

Opleiding Elektromechanica

Geïntegreerde proef



Het automatisch volgen van de zon met het oog op verbetering van energieopwekking met zonnepaneel

Quinten Vercruysse

Vakleerkracht: Reynders Erik

Opleiding: Elektromechanica

Studiejaar: 2021 - 2022

Inhoud

Voorwoord:	3
Verloop van het bouwproces	4
Hoe werkt alles?	5
De gelijkstroom motor:.....	5
Eindeloopschakelaar:.....	6
Transformator:.....	6
Zonnepanelen:.....	6
Voltmeter:.....	6
Arduino uno rev 3	6
Specificaties van de arduino	7
Arduino 4relay shield:.....	7
Tandwielen:	7
Kogellager:	7
LDR's:	7
Weerstanden:	8
Het programma:	8
Uitleg van het programma	8
Bronnen:	13
Bijlage:	14
Projectfoto's	14
Elektrisch schema	17

Voorwoord:

De geïntegreerde proef gaat over het automatisch volgen van de zon om daaruit volgend een verbetering in de energieopwekking te bekomen. Het project was erg leerrijk en was een verrijking ter afsluiting van mijn studie Elektromechanica. In de toekomst zal het eindwerk geplaatst worden in de achtertuin om verscheidene lampen 's avonds te verlichten. Gelijktijdig zal gezocht worden naar het maximaliseren van de energieopwekking.

Met het oog naar de nood op een duurzamere wereld en de klimaatveranderingen werd in het project voornamelijk gebruik gemaakt van recuperatiematerialen. Alles werd herbruikt, uitgezonderd de Arduino printplaat, voltmeter en zonnepaneeltjes. Verder werden door middel van 3D printers, die beschikbaar werden gesteld op het school, verschillende onderdelen optimaal op maat gemaakt.

Het ontwikkelingsproces heeft zowel tot veel plezier als irritatie geleid. Soms verliep het onderzoek vlot, maar soms stootte ik op uitdagingen die heel wat tijd vergden om tot een oplossing te komen. Tijdens de uitvoering diende mijn denkwijze aangepast te worden om problemen te kunnen oplossen, maar ook om de kostprijs van het project zo laag mogelijk te houden, aangezien het de bedoeling was om een energiebesparend project op basis van een ecologisch verantwoorde manier te creëren. Het plan van aanpak werd doorheen het onderzoeksproces aangepast om tot het eindresultaat te komen.

Graag wil ik nog een aantal personen bedanken voor hun hulp en ondersteuning tijdens mijn project. Ik dank Dhr. Reynders voor alle hulp en handige tips wanneer het even niet meer lukte. Verder ondervond ik ook ondersteuning van mijn vader die me soms tot andere inzichten bracht en door zijn opmerkingen mij verder aan het denken zette. Ik mocht tevens gebruik maken van het werkmateriaal van mijn vader. Het was fijn te ervaren hoe hij mij steunde net zoals mijn moeder en de rest van mijn familie.

Verloop van het bouwproces

Allereerst werd gestart met het opzoeken van vergelijkbare projecten. Al snel werden verscheidene voorbeelden gevonden op de Arduino-site. Arduino ontwerpt, produceert en ondersteunt elektronische apparaten en software, waardoor mensen over de hele wereld gemakkelijk toegang krijgen tot geavanceerde technologieën die interageren met de fysieke wereld (Arduino, 2021). Verschillende projecten van diverse personen werden vervolgens ingekeken om informatie uit te vergaren. Op basis van de informatie werd dan aan de slag gegaan.

Het eerste vraagstuk was hoe best het project gerealiseerd kon worden op basis van de beschikbare materialen. De volgende stap was om af te vragen hoe de opstelling verbeterd kon worden. Bij de afdruk van het online 3D-model werd opgemerkt dat de plaatsing niet handig was met de lichtgevoelige weerstanden, de Light-Dependent Resistors (LDR's). Dit werd daarom vervolgens vanaf nul in het tekenprogramma Solidworks gemaakt, waardoor wel gemakkelijk de LDR's stabiel geïnstalleerd konden worden.

Vervolgens werd gezocht naar de correcte motoren, maar een nieuwe motor kost al snel 100 euro. Zelf was er thuis nog een oud ziekenhuisbed met elektrische bediening in bezit. Bij nazicht bleek dat de motoren op het bed bruikbaar waren voor dit eindproject. Na de demontage van de motoren werd gestart met het bouwen van een ronddraaiende plaat waarop de zonnecellen zouden geplaatst worden. Om dit te verwezenlijken werden een as en aantal kogellagers, die reeds in bezit waren thuis gebruikt. Nadien werd het frame van het bed uitgeslepen om als onderstel te gebruiken. Op dat onderstel werd een houten plank vastgemaakt, waarop de onderdelen (de as en latere elektronica) gemonteerd konden worden.

Nadat de plaat in alle richtingen kon draaien, werd de motor voor de omhoog- en omlaag beweging gemonteerd. Om het geheel te laten roteren, werd onderzocht hoe best de overbrenging gemaakt kon worden. Er werd geconcludeerd dat dit verwezenlijkt kon worden door middel van een tandwiel overbrenging. Aangezien de motor 54 toeren per minuut behaalt en de plaat maar 270° moest draaien op een hele dag, werd een vertragende overbrenging ontwikkeld zodat deze niet te snel zou keren.

Daarbij werd opgemerkt dat een extra tool nodig was om de tandwielen te laten draaien. Een kapotte boormachine bracht de oplossing. Uit die boormachine werd het roterende onderdeel gedemonteerd. Het roterend onderdeel kon dan vervolgens vastgelijmd worden op de tandwielen, waardoor deze konden ronddraaien.

De volgende stap was om ervoor te zorgen dat alles correct zou aangestuurd worden door de Arduino unit. De eerste optie was om dat te doen via de relais, die in de sturingsdoos van de motoren zat. Na het doormeten van de printplaat waarop de relais stonden, werd vastgesteld dat deze te gecompliceerd was voor hetgeen bereikt moest worden.

Een andere oplossing was gebruik te maken van de afstandsbediening van het bed. Deze was erop voorzien de motoren van het bed aan te sturen. De uitgangen van de Arduino moesten dusdanig wel nog gekoppeld kunnen worden met de drukknoppen van de

afstandsbediening. Daarom werd de link tussen de 2 contactpunten van één van de drukknoppen met een relaisuitgang van de Arduino gelegd. Op de Arduino Uno Rev3 werd een printplaat met 4 relais van Arduino gekoppeld. Hierdoor konden de vier bewegingen (omhoog, omlaag, links en rechts) worden aangestuurd.

Via voorgaande stappen kon de aansturing van de motoren behouden worden. Het was alleen nog nodig om aan de achterkant van de afstandsbediening de kabels vast te solderen, het programma te schrijven en de LDR's aansluiten op de Arduino. Er werd ook een zoektocht gedaan naar wat een goede weerstand was voor de LDR's. De testen werden uitgevoerd met weerstanden van 100Ω, 470Ω, 1KΩ en 4,7KΩ. Daarbij bleek dat 1 kΩ het beste uitkwam.

De conclusie werd bekomen doordat de weerstandswaarde van de LDR kleiner wordt naargelang de LDR sterker wordt belicht.

Het schrijven van het programma was wel wat werk, maar na een aantal aanpassingen werd een werkende versie bekomen.

Als afwerking werden op het einde nog een aantal kabels ingekort zodat deze beter pasten in de verdeelkasten. Om te controleren hoeveel de zonnepanelen opleverden werd een voltmeter geplaatst. Zo werd gemerkt dat als er een hoekverschil is, dat toch een aanzienlijk verschil maakt in opbrengst. Waarschijnlijk zullen tijdens de zomer nog verbeteringen aangebracht worden, zodanig dat minder elektriciteit verbruikt wordt.

Hoe werkt alles?

De gelijkstroom motor:

Gelijkstroommotoren bestaan hoofdzakelijk uit twee delen: het stilstaande, buitenste deel wordt de stator genoemd, het draaibare, binnenste deel is de rotor.

De stator is een holle, ijzeren cilinder die aan de binnenzijde voorzien is van een even aantal magneetpolen. Deze magneetpolen wekken het bekrachtigingsveld op tussen de hoofdpolen en door de luchtspleet en de rotor heen. Bij kleine motoren worden hiervoor permanente magneten gebruikt, bij grotere motoren elektromagneten. Gelijkstroommotoren worden daarnaast voorzien van hulppolen en/of compensatiewikkelingen om de optredende ankerreactie tegen te werken.

De rotor is een ijzeren cilinder voorzien van axiale sleuven waarin de rotorwikkelingen liggen. De wikkelingen draaien in het magnetisch veld van de stator. De uiteinden van deze wikkelingen zijn verbonden met de lamellen van de commutator. Via koolborstels die contact maken met de commutatorlamellen wordt de rotor van stroom voorzien.

(Wikipedia, 2022)

Eindeloopschakelaar:

Een eindeloopschakelaar is een schakelaar die bediend wordt door een bewegend object, waardoor het schakelcontact ervan wordt geactiveerd en het object stopt met bewegen. (wikipedia, 2022)

Transformator:

Een transformator dient om spanningen en stromen om te vormen naar een andere hoeveelheid. In dit project wordt de spanning van 230volt naar 24V omgevormd. Dat gebeurt door middel van 2 spoelen waarbij op beide een ander aantal wikkelingen zit.

Zonnepanelen:

Een zonnepaneel heeft gemiddeld 17% tot 22% rendement. Dit betekent dat zeer veel verloren gaat van de zon die erop terecht komt. De zonnepanelen in deze geïntegreerde proef kunnen maximum 18 volt en 10Watt. Ze werden parallel geschakeld om de stroom hoger te krijgen.

Zonder belasting geven de panelen meer dan 18Vdc en lopen deze op tot maximaal 24Vdc

Voltmeter:

Deze voltmeter wordt gebruikt om de spanning die opgewekt wordt door de zonnepanelen te meten.

Arduino uno rev 3

Arduino Uno rev 3 is een microcontrollerbord gebaseerd op de ATmega328P ([datasheet](#)). Het heeft 14 digitale input/output-pinnen (waarvan 6 kunnen worden gebruikt als PWM-uitgangen), 6 analoge ingangen, een 16 MHz keramische resonator (CSTCE16M0V53-R0), een USB-aansluiting, een stroomaansluiting, een ICSP-header en een resetknop. Het bevat alles wat nodig is om de microcontroller te ondersteunen. De Arduino uno rev 3 kan eenvoudig aangesloten worden op een computer met een USB-kabel of door deze te voeden met een AC-naar-DC-adapter of batterij.

"Uno" betekent één in het Italiaans en werd gekozen om de release van Arduino Software (IDE) 1.0 te markeren. Het Uno-bord en versie 1.0 van Arduino Software (IDE) waren de referentieversies van Arduino, nu geëvolueerd naar nieuwere releases. Het Uno-bord is het eerste in een reeks USB Arduino-borden en het referentiemodel voor het Arduino-platform. (Arduino, 2022)

Specificaties van de arduino

Werkingsspanning : 5V

Ingang spanning aanbevolen: 7V tot 12V

Maximale ingang spanning: 6V tot 20V

Maximum DC stroom I/O: 30mA

Maximum DC stroom bij 3,3V pin : 50ma

Werksnelheid 16MHz

Arduino 4relay shield:

Het Arduino 4 Relays Shield is een oplossing voor het aansturen van hoge vermogensbelastingen die niet kunnen worden bestuurd door Arduino's digitale IO's, vanwege de stroom- en spanningslimieten van de controller. Het schild heeft vier relais, elk relais biedt 2-polige wisselcontacten (NO en NC). Om de stroomlimiet van elke uitgang te verhogen zijn de 2 wisselcontacten parallel geschakeld. Vier LED's geven de aan/uit-status van elk relais aan. (Arduino, 2022)

Tandwielen:

Er werd gekozen voor tandwielen van module 1.5 omdat hierbij de tanden nog groot genoeg zijn en de diameter van het tandwiel niet te groot is. Een tandwiel wordt gebruikt om het geheel krachtiger, sneller of trager te laten bewegen. In dit project werd gekozen om de tandwieloverbrenging in het geheel trager te laten bewegen. Dat gebeurt door meerdere tandwielen te gebruiken in de volgorde: 12 -> 40 / 10 -> 40 / 10 -> 80

=> $3.3 \times 4 \times 8 = 106$ omwentelingen van de motor geeft 1 omwenteling van het grootste tandwiel. Omdat er hier van klein naar groot naar klein en weer naar groter gegaan wordt, gaan de aantal omwentelingen per minuut vertragen.

Kogellager:

Een kogellager is een rollend lager dat de wrijving van een draai beweging vermindert door gebruik te maken van bolvormige kogels. (Wikipedia, 2022)

LDR's:

Er worden 4 LDR's gebruikt. Deze dienen om de het zonlicht te kunnen meten. Hoe meer licht er is, hoe groter de weerstand. Indien 2 LDR's vergeleken worden en de ene meting hoger is dan de andere, dan kan afgeleid worden welke motor aangestuurd moet worden.

Weerstanden:

Er worden 4 weerstanden van 1K Ohm gebruikt. Wanneer spanning over de weerstand en LDR gaat in plaats van enkel over de LDR, dan zal kortsluiting vermeden worden. Deze maatregel wordt genomen aangezien de weerstand van de LDR zeer laag kan worden (indien er geen zon aanwezig is).

Over de LDR en weerstand staat een spanning van 5Vdc.

Bij veel licht wordt 3.9V over de weerstand gemeten.

Bij weinig of geen licht wordt 1.7V over de weerstand gemeten.

Het programma:

Het programma werd gemaakt om aan de hand van de weerstand van de LDR's de motoren aan te sturen. Dit programma werkt volledig zelfstandig, waardoor éénmaal deze aan staat er geen handelingen meer nodig. Het programma is geschreven in de programmeer taal C++.

Uitleg van het programma

```
int analogInPin1 = A1;
int analogInPin2 = A2;
int analogInPin3 = A3;
int analogInPin4 = A4;
```

Bij deze lijn krijgt de analoge ingang pin A1 de naam analogInPin1. Dat gebeurt door tussen analogInPin1 en A1 een = teken te zetten. Dat teken betekent dat ze aan elkaar gelijk zijn.

Het commando int betekent op zijn beurt dan weer dat de computer bij het compileren er een geheel getal van moet maken. Dat getal situeert zich tussen de -32 000 en de 32 000.

Het ; symbool betekent dat men op het einde van de statement is. Dat komt achter elk statement.

```
int outPin1 = 8; .
int outPin2 = 12;
int outPin3 = 4; .
int outPin4 = 7; .
|
```

Hier worden dezelfde handelingen herhaald, maar dan voor digitale pins.


```
void setup() {
```

Door het commando void wordt er binnen de haakjes van void niks van informatie met het hele programma gedeeld, maar enkel binnen die void.

Binnen de haakjes van de Setup wordt alles 1 keer uitgevoerd bijvoorbeeld *pinMode(1, OUTPUT)*; zal ervoor zorgen dat de digitale pin1 een output wordt en kan niet meer veranderd worden, omdat het in de void staat. Als het programma weer opnieuw opgestart wordt dan overloopt het programma het proces opnieuw 1 keer voor dat gedeelte, maar het blijft zich niet continu herhalen zoals sommige andere delen van het programma.

{ dit haakje betekent dat het deel van de void setup() start.

} dit haakje betekent dan weer dat het deel stopt.

```
Serial.begin(9600);
```

“Serial.begin(9600);” dient om de signaal snelheid van de bits per seconde in te stellen. De meest frequente gebruikte is 9600 baud.

```
pinMode(outPin1, OUTPUT);  
pinMode(outPin2, OUTPUT);  
pinMode(outPin3, OUTPUT);  
pinMode(outPin4, OUTPUT);
```

Hier worden alle outputpins ingesteld als een output. Dit gebeurt door eerst het commando pinMode te gebruiken. Daarna worden de haakjes geopend. Binnen de haakjes wordt eerst de pin geplaatst, daarna een komma om aan te geven dat dit het einde van het woord is en na de komma wordt de soort I/O geplaatst in dit geval output. Vervolgens worden de haakjes gesloten.

```
void loop() {
```

Dit betekent dat alles zich zal blijven herhalen binnen de void.

```
int sensorValue1 = analogRead(analogInPin1);
```

int: zal het woord sensorValue1 als een geheel getal opslaan. *sensorValue1* benoemt dan analogRead (analogInPin1).

```
delay(100);
```

Dit commando zegt dat het programma even moet stoppen gedurende 100 miliseconde.

Dit herhaalt zich voor 3 extra keren maar dan voor de andere LDR's.

```
if (sensorValue2 +20 < sensorValue4) digitalWrite(outPin4, HIGH);  
if (sensorValue2 +20 < sensorValue4) digitalWrite(outPin2, LOW);  
if (sensorValue2 > 20 + sensorValue4) digitalWrite(outPin2, HIGH);  
if (sensorValue2 > 20 + sensorValue4) digitalWrite(outPin4, LOW);  
if (sensorValue1 +20 < sensorValue3) digitalWrite(outPin3, HIGH);  
if (sensorValue1 +20 < sensorValue3) digitalWrite(outPin1, LOW);  
if (sensorValue1 > 20 + sensorValue3) digitalWrite(outPin1, HIGH);  
if (sensorValue1 > 20 + sensorValue3) digitalWrite(outPin3, LOW);  
delay(500);|
```

Eerst wordt het commando if geplaatst. Dan worden de haakjes geopend. Binnen de haakjes wordt een vergelijking (groter of kleiner dan) gemaakt tussen de 3 waardes. Als de vergelijking klopt, zal de actie erachter uitgevoerd worden. Indien de vergelijking niet klopt, wordt de actie na de if niet uitgevoerd.

Het commando digitalWrite geeft aan dat een digitale pin van actie wordt veranderd. In dit geval wordt er voor gekozen dat de spanning hoog gaat op outpin4. Dat gebeurt door na digitalWrite de haakjes te openen, de pin te schrijven, daarna een komma. Na die komma volgt HIGH of LOW.

Bij HIGH komt er spanning uit de pin. Bij LOW komt er geen spanning uit de pin.

De delay zorgt er dan weer voor dat het programma niks doet voor 500 miliseconde. Dat zorgt ervoor als er een pin op staat dat deze zeker voor 500 miliseconde aan blijft.

```
digitalWrite(outPin1, LOW);  
digitalWrite(outPin2, LOW);  
digitalWrite(outPin3, LOW);  
digitalWrite(outPin4, LOW);
```

Zoals eerder verteld zorgt het commando ervoor dat de digitale pin oftewel aan of uit gaat gaan. In dit geval gaan alle digitale pins uit.

```
delay(1000);
```

Deze delay zorgt er voor dat het programma 1 seconde (1000 milliseconde) wacht. Dit wordt gedaan om een vertraging te verkrijgen.

```
Serial.print("Potentiometer 1=  :");  
Serial.println(sensorValue1);  
Serial.print("Potentiometer 2=  :");  
Serial.println(sensorValue2);  
Serial.print("Potentiometer 3=  :");  
Serial.println(sensorValue3);  
Serial.print("Potentiometer 4=  :");  
Serial.println(sensorValue4);  
`
```

Het commando "Serial.print" zal op de seriële monitor het woord *Potentiometer 1= :* tonen. Alles binnen "." wordt op het scherm getoond.

"Serial.println:" zal de waarde tonen van sensorValue1.

Hetzelfde geldt voor de 3 andere sensorValues.

```
}
```

Dit teken staat voor het einde van de void loop.

Op volgende pagina wordt het volledige programma getoond.

```

int analogInPin1 = A1;
int analogInPin2 = A2;
int analogInPin3 = A3;
int analogInPin4 = A4;

int outPin1 = 8;
int outPin2 = 12;
int outPin3 = 4;
int outPin4 = 7;

void setup() {

  Serial.begin(9600);

  pinMode(outPin1, OUTPUT);
  pinMode(outPin2, OUTPUT);
  pinMode(outPin3, OUTPUT);
  pinMode(outPin4, OUTPUT);
}

void loop() {
|

int sensorValue1 = analogRead(analogInPin1);

delay(100);
int sensorValue2 = analogRead(analogInPin2);

delay(100);
int sensorValue3 = analogRead(analogInPin3);

delay(100);

int sensorValue4 = analogRead(analogInPin4);

delay(100);

if (sensorValue2 +20 < sensorValue4) digitalWrite(outPin4, HIGH);
if (sensorValue2 +20 < sensorValue4) digitalWrite(outPin2, LOW);
if (sensorValue2 > 20 + sensorValue4) digitalWrite(outPin2, HIGH);
if (sensorValue2 > 20 + sensorValue4) digitalWrite(outPin4, LOW);
if (sensorValue1 +20 < sensorValue3) digitalWrite(outPin3, HIGH);
if (sensorValue1 +20 < sensorValue3) digitalWrite(outPin1, LOW);
if (sensorValue1 > 20 + sensorValue3) digitalWrite(outPin1, HIGH);
if (sensorValue1 > 20 + sensorValue3) digitalWrite(outPin3, LOW);
  delay(500);

digitalWrite(outPin1, LOW);
digitalWrite(outPin2, LOW);
digitalWrite(outPin3, LOW);
digitalWrite(outPin4, LOW);

delay(1000);

Serial.print("Potentiometer 1=  ");
Serial.println(sensorValue1);
Serial.print("Potentiometer 2=  ");
Serial.println(sensorValue2);
Serial.print("Potentiometer 3=  ");
Serial.println(sensorValue3);
Serial.print("Potentiometer 4=  ");
Serial.println(sensorValue4);
}

```

Bronnen:

Arduino. (2021, September 15). *About Arduino*. Récupéré sur Arduino.cc:
<https://www.arduino.cc/en/about>

Arduino. (2022, 06 12). *Arduino 4relay shield*. Récupéré sur Arduino:
<https://store.arduino.cc/collections/shields/products/arduino-4-relays-shield>

Arduino. (2022, 06 12). *Arduino Uno Rev 3*. Récupéré sur Arduino.cc:
<https://store.arduino.cc/products/arduino-uno-rev3>

Wikipedia. (2022, 06 12). *Gelijkstroommotor*. Récupéré sur Wikipedia:
<https://nl.wikipedia.org/wiki/Gelijkstroommotor>

Wikipedia. (2022, Maart 15). *Kogellager*. Récupéré sur Wikipedia:
<https://nl.wikipedia.org/wiki/Kogellager>

wikipedia. (2022, 06 12). *schakelaar*. Récupéré sur wikipedia:
<https://nl.wikipedia.org/wiki/Schakelaar#:~:text=Een%20eindschakelaar%20is%20een%20schakelaar,het%20object%20stopt%20met%20bewegen.&text=Met%20een%20enkelpolige%20schakelaar%20wordt%20%C3%A9%C3%A9n%20geleider%20onderbroken%20ofwel%20doorverbonden>.

Bijlage:
Projectfoto's



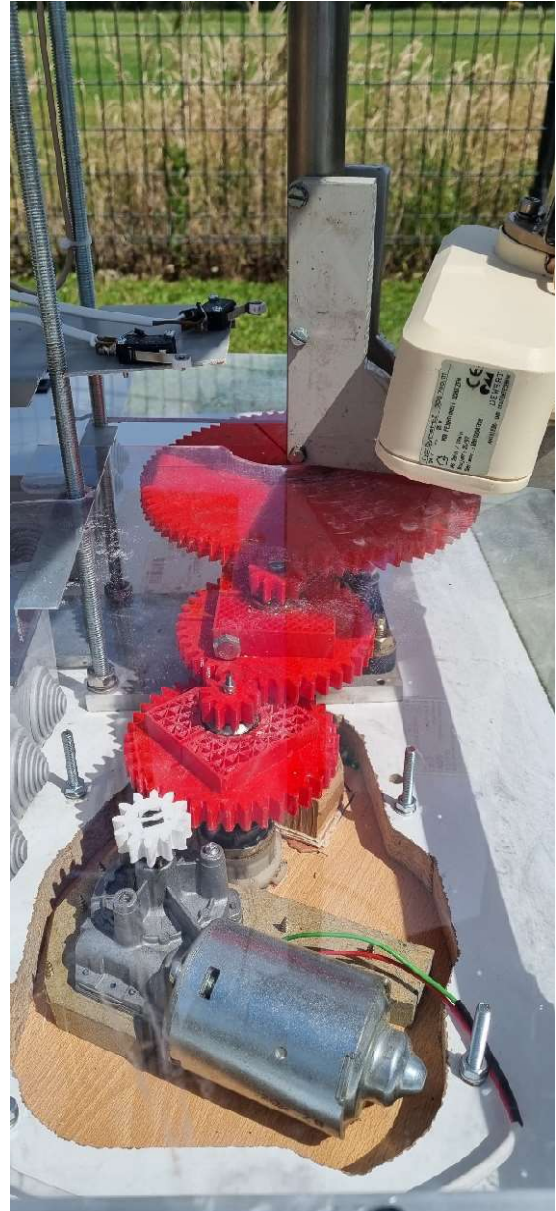
Figuur 1: achterzijde plaat



Figuur 2: voorzijde plaat



Figuur 3: LDR's



Figuur 4: Tandwielen



Figuur 5: Motor voor tandwielen.



Figuur 6: Inhoud verdeeldozen

Elektrisch schema

