



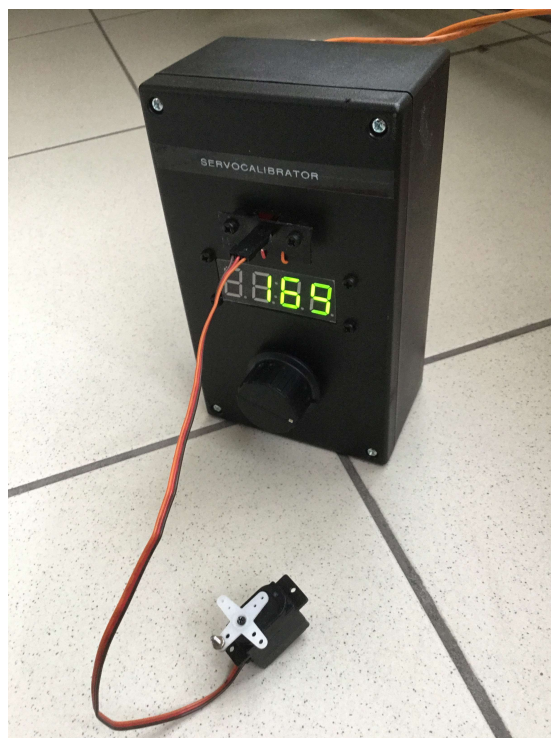
On East Belgian Railtracks / ELO

Servo-calibrator

Eigenlijk is het toestelletje dat hierna beschreven wordt geen echte calibrator aangezien het aan de servo's in kwestie geen aanpassingen toebrengt.

Wat doet het dan wel ?

Om met servo's wisseltongen, seinpaletten of ontkoppelaars in de juiste standen te kunnen zetten en houden is het zaak om de uiterste standen van de bewogen spoortoestellen te bepalen. In de aansturingsoftware moeten die uitwijkingen immers ingegeven kunnen worden, en bij voorkeur dan nog in een aantal graden tussen 0 en 180 omdat de gekende voorgeschreven functiebibliotheken de servo's meestal op deze manier aansturen (een aansturing gebaseerd op timings komt ook vaak voor).



Met deze servocalibrator kan men een servo in situ de gewenste uitwijkingen laten aannemen en op een display de overeenkomstige stand in graden aflezen.

Op de case vallen naast het display nog de grote draaiknop en de servoconnectie op.

Gebruikte componenten

- Arduino (of cloon genre Geekcreit of Robotdyn) UNO met gebruik van de Arduino.cc servo bibliotheek **Servo.h**

Deze bibliotheek is standaard al in de configuratiesoftware (de IDE) van Arduino.cc opgenomen en kan zonder meer aangeroepen worden.

Een UNO (revisie 3) is zo goedkoop te verkrijgen dat het niet de moeite loont om het processor-IC'tje en de componentjes die het minimaal nodig heeft om goed te werken (bvb het 16MHz kristal, of de 5V regulator) op een eigen bordje te gaan solderen. Bovendien hou je dankzij de USB-connectie de mogelijkheid om de sketch aan te passen, of om de schakeling via de seriële monitor van de Arduino IDE te monitoren en/of aan te sturen.

In dit project worden volgende pinnen gebruikt:

- *jack-connector : toevoer van 0v en 7v vanuit de spanningsregulator*

- *pin 0 en pin 1: voorbehouden voor de USB communicatie*

- *pin 9: PWM signaal voor de servomotor*

- *pin 11: CLK (kloksignaal) voor het display*

- *pin 12 : DIO (datastroom) voor het display*

- *pin 15: analoge pin A1 om de ingestelde waarde op de potentiometer mee in te lezen*

- de 'Seinen' UNO-printboard (dit is een niet commercieel en online vrij beschikbaar printontwerp)

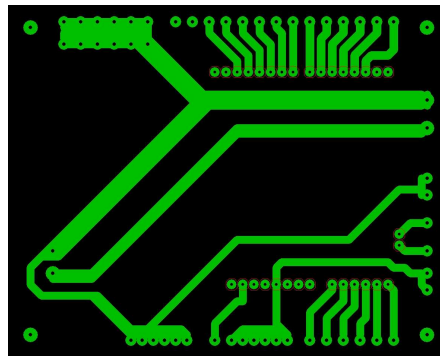
Download link:

<http://belgie-belgique-belgien.be/seinen-signaux-signale/arduino/pcb/unobreakoutv1.LAY>

Gewoon een en ander met de Arduino verbinden via pennen kan ook, omwille van de betrouwbaarheid van de verbindingen valt zo'n shield als dit hier toch te prefereren.

Opgelet, op dit OEER-bordje zit de Arduino op z'n kop, dus met de componenten naar het shield toe gericht, geprikt. Het resultaat is een compactere sandwich dan doorgaans met gewone shields gevormd wordt, en het opprikken lukt iets makkelijker dan met de lange pennen van een normaal shield. Bovendien heeft dit bordje 'normale' boorgaten die toelaten om het wat makkelijker in de behuizing vast te zetten dan het geval is met de wat

bizarre configuratie van de Arduino's.



- een potentiometer met knop

Bij voorkeur een lineaire potmeter van bvb. 100k

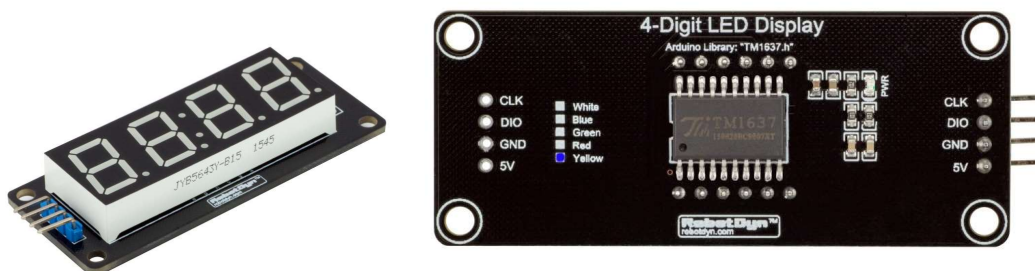
- een servoconnector

standaard 3 pinnetjes op 2.5 mm van elkaar

kleurencode en aansluiting op het Arduino-bordje:

<i>oranje</i>	<i>pwm</i>	<i>Arduino pin 9</i>
<i>rood</i>	<i>5v</i>	<i>Arduino uit 5v</i>
<i>bruin</i>	<i>0v</i>	<i>massa Arduino en voeding</i>

- een display, in dit geval werd gebruik gemaakt van het TM1637 led display met de bijhorende arduino-bibliotheek **TM1637.h** .



Op dit display zitten slechts 4 aansluitingen:

CLK: op deze ingang krijgt het display van de Arduino pulsjes om de datastroom te triggeren

-> Arduino PIN 11

DIO: de datastroom zelf komt serieel uit een tweede Arduino-pinnetje

-> Arduino PIN 12

GND: de massa-aansluiting wordt op de massa-aansluiting van het Arduino-bord gekoppeld

-> 0v massa

5v: en de voedingsspanning komt van de 5v uitgang van het Arduino-bordje

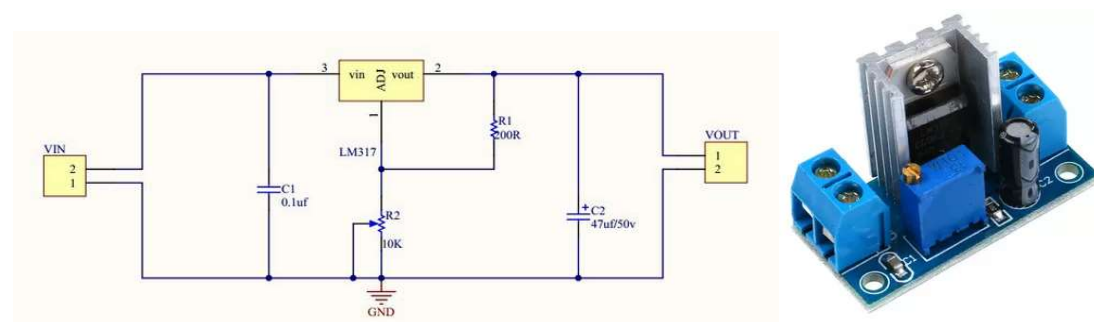
-> Arduino 5v uit

Het display vergt slechts 80mA om te werken. Dankzij de regulator (zie verderop) die voor het Arduino-bordje geschakeld wordt is er ruimschoots marge om ook het servomotortje nog aan te sturen (1A - 80mA = ruim 900 mA)

Download link library:

<https://robotdyn.com/pub/media/OG-00005632==Mod-LED-Display-4D-7S-TM1637-30x14mm/DOCS/TM1637.zip>

- spanningsregelaar om de Arduino niet met 12v maar veeleer met 7v te voeden, in dit geval werd gebruik gemaakt van een kant-en-klare module op basis van de LM317 (gewoon wegens beschikbaar en zo goedkoop dat zelf maken onnozel zou zijn)

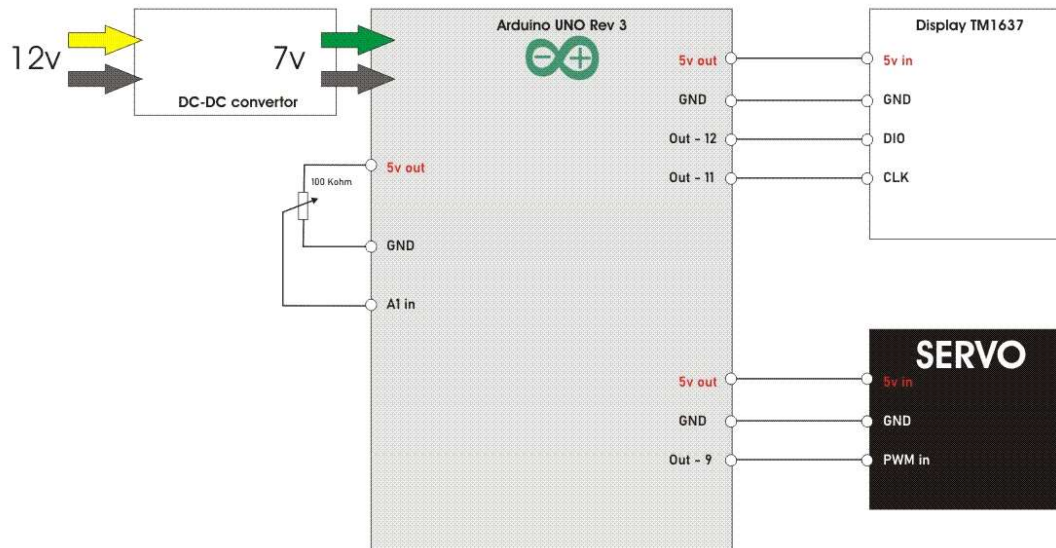


Op de Arduino zit sowieso ook een spanningsregelaartje, veelal van het type 7805 om de schakeling met 5v te voeden. Aangezien we de servo ook via deze regelaar gaan stroom leveren is het belangrijk om de spanning die de regelaar aan de ingang ziet zo laag mogelijk te houden. Idealiter wordt het Arduino-bordje op een externe spanning van zo'n 7v aangesloten, het kan dan gecumuleerd een stroom van bijna een ampère aan externe componenten leveren zonder dat de interne regelaar het begeeft onder de gedissipeerde warmte.


Deze extra regelaar plaatsen we dan ook bij om het verschil tussen de gangbare voedingsspanning van 12v en de gewenste 7v te kunnen verwerken. De gehele schakeling met 5v voeden is helaas niet mogelijk zonder aanpassingen aan het Arduino bordje, vandaar dat deze extra schakeling -die echter zo goed als niks kost- toch wordt opgenomen.

- een kleine behuizing, in dit geval van Ohmeron uit Aalst.

Schematische weergave van de opstelling



OEBR / Staf	
Arduino	Servo - calibrator
ELO	Circuit



De sketch

/* -----

OEBR - Servo calibratiehulp met TM1637 led display

display gebruikt pinnen 11 CLK en 12 DATA (DIO)

servo gebruikt pin 9 PWM

potentiometer aan A1 (pin 15)

adc meet van 0 tot 5v

en leest uit van 0 tot 1023

waarde wordt gemapt naar 0 tot 180 graden voor de servo

-----*/

```

// declaratiesequentie

#include "TM1637.h" // library voor led display
#define CLK 11 // pin 11 koppelen met CLK
#define DIO 12 // pin 12 koppelen met DIO

TM1637 disp(CLK,DIO); //aanmaken object voor displaysturing

#define SERVOPIN 9 // pin 9 wordt pwm uitgang

#include <Servo.h> // library voor aansturing servomotor
Servo testservo; // aanmaken object voor servo aansturing

#define potpin A1 // adc 1 leest potentiometer

int potval = 0; // variabele om adc waarde in te stockeren
int graden = 0; // variabele om geconverteerde waarde ts 0 en 180 in te stockeren

////////////////////////////////////

// eenmalig uit te voeren sequentie

void setup()
{

// helderheid display instellen tussen 0 en 7

```



```
disp.display(graden); // de uitlezing staat in graden
```

```
// einde van de uitvoersequentie
```

```
}
```