

Varning!

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

Automatisk Broöppnare

Beskrivning av utmaningen

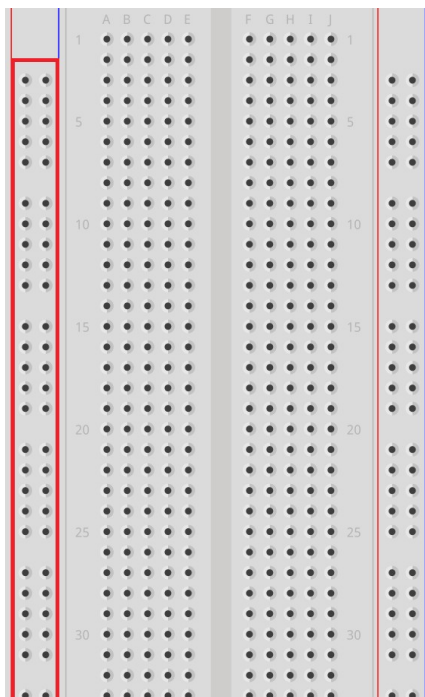
Idag är det många broar som öppnas manuellt vilket kan orsaka mycket väntetid, skapa onödiga kostnader och skapa stor frustration.

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 informations- och kommunikationsteknik (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.

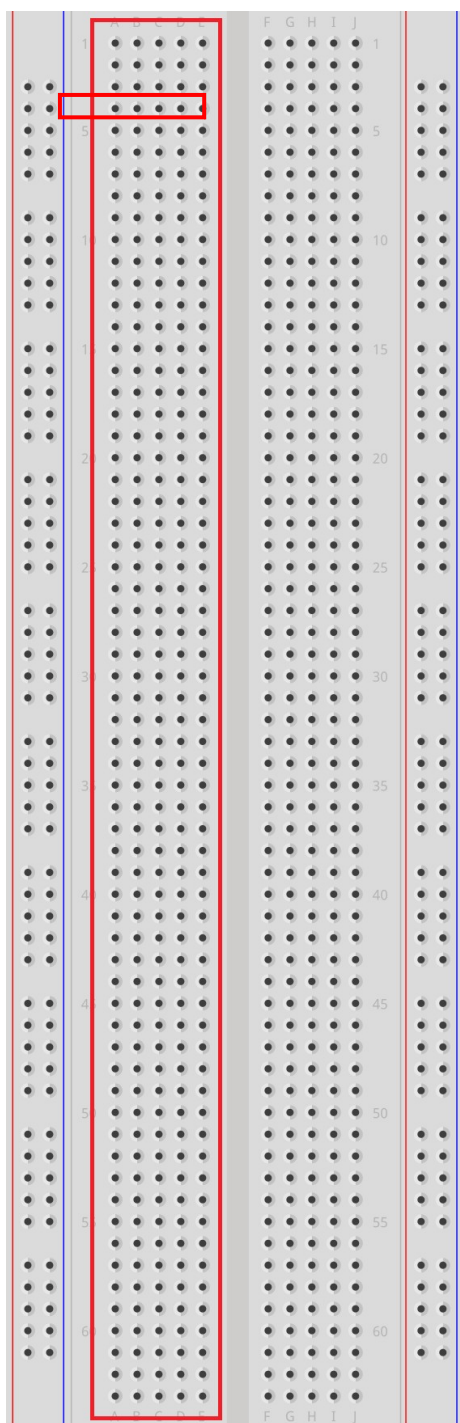
Bygga ihop kretsen

Till 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 vänstra kolumnen är kopplat till plus (+) och den högra kolumnen är kopplat till (-).



CHALMERS

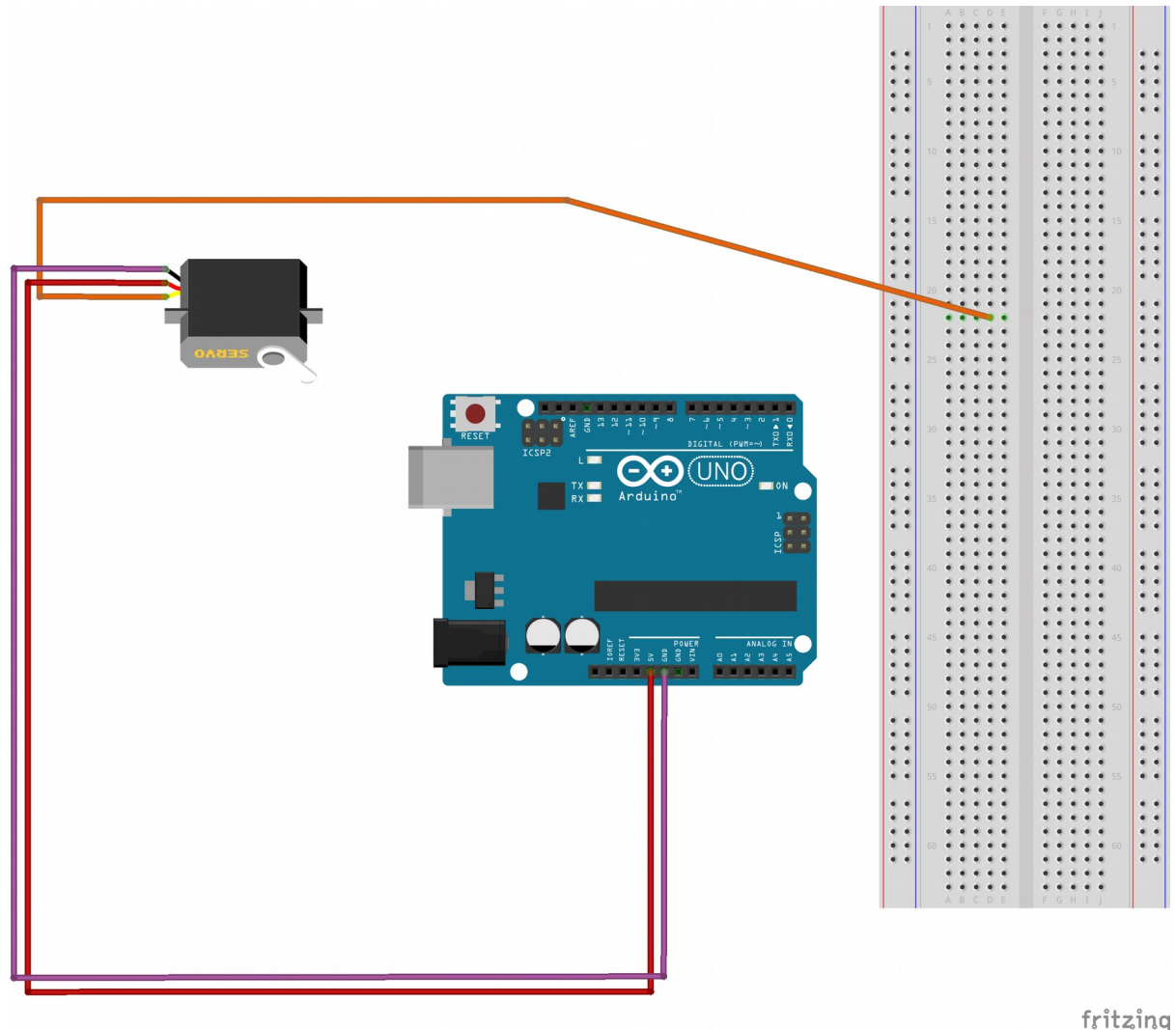
För att koppla samman flera komponenter måste de kopplas på samma rad i den större delen av kopplingsbrädet.



fritzing

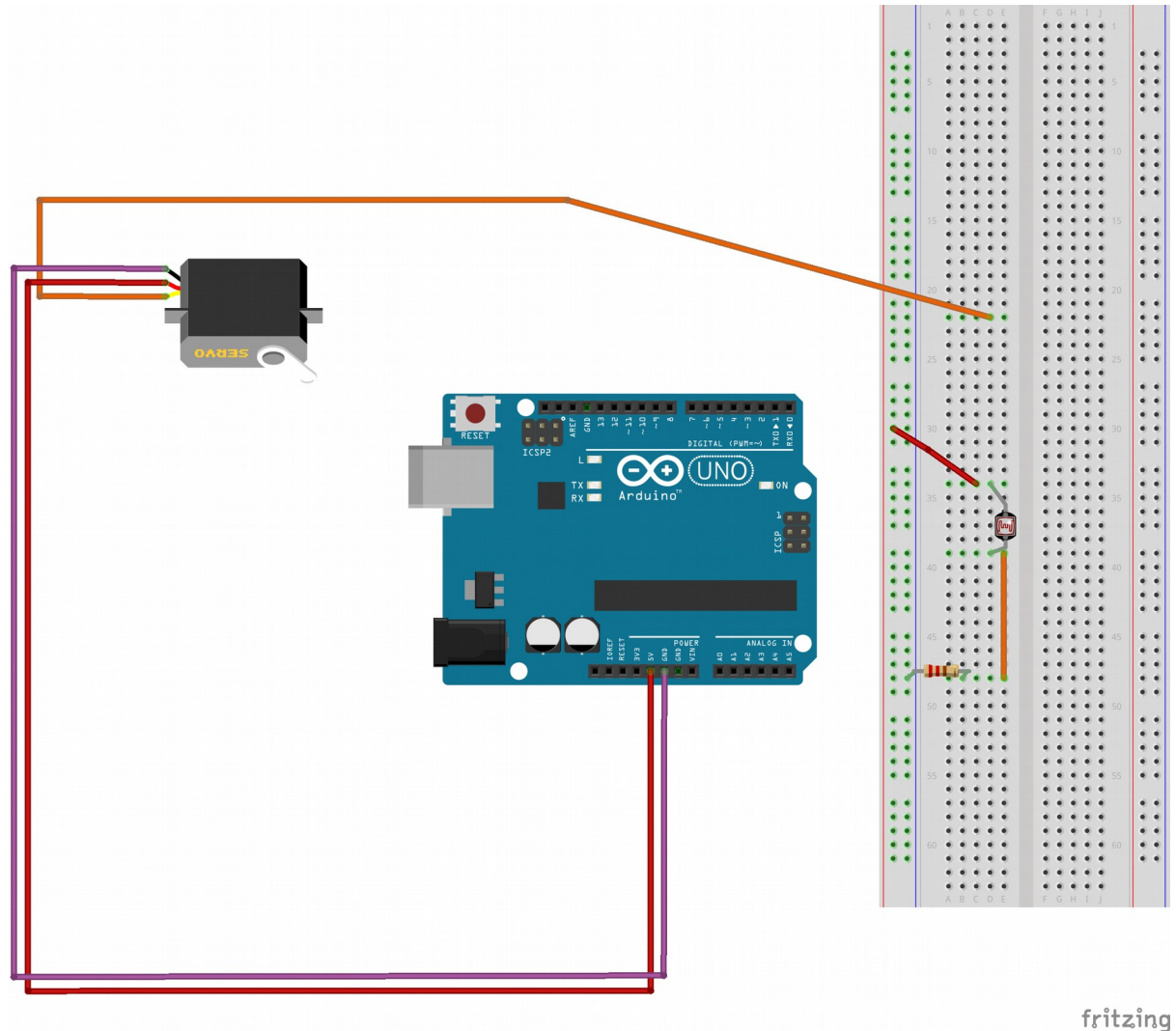
Vi börjar med att koppla in servot i breadboarden och Arduino. Denna sitter under bron för er.

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

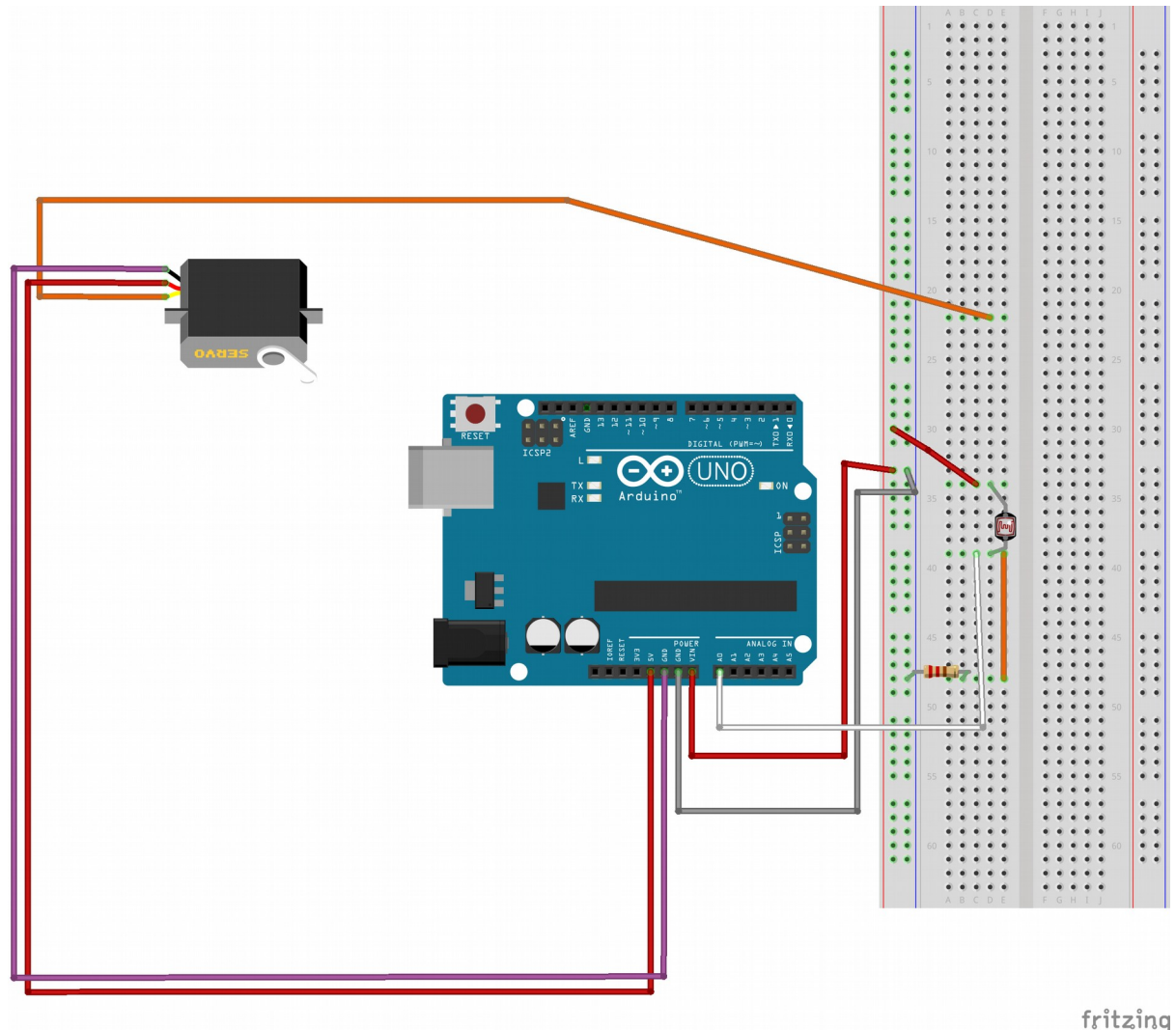


fritzing

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 ljus 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 Arduino mellan motståndet och LDR.

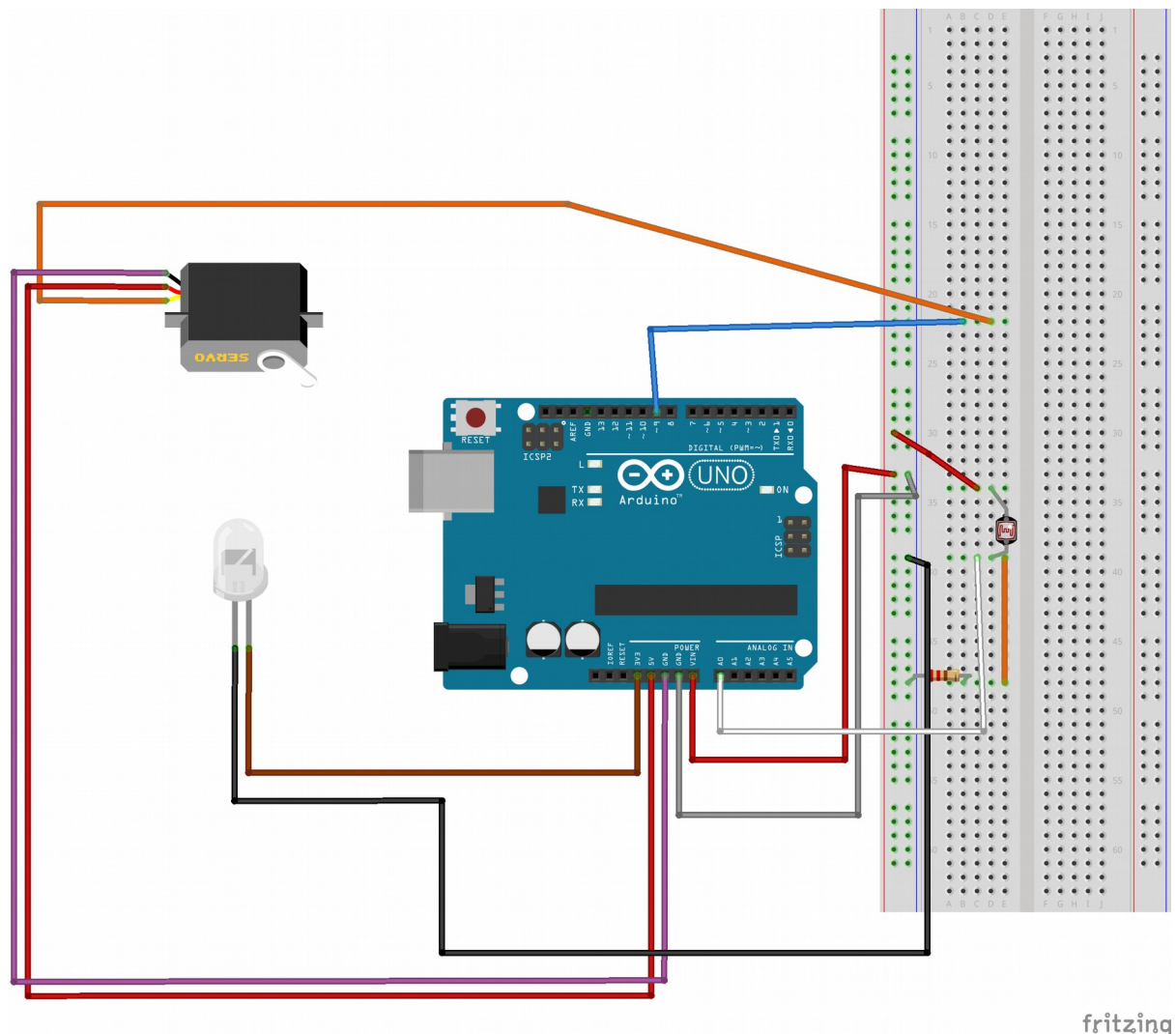


Nu är det dags att koppla in lasern och servot.

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 och förlängas med kablarna som finns i lådan. Den röda kabeln behöver kopplas ihop med +3,3V på Arduinon 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, det är denna som känner av förändringar i ljusstyrka.

Ofärdig Arduino Kod

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.

```
#include <Servo.h>
Servo servoBridge;
// Här ställer man in vilken referensvinkel som servot har för att enkelt bestämma
vinkel
const int refAngle = 150;

// Här ställer man in vilken pin som Arduinon använder för att skicka kommandon
till servon.
const int servoPin = 9;

// Här ska det finnas ett värde mellan 0 och 90 som anger hur många grader bron
ska lyftas till när den öppnar
const int openAngle = ???;

// Här ska det finnas ett värde mellan 0 och 90 som anger hur många grader bron
ska lyftas till när den stänger
const int closedAngle = ???;

// Här ska det finnas ett värde som styr hur länge bron ska hållas öppen. Målet är
att hålla bron öppet så lite tid som möjligt så att bilen kan passa. Värdet anges i
millisekunder (1000 millisekunder = 1 sekund)
const int openTime = ???;

const int activationValue = 880
const int analogInPin = A0;
int sensorValue = 0;

void setup() {
  // Här kommer koden som körs i början
  servoBridge.attach(servoPin);
  servoBridge.write(refAngle);
  delay(2000);
  Serial.begin(9600);
}
```

```

void loop() {
  // Här kommer koden som kontrollerar om något har kommit emellan lasern och
  sensorn.
  sensorValue = analogRead(analogInPin);
  Serial.println(sensorValue);

  // Om något kommer emellan lasern och sensorn, då kommer värdet
  sensorValue gå ner. Om den blir mindre än activationValue, då öppnar man bron.
  If (sensorValue < activationValue) {
    // Här behövs kod för att öppna bron.
    ???;
    delay(openTime); // Bron hålls öppen för den tid som satts tidigare i variablen
    openTime()
    // Och sedan stängs bron igen.
    ???;
  }
  delay(10);
}

// Den här koden öppnar bron. Skriv openBridge(); för att kalla denna kod.
void openBridge(){
  servoBridge.write(refAngle-openAngle);
}

// Den här koden stänger bron. Skriv closeBridge(); för att kalla denna kod.
void closeBridge(){
  servoBridge.write(refAngle-closedAngle);
}

```

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 refAngle = 150;
const int servoPin = 9;
const int openAngle = 90;
const int closedAngle = 0;
const int openTime = 1000;
const int activationValue = 880;
const int analogInPin = A0;
int sensorValue = 0;
void setup() {
  // put your setup code here, to run once:
  servoBridge.attach(servoPin);
  servoBridge.write(refAngle);
  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(refAngle-openAngle);
}
void closeBridge(){
  servoBridge.write(refAngle-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!