

BALANCER ROBOT

Berufsmaturitätsarbeit

Berufsmaturitätsschule Zürich
Technik, Architektur, Life Sciences

Betreuer: H.P. Lutz
Abgabe: 26.11.2019

Lacher Markus, BEL16a
markuslacher@hotmail.ch



Abstract

Im Projekt „Balancer Robot“ werden die physikalischen Effekte eines selbstbalancierenden Fahrzeugs analysiert. Ziel des Projekts ist die Entwicklung eines selbstbalancierenden Fahrzeugs.

Mithilfe von Testversuchen wird das Verhalten eines 3-Achsen Beschleunigungs- und Gyrosensors untersucht. Eine selbstentwickelte Elektronik steuert zwei DC-Bürstenmotoren an. Per Mikrocontroller wird der Sensor konfiguriert und ausgelesen. Nach der Verarbeitung der Daten werden die Motoren mithilfe eines Regelungsalgorithmus angesteuert mit dem Versuch, das Fahrzeug auszubalancieren.

Die Ergebnisse zeigen: Durch korrektes Verarbeiten der Sensorwerte lässt sich der Neigungswinkel des Fahrzeugs bestimmen. Durch eine PWM Ansteuerung können die Motoren reguliert werden. Doch erste Testversuche zeigen, dass für ein Ausbalancieren des Fahrzeugs nicht die Spannung über den Motoren, sondern die Stromaufnahme der Motoren reguliert werden sollte, da sich diese im Gegensatz zur Spannung proportional zum Drehmoment der Motoren verhält. Zudem wurde festgestellt, dass ein simpler PID-Regler als Regelungsalgorithmus nicht ausreicht, um das Fahrzeug auszubalancieren.



Inhalt

Abstract	1
Einleitung.....	4
Fragestellung	4
Projektphasen.....	4
Hauptteil.....	5
Konzept.....	5
Physikalisches Grundprinzip	6
Sensorik.....	8
Rotation.....	8
Beschleunigung	8
Konfiguration	8
Datenverarbeitung.....	9
Sensor Auswertung	10
Aktoren	12
Ansteuerung Motor	12
PWM.....	13
Motor Stromaufnahme	14
L298 als neuer Motortreiber.....	16
Layout	17
Akkumulatoren	21
Mechanik	22
Motor	22
Prototyp	23
3D-Model	24
Software.....	25
Architektur	25
Regelungsalgorithmus	26
Allgemein	26
PID-Regler	27
Umsetzung	28
Schluss	29
Quellenverzeichnis	31



Abbildungsverzeichnis.....	33
Dank.....	34
Anhang.....	35
Schema.....	35
Software.....	35
Balancer Robot.....	35
Python Programm	35



Einleitung

Fragestellung

Ist es möglich, mithilfe eines Beschleunigungs- und Gyrosensors und zwei DC-Motoren ein Fahrzeug auf 2 Rädern auszubalancieren? Welche physikalischen Kräfte wirken und wie funktioniert das Ausbalancieren?

In einem YouTube Video sah ich ein kleines selbstgebautes Fahrzeug, welches sich auf 2 Rädern ausbalanciert, indem es vorwärts und rückwärts fährt. Dieses Prinzip faszinierte mich so sehr, dass ich selber ein Fahrzeug bauen wollte, welches sich ausbalanciert.

Grosse Teile dieses Projekts befassen sich mit der Entwicklung einer Elektronik, welche die Sensorik und die Ansteuerung der Motoren zur Verfügung stellt, damit eine Regelungssoftware das Ausbalancieren des Fahrzeugs bewältigen kann. Es werden Überlegungen zur Physik angestellt, sowie Informationen über Regelungstechnik gesammelt, damit eine Elektronik entwickelt werden kann, die ein Ausbalancieren ermöglicht.

Projektphasen

Das Projekt lässt sich in folgende 4 Phasen unterteilen:

- 1) Berechnen, analysieren und verstehen der physikalischen Kräfte
Zu Beginn wurden physikalische Berechnungen durchgeführt, um die wirkenden Kräfte abzuschätzen.
- 2) Entwickeln der Elektronik und der Mechanik
Aufgrund der Überlegungen zur Physik wurden Motoren und Sensoren bestellt. Danach wurde ein erster Prototyp entwickelt.
- 3) Entwickeln der Embedded Software
Für den ersten Prototyp wurde Software entwickelt, welche die Hardware steuert. Dies beinhaltet das Konfigurieren und Auslesen des Sensors, das Ansteuern der Motoren, sowie das Überprüfen der Akku-Ladung. Zusätzlich wurde eine Software für den Computer entwickelt, dass die Messwerte des Sensors analysiert werden können.
- 4) Entwickeln eines Regelalgorithmus
Nachdem der erste Prototyp entwickelt wurde und die Software für das Steuern der Hardware programmiert und getestet wurde, konnten erste Balancier-Versuche mithilfe eines Regelungsalgorithmus durchgeführt werden. Ich informierte mich über PID-Regler und entwickelte einen ersten kleinen Regelungsalgorithmus. Leider mit begrenztem Erfolg, denn das Ausbalancieren funktionierte nur beschränkt.

Im Hauptteil werden diese 4 Phasen chronologisch abgearbeitet. Zuerst wird das grobe Konzept jeder Regelungselektronik sowie das physikalische Grundprinzip erläutert. Nachfolgend sind Information über die Sensorik und die Aktoren dokumentiert. Anschliessend ist die Entwicklung der Leiterplatte, der Mechanik und der Software beschrieben. Abschliessend sind erste Experimente eines Regelalgorithmus dokumentiert.



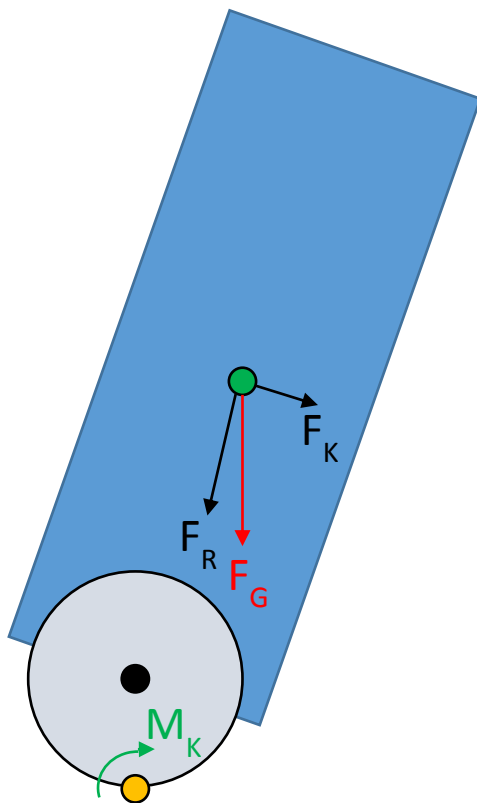
Hauptteil

Konzept

Der Balancer Robot ist im Prinzip nichts anderes als eine Regelungselektronik. Über Sensorik wird eine physikalische Grösse gemessen und mithilfe der Regelung wird ein Aktor angesteuert, sodass sich ein gewollter Sollwert einstellt.

Das Fahrzeug misst mithilfe eines 3-Achsen Gyrosensors und eines 3-Achsen Beschleunigungssensors die Kräfte, die auf das Fahrzeug wirken. Die Messwerte werden von einem Mikrocontroller eingelesen und verarbeitet. Dieser versucht dann die Motoren so anzusteuern, dass das Fahrzeug nicht umfällt.

Physikalisches Grundprinzip



- **Schwerpunkt**
- **Motorachse**
- **Kippachse**

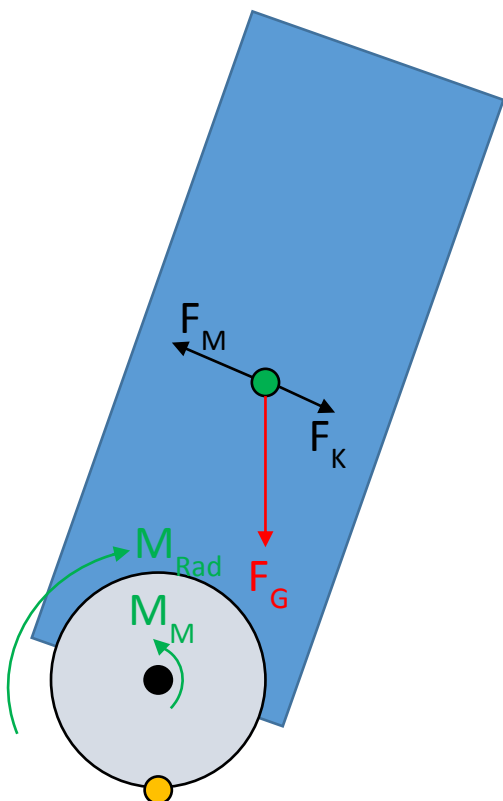
Die Erdanziehungskraft F_G wirkt auf den Schwerpunkt des Fahrzeugs. Die Kraft lässt sich aufteilen in F_R und F_K .

$$F_G = \sqrt{F_R^2 + F_K^2}$$

F_R wirkt diagonal in Richtung der Kippachse und ist für die Verschiebung des Fahrzeugs beim Umfallen verantwortlich.

F_K steht rechtwinklig zur Kippachse beschleunigt die Rotation um die Kippachse.

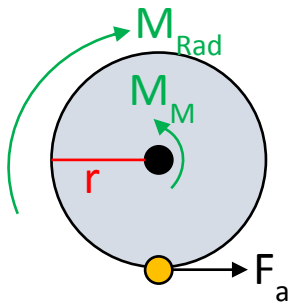
Diese Rotation resultiert in einem Drehmoment M_K auf der Kippachse. Je schief der Fahrzeug steht, desto grösser wird F_K und somit auch M_K .



Um nicht umzufallen muss das Fahrzeug vorwärts fahren. Der Motor erzeugt das Drehmoment M_{Rad} , das Rad beginnt sich im Uhrzeigersinn zu drehen. Laut Newtons drittem Gesetz folgt auf eine Kraft immer eine Gegenkraft. Dies bedeutet, dass das Drehmoment M_{Rad} eine Gegenkraft, bzw. Gegendrehmoment auf die Motorachse ausübt. Diese Gegenkraft wird mit M_M bezeichnet.

Das Drehmoment M_M der Motorachse erzeugt eine Kraft F_M im Schwerpunkt, welche das Fahrzeug wieder aufrichtet.

F_M grösser F_K resultiert im Aufrichten des Fahrzeugs, F_M kleiner F_K resultiert im Umfallen des Fahrzeugs.



Dividiert man das Drehmoment des Rades durch dessen Radius, erhält man die Kraft F_a , welche das Fahrzeug relativ zum Boden beschleunigt.

$$F_a = \frac{M_{Rad}}{r}$$

Die beschleunigende Kraft F_a ist wichtig, da die Reifen einen Reibungskoeffizienten besitzen. Die Gewichtskraft F_G multipliziert mit dem Reibungskoeffizienten definiert die maximal mögliche Beschleunigungskraft F_a .

$$F_a \leq F_G * \mu_{reifen}$$

Der erste Prototyp wiegt 788g. Der Reibungskoeffizient des Reifen ist abhängig von der Oberfläche, auf der dieser rollt. Die Schätzung des Reibungskoeffizienten wurde für einen Holzboden als Unterlage getroffen, da der Prototyp hauptsächlich auf selbigem getestet wurde.

Schätzung Reifen auf Holzboden: $\mu_{reifen} \cong 0.75$

$$F_{amax} = 0.788kg * 9.81 \frac{m}{s^2} * 0.75 = 5.798N$$

$$M_{Radmax} = 5.798N * 35mm = 0.203Nm$$

Die Rechnung zeigt auf, dass ein Drehmoment von 0.203Nm pro Reifen nicht überschritten werden sollte, da dieser ansonsten die Haftung zum Boden verliert und durchdreht.

Sensorik

Die rechts abgebildete Leiterplatte ist mit dem Sensor MPU-6050, einen sehr verbreiteten Gyro- und Beschleunigungssensor, bestückt. Dieser misst in allen 3-Achsen die Beschleunigung, sowie die Rotationsgeschwindigkeit. Über die digitale I2C Kommunikationsschnittstelle kann der Sensor konfiguriert und ausgelesen werden.

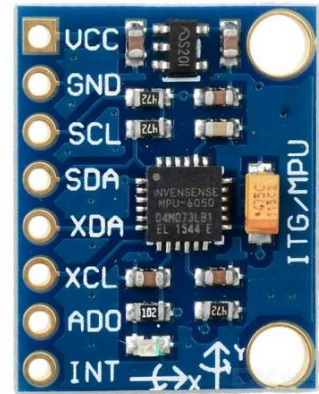


Abbildung 1: Sensorboard MPU-6050

Rotation

Der Gyrosensor misst die Rotation um eine Achse in Grad/s. Je nach Konfiguration kann der Sensor bis zu 1000°/s messen. Befindet sich der Sensor in Ruhelage, ist die Rotation aller Achsen gleich 0. Man könnte den Winkel bestimmen, indem man die Rotationsgeschwindigkeit pro Zeiteinheit aufsummiert, jedoch driftet die Berechnung des Winkels ab, da Messfehler addiert werden und somit das Ergebnis mehr und mehr verfälscht wird.

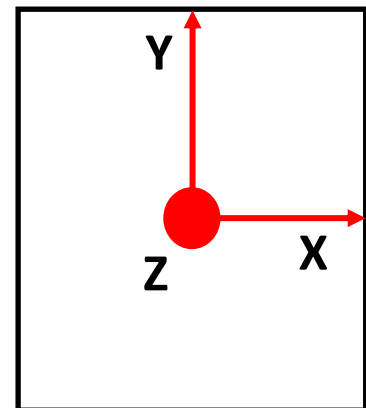


Abbildung 2: Achsenansicht von oben

Beschleunigung

Die Beschleunigung wird in m/s^2 gemessen. Je nach Konfiguration kann der Sensor bis zu $\pm 16G$ messen, dass entspricht der 16-fachen Erdbeschleunigung also ca. $160m/s^2$.

Liegt der Sensor still, ist die einzige Kraft, welche auf ihn wirkt, die Erdbeschleunigung. Würde der Sensor flach auf dem Tisch liegen wie in Abbildung 1, dann misst man auf der Z-Achse eine Beschleunigungskraft von $9.81m/s^2$. Auf der X- und Y-Achse liegt keine Kraft an. Beginnt man nun den Sensor zu kippen, teilt sich die Kraft der Erdbeschleunigung auf die Achsen auf. Mithilfe von Vektorgeometrie lässt sich dann der Winkel bestimmen.

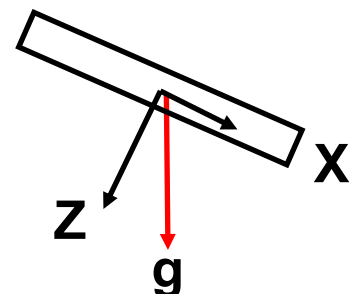


Abbildung 3: Kraftvektoren beim Kippen

$$\alpha = \arctan\left(\frac{Z}{X}\right)$$

Um den Roboter auszubalancieren versucht die Regelung den Winkel Alpha immer bei 90° zu halten.

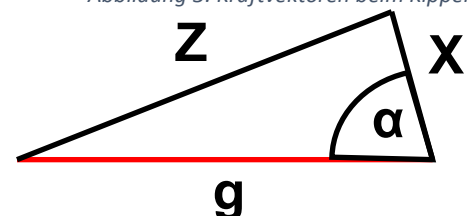


Abbildung 4: Kräftedreieck für Winkelberechnung

Konfiguration

Der Sensor wird so konfiguriert, dass er mit einer Frequenz von 100Hz die Messwerte der 6-Achsen aufzeichnet. Sobald diese bereit zum Auslesen sind, wird dies über den Pin INT (Interrupt) signalisiert. Die Messwerte werden dann sofort vom Mikrocontroller ausgelesen und verarbeitet.

Datenverarbeitung

Der Sensor besitzt einen internen DMP. Richtig konfiguriert filtert dieser die Messdaten mit einem Tiefpassfilter, damit Störungen bereits intern im Sensor herausgefiltert werden können.

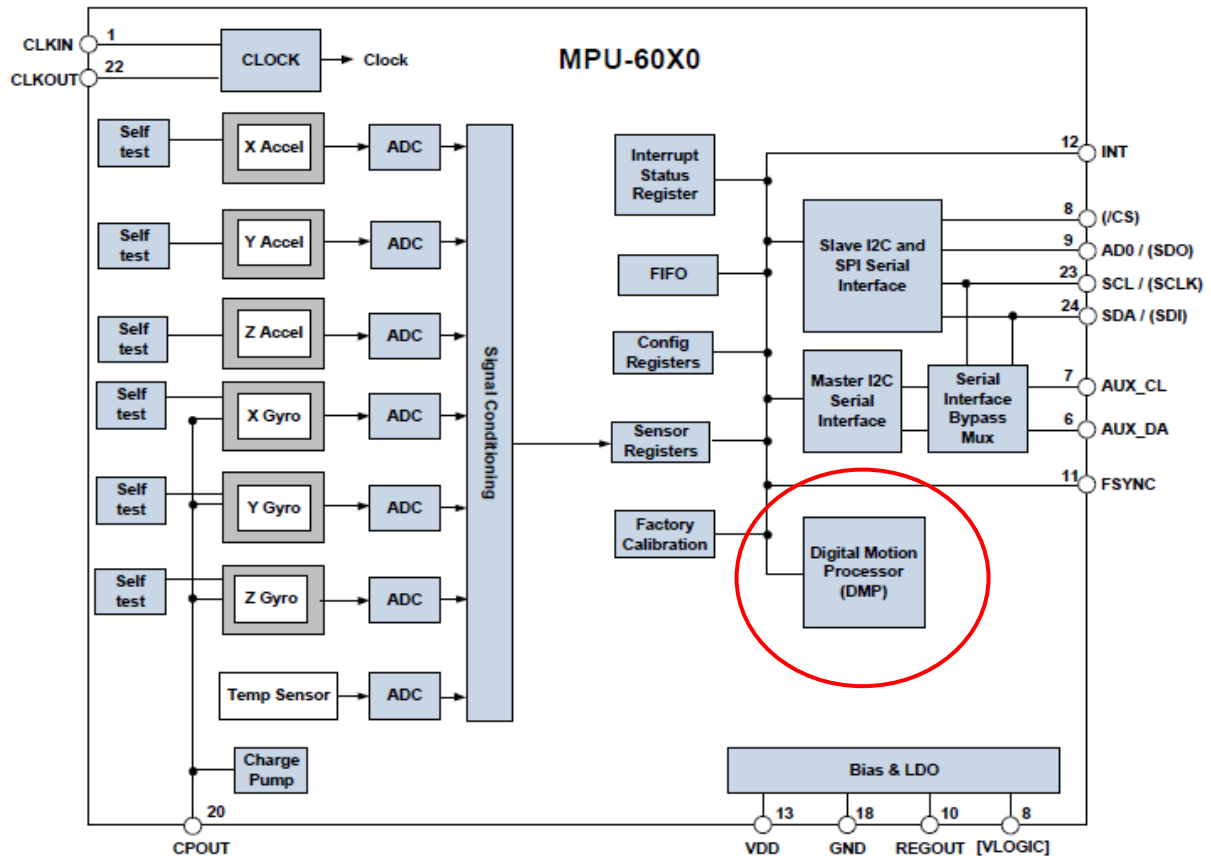


Abbildung 5: Blockschema MPU-6050 (InvenSense, 2019)

Auch per Software werden die Daten nochmals optimiert. Das Daten Verarbeiter Modul der Software glättet die Messwerte mithilfe eines gleitenden Mittelwerts.

Durch das Tiefpassfilter und den gleitenden Mittelwert werden Störungen, Messfehler und Rauschen des Sensor minimiert.

Einziger Nachteil: Durch das Verarbeiten der Daten wird die Zeitverzögerung der tatsächlichen Messung bis zu deren Verarbeitung durch Regler grösser. Dies kann das Verhalten des Reglers negativ beeinflussen.

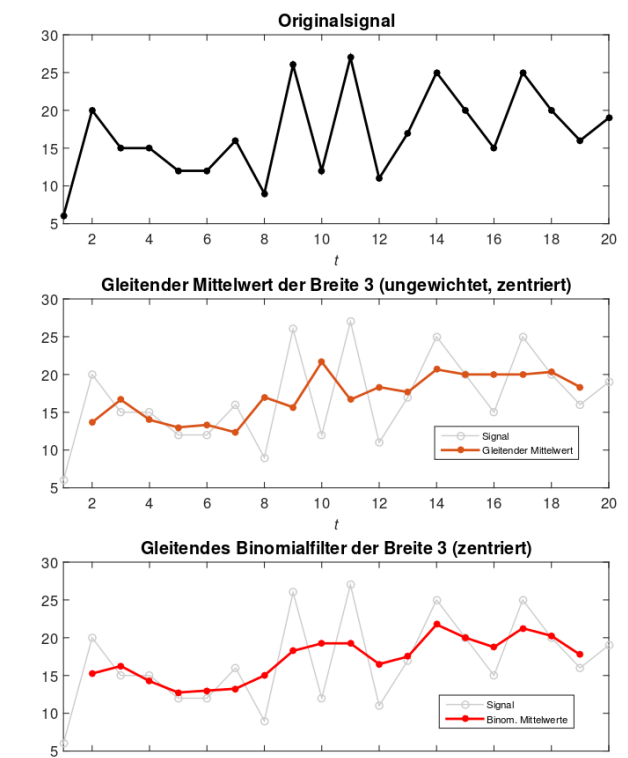


Abbildung 6: Gleitender Mittelwert (Wikipedia, 2019)

Sensor Auswertung

Um den Sensor zu testen wurde eine erste Testsoftware für den Mikrocontroller geschrieben, welche die gemessenen Werte über die UART-Schnittstelle des Mikrocontrollerboards an einen angeschlossenen PC sendet. Zusätzlich wurde eine Python Software entwickelt, welche die gesendeten Daten verarbeitet und diese in eine Excel Tabelle schreibt. Dadurch konnte das Verhalten des Sensors analysiert werden.

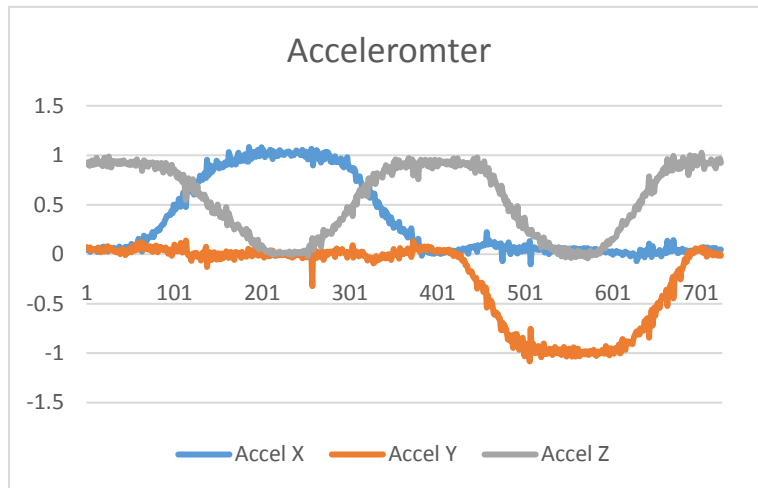


Abbildung 7: Messung des Accelerometers

Kippen von Hand

Der Sensor wurde von Hand um alle drei Achsen gedreht. Die Messwerte sind stark verrauscht, da einerseits die Handbewegungen nicht exakt sind, andererseits das digitale Tiefpassfilter des Sensors noch nicht aktiviert ist.

Die Y-Achse entspricht der Kraft in G und die X-Achse der Nummer des Messwerts.

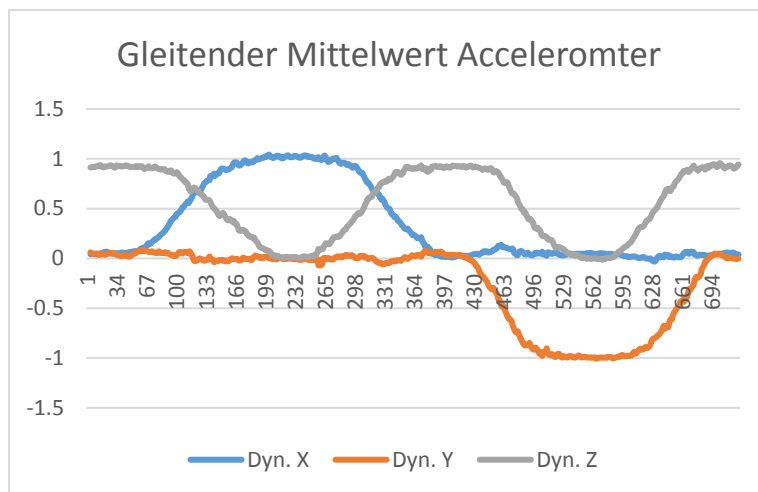


Abbildung 8: Messung mit gleitendem Mittelwert

Gleitender Mittelwert

Im Excel wurde der gleitende Mittelwert aus den letzten 5 Werten gebildet. Das Rauschen wurde gedämpft und Störungen bzw. Spitzen wurde geglättet.

Um zu prüfen ob der Sensor den Anforderungen entspricht und auch um ein Verständnis für das Verhalten des Sensors zu entwickeln wurden zwei weitere Tests durchgeführt.

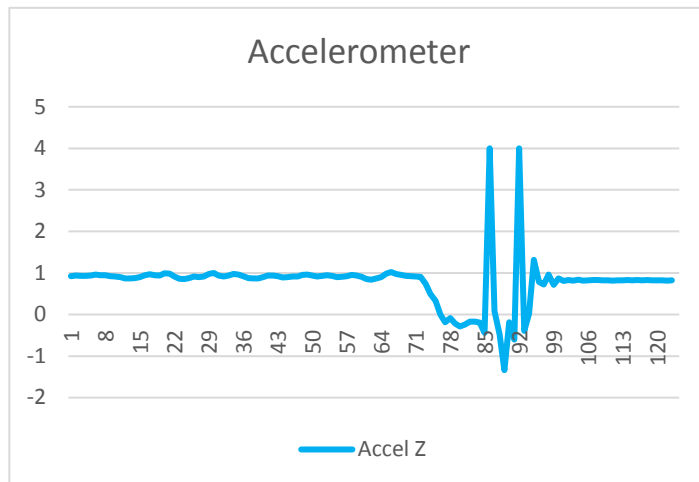


Abbildung 9: Messwerte der Z-Achse im freien Fall

Freier Fall

Der Sensor wurde aus 300mm fallen gelassen, um die Kräfte während dem Fall zu messen. Die Z-Achse (graue Linie) misst zu Beginn 1G. Sobald der Sensor fallengelassen wird, sinkt die Kraft auf 0G. Beim Aufprall wirken enorme Kräfte auf den Sensor, weshalb alle 3-Achsen starke Ausreisser aufweisen.

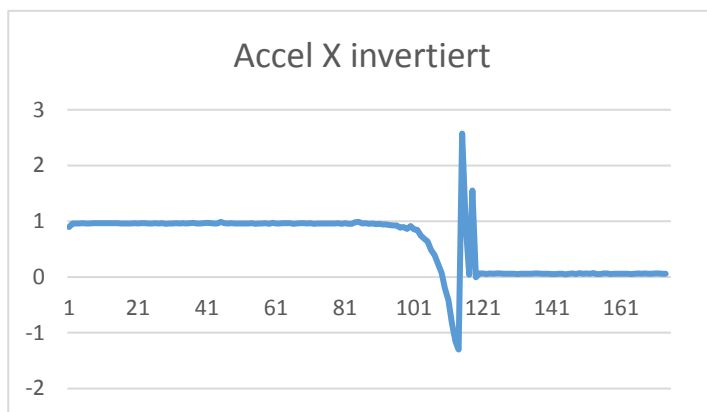


Abbildung 10: Messwerte X-Achse Accelerometer

Umfallen

Um zu testen, ob die Kräfte beim Umfallen im Gegensatz zum freien Fall messbar sind, wurde der Sensor auf ein 300mm langes Lineal montiert. Dieses wurde senkrecht aufgestellt, beim Umfallen wurden dann die Kräfte aufgezeichnet.

Beim Aufbau zeigte die X-Achse nach oben, weshalb diese zu Beginn die Erdanziehungskraft von einem G misst. Ab dem 90. Messwert beginnt die Kraft zu sinken, der Sensor beginnt zu kippen.

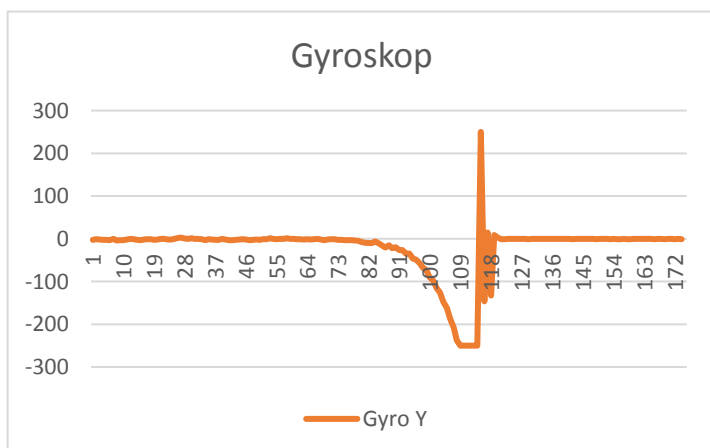


Abbildung 11: Messwerte der Y-Achse Gyroskop

Auch beim Gyroskop sieht man die steigende Rotationsgeschwindigkeit. Etwa beim 110. Messwert ist der Sensor aufgeprallt, die Messwerte danach sind Störungen, bzw. Messungen der enormen Aufprallkräfte.

Interessant zu beobachten ist, dass die Kurve beim Beschleunigungssensor nicht bei 0G stoppt, da dies ja einem Winkel α von 0° entsprechen würde. Dies ist auch auf die Kräfte beim Aufprall zurückzuführen. Nachdem der Sensor ruhig auf dem Boden liegt, sieht man auch die erwartete Kraft von 0G.

Aktoren

Ansteuerung Motor

Ein DC-Motor besitzt zwei Anschlüsse. Ändert die Richtung des Stromes, wechselt auch die Drehrichtung des Motors. Um dieses Wechseln des Stromes elektronisch umzusetzen wurde eine H-Brücke, auch Vierquadrantensteller genannt, verwendet.

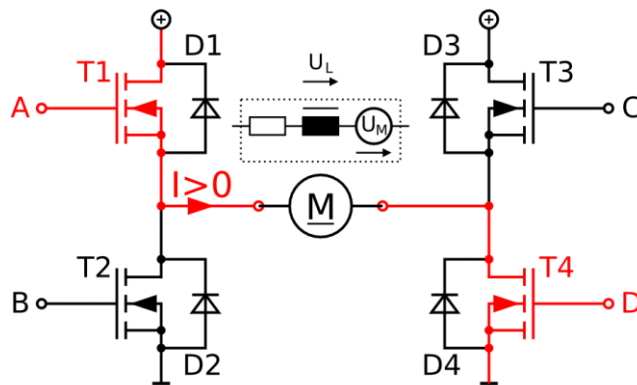


Abbildung 12: (Wikipedia, 2019)

Prinzip: Die Schaltung besitzt eine Motorspannung U_M und einen GND. Schaltet der Transistor T1 und T4 während T2 und T3 sperren, dann liegt vom linken zum rechten Anschluss des Motors gesehen eine positive Spannung an.

Sperrt T1 und T4 und leiten dafür T2 und T3, dann liegt eine negative Spannung vom linken zum rechten Anschluss an. Die Drehrichtung ändert.

Da ich die Anschlüsse A-D der H-Brücke mit CMOS-Logik (0 – 3.3V, gegeben durch Mikrocontroller) ansteuern möchte, die Motorspannung U_M jedoch bis zu 16V beträgt, müssten für den Anschluss A und C noch zusätzliche Komponenten für die Spannungstransformation verwendet werden. Ausserdem bräuchten die Bauteile einzeln

verbaut mehr Platz. Deswegen wird der IC **L293D** verwendet.

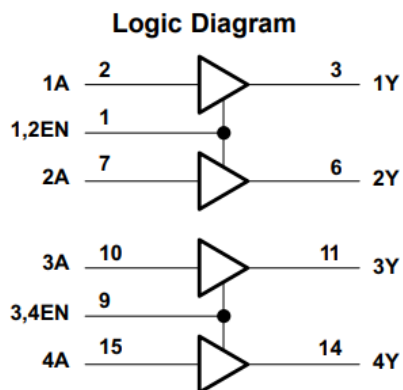


Abbildung 13: (Texas Instruments, 2019)

Der Baustein besitzt 4 High-Voltage Treiber. Über den Eingang A kann mit 0V oder 3.3V der Ausgang Y auf 0V oder auf den Level der Speisespannung geschaltet werden. Die Speisespannung kann bis zu 36V betragen. So kann an 1Y und 2Y der erste Motor und über 3Y und 4Y der zweite Motor angeschlossen werden.

Die Ausgänge können bis zu 1.2A liefern und auch „schlucken“. Mit den beiden Signalen 1,2EN und 3,4EN können die Treiber aktiviert werden.

PWM

Die Software soll die Leistungsabgabe des Motors steuern können, ansonsten ist eine Regelung nicht möglich. Um diese Steuerung zu erreichen wird die H-Brücke mit einem **PWM** Signal angesteuert. PWM steht für **P**ulse **W**ide **M**odulation.

Die Spannung über dem Motor soll reguliert werden. Da die Hardware jedoch nur zwei Zustände einstellen kann, (siehe H-Brücke) wird mit ständigem Ein- und Ausschalten die Spannung über dem Motor reguliert.

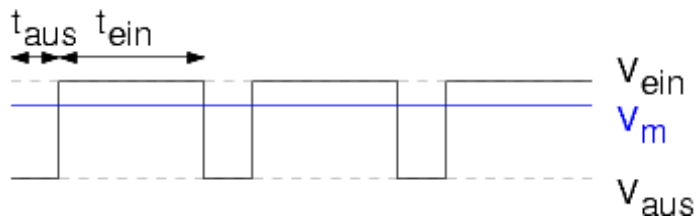


Abbildung 14: (Mikrocontroller.net, 2019)

$$T = t_{\text{ein}} + t_{\text{aus}}$$

$$f = \frac{1}{T}$$

$$\text{Tastgrad} = \frac{t_{\text{ein}}}{T}$$

Motor Stromaufnahme

Problem

Die ersten Tests mit dem L293D mit PWM Ansteuerung verliefen gut. Die Motoren liessen sich in beide Richtungen drehen. Der L293D wurde aber relativ heiss. Doch als ich den Roboter das erste Mal zusammengebaut hatte und diverse Probleme nach einer zeitintensiven Fehlersuche endlich beheben konnte, wollte ich die komplette Software mit einem einfachen Regelungsalgorithmus testen.

Der L293D Baustein wurde noch heisser als sonst und ein Motor arbeitete nicht richtig. Ich vermutete der Baustein sei beschädigt, da ich ihn anfangs falsch angeschlossen hatte. Als ich den Baustein austauschte in der Hoffnung, die Ansteuerung der Motoren würde funktionieren, ging dieser direkt bei Einschalten der Motoren kaputt, der Mikrocontroller funktionierte auch nicht mehr.

Theorie

Im Projekt werden Gleichstrom- Bürstenmotoren eingesetzt. Diese erzeugen durch die Rotation der Achse eine Gegeninduktion in der Spule, welche den Strom ausbremst. Je höher die Drehzahl des Motors, desto grösser ist die Gegenspannung und dementsprechend wird der Strom umso kleiner. Dies ist auch der Grund für die Tatsache, dass das Drehmoment bei Elektromotoren bei steigenden Drehzahlen abnimmt.

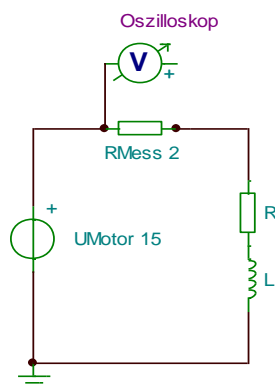


Abbildung 15: Messaufbau

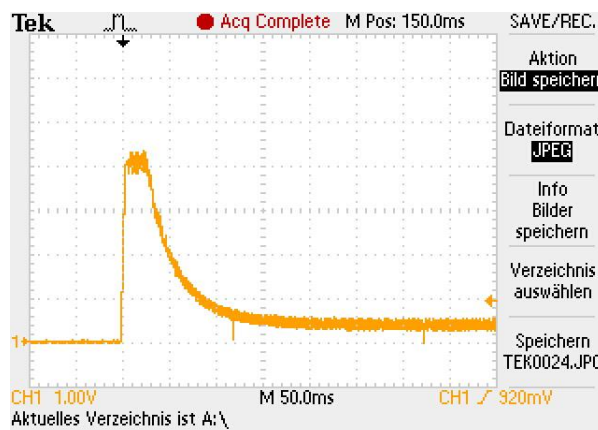


Abbildung 16: Messung 1, Start des Motors

Parameter:

U ₀ :	15V
R _{Mess}	2Ω
R _{Motor}	3Ω

Analyse

Um das Problem zu analysieren wurde ein Messversuch durchgeführt. Über einen 2 Ohm Messwiderstand wurde mithilfe eines Oszilloskops die Stromaufnahme des Motors aufgezeichnet. Dieser ist nicht ideal, da der Ohm'sche Widerstand der Motorspule 30hm beträgt. Dies führt bei grossen Strömen zur Verminderung der Spulenspannung über dem Motor, da über dem Messwiderstand eine zu grosse Spannung abfällt.

Trotzdem war der Fehler bereits nach der ersten Messung offensichtlich. Im Stillstand induziert die Motorwelle in der Spule noch keine Gegenspannung. Beim Einschalten wirkt somit ganz kurz die Impedanz der Spule, danach nur noch der Ohm'sche Widerstand von 3Ω. Bei 15V entsteht dann über einen Zeitbereich von 40ms ein Strom von über 4 Ampère. Danach sinkt der Strom innerhalb von 60ms auf 200mA. Dies entspricht ziemlich genau den Angaben des Datenblatts, welches bei keiner mechanischen Belastung einen Strom von 180mA angibt.

Der Motortreiber L293D kann pro Ausgang einen konstanten Strom von 1.2A Ampère liefern, was für den Normalbetrieb des Motors ausreichen würde. Doch der kurzzeitige Stromspitz beim Anfahren genügt, um den Motortreiber zu zerstören.

Um der hohen Stromanforderung gerecht zu werden, wurde ein neues IC getestet. Der L298 ist ähnlich aufgebaut wie der alte Treiber L293D, jedoch kann dieser einen Strom von 2A liefern und kurzzeitige Spitzen bis zu 3A Ampère. Dieser wird im Betrieb auch sehr warm, funktioniert jedoch.

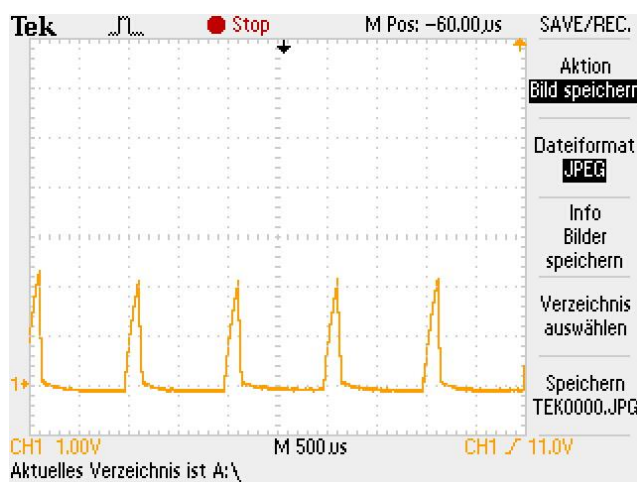


Abbildung 17: Messung des Stroms bei Anfahren mit PWM

Strommessung bei Anfahren mithilfe von PWM. Man sieht die PWM Frequenz von 1kHz sowie das Stromverhalten eines LC Gliedes, da die Kurve einer e-Funktion ähnelt. Da die Spannung nur kurzzeitig eingeschaltet ist, steigt der Strom nicht über 1.5A, weshalb der L293D intakt bleibt.

Gefährlich wird es, wenn der Motor mit einem hohen PWM Wert anfährt.

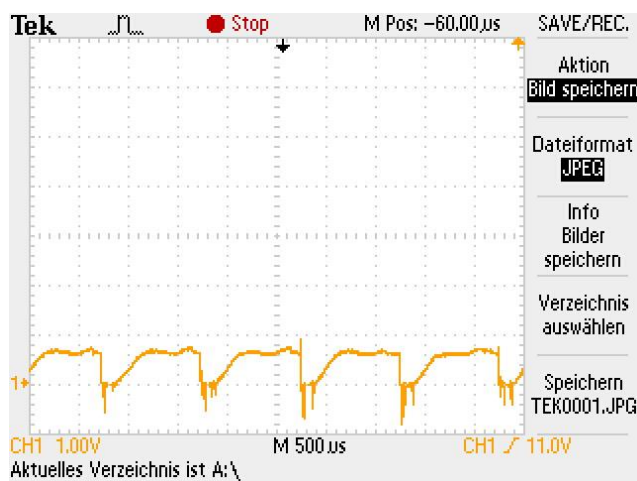


Abbildung 18: Messung des Stroms bei PWM 60%

Messung des Stroms bei laufendem Motor mit einem PWM von 60%. Man sieht gut, dass der Motor im Betrieb weniger Leistung benötigt als beim Anfahren. In der Messung sieht man auch gut, wie der PWM arbeitet. Die negativen Stromspitzen werden beim Abschalten der Spule durch deren Selbstinduktion erzeugt.

L298 als neuer Motortreiber

Die Messungen zeigten, dass sich der L298 aufgrund der grösseren Ausgangsleistung besser für die Ansteuerung der Motoren eignet. Dieser IC hat ein grösseres Gehäuse mit besserer Kühlfläche und 2 zusätzliche Anschlüsse. Im Gegensatz zum L293D sind die Emitteranschlüsse der beiden H-Brücken nicht direkt mit GND verbunden, sondern über zwei Pins hinausgeführt. Dies ermöglicht das Einsetzen zweier Shunt Widerstände um die Stromaufnahme der beiden Motoren zu messen.

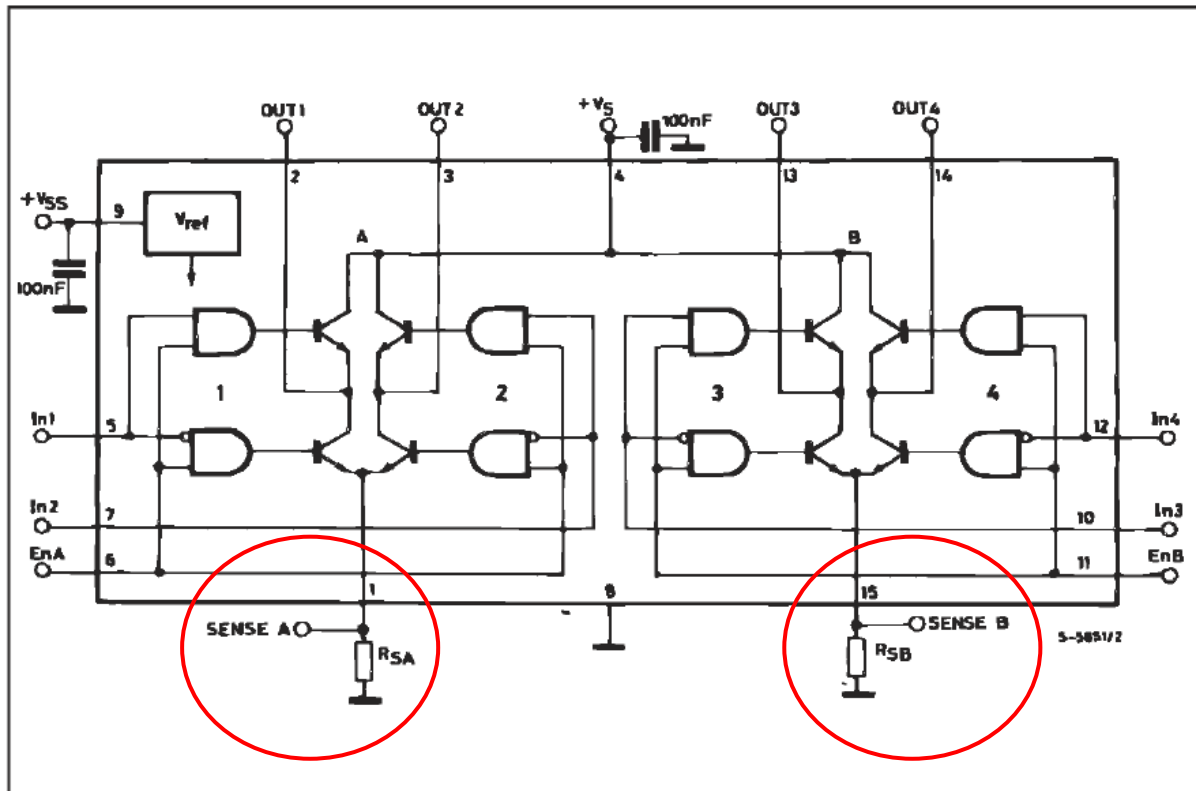


Abbildung 19: Block Diagramm aus L298 Datenblatt (ST Microelectronics, 2019). Die roten Kreise markieren die beiden Anschlüsse für das Messen des Stromes

Layout

Von Beginn an war klar, dass eine Leiterplatte entwickelt werden muss, damit die vielen Verbindungen nicht mit Kabel umgesetzt werden müssen. Die erste Idee war das Entwickeln zweier Leiterplatten. Eine Leiterplatte ist mit dem Motortreiber bestückt und wird vom Akku gespeisen. Auf der zweiten Leiterplatte befinden sich die Sensorik, der Mikrocontroller sowie der Taster für den Benutzer. Die beiden Leiterplatten sollten übereinander verbaut werden, der Motor Driver zwischen den Akkus und das Controller Board oberhalb davon.

Abmessung: 50x50mm

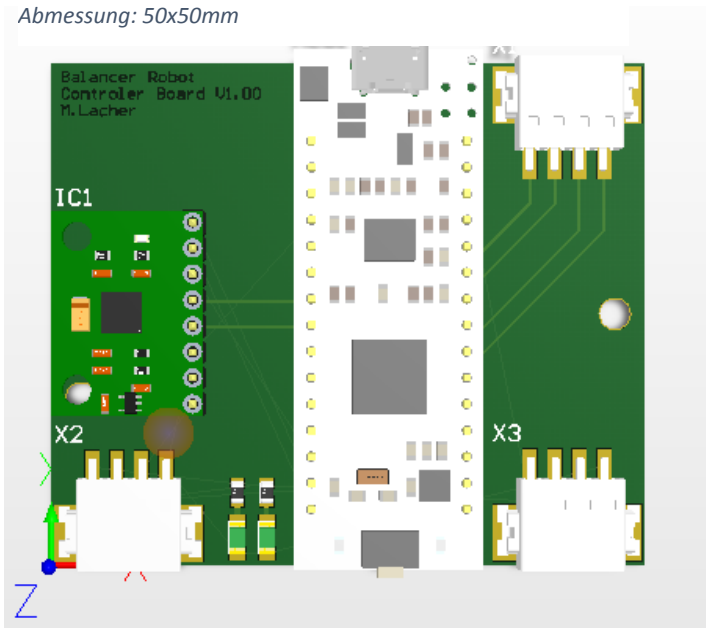


Abbildung 21: Leiterplatte „Controller Board V1.00“

Abmessung: 50x50mm

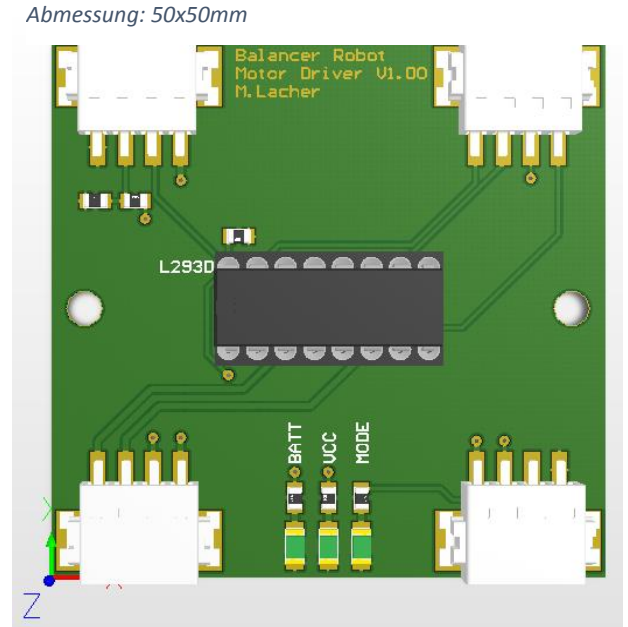


Abbildung 20: Leiterplatte „Motor Driver V1.00“

Da die Akkuhalter sehr lang sind würde das Fahrzeug viel zu breit werden. Deshalb wurde eine zweite Variante entwickelt, in der alle Komponenten auf einer Leiterplatte bestückt sind. Dies vereinfacht einerseits die Verdrahtung der einzelnen Fahrzeugkomponenten, andererseits konnte die Breite des Fahrzeugs um 50mm verringert werden.

Abmessung: 100x60mm

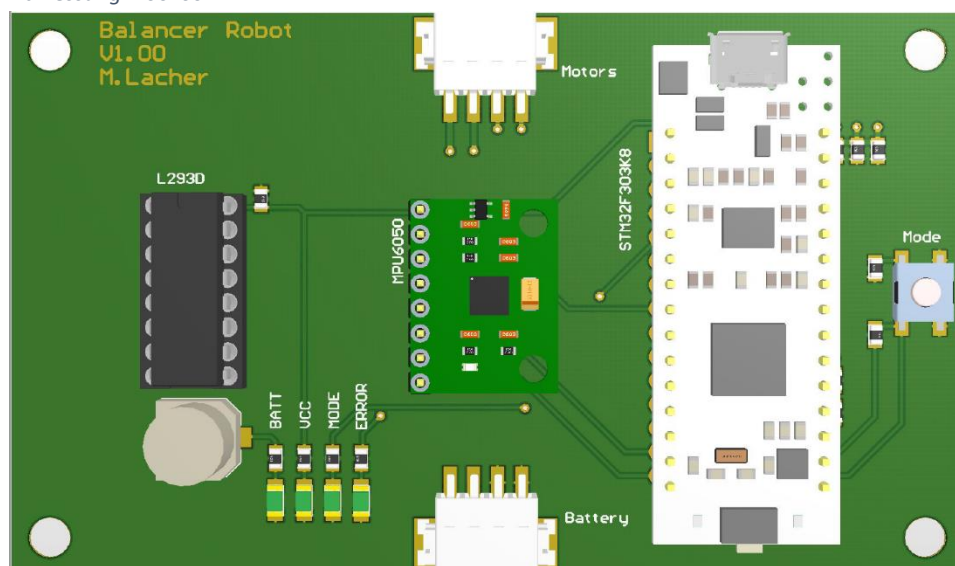


Abbildung 22: Leiterplatte "Balancer Robot V1.00"

Nachdem dem die Probleme mit dem Motortreiber L293D auftraten und ein neuer Treiber evaluiert wurde, musste eine neue Leiterplatte entwickelt werden. Rot eingekreist sieht man den neuen Motortreiber sowie die beiden Shunt-Widerstände für die Strommessung.

Die Bohrungen haben einen Durchmesser von 4.3mm und sind somit für M4 Schrauben ausgelegt. Das Zentrum der Bohrungen ist 4mm vom Rand entfernt.

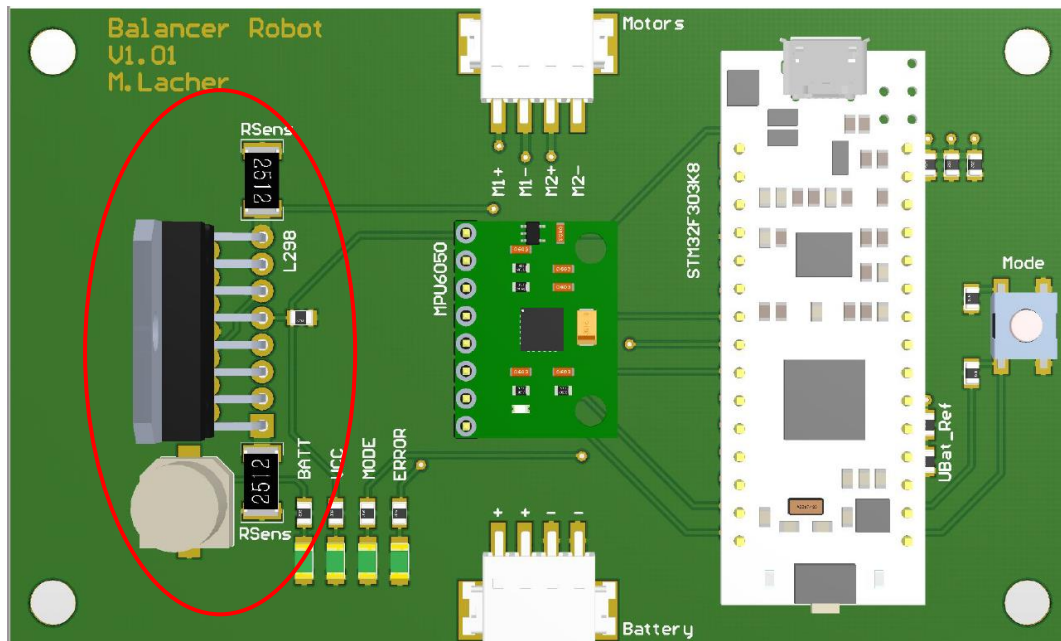


Abbildung 24: Leiterplatte "Balancer Robot V1.01"

Ganz rechts auf dem Layout befindet sich ein Taster. Mithilfe dieses Tasters kann der Benutzer die Software steuern. Im fertigen Produkt wird dann der Mode des Balancer Robots geändert.

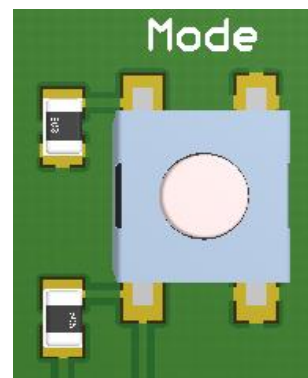


Abbildung 23: Mode-Taster

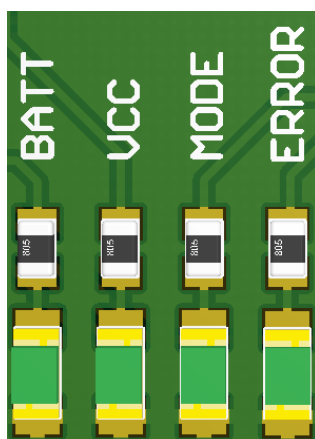


Abbildung 25: LED-Anzeige

Die 4 LED signalisieren dem Benutzer folgende Informationen:

Batt: Batteriespeisung vorhanden

VCC: 3.3V Speisung vorhanden

Mode: Signalisiert den aktuellen Modus mithilfe eines Blink-Codes

Error: Signalisiert verschiedene Fehler mithilfe eines Blink-Codes

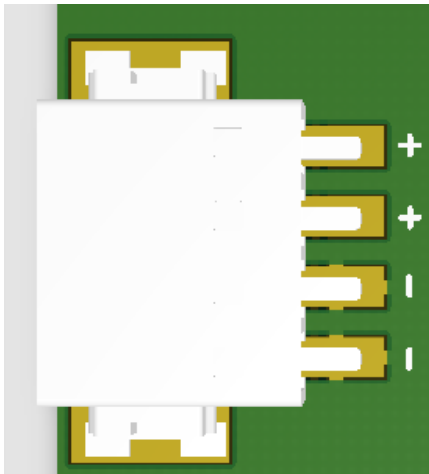


Abbildung 26: Akku-Stecker

Über den Akku-Stecker wird die Schaltung eingespeist. Jeweils zwei Anschlüsse für das positive Spannungspotential und zwei Anschlüsse für das negative Spannungspotential sind miteinander verbunden. Dies ermöglicht einerseits das Weiterführen der Akkuspannung, andererseits können bei grosser elektrischer Belastung zwei Drähte parallel zur Einspeisung verwendet werden.

An den 4 Anschlüssen des Motor-Steckers werden die beiden Motoren angeschlossen. Pro Motor ist ein Anschluss mit Plus und ein Anschluss mit Minus gekennzeichnet. Auf der mechanischen Grundplatte sind diese Bezeichnungen wiederzufinden.

Es ist wichtig, dass die Motoren korrekt angeschlossen werden, ansonsten funktioniert das Ausbalancieren nicht.

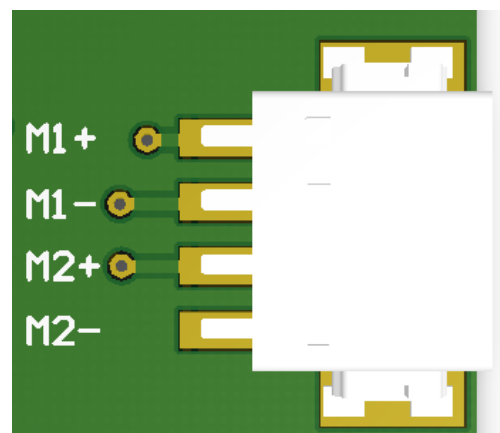


Abbildung 27: Motor-Stecker

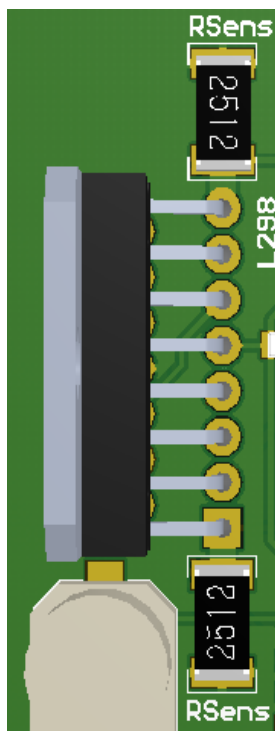


Abbildung 28: L298 Motor Treiber

Der neue Motortreiber L298 ist senkrecht aufgerichtet, damit auf der Rückseite ein Kühlkörper montiert werden kann.

Die beiden Shunt-Widerstände sind so nahe wie möglich am dazugehörigen Pin und danach direkt auf GND angeschlossen, damit der Widerstand der Leiterbahnen möglichst gering bleibt.

Ein SMD-Elektrolytkondensator stützt die Akkuspannung, direkt beim Motor Treiber, damit diese bei schnell ändernder Belastung nicht zusammenbricht.

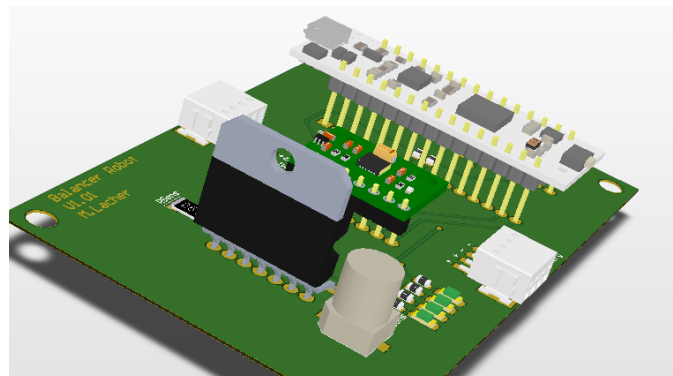


Abbildung 29: Seitenansicht, Rückseite des L298

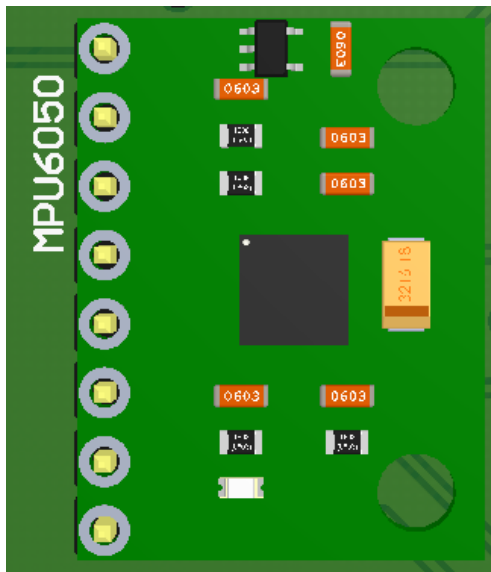


Abbildung 30: Sensorboard MPU-6050

Das Sensorboard kann in dieser Ausführung online bestellt werden. Es ist bestückt mit dem Beschleunigungs- und Gyrosensor MPU6050 sowie einem Längsregler für die Spannungsversorgung plus diverser zusätzlicher Hardware.

Dieses Board kann so direkt auf den Print gesteckt werden. Dies erleichtert ein allfälliges Ersetzen des Sensors bei einem Defekt. Zusätzlich könnte später ein eigenes Sensorboard entwickelt werden mit zusätzlichen Features, welches dann das aktuelle ersetzt.

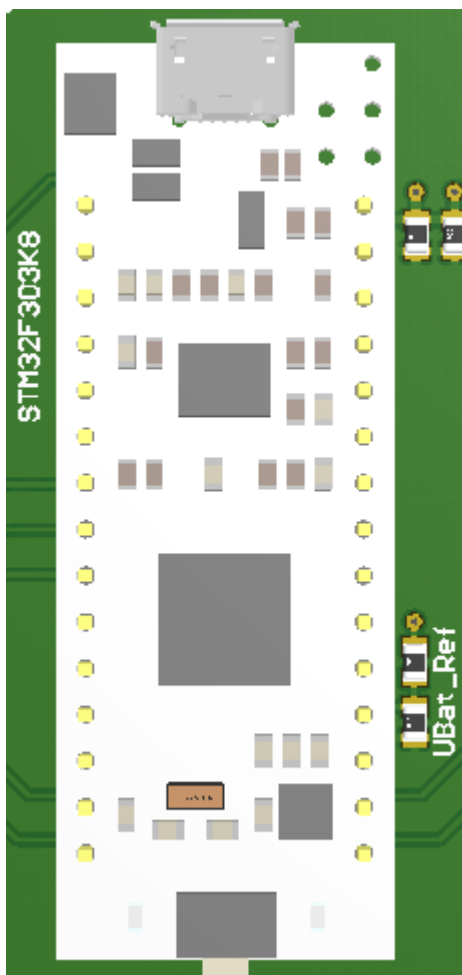


Abbildung 31: Mikrocontrollerboard STM32F303K8

Das Mikrocontrollerboard Nucleo STM32F303K8 ist ein Entwicklungsboard von STMicroelectronics. Es stellt alle nötigen Peripheriebauteile für das Verwenden eines Mikrocontrollers zur Verfügung.

Das Mikrocontrollerboard kann direkt auf den Print gesteckt werden. Dies erleichtert ein allfälliges Ersetzen des Mikrocontrollers bei einem Defekt.

Akkumulatoren

Für die Energieversorgung der Elektronik werden handelsübliche Lithium-Ionen Akkuzellen verwendet.

Lithium-Ionen Akkuzellen, auch Li-Ion Akkus genannt, zeichnen sich durch eine sehr hohe spezifische Energiedichte aus, etwa 150 Wh/kg. Auch die Energiedichte von 400 Wh/l macht die Li-Ion Akkus interessant für mobile Anwendungsbereiche. (Wikipedia, 2019)

Nachteil: Die Akkuzellen sind anfällig auf Tiefenentladung sowie Überladung. Es ist also Elektronik notwendig sowohl für das Laden der Akkus, als auch für die Überwachung während des Betriebs.

Die Li-Ion Akkus werden von der Firma NKON aus den Niederlanden geliefert. Der Hersteller garantiert einen maximalen konstanten Strom von 10A, was für die Anwendung völlig ausreichend ist. Obwohl Li-Ion Akkus theoretisch von 4.3V bis runter zu 2.5V schädigungsfrei betrieben werden könnten, empfiehlt der Hersteller eine Spannung von 3V nicht zu unterschreiten.

Ein Spannungsteiler auf der Leiterplatte teilt die Akkuspannung mit dem Verhältnis 1:4.3, sodass der Mikrocontroller per AD-Wandler die Spannung überwachen kann. Sollte die Spannung unter 3.2V pro Zelle sinken, deaktiviert die Software die Motoren, um den Energieverbrauch drastisch zu senken. Eine niedrige Akkuspannung wird mit einem 1Hz Blinken der Error LED signalisiert.

Ingesamt werden 4 Akkuzellen in Serie geschaltet, das ergibt eine Versorgungsspannung zwischen 12V – 16.8V. Mit einer Mindestkapazität von 2600mAh ergibt folgende ungefähre Energie:

$$W = U * Q = 14.4V * 2.6Ah = 37.44Wh$$

Bei einem geschätzten durchschnittlichen Verbrauch von 0.8A ergibt das eine Betriebsdauer von über 3h.

$$t = \frac{W}{P} = \frac{37.44Wh}{14.4V * 0.8A} = 3.25h$$



Abbildung 32: Li-Ion Akkuzelle Typ 18650

Marke	LG
EAN/GTIN	9009316342928
Modell:	LGEBM261865
Batteriegröße	18650
Batterie chemie	Li-ion
Batterie	Wiederaufladbar
Spannung	3.6 / 3.7V
Mindest. kapazität in mAh	2.600,00
Batterieversion	Flat top
Entladestrom	10,00
Lithium - Schutzschaltung	Ohne
Höhe in mm	65,00
Durchmesser in mm	18,00

Abbildung 33: Angaben der bestellten Akkuzellen (NKON, 2019)

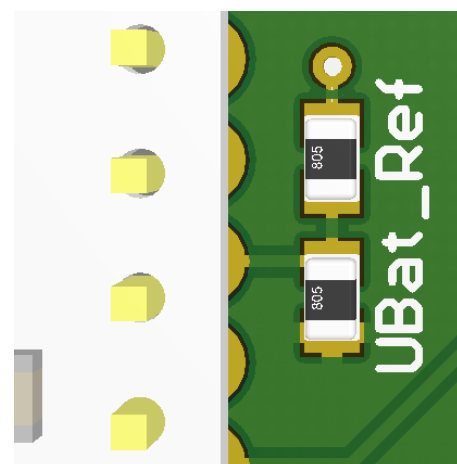


Abbildung 34: Spannungsteiler auf der Leiterplatte

Mechanik

Die Mechanik spielt eine entscheidende Rolle im Projekt. Der Roboter sollte möglichst leicht sein, damit das benötigte Drehmoment, um den Roboter wieder aufzurichten, möglichst klein bleibt. Ausserdem muss auf die Gewichtsverteilung geachtet werden, um den Schwerpunkt tief zu halten.

Für den ersten Prototyp habe ich eine Grundplatte gezeichnet, welche von einem Mechaniker angefertigt wurde. Auf der Unterseite wurden die Motoren befestigt. Die Oberseite ist bestückt mit den Akkuhaltern und den Distanzbolzen, um weiter oben die Leiterplatte zu montieren.

Motor

MFA 950D501	
Lieferant	Conrad.ch
Preis	25.95 CHF
Betriebsspannung	9-15V
Gewicht	156g
Wellen-Länge	15mm
Wellen-Durchmesser	6mm
Untersetzung	50:1
Maximale Effizienz	61.85%
Leistungsabgabe maximale Effizienz (12V)	6.21W
Stromaufnahme maximale Effizienz (12V)	0.837A
Anfahrdrehmoment (12V)	2.025Nm
Drehmoment bei maximaler Effizienz (12V)	0.346Nm
Maximale Drehzahl (12V)	252U/min
Drehzal bei maximaler Effizienz (12V)	185.6U/min



Abbildung 35: MFA 950D501 (Conrad, 2019)

Zu Beginn des Projekts wurden zwei verschiedene Motoren ohne Übersetzung auf Conrad.ch bestellt. Doch nach weiteren mechanischen Berechnungen wurde klar, dass das Drehmoment der bestellten Motoren viel klein ist (ca. 0.1Nm). Deswegen wurde ein Motor mit mechanischer Übersetzung bestellt. Dieser besitzt ein Getriebe mit einer Untersetzung von 50:1. Dadurch wird das Drehmoment des Elektromotors mit dem Faktor 50 erhöht, respektive die Drehzahl um den Faktor 50 reduziert. Die mechanischen Verluste des Getriebes sind dabei vernachlässigbar. Der Motor arbeitet linear von 6V bis 15V. Das heisst bei 15V Spannung beträgt das Anfahrdrehmoment sogar 2.531Nm.

Prototyp

Nachdem die Motoren bestellt waren, wurde zeitgleich zur Leiterplatte die Grundplatte aus Aluminium gezeichnet. Diese wurde dann von einem Mechaniker hergestellt. Nachdem die Leiterplatte bestückt wurde, konnte ein erster Prototyp zusammengebaut werden.

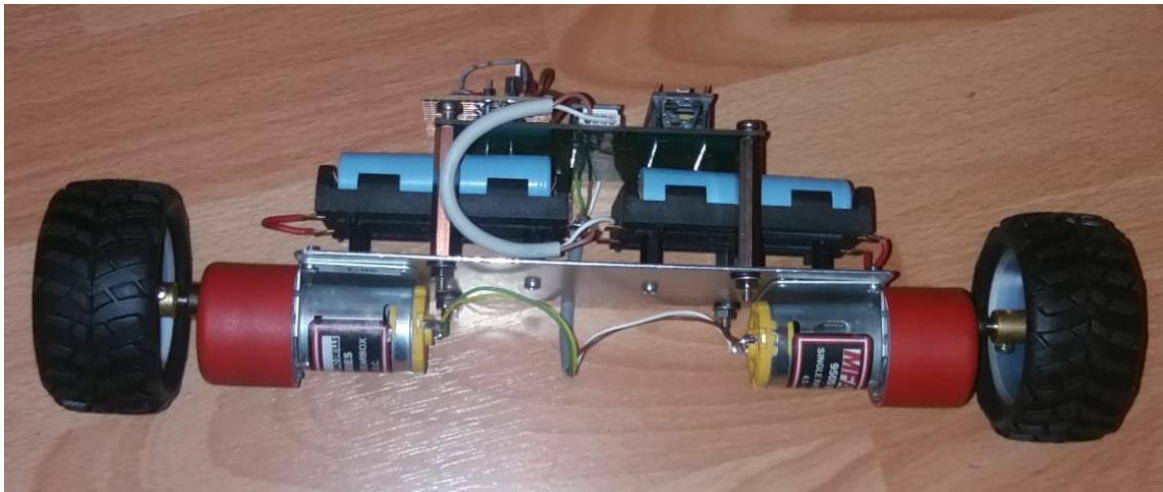


Abbildung 37: Erster Prototyp, Seitenansicht

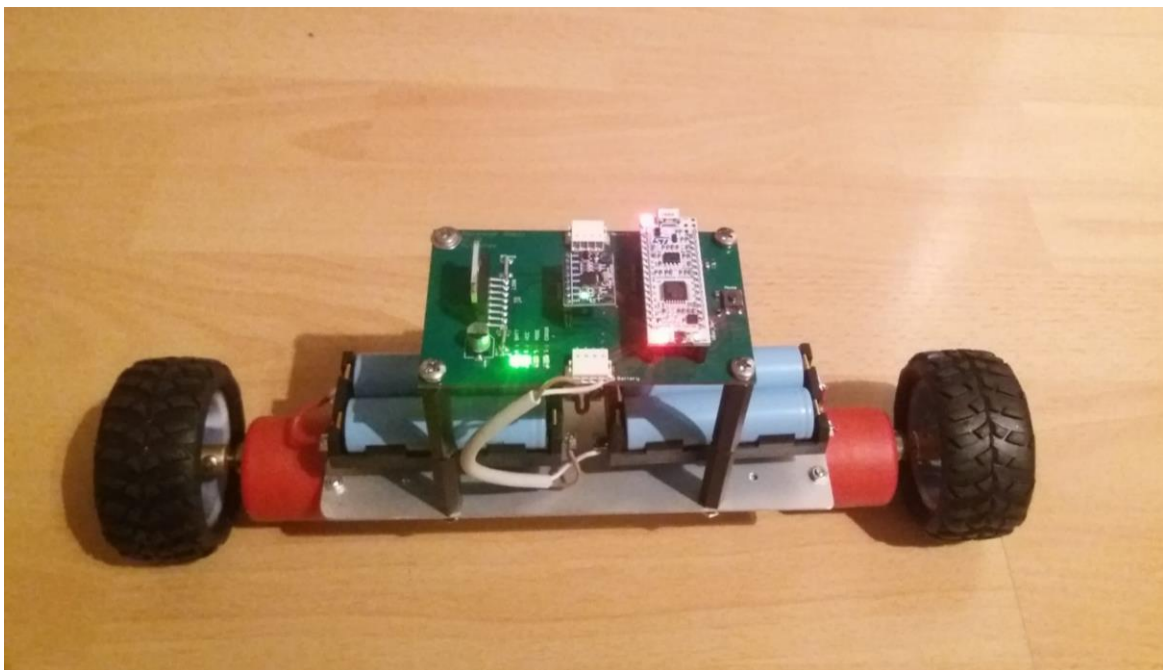


Abbildung 36: Erster Prototyp, Vogelperspektive

3D-Modell

Der Prototyp besteht aus folgenden Komponenten:

- 1) Leiterplatte (aktuelle Version 1.01)
- 2) Distanzbolzen, Kunststoff M4 (aktuelle Länge 8mm)
- 3) 2x Akkuhalter für 4 Akkuzellen Typ 18650
- 4) Grundplatte, Aluminium
- 5) 2x Reifen, Durchmesser 70mm + Adapter für die Motorachse
- 6) 2x Bürstenmotor MFA 950D501

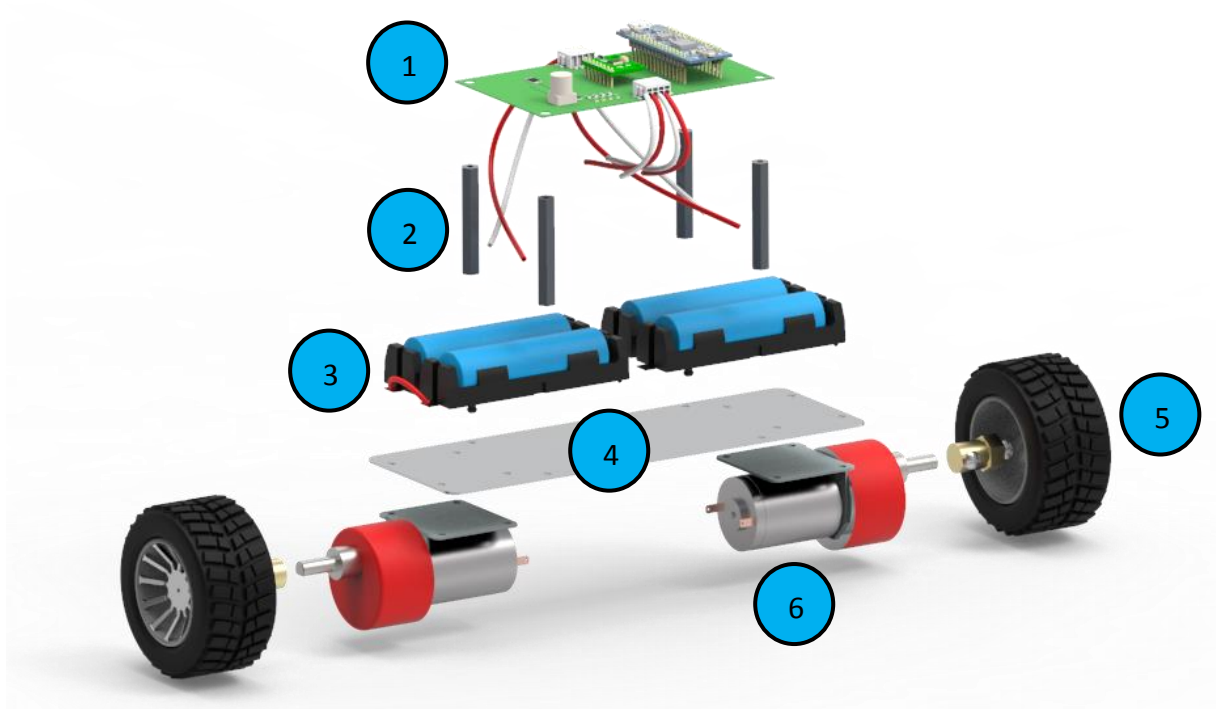
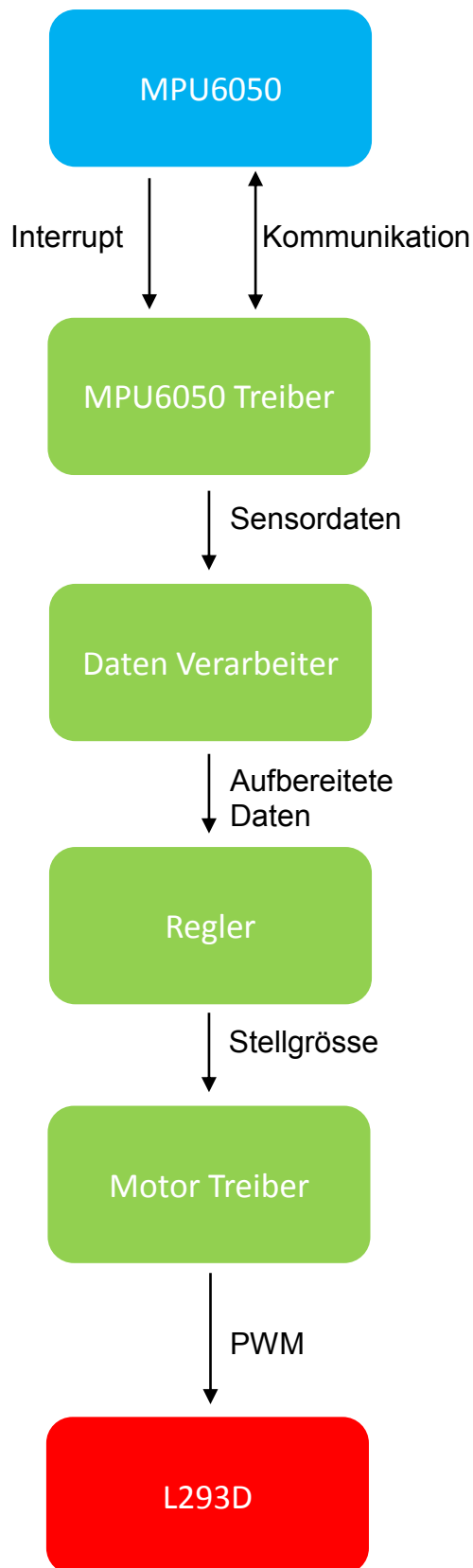


Abbildung 38: 3D-Modell Explosionszeichnung



Software



Architektur

Blau: Sensoren

Grün: Mikrocontroller (Software)

Rot: Aktoren

Der MPU6050-Treiber konfiguriert den Sensor, sodass dieser mit einer Frequenz von 100Hz die 3-Achsen des Gyrosensors, sowie die 3-Achsen des Accelerometers misst. Über einen Interrupt Pin wird dem Mikrocontroller signalisiert, dass die Daten bereitstehen. Der Mikrocontroller liest dann sofort die Daten aus.

Die Sensordaten werden dem Datenverarbeiter übergeben. Dieser berechnet den dynamischen Mittelwert der Daten, um diese zu Glätten und Störungen herauszufiltern.

Die aufbereiteten Daten werden dem Regler übergeben. Dieser errechnet mithilfe des Regelalgorithmus die Stellgrösse für die Motoren.

Die Stellgrösse wird dem Motortreiber übergeben. Aufgrund der Stellgrösse steuert der Motor Treiber die Motoren per PWM Signal an.

Regelungsalgorithmus

Allgemein

Die Regelungstechnik bildet einen grossen Teil des Projekts. Im Grundprinzip funktioniert jede Regelungsschaltung, ob digital oder analog, auf dem gleichem Grundprinzip. Die Differenz X_d zwischen der Führungsgrösse, also dem gewollten Sollwert und der Rückführgrösse, also die gemessene tatsächliche Grösse, wird einem Regler übergeben. Dieser errechnet die Stellgrösse für den Aktor, welcher in die Regelstrecke eingreift. Ein Sensor misst die Regelstrecke aus. Über die Messeinrichtung wird die Rückführgrösse erzeugt und rückgekoppelt.

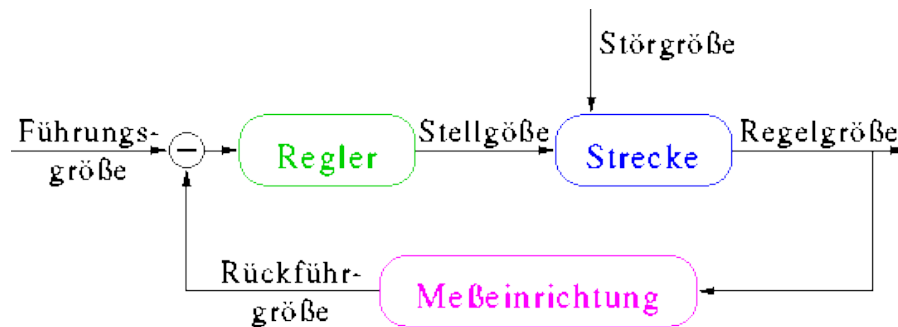


Abbildung 39: (Heuler, 2019)

Konkret auf den Balancer Robot angewendet lässt sich die Regelstrecke folgendermassen beschreiben: Abhängig von der errechneten Stellgrösse wird die Spannung über den Motoren reguliert. Durch äussere Störgrössen, wie z.B. die Erdanziehungskraft, kippt der Roboter in eine Richtung. Der Sensor detektiert die Veränderung des Winkels und somit die Abweichung vom Sollwert. Der Regler berechnet eine neue Stellgrösse um den falschen Winkel wieder zu korrigieren.

PID-Regler

PID-Regler sind weit verbreitet in der Regelungstechnik. Sie lassen sich in folgende drei Grundanteile einteilen:

Proportional-Anteil

Der P-Anteil gibt proportional zur Regeldifferenz x_d einen Ausgangswert y aus. P-Regler reagieren sofort auf Änderungen, können dafür die Regeldifferenz x_d nicht vollständig ausregeln. K_P definiert den P-Anteil der gesamten Regelung.

$$y = K_P * x_d$$

Integral-Anteil

Der I-Anteil integriert die Regeldifferenzen auf. Dadurch steigt bei einer konstanten Abweichung zwischen x_d und dem Sollwert der Ausgang so lange an, bis die Differenz ausgeglichen wurde. K_I definiert den I-Anteil der gesamten Regelung.

$$y = K_I * \int_0^t x_d \Delta t$$

Differenzial-Anteil

Der D-Anteil differenziert die Regeldifferenzen. Dadurch reagiert der D-Anteil auf schnelle Änderungen der Regeldifferenz x_d . Der D-Anteil alleine kann eine Regeldifferenz nicht ausregeln, wird jedoch verwendet, wenn schnelle Reaktionen auf Störsignale und Regeldifferenzabweichungen nötig sind. K_D definiert den D-Anteil der gesamten Regelung.

$$y = K_D * \frac{\Delta x_d}{\Delta t}$$

Kombiniert sieht die Formel für den PID-Regler folgendermassen aus:

$$y = K_P * x_d + K_I * \int_0^t x_d \Delta t + K_D * \frac{\Delta x_d}{\Delta t}$$

Die drei Regelanteile K_P , K_I und K_D müssen abhängig der Anforderungen evaluiert werden. Eine grundsätzliche Faustregel besagt, dass der P-Anteil etwa 90% der Regelung übernimmt, der I-Anteil die bleibende Regeldifferenz sowie konstante Abweichungen mit der Zeit korrigiert werden. Der D-Anteil sollte so ausgelegt werden, dass Störungen und schnelle Abweichungen vom Sollwert abgefangen werden.

Umsetzung

In den ersten Versuchen wurde ein simpler P-Regler implementiert. Es wurde nicht ein tatsächlicher Winkel verwendet, sondern direkt die beschleunigende Kraft auf der Y-Achse. Denn die Kraft auf der Y-Achse ist direkt proportional zum nötigen Drehmoment für das Aufrichten.

Erster Versuch

Der Balancer Robot zitterte unaufhörlich und konnte sich selber nicht stabilisieren. Nach diversen Spielereien und Tests an den Regelparametern und der Sensordatenverarbeitung wurde dann folgender Denkfehler klar: Die Regelung der Motoren funktionierte gut, wenn dieser von Hand gekippt wird, doch wenn das Fahrzeug sich selbst aufrichtet, entstehen beschleunigende Kräfte, die jener der Y-Achse gegengerichtet sind. Der Balancer Robot möchte also ständig den Winkel korrigieren, beeinflusst aber direkt den Messwert durch die Korrektur, was eine Richtungsänderung zur Folge hat.

Korrektur der Beschleunigungskräfte

Da kommt der Gyrosensor ins Spiel. Die Messwerte des Beschleunigungssensors müssen korrigiert werden, damit der Winkel trotz der wirkenden Kräfte beim Aufrichten korrigiert werden kann.

Durch die Rotationsgeschwindigkeit, gegeben durch den Gyrosensor, und dem Radius zur Motorachse kann die Bahngeschwindigkeit des Sensors bestimmt werden.

$$v = \omega * r$$

Aus der Geschwindigkeitsänderung pro Zeit kann dann die verursachte Beschleunigung errechnet werden.

$$a = \frac{v - v'}{\Delta t} = \frac{(\omega - \omega') * r}{\Delta t}$$

Subtrahiert man die durch die Aufrichtung entstehende Beschleunigung der Y-Achse, dann bleibt nur noch die Kraft der Erdbeschleunigung übrig.

Zweiter Versuch mit korrigiertem Messwert

Nachdem die Korrektur der Messwerte vorgenommen wurde, war eine klare Verbesserung zu sehen. Der Balancer Robot zitterte nicht mehr wie wild umher, sondern versuchte sich tatsächlich entgegen der Kipprichtung aufzurichten. Jedoch klappte das Ausbalancieren noch nicht, da der Roboter nach kurzer Zeit entweder durch ständiges Hin- und Herschwingen umfiel, oder aber bei Anpassungen des Proportionalfaktors zu schwach oder zu spät eingriff.

PID-Regler

Etwas Zeit war noch übrig, sodass noch ein PID-Regler programmiert werden konnte. Es wurden mehrere Versuche mit verschiedenen Werten der P-, I- und D-Anteile getestet. Doch das Ausbalancieren funktionierte auch dann nicht. Vielleicht wäre ein Ausbalancieren mit perfekt abgestimmten Regelanteilen möglich, jedoch fehlte die Zeit und auch das Know-how, um weitere Versuche gezielt durchführen zu können.

Schluss

Leider ist der Balancer Robot nach aktuellen Stand (21.11.2019) noch nicht fähig, sich selbst auszubalancieren. Die Elektronik läuft, die Mechanik ist auch fertig, sodass ein erster Prototyp gebaut werden konnte. Auch die Software ermöglicht das Konfigurieren des Sensors, die Ansteuerung der Motoren und auch die Überwachung der Akkuspannung. Mithilfe des Tasters auf der Leiterplatte können der Roboter zwischen dem Balancier Modus und dem Standby Modus wechseln.

Die Nachforschungen zeigten auf, dass mithilfe eines Beschleunigungs- und Gyrosensor der Neigungswinkel des Fahrzeugs bestimmt werden kann. Auch wurde nachgewiesen, wie wichtig das saubere Aufbereiten der Daten ist. Mir wurde folgendes bewusst: „Kann man ein System nicht korrekt messen, lässt es sich auch nicht Steuern, beziehungsweise Regeln.“

Die grössten Erkenntnisse wurden überraschenderweise beim Ansteuern der Motoren gewonnen. Obwohl für die Ansteuerung der Motoren nicht viel Aufwand eingeplant wurde, brachten diese mit am meisten Schwierigkeiten. Es wurde deutlich, dass bei DC-Bürstenmotoren sehr hoher Anfahrtsstrom benötigt wird. Dadurch musste der Motortreiber gewechselt werden. Auch wurde mir bewusst, dass sich das Drehmoment der Motoren proportional zum Strom verhält. Dies führt auch direkt zum ersten Optimierungsvorschlag: Der Mikrocontroller sollte den Strom der Motoren regulieren können und nicht wie aktuell umgesetzt die Spannung über den Motoren. Dies würde eine zweite Regelung benötigen, entweder per Software oder per Hardware. So eine Regelung nennt man auch Drehmomentregler. Die neueste Version der Leiterplatte (V1.01) ermöglicht dem Mikrocontroller das Messen der Motorströme. Jedoch muss die Software mit der dazugehörigen Regelung noch implementiert werden.

Zweiter Verbesserungspunkt wäre die Motorenauswahl. Anstelle von DC-Bürstenmotoren würde ich bei nächsten Mal auf bürstenlose Motoren wechseln. Diese zahlen sich durch einen besseren Wirkungsgrad und weniger Spiel der Motorachsen aus. Bei Bürstenmotoren müssen die internen Bürsten für die Kontaktierung jedes Mal die Richtung wechseln, wenn der Motor die Drehrichtung ändert. Dies resultiert in einer grösseren Verzögerungszeit beim Richtungswechsel, sowie einer geringeren Lebensdauer der Motoren durch die mechanische Belastung der Bürsten. Jedoch sind bürstenlose Motoren komplizierter zum Ansteuern, was eine grosse Änderung der Hardware in Anspruch nehmen würde.

Sehr zufrieden bin ich mit der Software des Balancer Robots. Diese nahm auch sehr viel Zeit in Anspruch, ist dafür voll funktionstüchtig und professionell umgesetzt. Alle gewünschten Features, wie z.B. die LED-Anzeige, das Messen der Akkuspannung, das Steuern der Motoren, die Kommunikation mit dem Sensor etc. konnten umgesetzt werden.

Interessant war auch das Ausprobieren des P-Regler und PID-Regler Prinzip für den Regelungsalgorithmus. Leider reichte dies nicht für ein vollständiges Ausbalancieren. Ein Elektroingenieur mit Masterabschluss in Regelungstechnik riet mir den Einsatz eines Zustadoreglers. Doch die Theorie dahinter wird erst auf dem Niveau eines Masters an der Fachhochschule behandelt.



Um meine Leitfrage zu beantworten: Ja, es ist definitiv möglich, mithilfe eines Beschleunigungs- und Gyrosensors ein Fahrzeug auf zwei Rädern auszubalancieren. Die Ergebnisse der verschiedenen Versuche und Experimente zeigen deutlich auf, dass der Winkel des Fahrzeugs bestimmt werden und die Motoren reguliert werden können. Mit einem geeigneten Regelungsalgorithmus müsste ein Ausbalancieren also möglich sein.



Quellenverzeichnis

Conrad: MFA 950D501 Getriebemotor

<https://www.conrad.ch/de/p/mfa-950d501-getriebemotor-12-v-dc-50-1-222373.html>

[Stand: 11.09.2019]

[Abrufdatum: 11.09.2019]

Heuler: Regelungstechnik

http://www.heuler.net/stud/pendel/html_netz/node9.html

[Stand: 07.01.1997]

[Abrufdatum: 23.10.2019]

Mikrocontroller.net: Pulsweitenmodulation.

<https://www.mikrocontroller.net/articles/Pulsweitenmodulation>

[Stand: 30.09.2019]

[Abrufdatum: 30.09.2019]

NKON: Lithium-Ionen Akku

<https://www.nkon.nl>

[Stand: 25.10.2019]

[Abrufdatum: 25.10.2019]

Texas Instruments: L293D

<http://www.ti.com/product/L293D>

[Stand: 24.10.2019]

[Abrufdatum: 24.10.2019]

Texas Instruments: L293D PDF

<http://www.ti.com/lit/ds/symlink/l293d.pdf>

[Stand: Januar 2016]

[Abrufdatum: 30.09.2019]

Wikipedia: Gleitender Mittelwert

https://de.wikipedia.org/wiki/Gleitender_Mittelwert

[Stand: 24.10.2019]

[Abrufdatum: 24.10.2019]

Wikipedia: Lithium-Ionen-Akkumulator

<https://de.wikipedia.org/wiki/Lithium-Ionen-Akkumulator>

[Stand: 14.11.2019]

[Abrufdatum: 14.11.2019]

Wikipedia: *Vierquadrantensteller*

<https://de.wikipedia.org/wiki/Vierquadrantensteller>

[Stand: 10.06.2017]

[Abrufdatum: 30.09.2019]



ST Microelectronics: L298 Datenblatt

PDF-Datei

[Abrufdatum: 24.10.2019]

InvenSense: MPU-6050 Datenblatt.

PDF-Datei

[Abrufdatum: 20.09.2019]



Abbildungsverzeichnis

Abbildung 1: Sensorboard MPU-6050.....	8
Abbildung 2: Achsenansicht von oben	8
Abbildung 3: Kraftvektoren beim Kippen.....	8
Abbildung 4: Kräftedreieck für Winkelberechnung	8
Abbildung 5: Blockschema MPU-6050 (InvenSense, 2019)	9
Abbildung 6: Gleitender Mittelwert (Wikipedia, 2019)	9
Abbildung 7: Messung des Accelerometers	10
Abbildung 8: Messung mit gleitendem Mittelwert.....	10
Abbildung 9: Messwerte der Z-Achse im freien Fall	11
Abbildung 10: Messwerte X-Achse Accelerometer.....	11
Abbildung 11: Messwerte der Y-Achse Gyroskop	11
Abbildung 12: (Wikipedia, 2019).....	12
Abbildung 13: (Texas Instruments, 2019)	12
Abbildung 14: (Mikrocontroller.net, 2019)	13
Abbildung 15: Messaufbau.....	14
Abbildung 16: Messung 1, Start des Motors.....	14
Abbildung 17: Messung des Stroms bei Anfahren mit PWM.....	15
Abbildung 18: Messung des Stroms bei PWM 60%	15
Abbildung 19: Block Diagramm aus L298 Datenblatt (ST Microelectronics, 2019)	16
Abbildung 20: Leiterplatte „Motor Driver V1.00“	17
Abbildung 21: Leiterplatte „Controller Board V1.00“	17
Abbildung 22: Leiterplatte "Balancer Robot V1.00"	17
Abbildung 23: Mode-Taster	18
Abbildung 24: Leiterplatte "Balancer Robot V1.01"	18
Abbildung 25: LED-Anzeige	18
Abbildung 26: Akku-Stecker	19
Abbildung 27: Motor-Stecker.....	19
Abbildung 28: L298 Motor Treiber.....	19
Abbildung 29: Seitenansicht, Rückseite des L298.....	19
Abbildung 30: Sensorboard MPU-6050.....	20
Abbildung 31: Mikrocontrollerboard STM32F303K8	20
Abbildung 32: Li-Ion Akkuzelle Typ 18650	21
Abbildung 33: Angaben der bestellten Akkuzellen (NKON, 2019)	21
Abbildung 34: Spannungsteiler auf der Leiterplatte	21
Abbildung 35: MFA 950D501 (Conrad, 2019)	22
Abbildung 36: Erster Prototyp, Vogelperspektive.....	23
Abbildung 37: Erster Prototyp, Seitenansicht	23
Abbildung 38: 3D-Modell Explosionszeichnung.....	24
Abbildung 39: (Heuler, 2019)	26



Dank

Einen grossen Dank möchte ich an folgende Personen aussprechen:

Enrico Malancarne

Mein Elektrotechniklehrer unterstütze mich tatkräftig in der Informationsbeschaffung über Regelungstechnik. Er war meine erste Anlaufstelle, wenn Probleme in der Elektronik auftauchten, oder wenn ich Tipps brauchte für das Entwickeln des Fahrzeugs.

Hanspeter Lutz

Durch Herrn Lutz als Betreuer erfuhr ich Unterstützung, wenn es um das Verstehen der Physik des Fahrzeugs ging. Er betreute meine Klasse während der Berufsmaturitätsarbeit und gab uns einen lockeren, dennoch guten Leitfaden für das Bewältigen der BMA.

Werner Wahl

Zeitgleich zur BMA wurde ich im Geschäft im Programmieren von Embedded Software geschult. Auch wenn ich bereits vorher einige Erfahrungen im Bereich der Mikrocontrollertechnik gemacht hatte, wäre die Software des Balancer Robots nie so professionell geworden, hätte Werner mich nicht geschult und gefördert.

Christian Walser

Ohne die Einwilligung meines Lehrmeisters Christian hätte ich das ganze Projekt von Zuhause aus und auf eigene Kosten entwickeln müssen. Doch durch seine Grosszügigkeit konnte ich das Equipment der Firma für diverse Testversuche nutzen. Auch durfte ich viele Komponenten des Prototyps auf Kosten der Firma bestellen.

Aaron Scheiweiler

Meinem Lehrlingskollegen Aaron möchte ich danken für das Erstellen der 3D-Modelle und der Explosionszeichnung für die Dokumentation.

Erwin Lacher

Zuletzt möchte ich meinem Vater für das Korrekturlesen der BMA danken.



Anhang

Schema

Balancer Robot.PDF

Software

Balancer Robot

main.h

main.c

system.c

system.h

MPU6050.h

MPU6050.c

Motors.h

Motors.c

GPIO.h

GPIO.c

I2C.h

I2C.c

LED.h

LED.c

Battery.h

Battery.c

Button.h

Button.c

Controller.h

Controller.c

DataProcessor.h

DataProcessor.c

TimingHandler.h

TimingHandler.c

ErrorHandler.h

ErrorHandler.c

Python Programm

UART Reader