Quelle der Inspiration:  
<https://wiki.hshl.de/wiki/index.php/Projekt_18b:_Sonnen-Nachf%C3%BChrung>  
<https://www.az-delivery.de/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/solar-tracker-mit-schrittmotoren-und-oled-display>  
[Solar Tracker by Pontifex42 - Thingiverse](https://www.thingiverse.com/thing:5437654)

**Sonnenstand Simulation mit 3W LED und Nachführung des Solarmodul (Arduino Uno)**

### Verwendete Bauteile:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Anzahl | Bezeichnung | Zweck | Komm. | analog IN | analog OUT | Digital IN | Digital OUT | RESET | vorhanden | bestellen |
| 4 | LDR 07 (lichtempfindliche Widerstände) | Erkennen der stärksten Licht-Einstrahlung |  | 4 |  |  |  |  | 2G |  |
| 4 | Widerstände 10kOhm | Pulldown der analogen Eingänge (gegen GND) |  |  |  | 1 |  |  | G |  |
| 1 | Widerstand 220 Ohm | Pulldown des digitalen Eingang (gegen GND) |  |  |  |  |  |  | G |  |
| 2 | Steppmotoren | Antrieb X / Y Achse Solarpanel mit Teiber |  |  |  |  | 8 |  |  |  |
| 2 | Endschalter | für Positionsbestimmung Solarzelle horizontal |  |  |  | 1 |  | 1 |  | 2X |
| 2 | Endschalter | für Positionsbestimmung Solarzelle vertikal (oberer und unterer max. Punkt) |  |  |  | 2 |  |  |  | 2X |
| 1 | Solarzelle | Objekt der Ausrichtung und Spannungserzeugung |  | 1 |  |  |  |  | D |  |
| 1 | Strommesswandler | Spannung und Strom der Solarzelle | I2C |  |  |  |  |  | G |  |
| 1 | OLED Display | Anzeige Spannung und Strom der Solarzelle | I2C |  |  |  |  |  | G |  |
| 1 | Spannungsquelle | Leistungsstarke Versorgung für Sonne und Motoren |  |  |  |  |  |  | G |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 1 | LED 3 W weiß | **Simulation der Sonne mit starker Lichtquelle** |  |  |  |  |  |  | G |  |
| 1 | Relais Ausgang | **Sonne ein- oder ausschalten (zuviel Leistung für Arduino)** |  |  |  |  |  |  | G |  |
| 1 | Steppmotor mit Treiber | Bewegung Tragarm der Sonne über Firmament |  |  |  |  | 4 |  | G |  |
| 1 | Endschalter | Positionserkennung Sonne „Aufgang“ |  |  |  |  |  |  |  | 1X |
| 1 | IC 4051 | Um aus einem analogen Eingang **acht analoge Eingänge** zu machen |  | 1 |  |  | 3 |  |  | 1X |
| 1 | MCP23017 I2C I/O Port Expander | Über I2C angesteuert auf **16 digital Ausgänge** | I2C 0x20 |  |  |  |  |  |  | 1X |
| 1 | Arduino Uno | Zentrale Steuereinheit | I2C und I/O |  |  |  |  |  |  |  |

* 1 Steckplatine
* 3D Drucker zum drucken der Gehäusebauteile

### Verwendete Software

* Arduino Software IDE 2.2.1
* Fritzing

### Lasten und Pflichtenheft

|  |  |
| --- | --- |
| Lasten | Pflichten |
| 1. Entwurf passender Gehäuse für Servomotoren und zur Unterbringung der Fotosensoren. Einarbeitung in die dafür notwendigen Aktoren und Sensoren und deren Schaltung. | Betrachtung bereits existierender Lösungen und Anpassung dieser an eigene Pläne. |
| 2. Erstellung des Schaltplans mit Fritzing und erste Tests in real der Step-Motoren und Sensoren mit Hilfe von Breadboard | Konfigurationsaufbau und Analyse des Aufbaus. |
|  |  |
| 4. Einarbeitung und Erstellung der benötigten Softwaregrundlagen mit Hilfe der Arduino IDE Software | Test verschiedener Gehäusekonfigurationen mit der genuinen Arduino IDE, um auf die angepassten Bibliotheken und Befehlssätze dieser zugreifen zu können, was einen leichteren und direkteren Zugang zum System ermöglicht und sich als zuverlässiger erweist. |
| 5. Fertigstellung und Test des ersten voll lauffähigen Prototyps in Hard- und Software. | Wichtiger Meilenstein erreicht. Grundanforderung an Projekt erfüllt, so dass sich auf Feinjustierung des Systems konzentriert werden kann. |
| 6. Erstellung der Dokumentation des Projekts | Dokumentation des Projekts zur Transparenz des Projekts für dritte. |
| 7. Vorbereitung auf die Präsentation im Rahmen der Abschlussveranstaltung | Erfüllung aller geforderten Standards im Rahmen der Projektarbeit. |

# **Datei:Projektplan final.png**

Ein Bild, das Text, Diagramm, technische Zeichnung, Plan enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Zahl, Reihe enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Reihe, Zahl enthält.

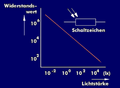
Automatisch generierte Beschreibung

### Projektdurchführung

**Theoretische Grundlagen und Praxis der Fotosensoren**

Die grundlegende Theorie hinter den Fotosensoren ist relativ einfach. Auf dem Auflegerarm werden die 4 Fotowiderstände jeweils zu zwei Paaren oben-unten, links-rechts montiert und an die A0 bis A3 Analogeingänge angeschlossen. Zur Spannungsversorgung dient der 5V Ausgang des Arduino. Diese werden jeweils über 10K Ohm-Widerstände an die Fotosensoren angelegt. Unsere Fotowiderstände gehören zu den Halbleiterbauelementen. Die Funktionsweise beruht dabei auf den inneren Lichtelektrischen Effekt. Treffen die Lichtquanten auf die Halbleiterschicht, werden Elektronen aus den Atomhüllen der Halbleiteratome "herausgeschlagen" und stehen somit als frei bewegliche Ladungsträger zur weiteren Möglichkeit der Ladungsweitergabe zur Verfügung. Somit wird bei steigender Lichtintensität der ohmsche Widerstand gesenkt. Die Fotowiderstände verringern bei steigender Lichtintensität ihren inneren Widerstand, so dass ein höherer Strom gemessen werden kann an den Analogeingängen. Somit kann das Arduino mit jeweils zwei Widerständen für oben und unten und links und rechts und deren Spannungsunterschied, per passenden Softwarebefehlen und if-Anweisungen diese Werte vergleichen und Steuersignale an die Servos geben zur Nachverfolgung der Lichtquelle.

Eingesetzter Fotoresistor

* [](https://wiki.hshl.de/wiki/index.php/Datei:Photo_resistor.jpg)[](https://wiki.hshl.de/wiki/index.php/Datei:Widerstand_fotosensor.png)

Lichtintensität und Spannungsverlauf (idealisiert)

* [Beispielcode Arduino und Fotosensor](https://wiki.hshl.de/wiki/index.php/Datei:Photorescode.png)

Beispielcode Arduino und Fotosensor

**Theoretische Grundlagen und Praxis der Servos**

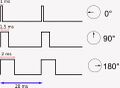
Die Servos werden ebenfalls über die 5V versorgung des Arduino mit Spannung versorgt. Die Steuerungssignale erfolgen hierbei über Pin 9 und 10, die die Ausgabe eines PWM Signals ermöglichen. Hierbei ist auch auf die Verkabelung der Servos zu achten, da in unserem Fall die Ansteuerung nur über das Signalkabel des Servos erfolgen kann. Dabei sorgt ein Servo für die horizontale Drehung im Bereich von 180° bis 45°. Hierbei haben wir die horizontale Bewegungsfreiheit der Servos per Arduino Code künstlich eingeschränkt, da die volle Bewegungsfreiheit von 180° zu Problemen führte, ausgelöst durch zu langen Input der Servos. Der zweite Servo sorgt für die vertikale Rotation, wobei hierbei die Rotation nach unten auch eingeschränkt wurde, da für gewöhnlich keine Sonne von unten auf die Sensoren scheint. Die Abbildungen aus der Bildergalerie verdeutlichen dies nochmal. Für gewöhnlich werden in der Industrie bei Servos als Positions Feedbackgeber hochpräzisionsencoder verwendet. Im Rahmen von Arduino und den Servos, handelt es sich um ein simples Potentiometer, dessen Funktion leicht zu erklären ist. Dabei wird die aktuelle Position, die vom Ponti erfasst und an die Arduino Steuerung weiter gegeben wird, verglichen mit der gewünschten Soll-Position. Per einfachen if-Schleifen und Vergleich der Werte, haben wir die Sonnennachführung so programmiert, dass solange ein Wert der Fotosensoren (horizontal-vertikal) grösser oder kleiner ist, ein Steuerbefehl an die Servos erfolgt, zur Bewegung in die Richtung. Hierbei habne wir auch eine gewisse Toleranz eingebaut, da die Fotosensoren bauartbedingt keine allzu präzise Messung ermöglichen. Innerhalb der Servos befindet sich eine Steuerelektronik, die per H-Brücke Soll und Ist-Wert der Spannung vergleicht und regelt, bis eine Differenz von 0 erreicht ist.

* [Aufbau eines Arduino Servomotors](https://wiki.hshl.de/wiki/index.php/Datei:Servo_end.jpg)

Aufbau eines Arduino Servomotors

* [Regelkreis der Servomotoren](https://wiki.hshl.de/wiki/index.php/Datei:Servoregelkreis.png)

Regelkreis der Servomotoren

* [](https://wiki.hshl.de/wiki/index.php/Datei:ServoMotor2.jpg)

Funktionsweise PWM am Servo

* [](https://wiki.hshl.de/wiki/index.php/Datei:2019-01-18_00_12_58-How_Servo_Motors_Work_%26_How_To_Control_Servos_using_Arduino_-_HowToMechatronics_.png)

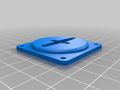
Beispielcode Arduino und Servo

**Gehäuse und 3D-Druck**

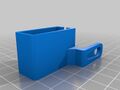
In unserer ersten Konstruktion des Gehäuses zeigte sich leider dass die 3D Druck Modelle nicht optimal geeignet waren für den Einsatz, da die Servomotoren ständig aus ihren Lagerungen sprangen und so ein zuverlässiger Betrieb nicht möglich war. Auch gab es Probleme mit den Toleranzen des Gehäuses, so dass doch ein DIY Ansatz für das Gehäuse gewählt wurde. So war es vor allen Dingen leichter und unproblematischer die Verbindung Servo zu Lagerung zu realisieren, welche sich bei den im 3D Drucker ausgedruckten Gehäuseteilen als immer wieder unzuverlässig erwiesen.

* [](https://wiki.hshl.de/wiki/index.php/Datei:3d_fuss.jpg)

3D Modell Ausleger

* [](https://wiki.hshl.de/wiki/index.php/Datei:3D_plattform.jpg)

3D Modell Plattform

* [](https://wiki.hshl.de/wiki/index.php/Datei:3D_hand2.jpg)

3D Modell Stativ

* [](https://wiki.hshl.de/wiki/index.php/Datei:3D_alle_3.jpeg)

3D Druck der 3 Teile

* [](https://wiki.hshl.de/wiki/index.php/Datei:GK2.jpeg)

Alternatives Gehäuse aus Holz

**Matlab und Simulink**

Prinzipiell war es möglich das Projekt mit Simulink zu verwirklichen. Leider führte die Benutzung von Matlab und Arduino immer wieder zu Abstürzen und Fehlern, egal wie die Testkonfiguration aussah, so dass wir wieder zur Arduino IDE wechselten, welche eine problemlose Nutzung und Funktion ermöglichte.

* [](https://wiki.hshl.de/wiki/index.php/Datei:Simulinksun.png)

Arduino und Simulink

Ergebnis

Im Ergbenis waren wir mit den Resultaten insgesamt zufrieden. Die Schaltung benötigte insgesamt Sorgfalt im Zusammenbau, alle Leitungen und WIderstände mussten durchgemessen werden, um die Senorwerte nicht zu verfälschen. Auch mussten verschiedene Konfigurationen aus Gehäuse/Software/Schaltungen/Servos/Fotoresistoren getestet werden, da sich viele Konfigurationen als problematisch erwiesen. Gerade der Einsatz der Fotosresistoren zeigte, dass hier auf Qualität geachtet werden muss, da die billigen Bauteile leicht zu Fehlmessungen führen, welche das ganze Projekt gefährden. So mussten wir alle benutzen Fotosensoren ausmessen und die zuverlässigsten wurden von uns ausgesucht.

Nach einiger Anpassung und Konfiguration der Software waren und sind wir aber mit dem Ergebnis der "Sonnen-Nachführung" zufrieden.

### Lessons Learned

* Teamarbeit (international)
* Projektplanung und Umsetzung dieser im Detail wie Materialbeschaffung, Recherche etc.
* Konstruktion von Gehäuseteilen per CAD und 3D-Druck dieser Teile
* Konstruktion und Verschaltung von Aktoren und Sensoren samt Schaltplan
* Aneignung von regeltechnischen Grundlagen bezüglich Aktoren im Regelkreis
* Progarmmierung von Arduino mit Matlab Arduino IDE und Anwendung dessen im mechatronischen System
* Präsentation und Dokumentation nicht nur der Ergebnise sondern auch der Zwischenschritte (nach aussen als auch im Team)

## Projektunterlagen

* <https://svn.hshl.de/svn/Elektrotechnik_Fachpraktikum/trunk/Projekte/18b_Sonnennachfuehrung/Unterlagen%20Projekt%2018b/>

## YouTube Video

<https://www.youtube.com/watch?v=gDTOf11C4Cs>

## Weblinks

* <https://howtomechatronics.com/>
* <https://www.arduino.cc/>
* <https://funduino.de/nr-6-fotowiderstand>
* <https://pm-blog.com/2008/05/08/projekte_strukturieren/>
* <https://www.photovoltaik-web.de/photovoltaik/dacheignung/solar-tracker-nachfuehrung-nachfuehrsysteme>

## Literatur