



DEUTSCHE
GESELLSCHAFT FÜR
ZERSTÖRUNGSFREIE
PRÜFUNG E.V.

ZfP-Sonderpreis der DGZfP beim Regionalwettbewerb Jugend forscht

LAMPERTHEIM

Intelligente Solarzelle 2.0

Niklas Dillen
Benjamin Eicher

Schule:
Schuldorf Bergstraße

Jugend forscht 2017

Intelligente Solarzelle 2.0

Ben Eicher und Niklas Dillen

Schuldorf Bergstraße

Sparte: Technik

Inhalt

1	Kurzbeschreibung	2
1.1	Kurzbeschreibung des Projekts.....	2
2	Aufgabenstellung	3
2.1	Was soll erreicht werden?.....	3
2.2	Wie sind wir auf das Projekt gekommen?	3
3	Experiment.....	3
3.1	Aufbau des Experiments	3
4	Das Modul.....	4
4.1	Die verwendeten Einzelteile	4
4.2	Das erste Modell	6
4.3	Vorherige Modelle.....	7
4.4	Andere Lösungen	8
4.5	Erweiterungs- und Verbesserungsmöglichkeiten	10
5	Die Programmierung	11
5.1	Programmierung der Lichtsensoren	11
5.2	Weitere Möglichkeiten.....	12
6	Quellenverzeichnis.....	13

1 Kurzbeschreibung

1.1 Kurzbeschreibung des Projekts

Heutzutage ist die Umweltverschmutzung und globale Erderwärmung ein großes Thema, vor allem im Anbetracht der Energiegewinnung. Ein weitverbreiteter Lösungsansatz ist die Solarenergie. Jedoch sind Solarzellen bei weitem nicht so effizient, wie sie es sein könnten, da diese durch ihre statische Bauweise so gut wie nie optimal zur Sonne ausgerichtet sind. Mit unserem Projekt mobilisieren wir die Solarzelle und mit Hilfe von Lichtsensoren ermöglichen wir eine automatische Positionierung und Ausrichtung derer im unmittelbaren Sonnenlicht. Unser Roboter, der auf einem Arduino basiert, steigert somit die Effizienz der Solarzelle auf ein neues Niveau.

2 Aufgabenstellung

2.1 Was soll erreicht werden?

Unser Projekt dient dazu die Effizienz von herkömmlichen Solarzellen zu steigern, indem wir nicht die eigentliche Solarzelle modifizieren, sondern die Positionierung und Ausrichtung automatisieren und perfektionieren. Mit Hilfe von Lichtsensoren, die an allen vier Seiten vorhanden sind, wird die Lichtintensität gemessen und verglichen. Diese Daten werden dann benutzt, um die Solarzelle für eine optimale Sonneneinstrahlung auszurichten.

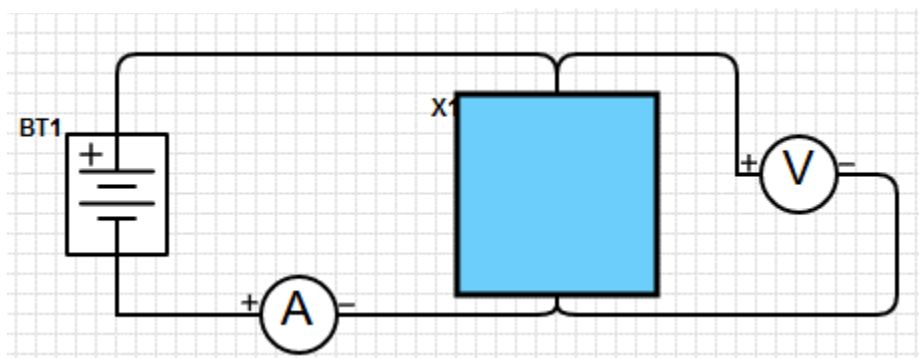
2.2 Wie sind wir auf das Projekt gekommen?

Letztes Jahr haben wir an dem Regionalwettbewerb Schüler experimentieren teilgenommen und mit diesem Projekt versuchen wir das Projekt des letzten Jahres zu verbessern.

3 Experiment

3.1 Aufbau des Experiments

Bild 1



Wir haben zweimal dieselbe Solarzelle, einmal auf dem Roboter, und einmal daneben auf dem Boden. Beide sind an einer Batterie, einem Amperemeter und einem Voltmeter angeschlossen, damit wir die Leistung beider testen können. Der Schaltplan in Bild 1 (oben) zeigt wie wir die Leistung der Solarzellen testen. Nun wird eine Lampe an einer vorgegebenen Strecke entlang bewegt, um die Sonne zu simulieren. Danach können wir anhand der Resultate unserer Messgeräte erkennen,

welche Solarzelle effizienter gearbeitet hat. Natürlich sollten der Roboter und die Solarzelle sollten den gleichen Bedingungen ausgesetzt sein, um genaue Ergebnisse zu erzielen.

Noch ist das Experiment nur in Planung aber hoffentlich werden wir bis zu dem Wettbewerb die Gelegenheit haben das Experiment auch in die Tat umzusetzen.

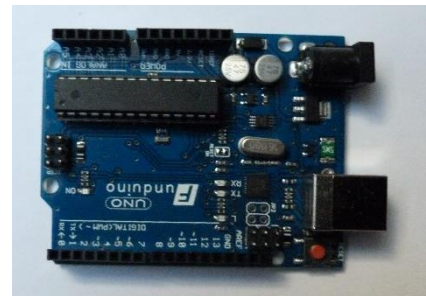
4 Das Modul

4.1 Die verwendeten Einzelteile

Arduino Uno:

Der Arduino Uno ist ein flexibles System zur Steuerung von elektronischen Komponenten, welches auf dem ATmega328P basiert. Die Programmierung ist einfach zu handhaben, da sie über ein USB Kabel läuft. Der Arduino wird mit einer abgewandelten Form von C programmiert, die nicht ganz so schwierig ist wie C oder Assembler (beides sind übliche Programmiersprachen für Microcontroller).

Bild 2



Steuerungsplatine:

Wir haben uns für diesen DC-Motoren Treiber von Arduino (Arduino Motor shield) entschieden, da er direkt als Aufsatz für den Arduino fungiert, und gleichzeitig noch Zugriff auf die restlichen Pins des Arduinos ermöglicht. Außerdem kann man mit dieser Platine zwei Motoren gleichzeitig steuern.

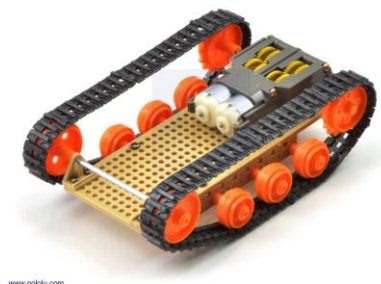
Bild 3



DC-Motoren mit Getriebe und Ketten:

Dies ist ein Modell von Pololu, welches wir zusammengebaut haben. Es besteht aus zwei DC-Motoren, einem Getriebe mit einer Übersetzung von 85:1 und dazugehörige Ketten und Räder.

Bild 4



Powerbank:

Diese Powerbank wird für die Experimente genutzt um den Arduino mobil mit Strom zu versorgen. Unser Modell hat eine Kapazität von 10400mAh.

Bild 5



Servomotor:

Dies ist ein standard Servomotor den wir zur seitlichen Ausrichtung der Solarzelle nutzen. Wir benutzen einen Servomotor, weil er die Solarzelle mit viel Präzision ausrichten kann.

Bild 6



Verbindungsplatine:

Die Verbindungsplatine ist eine selbst gelötete Platine, womit wir die Einzelteile verkabeln und an unser Steuermodul anschließen. Bild 7 zeigt einen Plan unserer Platine:

Bild 7

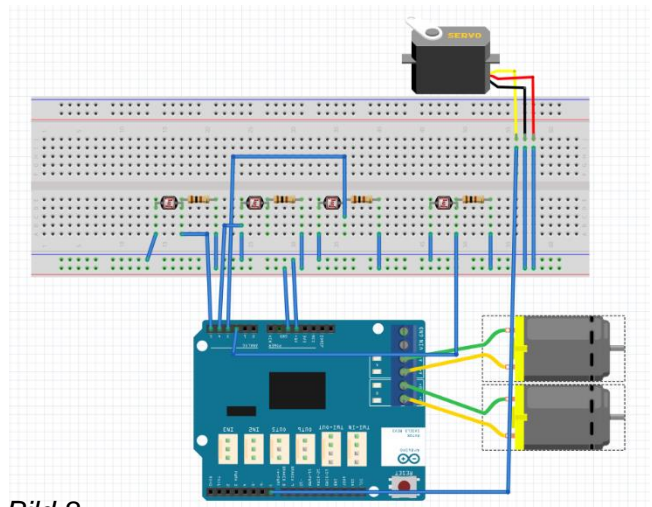
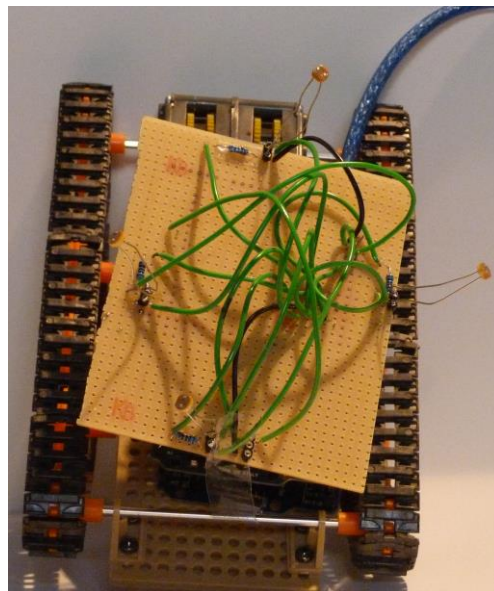


Bild 8

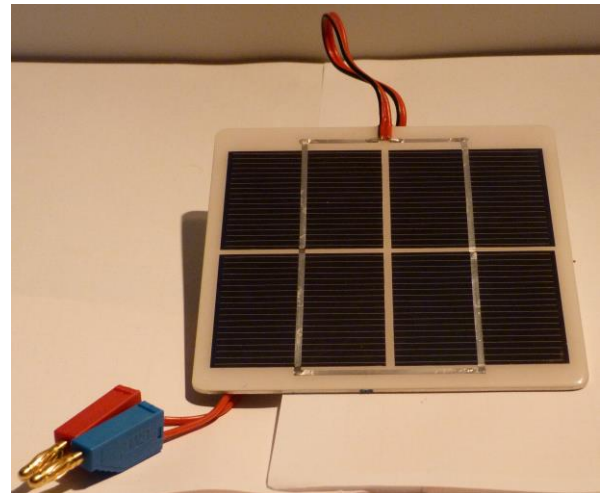
Bild 8 zeigt die fertiggestellte Platine auf dem Roboter.



Solarzelle:

- Für unser Modell benutzen wir eine Solarzelle mit 4 Zellen von Phywe [1]. Sie hat eine maximale Stromspannung von 700mA und eine Gleichspannung von ca. 2V
- Zum Messen der Spannungen benutzen wir das Digitale Messgerät Kobra 4 von Phywe.

Bild 9



4.2 Das Modell

Das Modell besteht aus den oben genannten Einzelteilen. Der Roboter benutzt außerdem noch 4 Umgebungslicht-Sensoren, um die optimale Position zur Stromproduktion zu identifizieren. Bild 10 zeigt nochmal den ungefähren Aufbau des Roboters.

Das folgende Bild zeigt wie der Roboter in etwa nach der Fertigstellung aussehen soll:

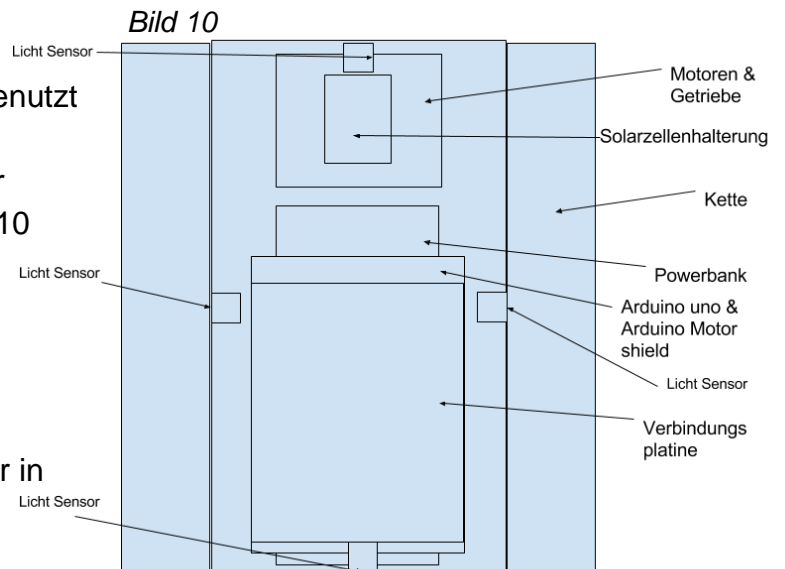


Bild 11

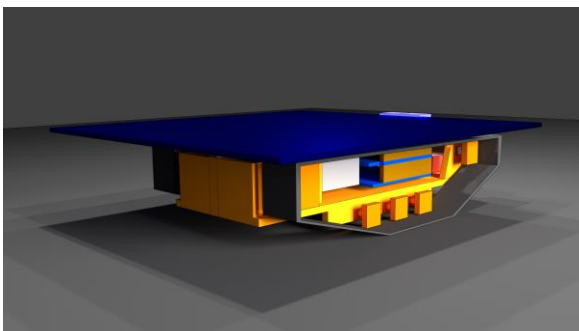
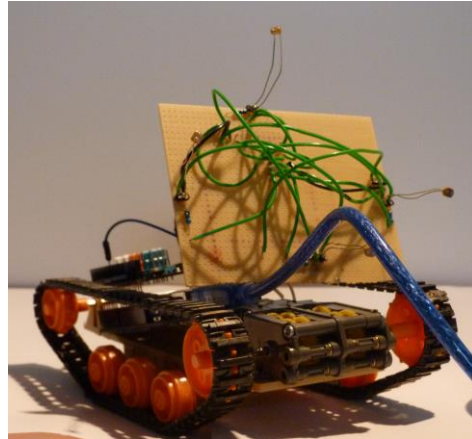


Bild 12

Bild 12 zeigt den Roboter im Zusammenbau:



4.3 Vorherige Modelle

Wir haben für den vorherigen Wettbewerb ein Modell mit dem Lego Mindstorms EV3 gebaut und programmiert. Wir haben auch Tests mit diesem ausgeführt.

Bilder 13 und 14 zeigen den Roboter von letztem Jahr:

Bild 14

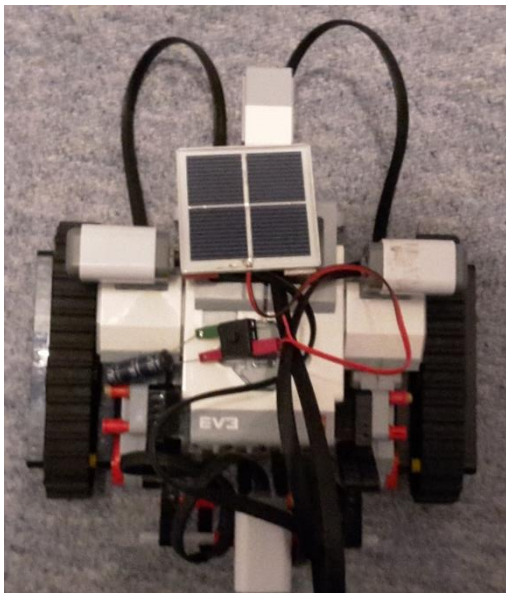
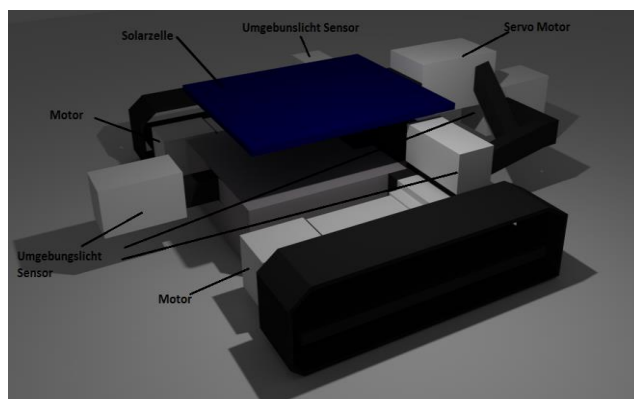


Bild 13



Das Diagramm in Bild 15 zeigt den Aufbau des vorherigen Experiments:

Bild 15

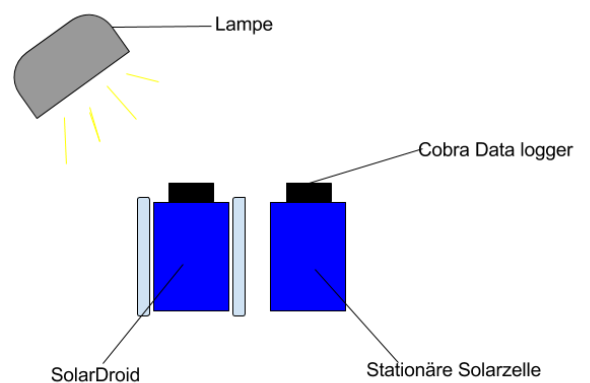
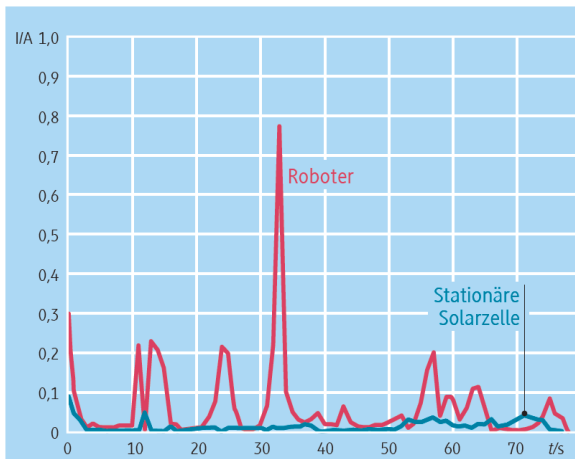


Bild 16



Dieses Diagramm in Bild 16 zeigt die Auswertung der Ergebnisse von dem alten Roboter.

Wie man sehen kann ist der alte Roboter effizienter als die Solarzelle. Unseren Berechnungen zufolge, war der Roboter 5,56 mal effizienter. Jedoch ist dieses Ergebnis entstanden ohne den Stromverbrauch des Roboters an sich mit einzurechnen.

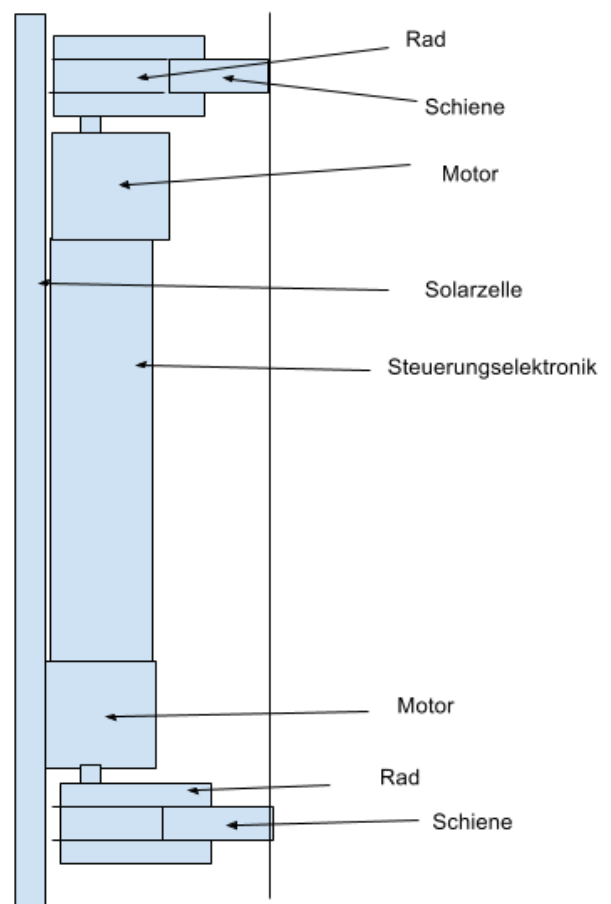
4.4 Andere Lösungen

Satteldach Lösung

Dieses Modell nutzt nur zwei Umgebungslicht-Sensoren, welche die Richtung des Roboters angeben. Dieses Modell läuft auf Schienen und ist dafür ausgelegt große Zellen zu positionieren (siehe Bild 17). Da sich dieses Modell nur linear in 2 Richtungen fortbewegen kann wird es wahrscheinlich nicht ganz so leicht den potentiellen Schatten von Bäumen ausweichen können wie das andere Modell. Dass das Modell auf Schienen läuft, macht es auch für die Benutzung an Wänden brauchbar.

Außerdem sollte das Satteldach nach Süden ausgerichtet sein, um die größtmögliche Effizienz zu gewährleisten.

Bild 17

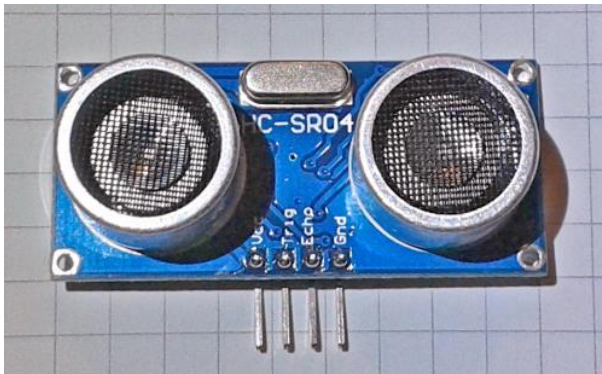


Flachdach Lösung

Dieses Modell ist fast so wie das normale Modell, mit dem einzigen Unterschied, dass es einen Ultraschall Entfernung-Sensor hat. Dieser kümmert sich darum, dass der Roboter nicht vom Dach herunterfährt.

Hier ein Bild eines Ultraschall Entfernung-Sensors:

Bild 18



4.5 Erweiterungs- und Verbesserungsmöglichkeiten

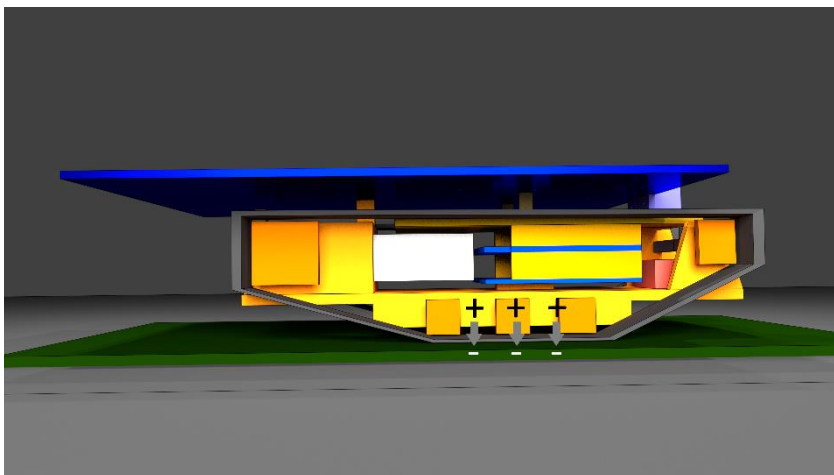
Dieses Modell ist natürlich auch ausbaufähig. Die oberen Modelle sind Abwandlungen von dem ersten Modell, die nur ein wenig Umbau und Umprogrammierung benötigen. Es gibt einige kleinere Probleme mit der Version der Sensoren. Wir benutzen Fotowiderstände, um das Umgebungslicht auf jeder Seite des Roboters zu messen, aber dies kann zu Fehlmessungen führen, da die Widerstände nach einer Zeit beim gleichen Licht nicht mehr den gleichen Wert anzeigen was dann dazu führen kann, dass der Roboter in die falsche Richtung fährt. Man müsste daher nach anderen Lösungen für die Messungen der Lichtintensität suchen.

Überdies könnte man den Arduino auch Daten an ein Netzwerk senden lassen, wo dann beispielsweise ein RaspberryPi die Daten auswertet. Somit bieten sich viele neue Möglichkeiten, wie zum Beispiel die Einprägung eines Tageszyklus, die Programmierung der Dachgröße, Steuerung im Schwarm oder die Analyse von Fahrgewohnheiten der Roboter.

Eine weitere Funktion die man einbauen könnte ist ein Algorithmus zur Einstellung der Messintervalle. Außerdem besteht noch die Möglichkeit die genaue Ausrichtung der Solarzelle zu errechnen. Momentan wird die Solarzelle nur um 45 Grad ausgerichtet.

Man könnte außerdem eine Station zum automatischen Entladen einrichten, die mit Induktion den Strom weitergibt (siehe Bild 19). Bei der Satteldach-Lösung kann man das Abgeben des Stromes durch die Schienen recht einfach ermöglichen. Bei der Flachdachlösung wäre die Energieübertragung auf drahtlosem Weg etwas schwieriger, da potentielle Flüssigkeiten auf dem Dach die Energieübertragung stören können. Letztlich könnte man die Licht-Resistoren durch winzige Solarzellen ersetzen, mit denen man dann etwas Energieeffizienter die Richtung mit der höchsten Lichtintensität identifizieren kann.

Bild 19



5 Die Programmierung

5.1 Programmierung der Lichtsensoren

Zur Programmierung des Roboters haben wir die Arduino IDE benutzt. Diese IDE die auf Java basiert erlaubt es einem in der C-ähnlichen Programmiersprache Arduino zu programmieren, zu kompilieren und die vervollständigten Programme auf den Arduino mittels eines simulierten Serial-Ports über USB hochzuladen, über den man später auch kommunizieren kann.

Das Programm vergleicht ständig die verschiedenen Sensoren und hat für jedes Szenario eine Aktion. So wird zum Beispiel, wenn der rechte Sensor am hellsten ist, der Roboter nach rechts fahren. Wenn dann der vordere Sensor am hellsten ist fährt der Roboter nach vorne.

Das Programm nutzt außerdem einen Timer der einen Messintervall enthält, dessen Länge von der Lichtstärke abhängt. Nach einer Messung wird der Roboter in die entsprechende Richtung gefahren und dann in die richtige Richtung gedreht und die Solarzelle ausgerichtet. Wenn die Lichtintensität jedoch über einem gewissen Wert (700), liegt, hört der Roboter auf zu fahren und der Messintervall wird verlängert.

Die folgende Tabelle zeigt wie die Längen der Messintervalle variiert je nach dem wie hoch die gemessene Lichtintensität ist. Natürlich ist die Länge der Messintervalle variabel und kann noch angepasst werden.

Tabelle 1

Lichtintensität (0-1023)	Intervall Länge (s)
0-400	5
401-600	15
601-1023	30

Die Arduino Programmiersprache erlaubt es einem die Werte von analogen Pins (0 - 1023) und von digitalen Pins (0 oder 1) zu messen, diese aber auch auf eine gewisse Stromstärke zu stellen, so werden zum Beispiel Pins A2 bis A5 analog gemessen, um die Stärke des Umgebungslichts heraus zu finden und die Motoren durch die digitalen Pins 12, 13, 3, 11, 9 und 8 gesteuert und Strommessungen für diese auf dem analogen Pin 0 und 1 durchgeführt.

Tabelle 2 zeigt welcher Pin bei der DC-Motoren Steuerung welche Rolle übernimmt:

Tabelle 2

Funktion	Pins Motor A	Pins Motor B
Drehrichtung	Digital_12	Digital_13
PWM	Digital_3	Digital_11
Bremsen (HIGH=bremse)	Digital_9	Digital_8
Strommessung	Analog_0	Analog_1

Um den Servomotor zu steuern wird eine Bibliothek genutzt (Sammlung vorgefertigter Programme). Diese nimmt viel Schwierigkeit aus dem Programmieren heraus. Der Servomotor wird mit einer bestimmten Länge eines Signals auf dem Signal-Pin (orange) gesteuert. Damit der Programmierer diese Werte nicht selbst ausrechnen muss, wird in der Bibliothek die Grad Zahl automatisch umgewandelt.

Den Gesamten Quelltext der für den Roboter genutzt wurde befindet sich im Anhang.

5.2 Weitere Möglichkeiten

Für die bereits erwähnten Variationen des Roboters müsste man die Programmierung etwas anpassen. Wir haben deshalb hier die anderen Modelle aufgelistet und wie man diese umprogrammieren müsste.

Flachdach Lösung

Der Ultraschall Sensor der in dem Roboter eingebaut ist funktioniert aus etwa 50 cm Entfernung. Wenn der Infrarotsensor einen Gegenstand ermittelt fährt er in die Richtung, die der linke oder der rechte Sensor als die hellste Richtung einstufen.

Satteldach Lösung

In diesem Fall wird nur von zwei Sensoren entschieden in welche Richtung der Roboter fährt. Wenn links heller ist, wird nach links gefahren und auch anders herum.

6 Quellenverzeichnis

- [1] PHYWE: Solarzelle
<https://www.phywe.de/de/06752-20>
- [2] Wikipedia: Drahtlose Energieübertragung
de.wikipedia.org/wiki/Drahtlose_Energie%C3%BCbertragung
- Informationen zur Programmierung:
arduinospiegelwiese.de
- Solarzellenhalterung:
<http://www.exp-tech.de/dagu-pan-tilt-kit-with-servos>
- Arduino und Arduino Motor shield:
www.arduino.cc
- Getriebe und Motor:
<https://www.pololu.com/product/61>
- Pololu Platte:
<https://www.pololu.com/product/79>
- Räder und Ketten:
<https://www.pololu.com/product/106>
- Power bank:
<http://www.ravpower.de/10400mah-power-bank-black.html>

7 Anhang

8.1 Quelltext

```
/*
a0 = rechts = wert1
a1 = links = wert2
a2 = hintern = wert3
a3 = vorne = wert4

*/

#include <Servo.h>

int zeit = 5000; //falls nötig ändern

int wert1;
int wert2;
int wert3;
int wert4;

Servo meinServomotor;

void setup()
{
  pinMode(12, OUTPUT); // Drehrichtung
  pinMode( 3, OUTPUT); // PWM

  pinMode( 9, OUTPUT); // Bremse
  digitalWrite(9, LOW); // nicht bremsen

  pinMode(13, OUTPUT); // Drehrichtung
  pinMode( 11, OUTPUT); // PWM

  pinMode( 8, OUTPUT); // Bremse
  digitalWrite(8, LOW); // nicht bremsen

  meinServomotor.attach(7); //Servo auf 7

  pinMode(13, OUTPUT);
}

void vorwaerts(){

  digitalWrite(12, LOW);
  digitalWrite(13, LOW);

  analogWrite(3, 50);
```



```

    analogWrite(8, 50);

    delay(zeit);

    analogWrite(3, 0);
    analogWrite(8, 0);
}

void rueckwaerts(){
    digitalWrite(12, HIGH);
    digitalWrite(13, HIGH);

    analogWrite(3, 50);
    analogWrite(8, 50);

    delay(zeit);

    analogWrite(3, 0);
    analogWrite(8, 0);
}

void links(){
    digitalWrite(12, LOW);
    digitalWrite(13, HIGH);

    analogWrite(3, 50);
    analogWrite(8, 50);

    delay(zeit);

    analogWrite(3, 0);
    analogWrite(8, 0);
}

void rechts(){

    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);

    analogWrite(3, 50);
    analogWrite(8, 50);

    delay(zeit);

    analogWrite(3, 0);
    analogWrite(8, 0);
}

void loop() {

```

```

//LESE SENSOREN
wert1 = analogRead(A2);
wert2 = analogRead(A3);
wert3 = analogRead(A4);
wert4 = analogRead(A5);

// SUCHE DEN GROECHSTEN WERT
if (wert1 > wert2){
  if (wert1 > wert3){
    if (wert1 > wert4){
      //1
      rechts();
    }else{
      //4
      vorwaerts();
    }
  }else{
    if (wert3 > wert4){
      //3
      rueckwaerts();
    }else{
      //4
      vorwaerts();
    }
  }
}
}else{
  if (wert2 > wert3){
    if (wert2 > wert4){
      //2
      links();
    }else{
      //4
      vorwaerts();
    }
  }else{
    if (wert3 > wert4){
      //3
      rueckwaerts();
    }else{
      //4
      vorwaerts();
    }
  }
}
}
}

```