	I	
Constant	↑	↓	$R \propto \frac{1}{I}$
↑	Constant	↑	$V \propto I$
↓	↑	Constant	$I \propto \frac{1}{R}$

- ✓ Light has Wave-particle duality nature. Which means that light has both particle-like and wave-like nature.
- ✓ When light falls on semiconductor material, photons present in light are absorbed by electrons and they get excited to higher energy bands.
- ✓ A photoresistor is a type of light-dependent resistor that varies its resistance values based on the light incident on it. These photoresistors tend to decrease their resistance values with an increase in the intensity of the incident light.
- ✓ Photoresistors exhibit photoconductivity. These are less photo-sensitive devices compared to photodiodes and phototransistors. Photoresistivity of a photoresistor varies with change in ambient temperature.

https://www.elprocus.com/

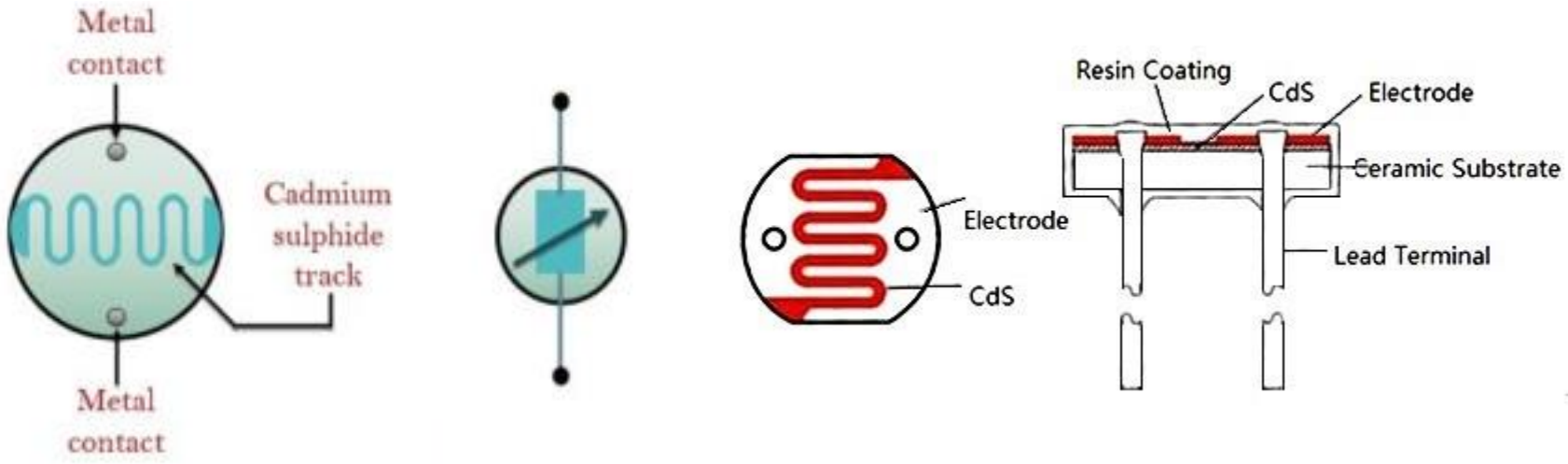
Working Principle

- ✓ The photoresistor doesn't have a P-N junction like photodiodes. It is a passive component. These are made up of high resistance semiconductor materials.
- ✓ When light is incident on the photoresistor, photons get absorbed by the semiconductor material.
- ✓ The energy from the photon gets absorbed by the electrons.
- ✓ When these electrons acquire sufficient energy to break the bond, they jump into the conduction band. Due to this, the resistance of the photoresistor decreases. With the decrease in resistance, conductivity increases.
- ✓ Depending upon the type of semiconductor material used for photoresistor, their resistance range and sensitivity differs.



https://www.elprocus.com/

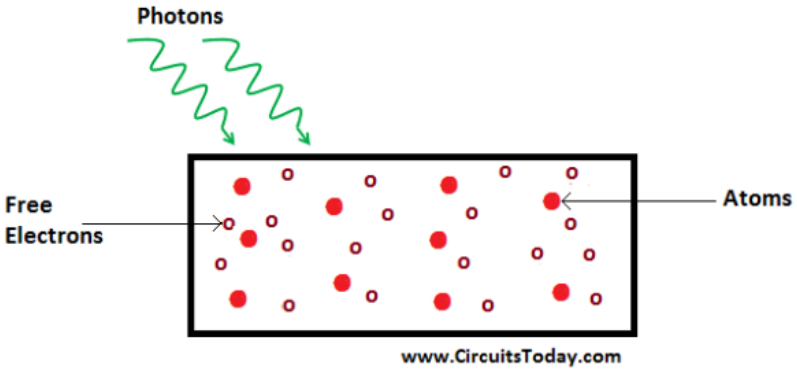
Photoresistor schematic

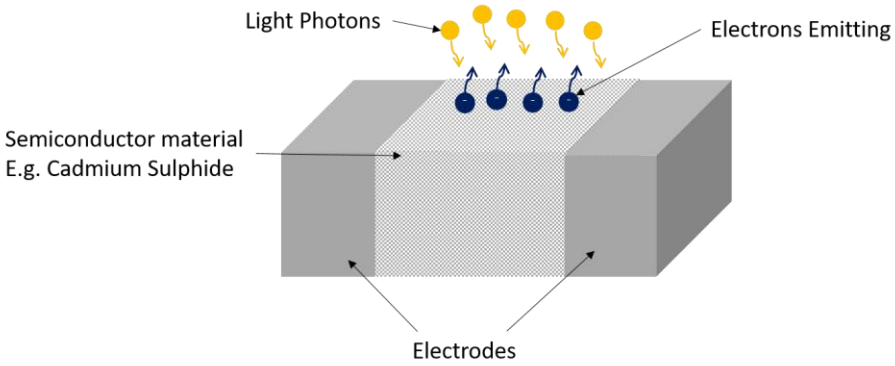


- ✓ Photoresistors are special resistors made of vulcanized or selenized semiconductor materials. The surface is also coated with a moisture-proof resin, which has a photoconductive effect. The working principle of the photoresistor is based on the internal photoelectric effect, that is, the electrode leads are mounted at both ends of the semiconductor photosensitive material, and the photoresistor is formed by packaging it in a tube case with a transparent window. To increase sensitivity, the two electrodes are often comb-shaped.

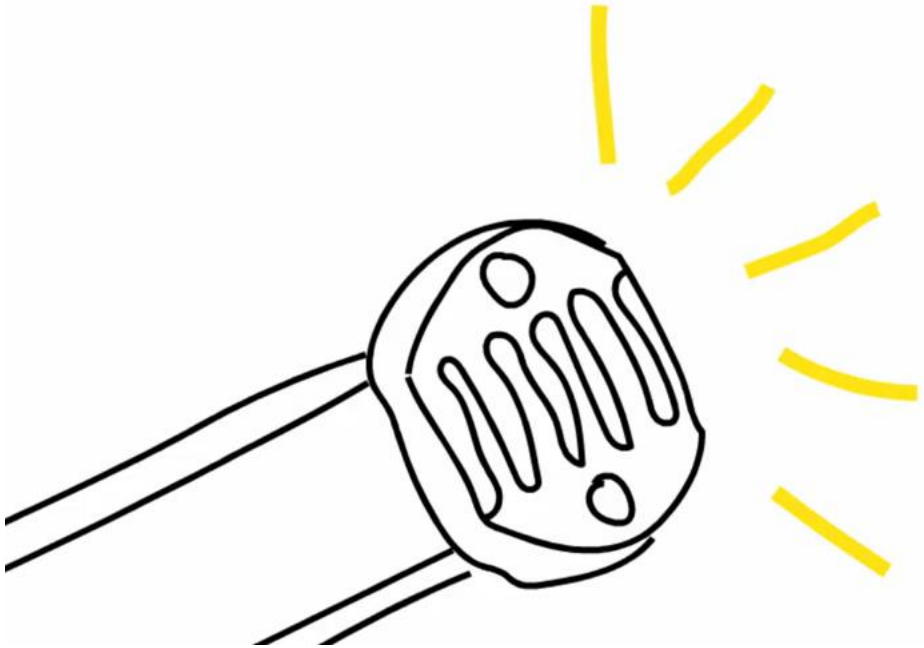
www.knowelectronic.com

- ✓ The conductivity of a semiconductor depends on the number of carriers in the semiconductor's conduction band. When the photoresistor is illuminated, the electrons in the valence band absorb the photon energy and then jump to the conduction band and become free electrons. At the same time, holes are generated. The appearance of the electron-hole pair makes the resistivity smaller. The stronger the light, the more photo-generated electron-hole pairs and the lower the resistance value. When a voltage is applied across the photoresistor, the current flowing through the photoresistor increases with increasing light. The incident light disappears, the electron-hole pair gradually recombines, the resistance gradually returns to its original value, and the current gradually decreases.

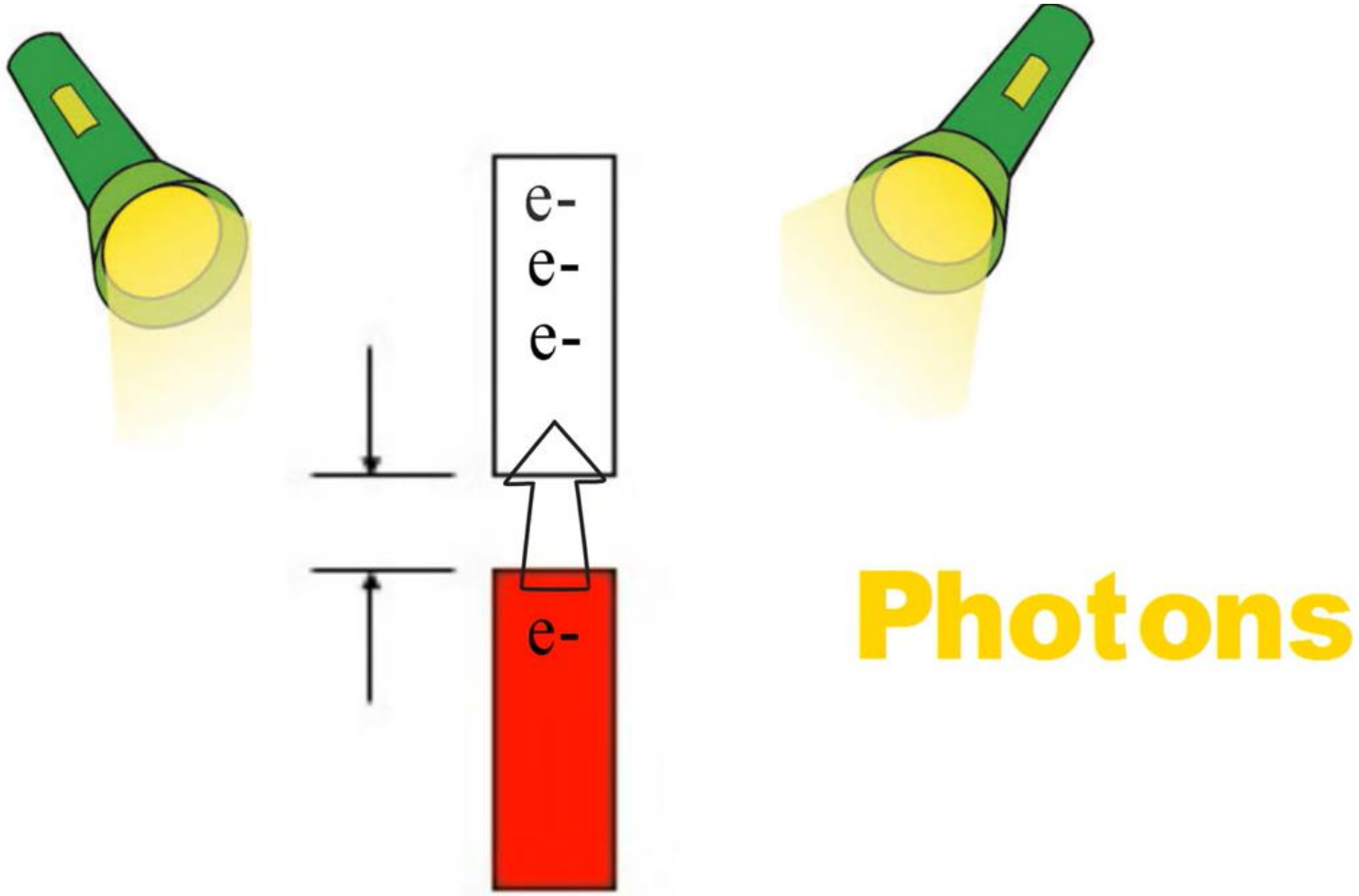




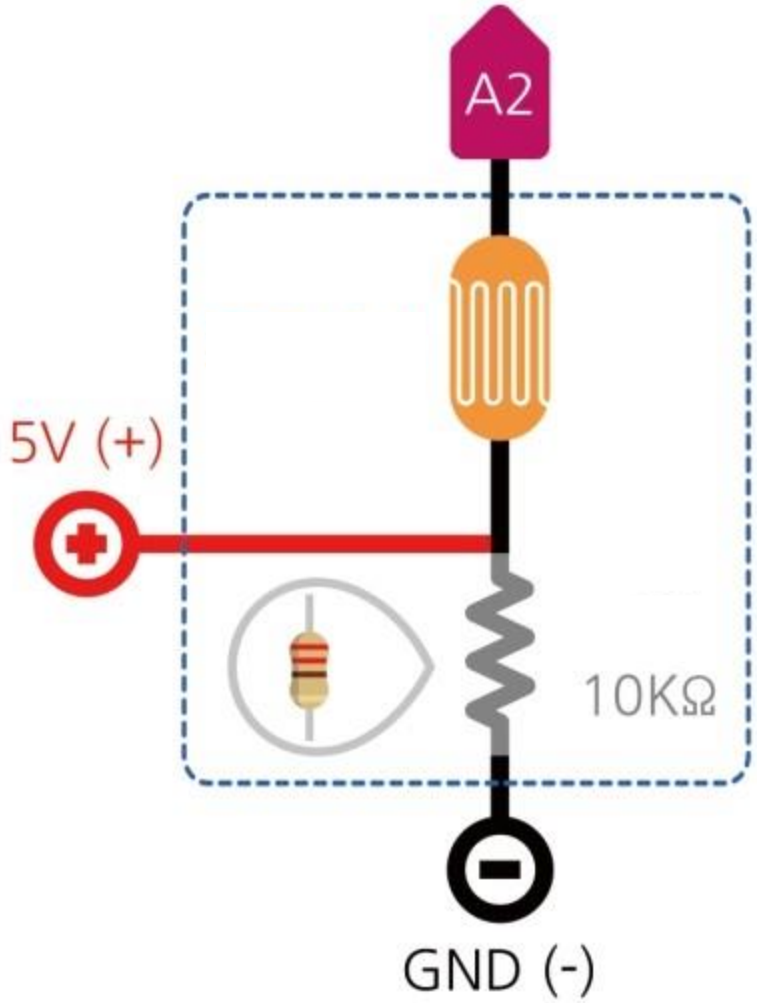
How does a photoresistor work?

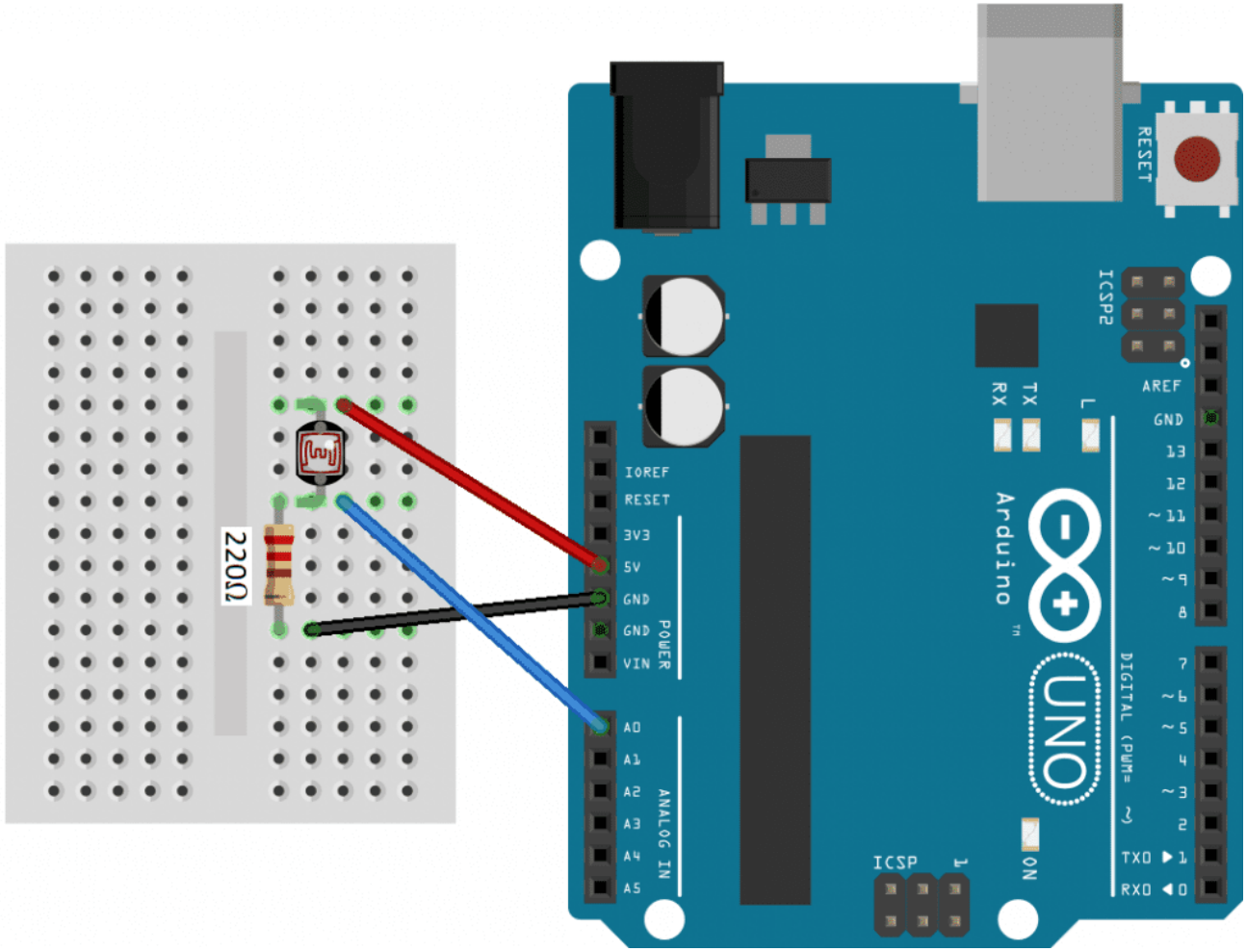


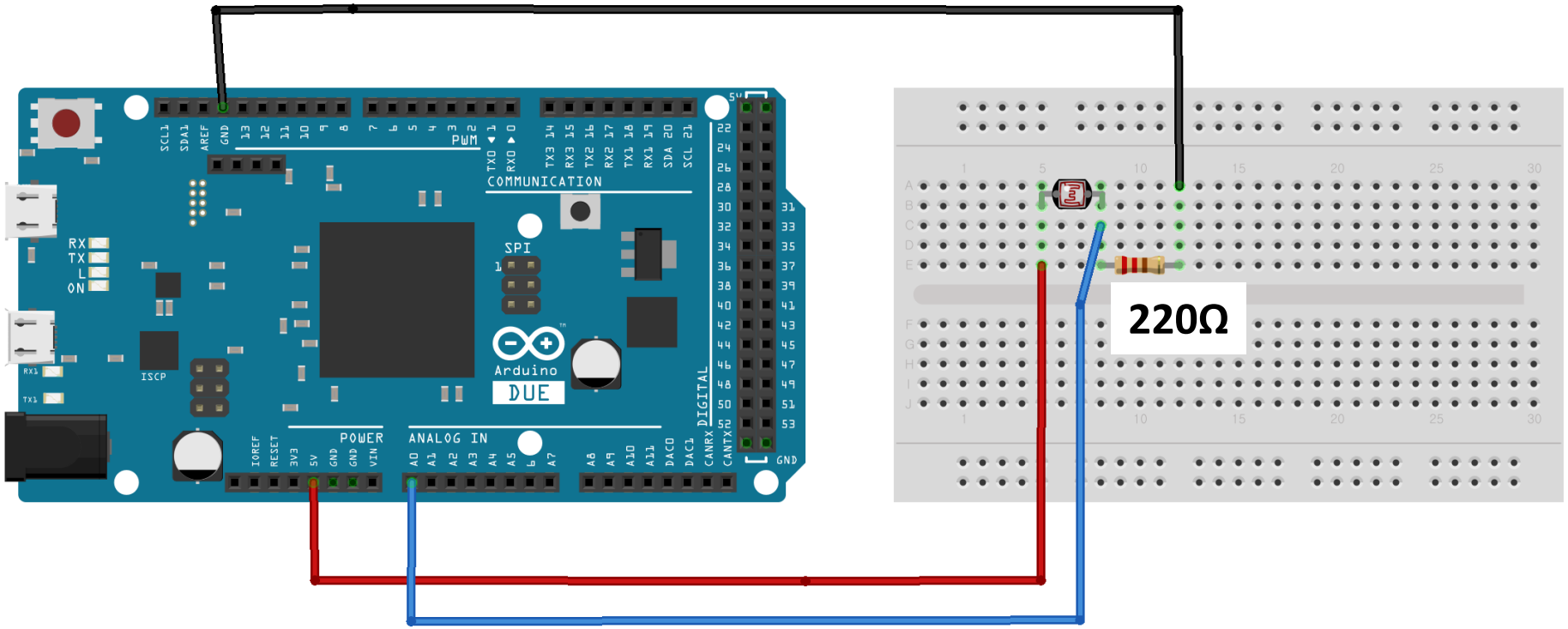
https://www.youtube.com/watch?v=u9Riurh4y9U

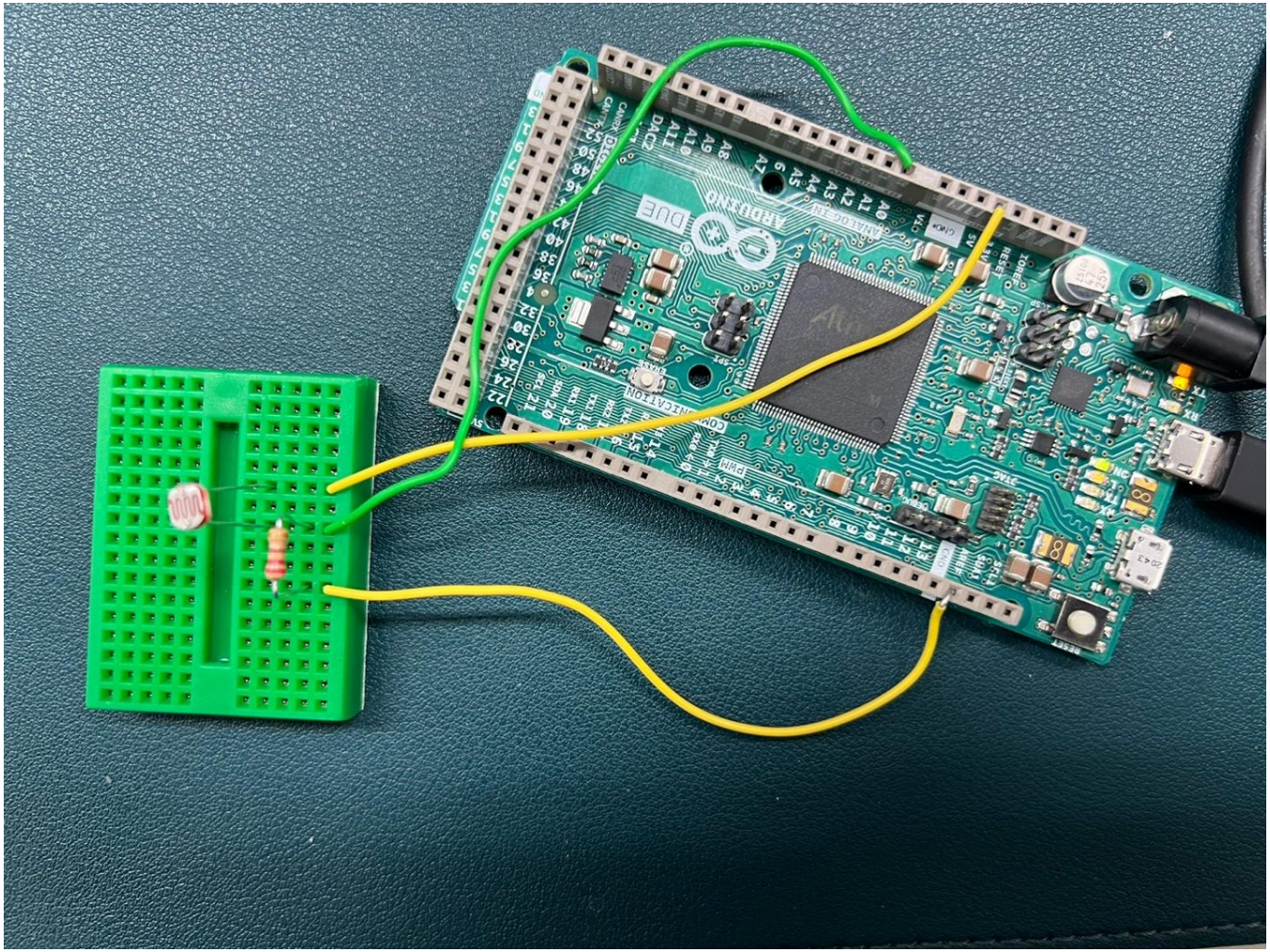


https://www.youtube.com/watch?v=WeC7cR6n48M

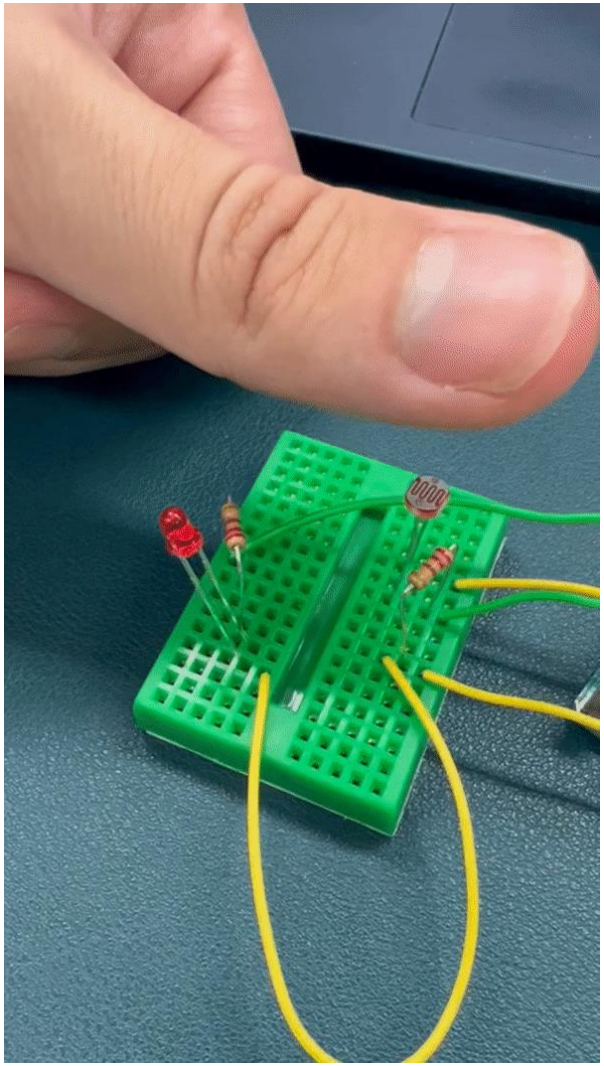

A2
5V (+)
10KΩ
GND (-)

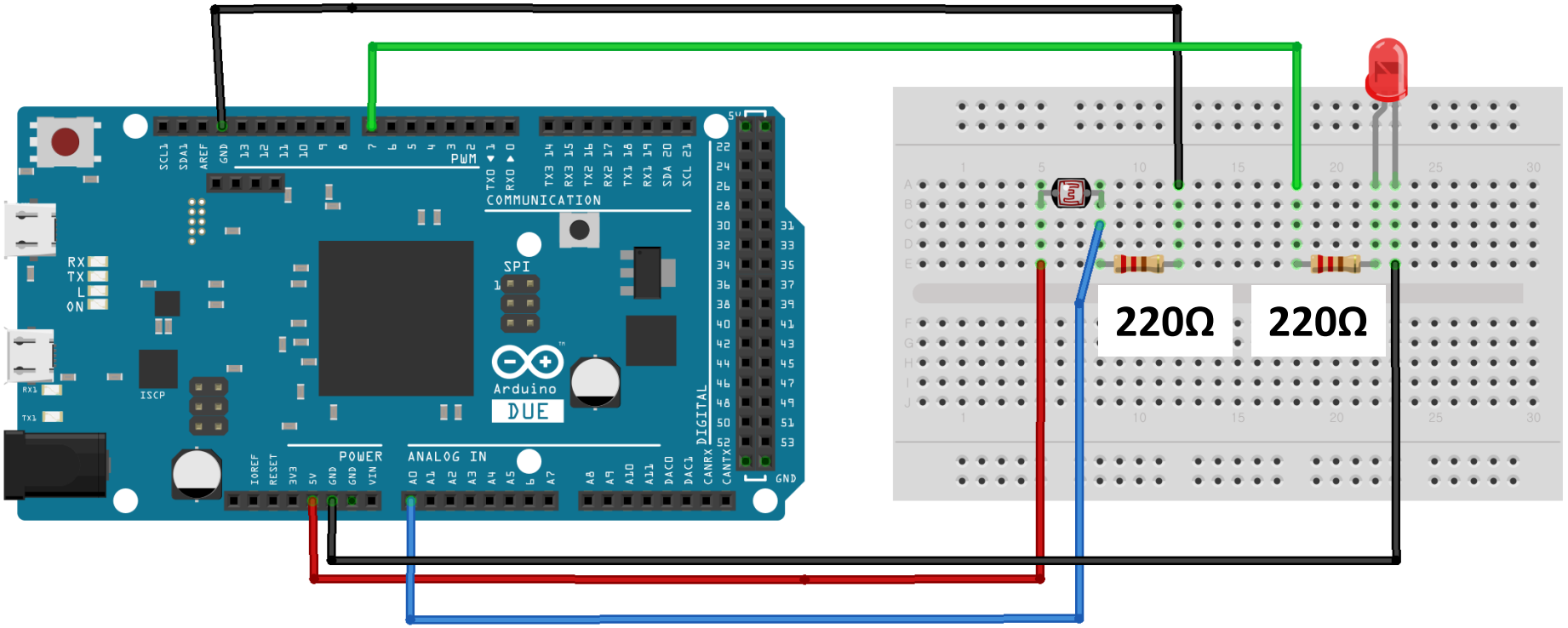

220Ω
RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
ON
Arduino
UNO
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
ICSP


220Ω
Arduino
DUE
PWM
COMMUNICATION
SPI
POWER
ANALOG IN
DIGITAL

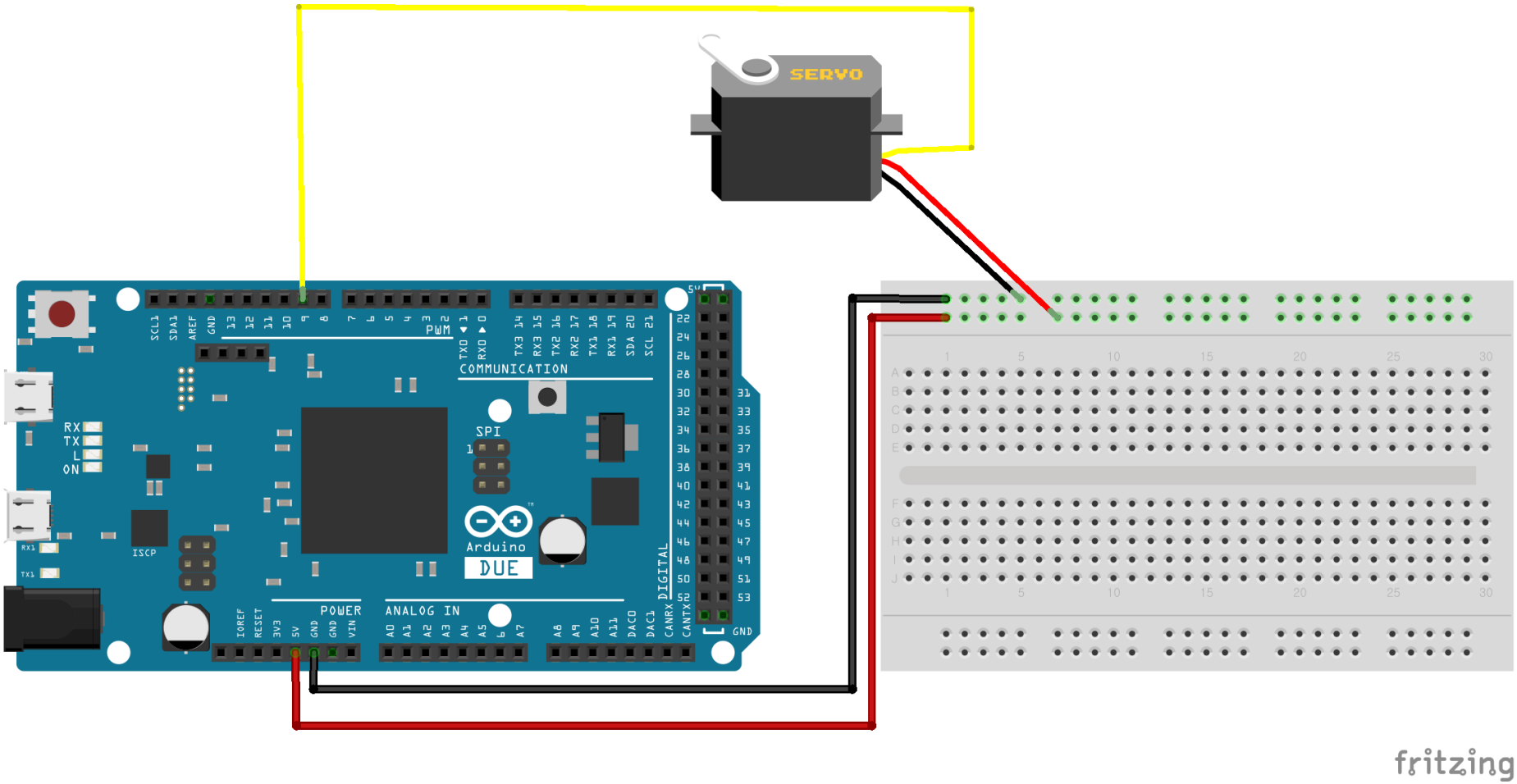

DUE
ARDUINO

```
int photoPin = A0;

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

void loop() {
  int light = analogRead(photoPin);
  Serial.println(light);
  delay(100);
}
```

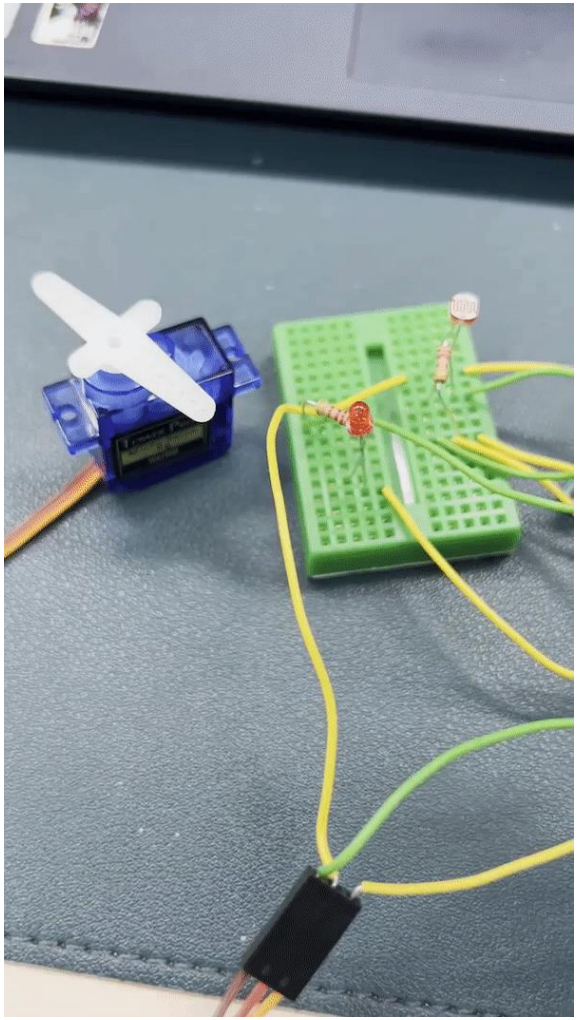


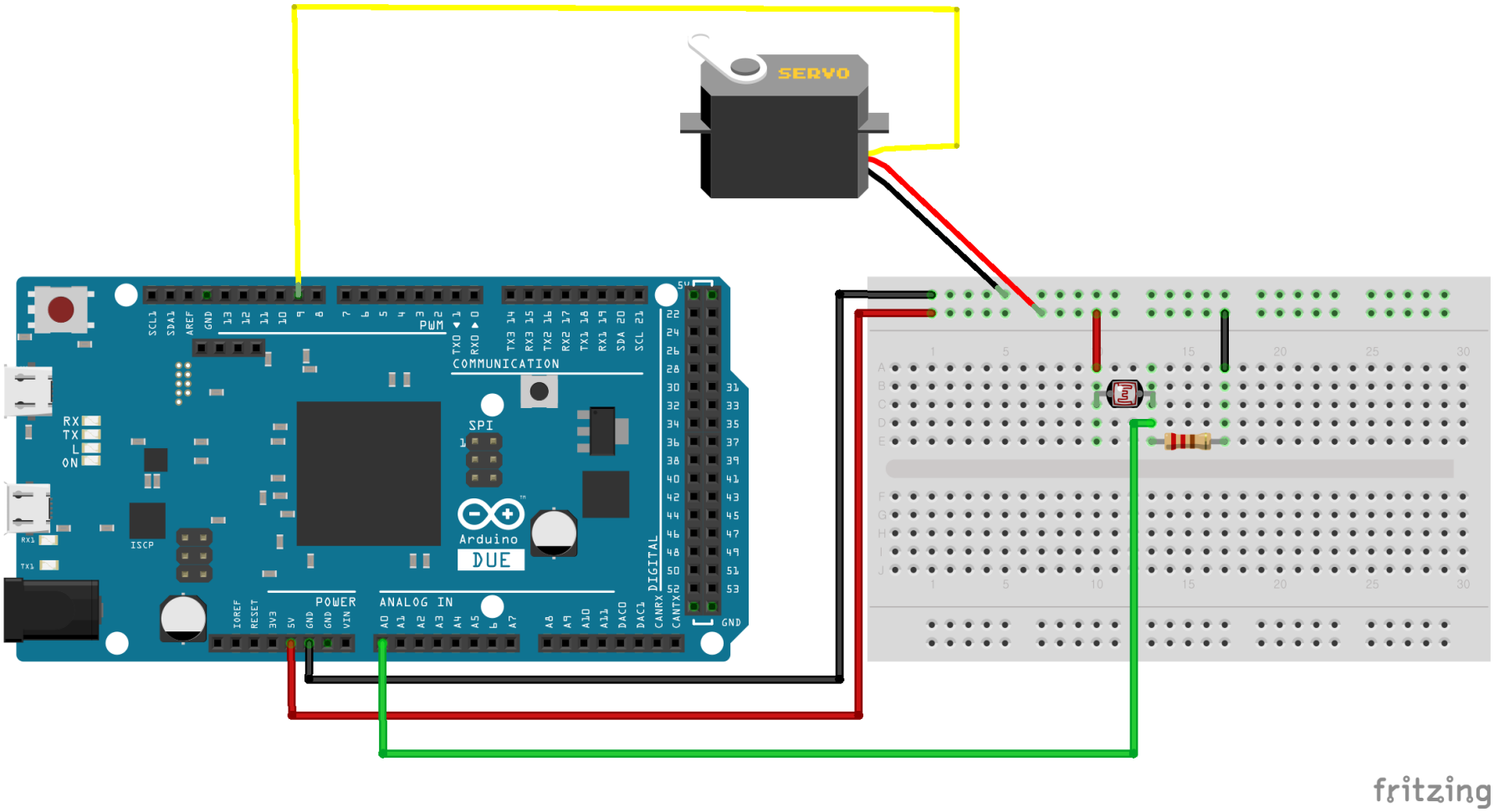

220Ω
220Ω
Arduino
DUE

```
int photoPin = A0;
int ledPin = 7;

void setup(){
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);

}

void loop(){
  int lightRaw = analogRead(photoPin);
  int light = map(lightRaw, 0, 1023, 0, 100);
  Serial.println(light);

  if (light < 5) {
    digitalWrite(ledPin, HIGH);
  }

  else {
    digitalWrite(ledPin, LOW);
  }
}
```

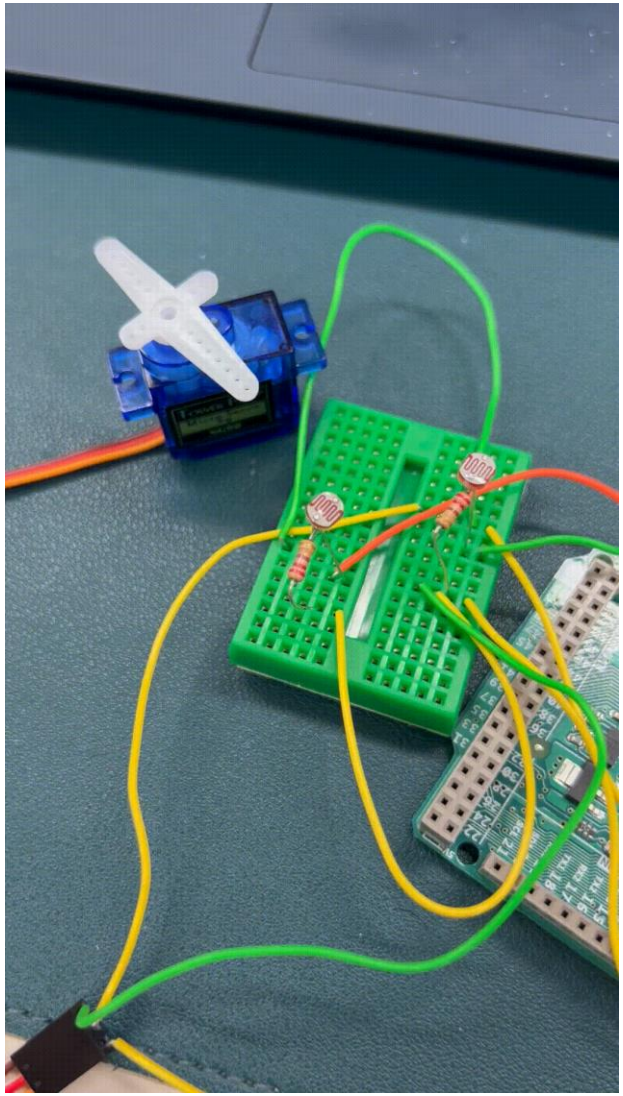

SERVO
Arduino
DUE
COMMUNICATION
SPI
POWER
ANALOG IN
DIGITAL
PWM
fritzing

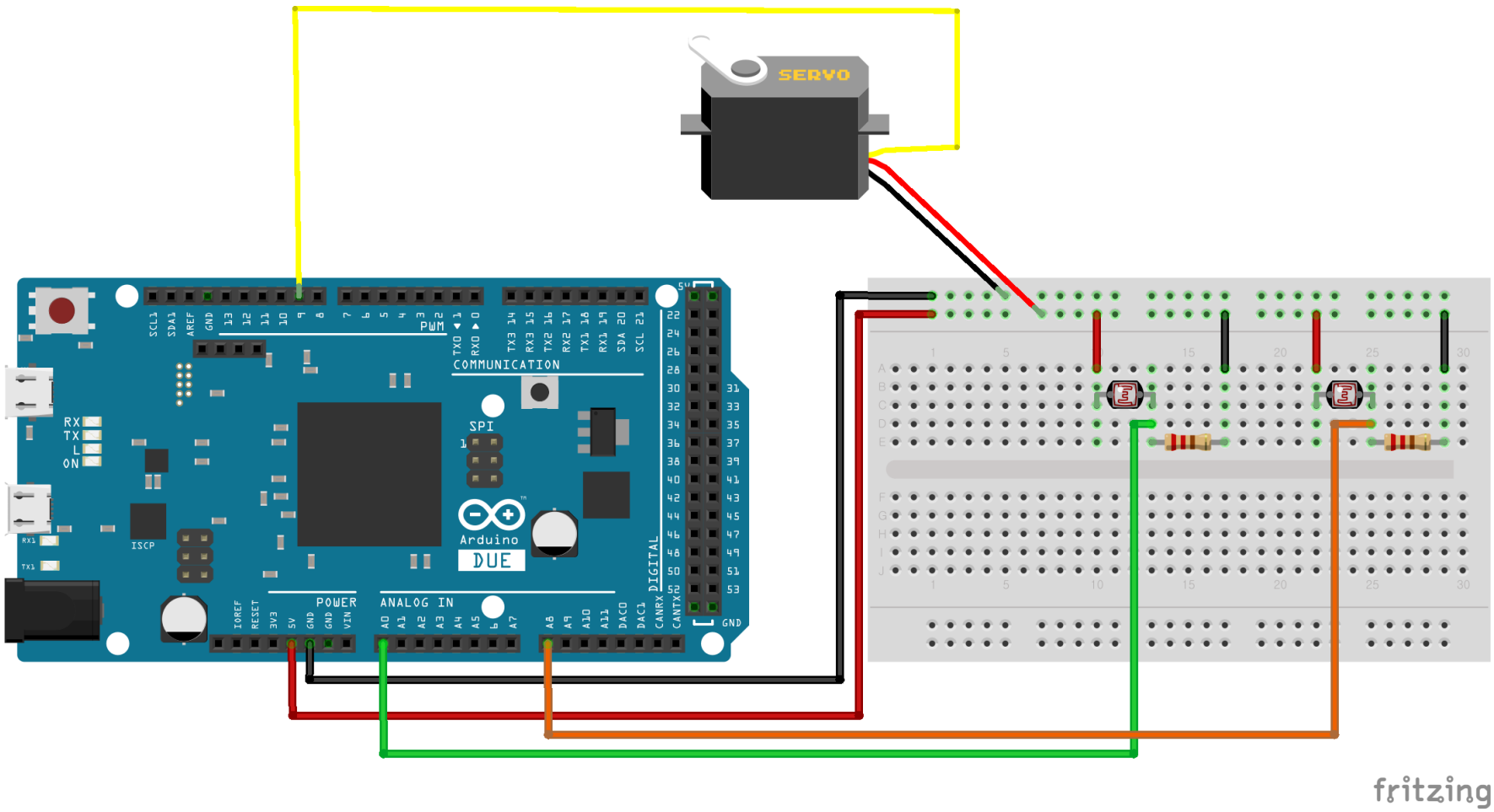
```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```




SERVO
PWM
COMMUNICATION
SPI
Arduino
DUE
POWER
ANALOG IN
DIGITAL
fritzing

```
#include <Servo.h>
int photoPin = A0;
Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
 Serial.begin(9600);
}

void loop() {
 int lightValue = analogRead(A0);
 int lightRaw = analogRead(photoPin);
 lightValue = map(lightValue,0,1023,0,180);
 myservo.write(lightValue);
 Serial.println(lightRaw);
}
```




SERVO
Arduino
DUE
COMMUNICATION
PWM
SPI
POWER
ANALOG IN
DIGITAL
fritzing

```
#include <Servo.h>
int photoPin1 = A0;
int photoPin2 = A8;


Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
 Serial.begin(9600);
}

void loop() {
 int lightValue1 = analogRead(A0);
 int lightRaw1 = analogRead(photoPin1);
 int lightValue2 = analogRead(A8);
 int lightRaw2 = analogRead(photoPin2);


 lightValue1 = map(lightValue1,0,1023,0,180);
 lightValue2 = map(lightValue2,0,1023,0,180);

 Serial.print(lightValue1); Serial.print(',');Serial.println(lightValue2);
 myservo.write(lightValue1);

if(lightValue1<10){
 for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
   // in steps of 1 degree
   myservo.write(pos);           // tell servo to go to position in variable 'pos'
   delay(15);                    // waits 15ms for the servo to reach the position
  }
  }
 if(lightValue2<10){
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
   myservo.write(pos);           // tell servo to go to position in variable 'pos'
   delay(15);                    // waits 15ms for the servo to reach the position
  }
  }
}
```