

Beschrijving

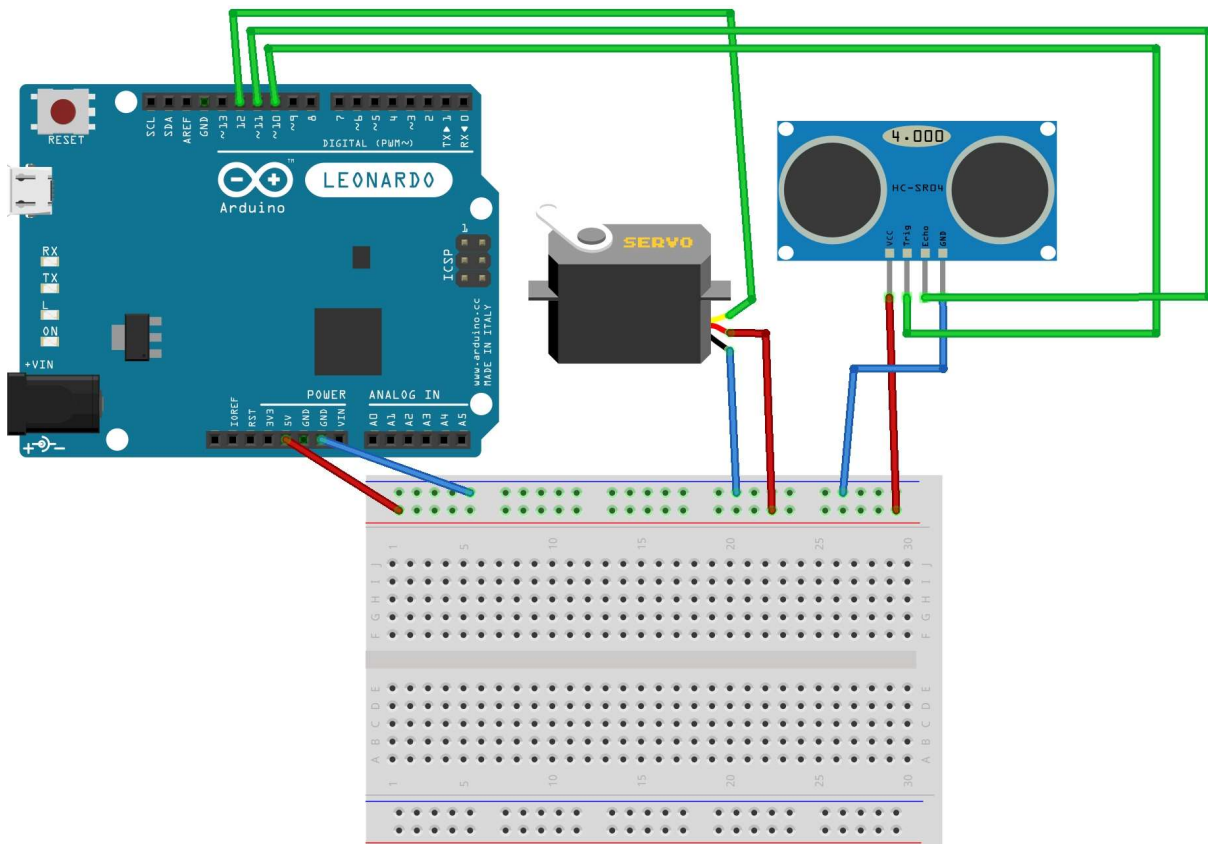
Bouw een radar met de HC-SR04 en een servomotor.

De HC-SR04 zijn de ogen, de servo laat de ogen draaien, zodat de omgeving wordt afgespeurd.

We gebruiken het "Processing" om de radar-beelden weer te geven, want onze Arduino schermen zijn hiervoor te klein. Vraag hierbij om hulp. De code voor processing staat niet in deze les.



Schakeling



fritzing

De code

```
// Includes the Servo library
#include <Servo.h>

// Defines Trig and Echo pins of the Ultrasonic Sensor
const int trigPin = 10;
const int echoPin = 11;
// Variables for the duration and the distance
long duration;
int distance;

Servo myServo; // Creates a servo object for controlling the servo motor
```

```
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600);
  myServo.attach(12); // Defines on which pin is the servo motor attached
}
void loop() {
  // rotates the servo motor from 15 to 165 degrees
  for(int i=15;i<=165;i++){
    myServo.write(i);
    delay(30);
    distance = calculateDistance();// Calls a function for calculating the distance measured by
    the Ultrasonic sensor for each degree

    Serial.print(i); // Sends the current degree into the Serial Port
    Serial.print(","); // Sends addition character right next to the previous value needed later in
    the Processing IDE for indexing

    Serial.print(distance); // Sends the distance value into the Serial Port
    Serial.print("."); // Sends addition character right next to the previous value needed later in
    the Processing IDE for indexing
  }
  // Repeats the previous lines from 165 to 15 degrees
  for(int i=165;i>15;i--){
    myServo.write(i);
    delay(30);
    distance = calculateDistance();
    Serial.print(i);
    Serial.print(",");
    Serial.print(distance);
    Serial.print(".");
  }
}
// Function for calculating the distance measured by the Ultrasonic sensor
int calculateDistance(){

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in
  microseconds

  distance= duration*0.034/2;
  return distance;
}
```
