

In dit project maken we een spoorboom, die automatisch naar beneden gaat.

Ons project kent twee onderdelen.

1 – Ontdekken of er een trein aankomt.

2 – De spoorboom sluiten.

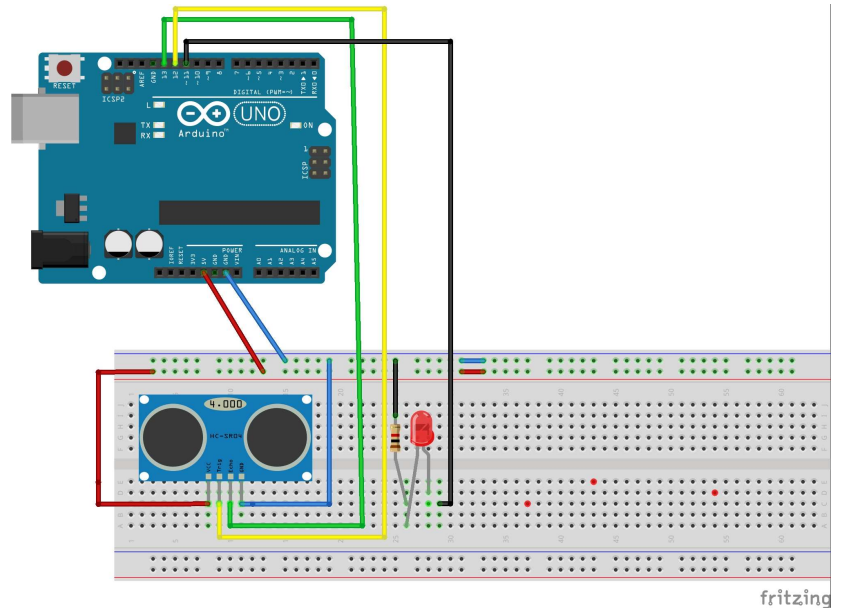
Best wel ingewikkeld, dus stap voor stap. Eerst maken we de onderdelen apart en testen deze ook apart. Als we precies weten hoe de onderdelen werken, maken we er één project van.

## 1 – Ontdekken of er een trein aankomt.

Doel: laat ledje 30 seconden branden als

### Schakeling

Maak de schakeling aan de linkerkant van het breadboard.



### Code

```
int ledPin = 10;
int trigPin = 11; // Trigger
int echoPin = 12; // Echo
long duration, cm;
void setup() {
  pinMode(ledPin,
OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
  cm = (duration/2) / 29.1;
  if (cm <= 10){
    digitalWrite(ledPin, HIGH);
    delay(5000);
    digitalWrite(ledPin, LOW);
  }
  delay(250);
}
```

### Hoe werkt het?

Als de trein (of je hand) dichterbij dan 10 cm bij de HC-SR04 komt, gaat het ledje branden. 5 sec lang. Daarna gaat het ledje weer uit.

Je hebt nu een sketch, die de trein kan zien.

Sla de sketch op met opslaan als en geef deze een toepasselijke naam, 'trein-komt' ofzo.

Nu gaan we eerst de 'spoorboom sluiten' maken.

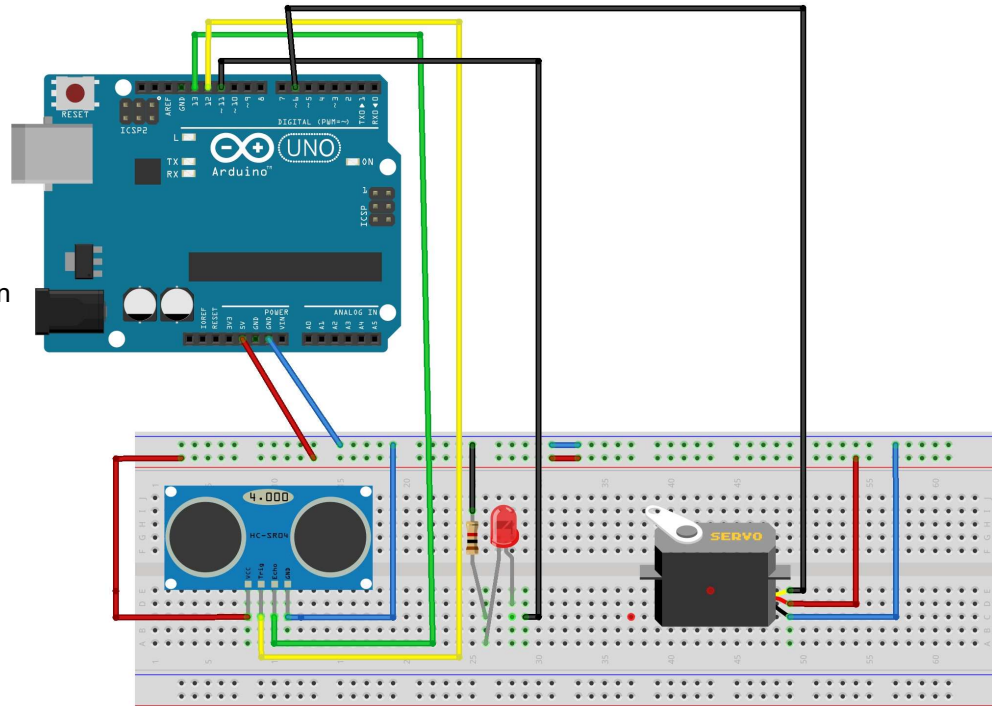
## 2 – De spoorboom sluiten.

Een goede programmeur werkt in stapjes. Elk stapje wordt eerst uitgetest, pas daarna worden de stapjes gekoppeld.

Deze sketch zet de spoorboom omhoog, wacht 5 seconden en doet dan de spoorboom weer naar beneden. Weer 5 seconden wachten en dan weer opnieuw.

### Schakeling

Maak de schakeling aan de rechterkant van het breadboard.



fritzing

### Code

```
#include <Servo.h>
Servo mijn_servo;
void setup()
{
  mijn_servo.attach(6);
}

void loop()
{
  mijn_servo.write(50);
  delay(5000);
  mijn_servo.write(160);
  delay(5000);
}
```

### Pas op!

De getallen 50 en 160 kunnen voor elke servo anders zijn, dus controleren en als nodig aanpassen.

Klaar? Mooi, dan weten we nu zeker dat de spoorboom zelf goed werkt. Sla de sketch op met opslaan als en geef deze een toepasselijke naam, 'sluit-spoorboom' ofzo.

## 3 – Het eindproject

### Schakeling

Aan de schakeling hoef je niets meer te veranderen.

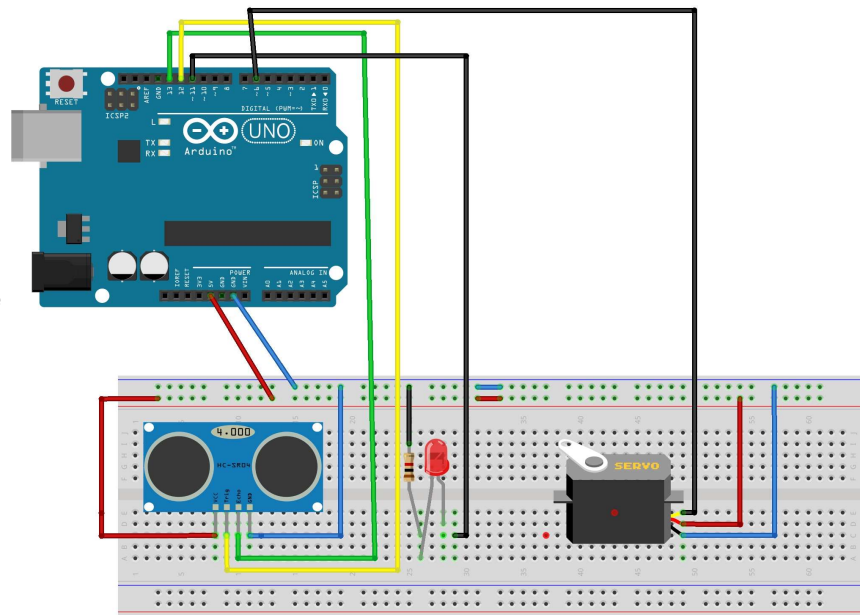
### Code

We gaan nu de twee stukjes code met elkaar koppelen.

De nieuwe code lijkt het meest op de eerste sketch van dit project.

Dus open nu de sketch 'trein komt' en sla deze op onder een andere naam, bijvoorbeeld 'spoorboom compleet'.

De grijze regels zijn overgenomen uit 'sluit spoorboom' en moet je toevoegen.



fritzing

```
#include <Servo.h>
Servo mijn_servo;
int ledPin = 10;
int trigPin = 11; // Trigger
int echoPin = 12; // Echo
long duration, cm;
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  mijn_servo.attach(6);
  mijn_servo.write(30);
}
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
  cm = (duration/2) / 29.1;
  if (cm <= 10){
    digitalWrite(ledPin, HIGH);
    mijn_servo.write(150);
    delay(5000);
    digitalWrite(ledPin, LOW);
    mijn_servo.write(30);
  }
  delay(250);
}
```

Klaar? Geweldig, goed gedaan.

-- Hoe kan je de code veranderen, zodat de trein eerder wordt gezien?

-- En kan de boom ook langer dichtblijven?