Varning!

Rikta inte lasern in i någons ögon då personen kan få permanenta skador.

Automatisk Broöppnare

**Beskrivning av utmaningen**

Idag är det många broar som öppnas manuellt vilket kan orsaka mycket väntetid, skapa onödiga kostnader och skapa stor frustration.

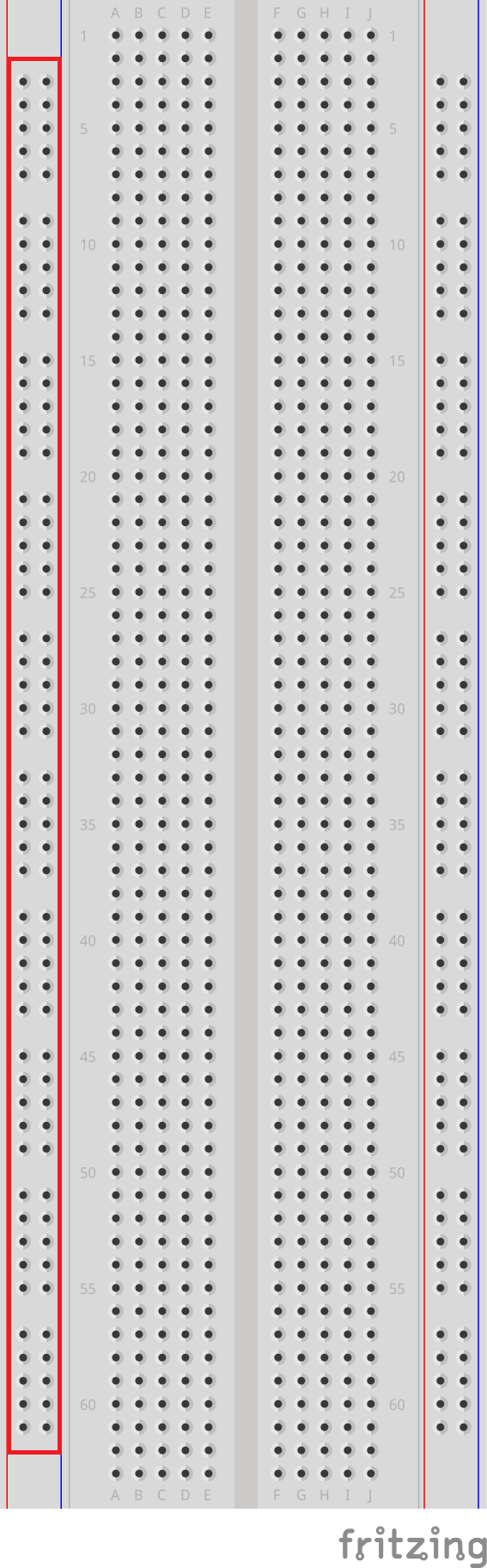
Fokus på det vi ska göra idag är att bygga systemet som kommer skapa en automatisk broöppnare. Denna skall kunna känna av ifall en bil är för hög för bron och skall i sådana fall öppna bron. Bron får inte vara öppen för länge då det kommer irritera bilister som vill över bron.

Inom IKT får man ofta utmaningar som handlar om att skriva kod som påverkar saker i den fysiska världen. I denna utmaning kommer ni att arbeta med att bygga en bro och fixa kod som får den att öppna automatiskt när ett högt fordon kommer förbi.

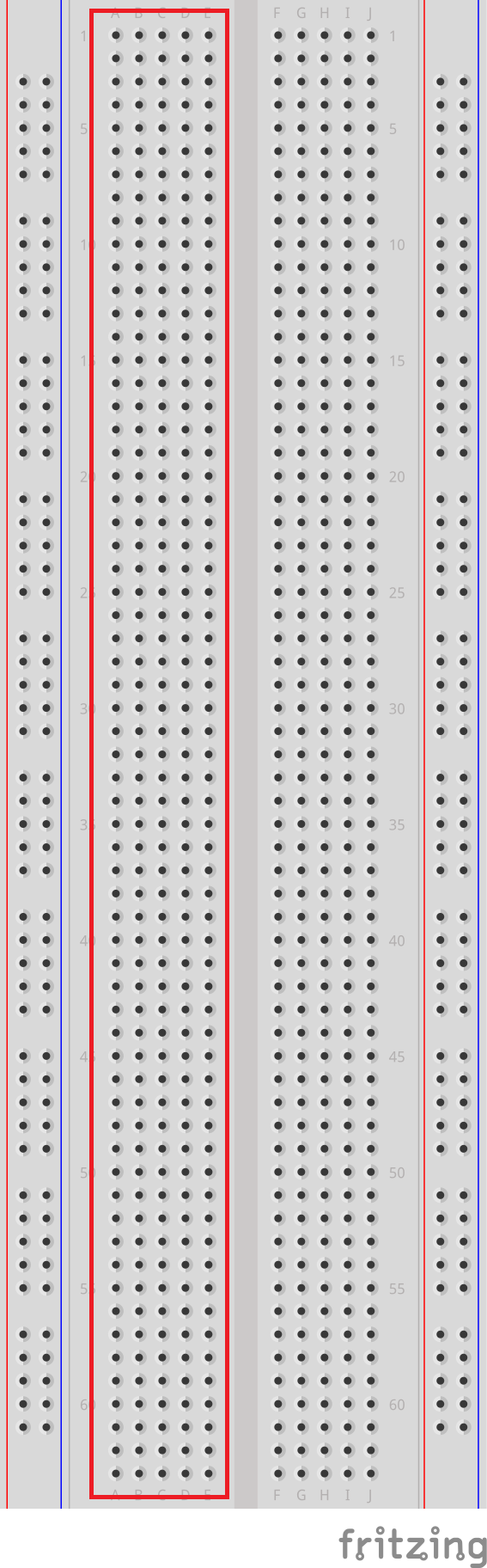
**Bygga ihop kretsen**

T

ill att börja med har ni en s.k. kopplingsbräda, i denna skall ni koppla kretsen som kommer ta emot och skicka laserstrålen. Sidorna på kopplingsbrädet har två kolumner, den västra kolumnen är kopplat till plus (+) och den högra kolumnen är kopplat till (-).

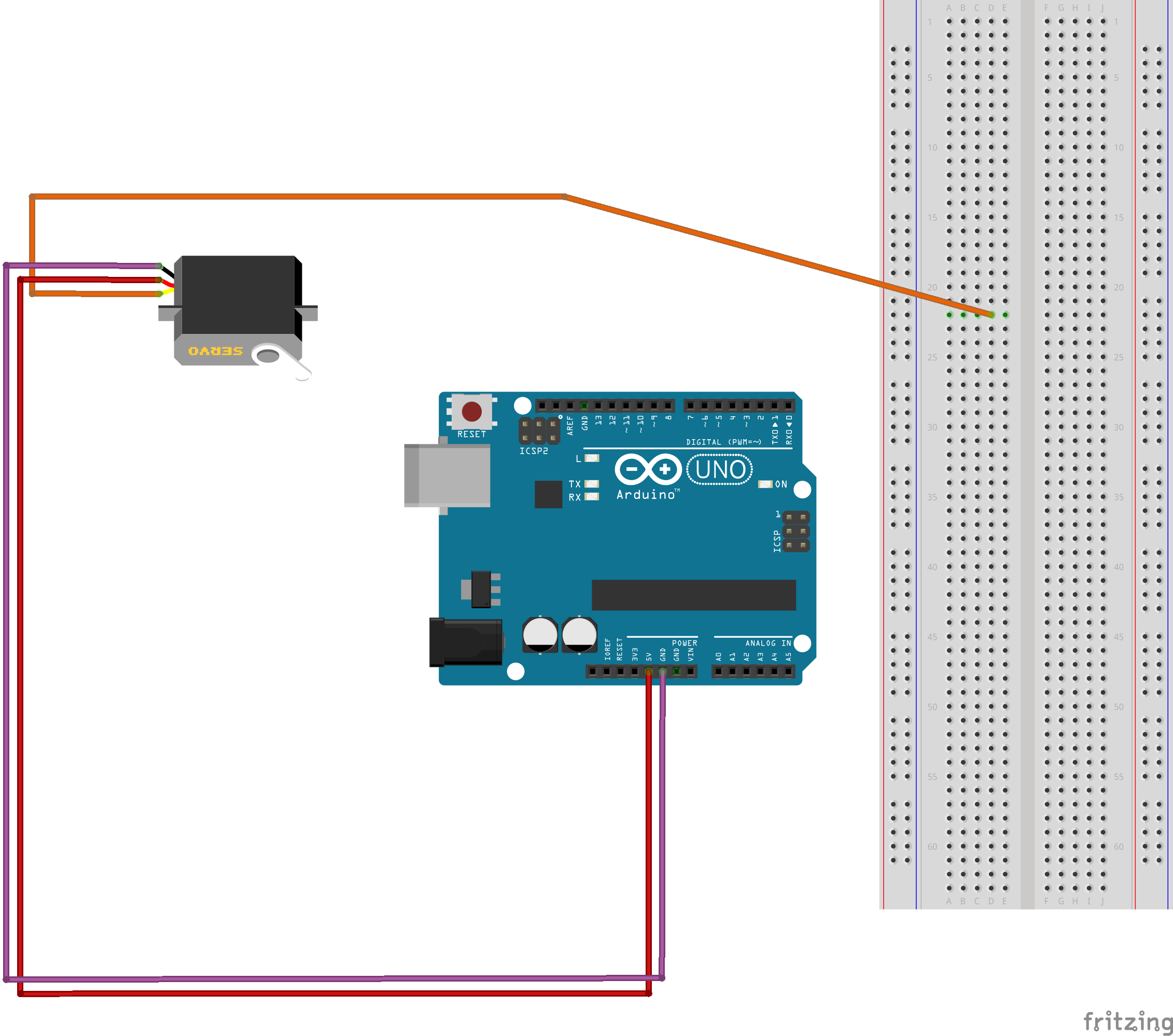


För att koppla samman flera komponenter måste de kopplas på samma rad i den större delen av kopplingsbrädet.

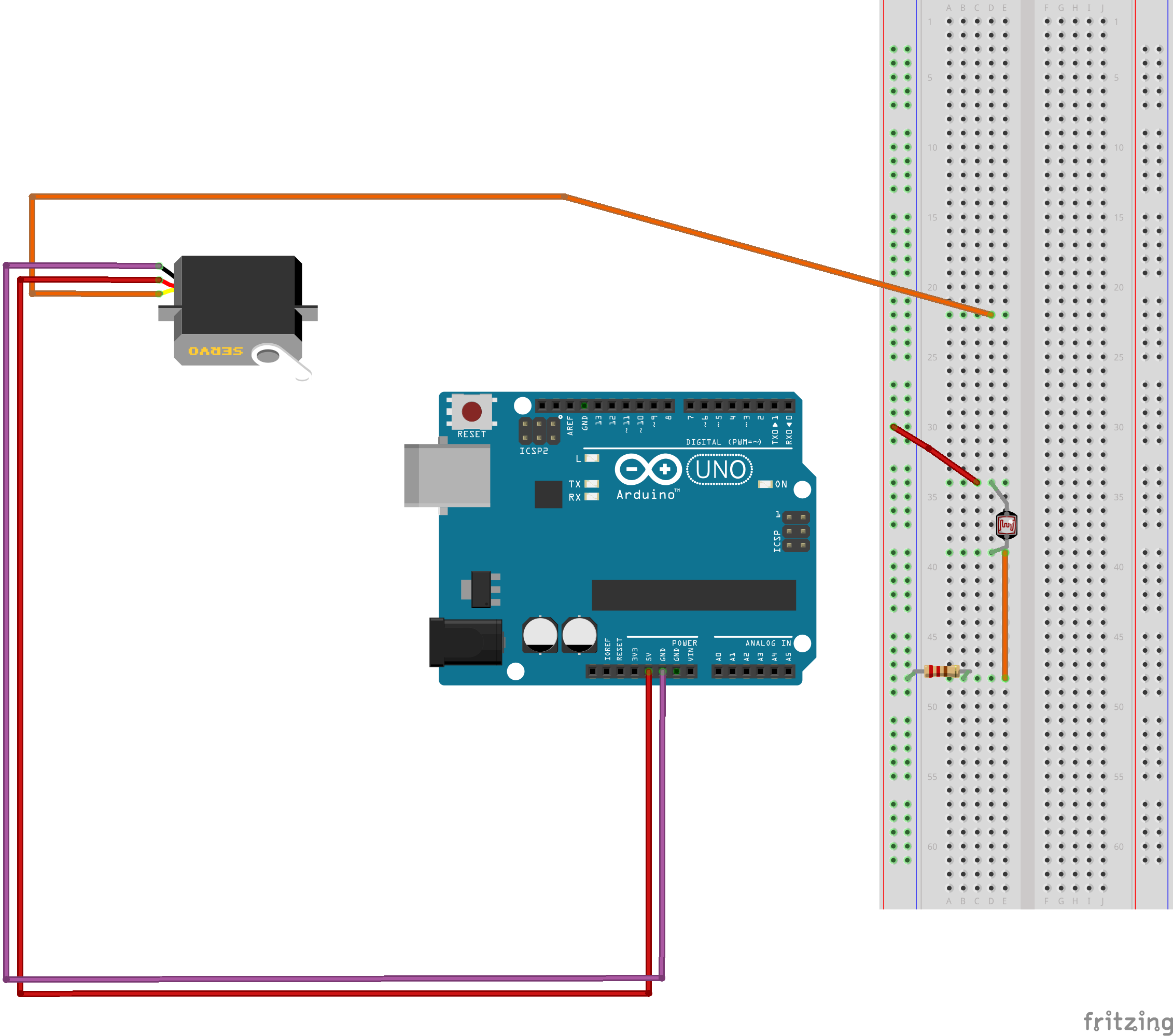


Vi börjar med att koppla in servot i breadboarden och Arduinon. Denna sitter under bron för er.

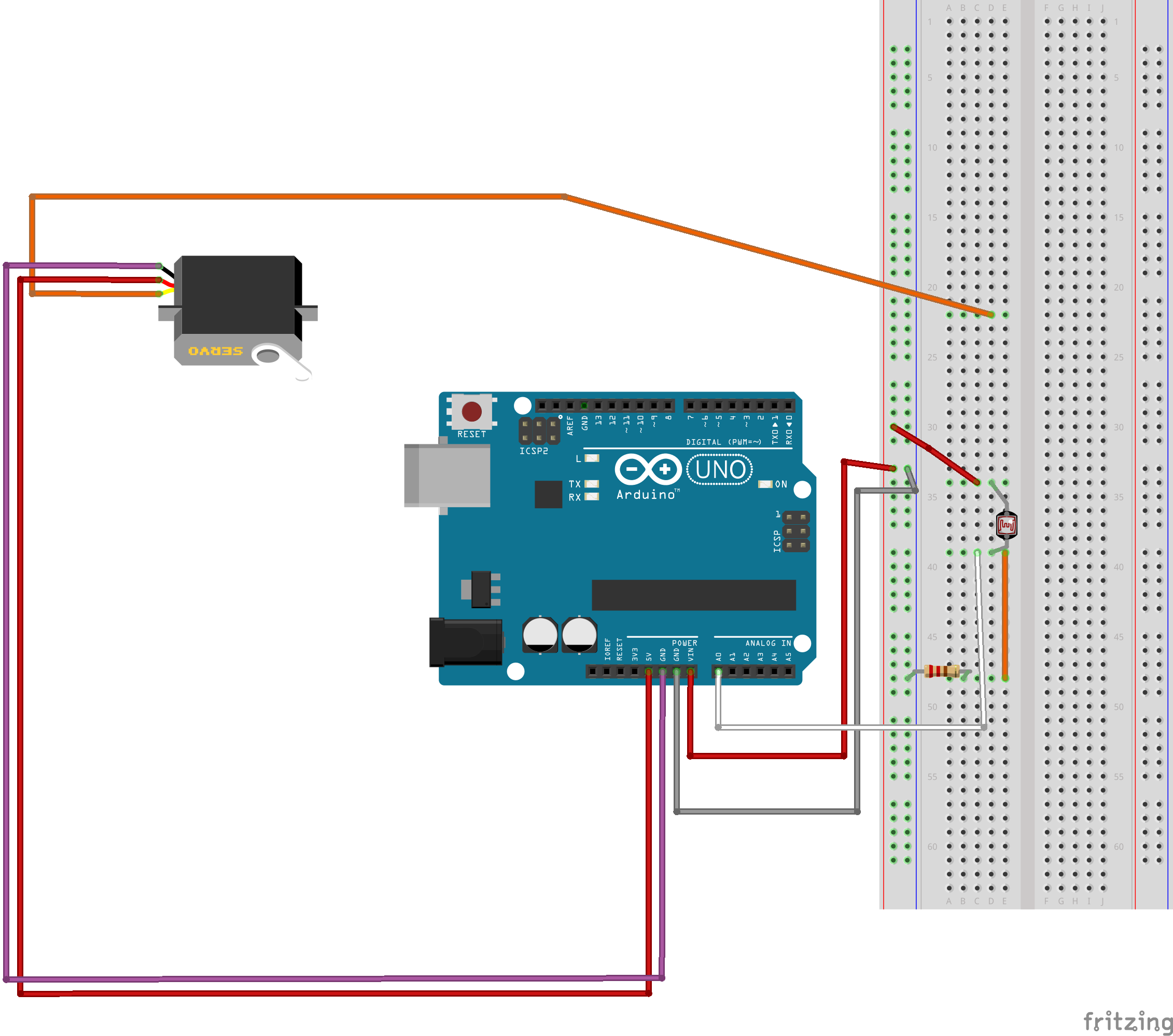
Arduinon är en "minidator" som kommer sköta höjningen och sänkningen av bron samt känna av laserstrålen.



För att se om bilen som passerar är för hög använder vi oss av en laserstråle, om denna bryts så måste bron öppna sig. För att känna av om den bryts så används en så kallad LDR (Light dependant resistor), denna LDR varierar spänning beroende på hur ljust det är.



För att den ska funka korrekt så måste den vara inkopplad tillsammans med ett motstånd. Motståndet kopplas in mellan jord och LDR och sen kopplar man från LDR till +5V. Mätningen görs av Arduinon mellan motståndet och LDR.

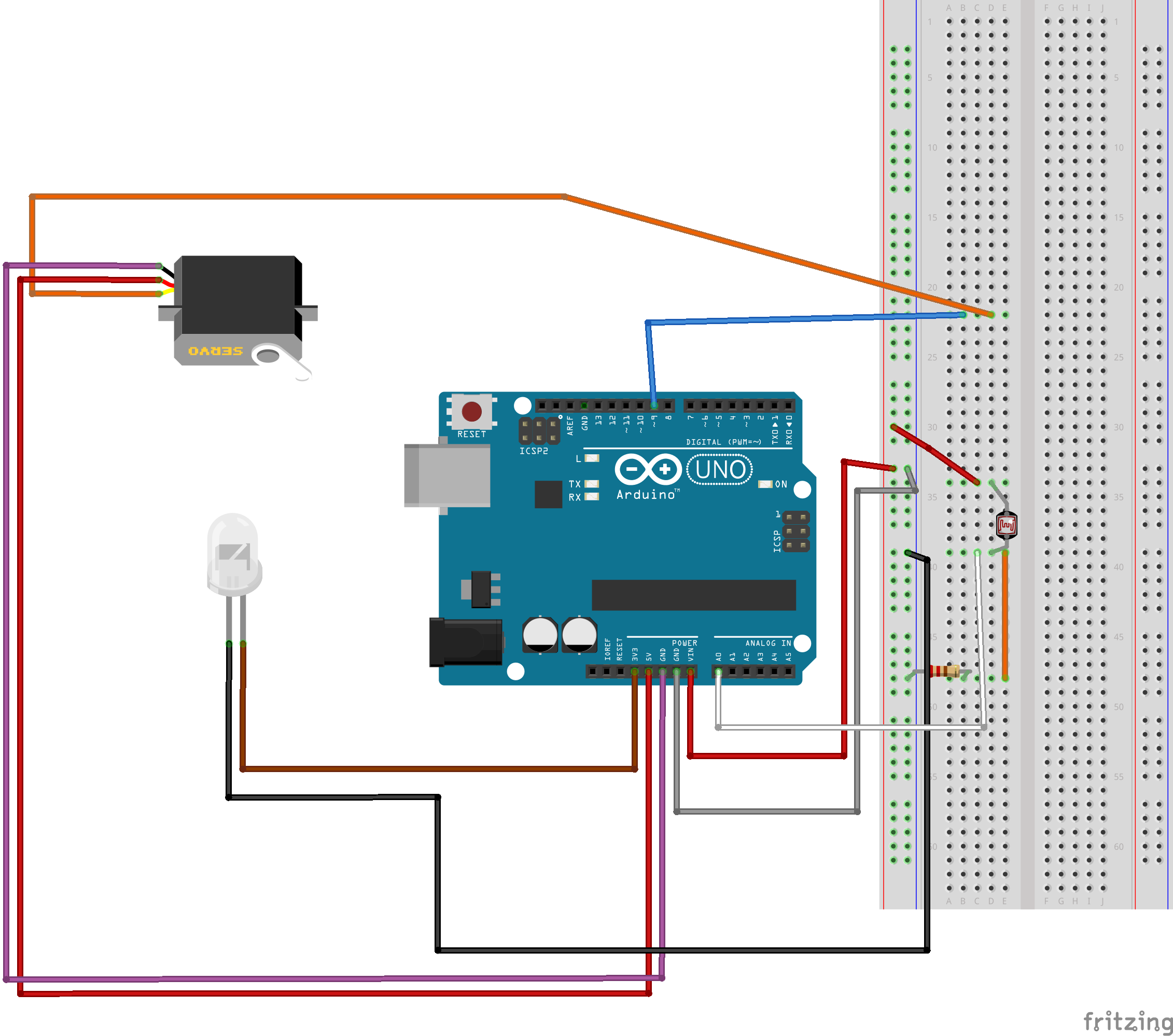


Nu kopplar vi in lasern och servot till pin 9.

Vi vill åter igen påminna om att inte lysa med lasern i någons ögon då det kan orsaka permanent skada.

Laserns röda och svarta kabel kommer behöva kopplas ihop med krokodilkablar. Den röda kabeln behöver kopplas ihop med +3,3V och den svarta kabeln måste kopplas till minus (-) på breadboarden.

Koppla även en kabel in mellan servots orangea kabel och pin 9 på er Arduino.



**Kodbeskrivning**

Variabeln sensorValue känner av om något har korsat sensorn eller ej. Det kollas konstant i "loop()" ifall sensorValue har blivit mindre än activationValue (detta är ett värde som ställs in för att inte känna av förändringar i takbelysningen, utan bara vår laser).

När if-satsen är sann (något korsar sensorn) vill vi att servots vinkel ökas och bron öppnas, står still ett tag, och vrider sig sedan tillbaka och sänker bron.

Lasern måste peka på sensorn på brädet, det är denna som känner av förändringar i ljusstyrka.

**Ofärdig Arduino Kod**

#include <Servo.h>

Servo servoBridge;

const int servoPin = 9;

const int openAngle = ???;

const int closedAngle = ???;

const int openTime = ???;

const int activationValue = ???;

const int analogInPin = A0;

int sensorValue = 0;

void setup() {

// put your setup code here, to run once:

servoBridge.attach(servoPin);

servoBridge.write(0);

delay(2000);

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

sensorValue = analogRead(analogInPin);

Serial.println(sensorValue);

if(sensorValue < activationValue){

???;

delay(openTime);

???;

}

delay(10);

}

void openBridge(){

servoBridge.write(openAngle);

}

void closeBridge(){

servoBridge.write(closedAngle);

}

Vi har hjälpt till med att skapa en del färdig kod, men en del har vi inte skrivit ned med flit. Försök härma koden som är ovan och ändra koden där det står "???" för att få det att fungera ordentligt.

På nästa sida kommer det finnas ett färdigt exempel. Gå dit om du fastnat eller vill jämföra med koden vi skrev. Men testa först att skriva kod själva!

**Färdig Arduino Kod**

#include <Servo.h>

Servo servoBridge;

const int servoPin = 9;

const int openAngle = 120;

const int closedAngle = 20;

const int openTime = 1000;

const int activationValue = 850;

const int analogInPin = A0;

int sensorValue = 0;

void setup() {

// put your setup code here, to run once:

servoBridge.attach(servoPin);

servoBridge.write(0);

delay(2000);

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

sensorValue = analogRead(analogInPin);

Serial.println(sensorValue);

if(sensorValue < activationValue){

openBridge();

delay(openTime);

closeBridge();

}

delay(10);

}

void openBridge(){

servoBridge.write(openAngle);

}

void closeBridge(){

servoBridge.write(closedAngle);

}

Detta är ett förslag på fungerande kod i det ljus som vi testade i. Jämför med din egen och se vilken som fungerar bäst!