# Energieversorgung

## Akku

Anhand der Konzeptfindung wurde entschieden, dass ein Lithium-Polymer Akku eingesetzt wird. Für die Dimensionierung wurden alle Verbraucher aufgelistet und ihre Leistungen berechnet. In unserer Anforderungsliste ist als Festanforderung eine Akkulaufzeit von mindestens 8 Minuten definiert. Als Wunschzeit wurden 30 Minuten definiert. Für die Berechnung der Akkukapazität wurde eine Akkulaufzeit von 20 Minuten angenommen.

|  |  |  |  |
| --- | --- | --- | --- |
| Komponente | Leistung [W] |  | Zeit t in h |
| Raspberry Pi 1 | 5 |  | 0.35 |
| Raspberry Pi 2 | 5 |  |  |
| Antriebsmotor | 60 |  |  |
| Servo Lenkung | 5 |  |  |
| Servo Greifer | 5 |  |  |
| Servo Heber | 5 |  |  |
| Servo Mulde | 5 |  |  |
| Sensoren | 5 |  |  |
| Webcam | 7 |  |  |
|  | **102** | **W** |  |
|  |  |  |  |
| Strom | 7.29 | A @14V Akkuspannung | |
|  |  |  |  |
| **Akkukapazität** | **2550.00** | **mA/h** |  |

Tabelle - Leistungen der Komponenten

Für die meisten Komponenten gab es nicht wirkliche eine Leistungsangabe. Meist musste diese selbst gerechnet oder gar geschätzt werden. Auch werden nicht alle Komponenten die ganze Zeit voll belastet, was eine geringere Leistungsaufnahme zur Folge hat. Zum Beispiel werden die Servo für den Greifer, Heber und Mulde nie gleichzeitig mit dem Antriebsmotor laufen. Anhand diesen Berechnungen und Überlegungen muss der Akku mindestens 20 Minuten halten.

## Akkuüberwachung

Wie schon in der Konzeptfindung erwähnt, sind die Lithium-Polymer Akkus sehr empfindlich gegenüber Tiefentladung. Dabei muss der Akku überwacht werden und sobald die Zellspannung unter 3.6V sinkt abgeschaltet werden. Diese Funktion wird eine Ladeüberwachungsschaltung mit OPV übernehmen. Die Schaltung ist im Bild unten ersichtlich. Mit dem Potentiometer kann die gewünschte Schaltspannung eingestellt werden. Sobald die Akkuspannung unter sinkt wird ein digitales High Signal am GPIO Eingang des Raspberry Pi angelegt. Dieser gibt den Alarm weiter an das zweite Raspberry Pi. Beide Rechenwerke führen noch Sicherungen durch und fahren herunter.



Abbildung - Schaltung Akkuüberwachung

## Spannungswandlung

Der Akku liefert bei voller Aufladung 16.8V. Bei spätestens 14.4V muss der Akku wieder aufgeladen werden. Somit ist die Eingangsspannung zwischen 14.4 bis 16.8V. Aus dieser Spannung muss die 5V Speisespannung für das Raspberry und die 7V Spannung für die Servos generiert werden. Dies wird mit Buck-Converter gemacht. Es gibt jeweils für die 5V Spannung und die 7V Spannungen eine Buck-Converter Schaltung.

LiPo Akku

14.4V – 16.8V

5V 4A

7V 5A

Raspberry1, Raspberry2, Encoder

Servo Bricklet

Sicherungen

14.4V – 16.8V

Spannug

Motor

Abbildung 2 - Blockschaltbild Spannungen

Die Buck-Converter haben einen sehr hohen Wirkungsgrad und sind preiswert. Die Dimensionierung einer solchen Schalung ist jedoch sehr aufwendig. Deshalb wurde das Webench-Tool von Texas Instruments eingesetzt. Mit diesem Tool kann der Eingangsspannungsbereich, die Ausgangsspannung und der Ausgangsstrom vorgegeben werden und das Tool gibt verschiedene Schaltungsvorschläge. Die daraus resultierenden Schemas sind im Anhang ersichtlich.

# Motoren

## Auswahl

In der Auswahl standen lange zwei verschiedene DC Motoren. In der untenstehenden Tabelle sind die Motoren einander gegenübergestellt:

|  |  |  |
| --- | --- | --- |
|  | Modelcraft RB350050 | Maxon RE30 |
| Bauart | DC Motor | DC Motor |
| Getriebe | 50:1 | 4.8:1 |
| Encoder | Nein | Ja, 512 Puls/U |
| Betriebsspannung | 12V | 24V |
| Leistung | 10W | 60W |
| Drehmoment | 5.39 Nm | 85.6mNm |
| Dimensionen | 106mm x 36mm | 125mm x 30mm |

Tabelle - Vergleich DC Motoren

Ein sehr wichtiges Kriterium für die Regelung eines Motors ist die Rückführung. Beim Maxon Motor ist ein Encoder direkt auf die Motorenwelle geschraubt und liefert 512 Pulse/U. Der Modelcraft Motor ist nicht mit einem Encoder lieferbar. Encoder für die nachträgliche Montage sind grundsätzlich sehr teuer und passen kaum in unser Budget. Weiter kommt dazu, dass der Encoder zusätzlich Mechanisch auf die Antriebswelle angekoppelt werden muss. Dies ist aus Platzgründen aufwendig. Deshalb fiel die Wahl auf den Maxon Motor mit integriertem Encoder.

# Motorensteuerung und Regelung

Um mit dem Fahrzeug Positions- und Geschwindigkeitsgesteuert fahren zu können, wird eine Regelung benötigt. Dazu wird eine Sollgeschwindigkeit vorgegeben und die Geschwindigkeit mithilfe dem Encoder wieder eingelesen. Der Regelkreis ist in der untenstehenden Grafik sichtbar:

Abbildung 3 - Regelkreis

PID Regler

Störungen

UM

ωS

ωM

Encoder

Meistens ist unser Fahrzeug im geschwindigkeitsgeregelten Zustand unterwegs. Die Sollgeschwindigkeit wird von unserer Steuerung vorgegeben und danach über den PID-Regler geregelt. Wenn die Steuerung jedoch einen Container sieht, kann sie die Distanz abschätzen. Nun wird Positionsgesteuert gefahren. Das heisst das Fahrzeugt fährt eine definierte Strecke ab. Dies kann gemacht werden, indem die Pulse gezählt werden. Anhand der Berechnung unten, ergeben sich 355 Pulse für eine Umdrehung.

## Encoder

Der Encoder welcher am Maxon Motor befestigt ist liefert 512 Impulse/Umdrehung. Diese Pulse müssen alle aufgezeichnet werden, damit genau gefahren werden kann. Dies wären bei einer Motorendrehzahl von 600 Umdrehungen (bei geschätzter maximalen Geschwindigkeit) pro Sekunde 300‘000 Pulse pro Sekunde. Dies würde eine sehr grosse Abtastrate zur Folge haben. Deshalb wird ein Schnittstellen-IC verwendet. Dieses beinhaltet einen 32bit Counter, welche vom Encoder aufwärts/herunter gezählt wird. Nun kann mit dem Raspberry periodisch die Werte herausgelesen werden. Der Zeitpunkt dazu kann selber festgelegt werden, ohne dass Pulse verloren gehen.