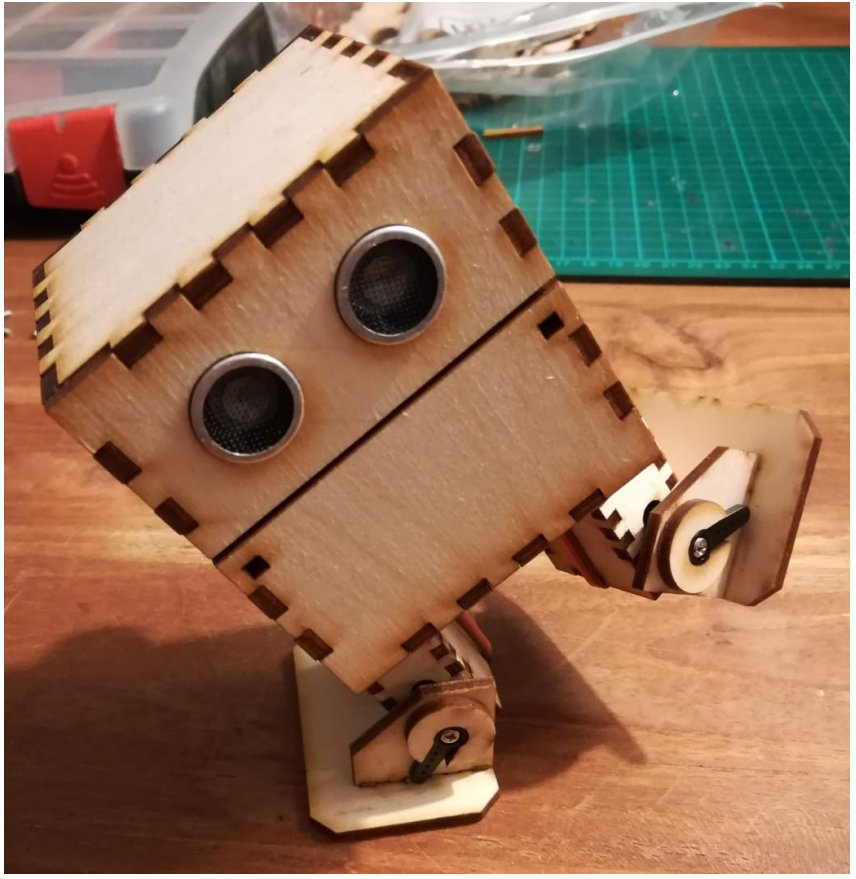
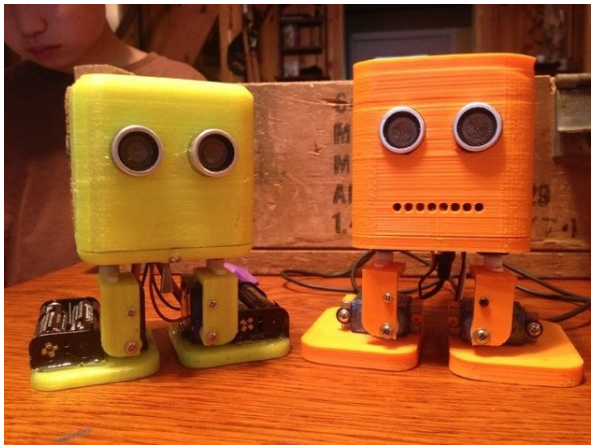
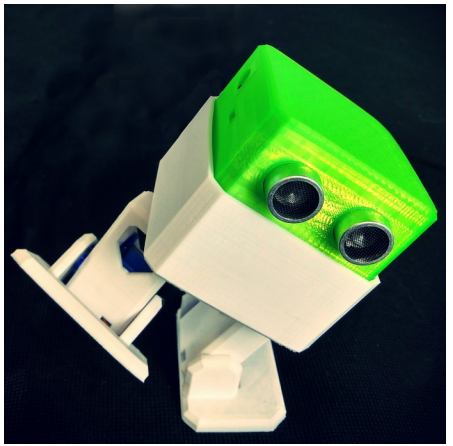
Berend



Oorsprong



Een van de eerste versies van dit lopende robotje was BOB the Biped.
Een goedkoop en simpel 3D te printen project.



Door members van Hackaday werd daar een nieuw ontwerp van gemaakt : Otto.
Steviger en makkelijker in gebruik voor schoolprojecten.

Door Backspace werd dit ontwerp omgezet naar een versie die met de lasersnijder uit hout kan worden gemaakt : Otto LC.

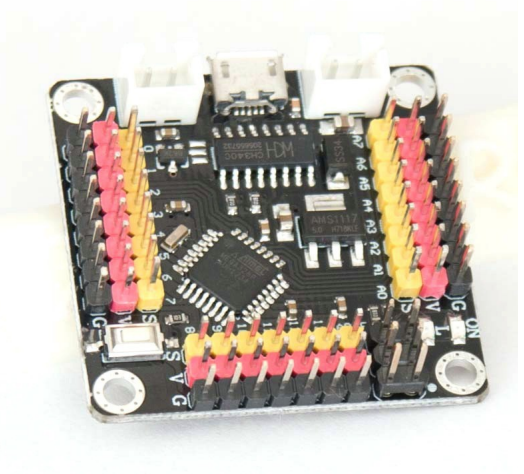
Dit ontwerp is door mij aangepast naar sterkere servo’s en een ander soort arduino.

De electronica – het brein

Arduino NANO Strong

* 5v Logic
* USB Micro
* Elke I/O pin heeft ook +5v en GND
  (toevallig? in servo volgorde)
* 7-12v externe voeding (2S LiPo 7.2v)
* ch340 USB2Serial chip.
* Atmega 328P microcontroller

32K Flash, 1K EEPROM, 2K RAM,
20 MHz, 14 I/O, 8 ADC In / PWM Out



De electronica – de beweging

TowerPRO MG90S

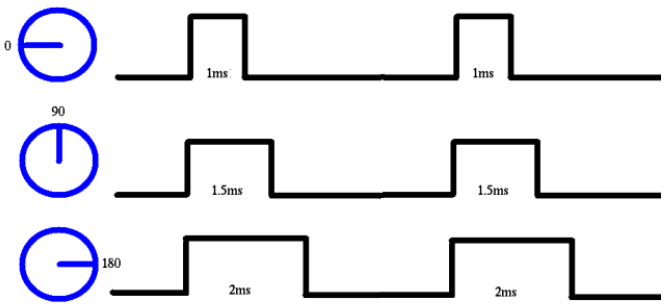
* 4.8 – 6v
* 180 graden
* redelijk snel
* goedkoop!
* klein!
* metalen tandwielen en as, veel steviger dan de SG90

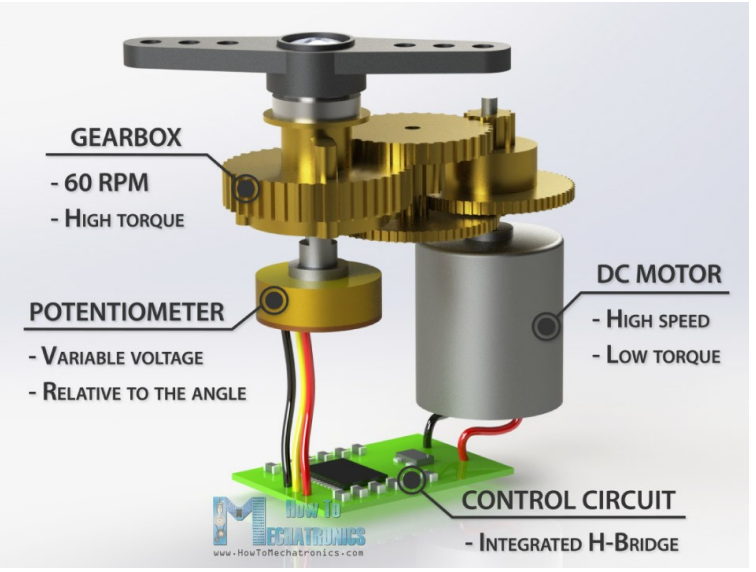
– iets afwijkende maten, dus moest Otto LC worden aangepast.

De electronica – de beweging

Hoe werkt het

Een PWM signaal van 2 Khz. Afhankelijk van de pulsbreedte veranderd de hoek van de servo.





De electronica – de beweging

Programmering

Standaard arduino servo library.
Positie wordt in graden opgegeven.

Servo “objectnaam”, bijvoorbeeld:

```
#include <Servo.h>

Servo knieGewricht;

knieGewricht.attach(2);

knieGwricht.write(90);
```

```
#include <Servo.h>

Servo mijnServo;  // creëer het servo object

int pos = 0;      // variabele om de positie mee te bepalen

void setup() {
  mijnServo.attach(9);  // koppel het servo object aan Digitale pin 9
}

void loop() {   // beweeg de servo steeds heen en weer tussen 0 en 180 graden
  for (pos = 0; pos <= 180; pos += 1) { // van "pos" is 0 tot 180 graden
    mijnServo.write(pos);               // zet de servo in positie "pos"
    delay(15);                          // wacht 15 miliseconden
  }
  for (pos = 180; pos >= 0; pos -= 1) { // van "pos" is 180 tot 0 graden
    mijnServo.write(pos);               // zet de servo in positie "pos"
    delay(15);                          // wacht 15 miliseconden
  }
}
```

De electronica – de ogen

HC-SR04 ultrasone afstandmeter

* 5V
* 2cm – 4m (ahem) met een uitstraalhoek van 15 graden

De electronica – de ogen

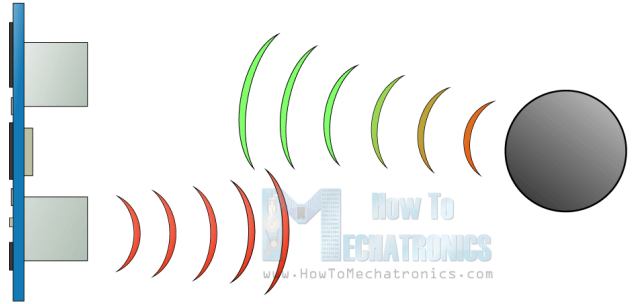
Hoe werkt het

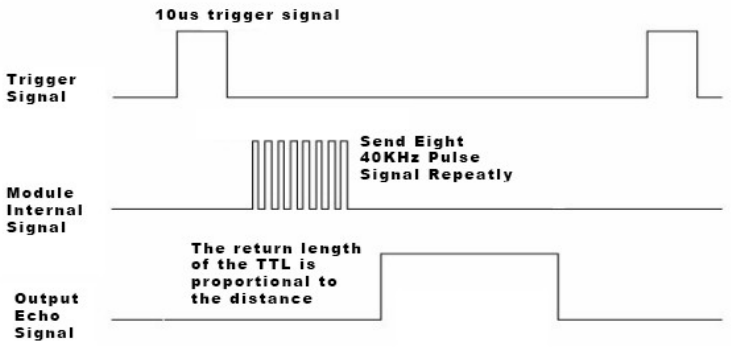
* Door een positieve puls van 10 microseconden op de “trigger” ingang zal de module 8 pulsen van 40 Khz verzenden.

* Als er door de ontvanger iets wordt terugontvangen zal de “echo” uitgang hoog worden en blijven afhankelijk van de gemeten afstand.

* De afstand in m is de helft van het aantal microseconden maal snelheid van het geluid in m/s

$$s = (t / 2) * 343$$





De electronica – de ogen

Programmering

Geen library nodig, mag uiteraard wel.

* 10µs trigger “hoog” maken

* luister op echo en lees de pulslengte

* deel de halve pulslengte[1] door 29.1

* of vermenigvuldig de halve pulslengte[1] met 0.0343

[1] de afstand is gedeeld door 2 vanwege het heen-en-terug gaan van de geluidsgolf.

```
#define trigPin 11          // Trigger pin
#define echoPin 12          // Echo pin
long afstand, tijdsDuur;    // duur van de echo, afstand en tijdsduur

void setup() {
    Serial.begin (9600);        // begin seriële communicatie
    pinMode(trigPin, OUTPUT);   // trigPin is een output
    pinMode(echoPin, INPUT);    // echoPin is een input
    digitalWrite(trigPin, LOW); // zet trigPin "logisch 0"
}

void loop() {     // genereer de puls en echo bij antwoord de afstand
  digitalWrite(trigPin, HIGH); // zet trigPin "logisch 1"
  delayMicroseconds(10);       // wacht 10 microseconden
  digitalWrite(trigPin, LOW);  // zet trigPin "logisch 0"

  tijdsDuur = pulseIn(echoPin, HIGH);  // lees de pulslengte op echoPin

  afstand = (tijdsDuur/2) / 29.1;  // Deel de halve pulslengte door 29.1
                                   // of vermenigvuldig met 0.0343
  Serial.println(afstand + "cm");
  delay(250);
}
```