Practicum Talking Servos

[7-4-2022, door Marius Versteegen]

# Inleiding

Als je dingen aanstuurt of meet, wil je meestal niet dat die dingen elkaar beinvloeden.  
Als je robotarm bijvoorbeeld omhoog moet bewegen, wil je niet dat hij soms horizontaal gaat trillen.  
Bij dit practicum gaan we ervaring opdoen met de mogelijke oorzaken van zo’n naar bijverschijnsel.

Als testcase gaan we twee servos aansturen. Van de eerste servo is de hoek in te stellen met een potentiometer. De tweede servo kunnen we met een knop naar de ene of de andere kant laten sweepen. Als we de potentiometer met rust laten mag de eerste servo idealiter niet gaan bewegen als we met de knop de sweep van de tweede servo activeren.

Voor de experimenten in dit practicum is het belangrijk om de aansluitdraden op ongeveer dezelfde plekken in het breadboard en de arduino te prikken. Het kan dus bijvoorbeeld ook uit maken waar in de voedingsrail of groundrail je een draad prikt.

## Benodigdheden:

* Een Arduino en een breadboard
* Een 10k (of zo) potentiometer
* Een weerstand (1k tot 50k of zo)
* Een elco (470uF of zo)
* Twee kleine servos
* Een tweede 5V voeding (zoals bijvoorbeeld een tweede Arduino)

# Een “goede start”

Afbeelding met tekst, elektronica

Description automatically generated

Probeer bovenstaande fritzing schema na te bouwen en upload de Arduino sketch uit de appendix naar je Arduino via de Arduino IDE.

Controleer vervolgens dat het werkt:

* Dat je met de potentiometer de eerste servo in elke mogelijke hoek kunt zetten.
* Dat je met de drukknop de tweede servo kunt laten sweepen.
* Dat de eerste servo tijdens dat sweepen netjes op zijn plaats blijft (niet gaat trillen)

# Experiment 1 : Aref verleggen

Ga uit van het schema van de “goede start”, maar sluit Aref op een ander punt aan:

Afbeelding met tekst, elektronica, schermafbeelding

Description automatically generated

Vermoedelijk zal nu in sommige standen van de potentiometer (niet op, maar in de buurt van de extreme hoeken) de eerste servo (licht) gaan trillen tijdens het bedienen van de sweep-knop van de tweede servo. Controleer dat.

**Vraag 1: Verklaar hoe dat komt.**

**Bij mij gaat die niet trillen, maar het zou kunnen omdat de Aref pin kijk naar het hoogste voltage en de input daarmee vergelijkt. Omdat de aherf nu is aangesloten na het gebruik van servo’s die pieken kunnen creëren kan het zijn dat het vergelijken dus niet klopt en de andere arduino gaat trillen**

# Experiment 2 : Aref loshalen

Ga uit van het schema van de “goede start”, maar haal dit keer Aref geheel los.

Afbeelding met tekst, elektronica

Description automatically generated

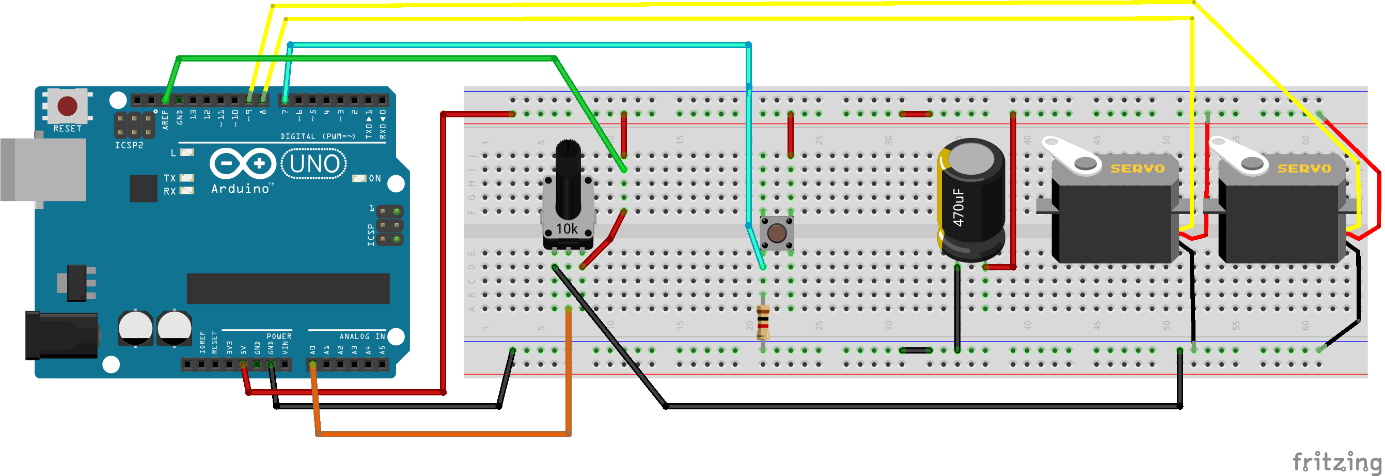
Vermoedelijk zal nu in sommige standen van de potentiometer (niet op, maar in de buurt van de extreme hoeken) de eerste servo (licht) gaan trillen tijdens het bedienen van de sweep-knop van de tweede servo. Controleer dat.

**Vraag 2: Verklaar hoe dat komt.**

**Dit klopt wel, Dit komt doordat we nu geen refrence voltage hebben**

# Experiment 3 : Potmeter ground verleggen

Herstel het schema van de “goede start” weer (Aref weer op de oude plek aansluiten), maar verleg nu de ground van de potmeter:



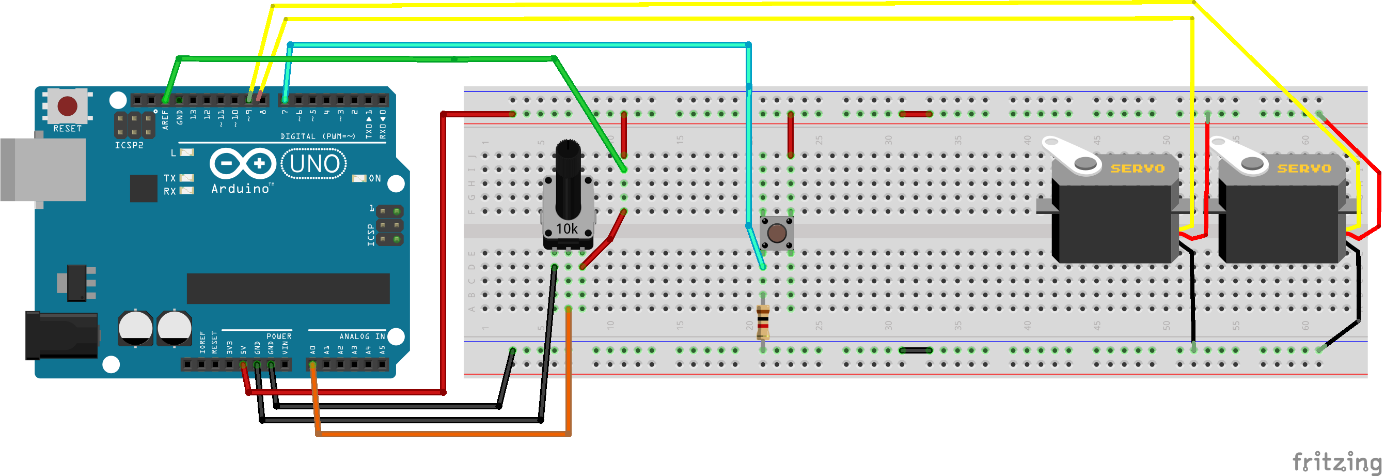
Vermoedelijk zal nu in sommige standen van de potentiometer (niet op, maar in de buurt van de extreme hoeken) de eerste servo (licht) ook in deze situatie gaan trillen tijdens het bedienen van de sweep-knop van de tweede servo. Controleer dat.

**Vraag 3: Verklaar hoe dat komt.**

**Klopt, dit komt doordat de ground vervuild is, elk draadje kan je ook zien als een hele kleine weerstand kan het zijn dat op dat punt ground niet meer echt de ground is**

# Experiment 4 : Elco weglaten

Herstel het schema van de “goede start” weer (grounds weer op de oude plek aansluiten), maar verwijder nu de elco:



Vermoedelijk zal nu in sommige standen van de potentiometer (niet op, maar in de buurt van de extreme hoeken) de eerste servo (licht) ook in deze situatie gaan trillen tijdens het bedienen van de sweep-knop van de tweede servo. Controleer dat.

**Vraag 4: Verklaar hoe dat komt.**

**Dit doet die bij mij niet, maar de elco vangt in dit geval piekstromen op en nu is die weg dus dat zou ervoor kunnen zorgen dat die piekstromen de andere servo ook iets laten trillen**

# Experiment 5 : Tweede voeding gebruiken

Zet nu de elco weer terug, en voed de servos als onderstaand uit een tweede voeding. Bijvoorbeeld een tweede Arduino (die kan evt aan een andere laptop hangen):

Afbeelding met tekst, circuit, schermafbeelding

Description automatically generated

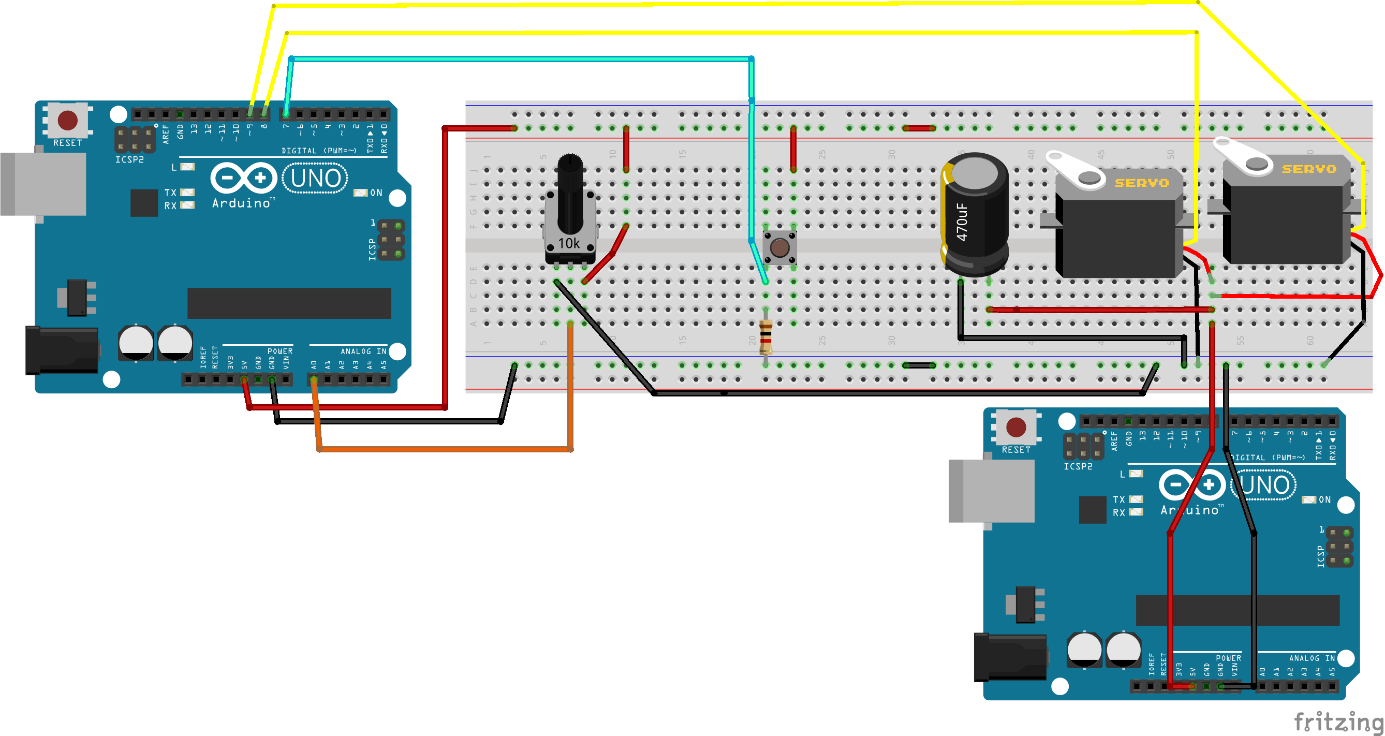
Controleer dat het nog steeds goed (zonder trillen) werkt.

**Vraag 5: Wat is het voordeel van het toevoegen van (een of meer) voedingen?**

**Alles werkt goed, dit komt doordat je een gedeelde grond hebt en zo alles met elkaar kan communiceren maar tegelijkertijd kunnen de servo’s geen ruis veroorzaken die de metingen bij het arduino deel vestoren**

# Experiment 6 : Tweede voeding gebruiken – met Aref losgehaald en verlegde ground

Nu gaan we zowel Aref loshalen als ground verleggen (op dezelfde manier als tevoren).



**Vraag 6a: Welk resultaat verwacht je als je weer gaat controleren of de eerste servo in bepaalde standen gaat trillen als je het sweep effect van de tweede servo activeert?**ik verwacht dat alles werkt zoals het hoort

**Vraag 6b: Welk resultaat nam je waar?**

Alles gaat goed, geen trillingen

**Vraag 6c: Verklaar hoe dat komt.**

Aref lijke me minder nodig aangezein de ground nu niet zo wordt vervuild aangezien de servo’s op een andere power suply zitten

# Experiment 7: Software aanpassing

Sluit de zaak aan op een manier waarbij de servo’s elkaar zo veel mogelijk beinvloeden.  
Bijvoorbeeld zowel aref weglaten als ground verleggen als condensator weglaten.  
  
Probeer de software aan te passen zodanig dat de beinvloeding minder wordt.  
(Hint: als een servo ineens een grote sprong moet maken, trekt hij een grote piekstroom..)

**Vraag 7a: Hoe ziet je software aanpassing eruit, en waarom?**

**Ik heb een grotere delay erin gezet en ook na elke update van myservo1 een delay erin gezet, dan heeft die langer de tijd om de piek niet mee te pakken**

**Verder heb ik myservo1 ook aangepast dat die stapjes van 4 maakt ipv van 0 naar 180 naar 0**

**Vraag 7b: Komt het resultaat overeen met je verwachting?**

**Ja het resultaat word zeker beter maar is niet volledig weg, waneer beide servo’s op het hoogst staan krijgt servo0 nog wel wat trillingen, maar zeker niet zo erg als voor de software aanpassingen**

Appendix 1 - Arduino sketch

#include <Servo.h>

Servo myservo0**;** // create servo object to control a servo

Servo myservo1**;** // create servo object to control a servo

int potpin **=** A0**;** // analog pin used to connect the potentiometer

int val**;** // variable to read the value from the analog pin

int buttonPin **=** 7**;**

int pwmPinServo0 **=** 8**;**

int pwmPinServo1 **=** 9**;**

void setup**()** **{**

pinMode**(**buttonPin**,**INPUT**);**

myservo0**.**attach**(**pwmPinServo0**);** // attaches the servo on pin 9 to the servo object

myservo1**.**attach**(**pwmPinServo1**);** // attaches the servo on pin 8 to the servo object

**}**

void loop**()** **{**

val **=** analogRead**(**potpin**);** // reads the value of the potentiometer (value between 0 and 1023)

val **=** map**(**val**,** 0**,** 1023**,** 0**,** 180**);** // scale it for use with the servo (value between 0 and 180)

myservo0**.**write**(**val**);** // sets the servo position according to the scaled value

delay**(**15**);** // waits for the servo to get there

**if(**digitalRead**(**buttonPin**))**

**{**

myservo1**.**write**(**180**);**

**}**

**else**

**{**

myservo1**.**write**(**0**);**

**}**

**}**

Mijn code na aanpassingen

#include <Servo.h>

Servo myservo0; // create servo object to control a servo

Servo myservo1; // create servo object to control a servo

int potpin = A0; // analog pin used to connect the potentiometer

int val = -1; // variable to read the value from the analog pin

int prev\_val = -1;

int writen = 0;

int buttonPin = 7;

int pwmPinServo0 = 8;

int pwmPinServo1 = 9;

void setup() {

pinMode(buttonPin,INPUT);

myservo0.attach(pwmPinServo0); // attaches the servo on pin 9 to the servo object

myservo1.attach(pwmPinServo1); // attaches the servo on pin 8 to the servo object

myservo1.write(writen);

}

void loop() {

val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)

val = map(val, 0, 1023, 0, 180); // scale it for use with the servo (value between 0 and 180)

myservo0.write(val); // sets the servo position according to the scaled value

delay(50); // waits for the servo to get there

if(digitalRead(buttonPin)) {

if(writen < 180){

writen += 4;

myservo1.write(writen);

delay(50);

}

}

else {

if(writen > 0){

writen -= 4;

myservo1.write(writen);

delay(50);

}

}

}