

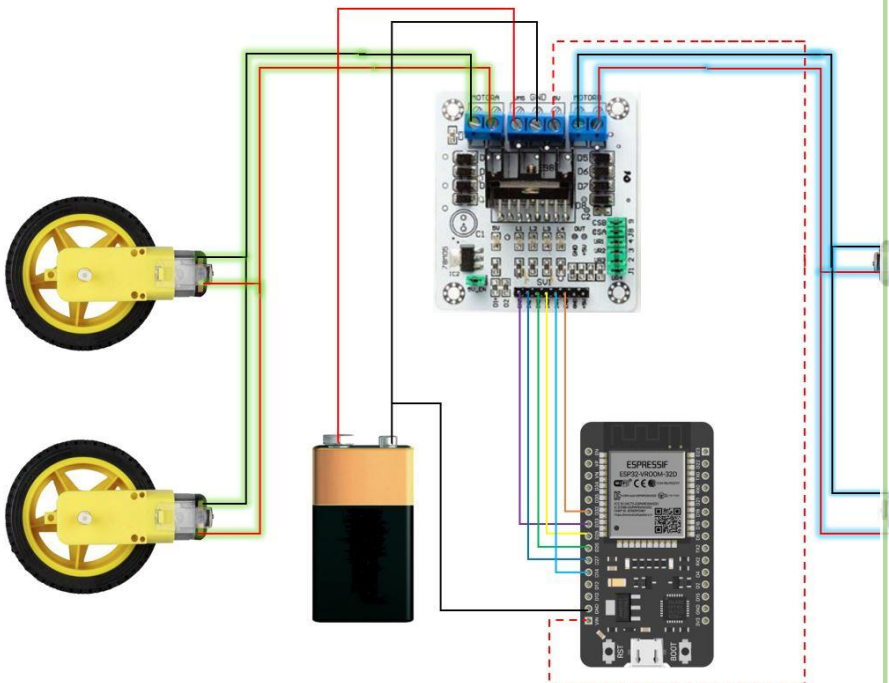
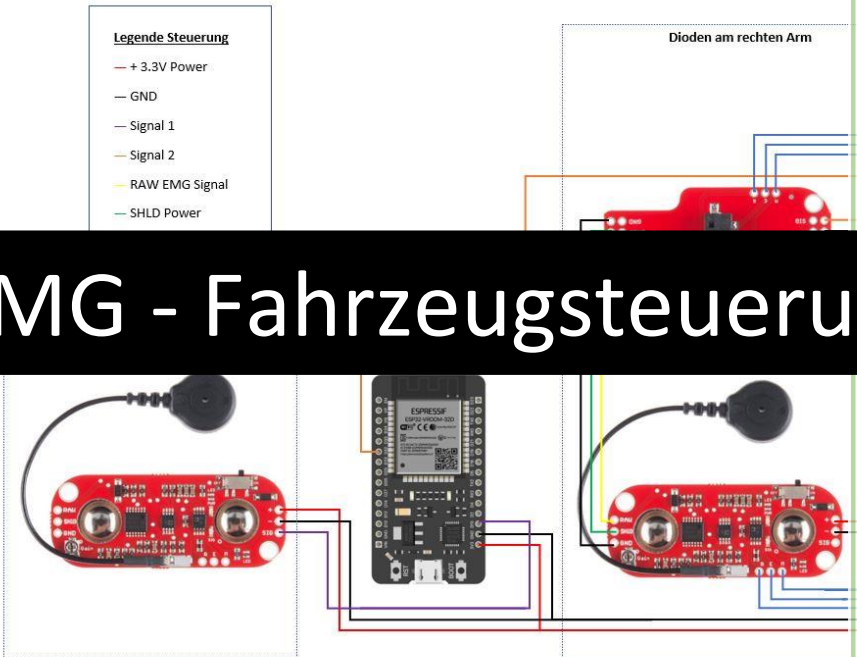
2020

EMG - Fahrzeugsteuerung

Legende Steuerung

- + 3.3V Power
- GND
- Signal 1
- Signal 2
- RAW EMG Signal
- SHLD Power

Dioden am rechten Arm



Schweizerische
Fachschule

TEKO

Diplomarbeit

Dipl. Elektrotechniker HF

Guillermo Dietrich

12.10.2020

Inhaltsverzeichnis

Verdankung	3
Eigenständigkeitserklärung	3
Management Summary.....	4
Lebenslauf	5
1 Projektinitialisierung	7
1.1 Zielsetzung.....	7
1.2 Was gibt es schon.....	7
1.3 Fachexperte	7
1.4 Projektauftrag Diplomarbeit	8
1.5 Zielscheibe.....	11
1.6 Erstellen der Teilaufgaben.....	12
2 Projektplanung	13
2.1 Projektablaufplan	13
2.2 Projektstruktur Diplomarbeit	14
3 Projektrealisierung	15
3.1 Lösungsvarianten	15
Lösungsfindung Teilaufgabe 1.....	15
Lösungsfindung Teilaufgabe 2.....	19
Lösungsfindung Teilaufgabe 3.....	22
3.2 Dokumentation	23
Steuerung	23
Fahrzeug	25
3.3 Programmierung	27
Grundlage Kommunikation	27
Grundlage Steuerung	28
Programmcode ESP32 der Steuerung	30
Der Code von ESP32 auf dem Fahrzeug.	39
3.4 Testbericht.....	51
4 Projektabschluss.....	54
4.1 Soll / Ist Vergleich.....	54
Soll / Ist Vergleich Ablaufplan	54
Soll / Ist Vergleich anhand der Zielscheibe.....	55
Statusberichte	56

4.2 Lessons Learned	61
Abbildungsverzeichnis.....	62
Tabellenverzeichnis	63
Literaturverzeichnis.....	63

Verdankung

Damit diese Arbeit umgesetzt werden konnte, möchte ich mich bei Personen/ Firmen bedanken, welche mich für die Umsetzung unterstützt haben.

Stromplan AG, Stans: Für die mögliche Nutzung der CAD Zeichnungsprogramme

Mario Bubalovic: Für die fachtechnische Unterstützung

Eigenständigkeitserklärung

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbständig erarbeitet habe und keine anderen als die im Verzeichnis angegebenen Hilfsmittel benutzt habe.

Ort, Datum

Luzern, 09.10.2020

Unterschrift Guillermo Dietrich



The image shows a handwritten signature in blue ink. The signature is written in a cursive style and appears to read 'Guillermo Dietrich'. The signature is written over a horizontal line.

Management Summary

Als Abschluss zum Studium Elektrotechniker wird eine Diplomarbeit gefordert, an einer selbstdefinierten Problemstellung, um das aufgebaute Wissen praxisorientiert anzuwenden. Dabei geht es im Wesentlichen um die konsequente Zielorientierung, die Anwendung der Phasen des Projektmanagements auf der Grundlage des vermittelten Unterrichtsstoffes an der HF sowie der konsequenten Umsetzung der definierten Rahmenbedingungen.

Die vorliegende Diplomarbeit beschäftigt sich deshalb mit dem Entwickeln und Bauen von elektronischen und digitalen Schaltungen und dem Programmieren von Mikroprozessoren. Mit einer strukturierten Herangehensweise nach den Grundsätzen der vermittelten Projektmanagementgrundlagen.

Das definierte Projektziel: **Modellautosteuerung mit EMG-Sensoren inklusiv komplettem Entwicklungsbericht.**

Es wurde eine Steuerung entwickelt, die eine elektrische Aktivität im ruhenden und kontrahierten Oberarmmuskel misst, und diese Messungen der EMG-Sensoren in brauchbare digitale Werte umwandelt. Diese Werte werden an ein Modellfahrzeug gesendet. Das Modellfahrzeug hat vier Motoren das es ansteuert, ein Motor pro Rad. Abhängig von den zugestellten Werten drehen die Motoren in die einte oder andere Richtung mit einer unterschiedlichen Geschwindigkeit. So kann das Fahrzeug in alle Richtungen fahren.

Die Arbeit Unterteilt sich in zwei Hauptbereiche, zum einten in der Aufnahme der Muskelsignale und der Steuerung, die mit Hilfe dieser Signale arbeiten muss. Zum andern mit einem Modellfahrzeug, das mit den Signalen, die es von der Steuerung erhält, seine Motoren in unterschiedlich Richtungen bewegt.

Das Projektziel konnte gemäss den Vorgaben erreicht werden.

Mann könnte das Projekt Weiterentwickeln, in dem man weitere Bauteile auf dem Modellfahrzeug integriert. Einer Kamera und ein Neigungssensor damit man mit dem Fahrzeug an einem Ort fahren könnte, an den man nicht hinsieht. Des Weiteren könnte die Steuerung auch etwas anderes Steuern, eventuell sogar für Leute mit einem Handicap.

Ich halte eine Steuerung mit EMG-Sensoren für sehr sinnvoll, die Anwendungsmöglichkeiten sind vielseitig. Mit der rasanten Entwicklung der Industrialisierung 4.0 und der immer preiswerteren Elektronik Komponenten, kann man für viele aktuelle Problemstellungen eine Steuerung auf der Grundlage dieser Arbeit entwickeln.

CURRICULUM VITAE



GUILLERMO DIETRICH

Angestrebter Abschluss
Dipl. Techniker HF
Elektrotechnik

PERSÖNLICHE DATEN

Geboren am: 13.11.1987

Familienstand: ledig

Bürgerort: Vilters-Wangs SG
Horwerstrasse 34
6005 Luzern

+41 (0)78 901 70 66
chulio.dietrich@gmail.com

BERUFLICHER WERDEGANG

- 05/2019 – bis jetzt **Projektleiter Planung**
Stromplan AG, Stans
- Planung und Projektleitung von Spezialobjekten
 - Devisierung / Ausschreibung und Angebotsvergleiche
 - Fachbauleitung Termin- und Kostenkontrolle
- 03/2014 – 04/2019 **Bauleiter**
Elektro Maréchaux AG, Luzern
- Bau- oder Teilbauleitung mit Personalführung
 - Offerieren und Abrechnen der erbrachten Arbeiten bis zur Rechnungsstellung
 - Abteilungs-Lehrlingsverantwortlicher
- 01/2012 – 01/2014 **Junior Projektleiter**
Elektro Zentral, Littau
- Führen und koordinieren des Teilbereichs Service und Unterhalt
 - Kunden beraten, Offerieren und Abrechnen der Leistungen und administrative Arbeiten
 - Verantwortlich für die technisch richtige und sachgemässe Ausführung
- 02/2011 – 12/2011 **Servicemonteur**
Elektro Frey+Cie AG, Luzern
- Service und Kundenspezifische Aufträge. Selbständiges betreuen des Kundenstammes
- 08/2009 – 01/2011 **Elektromonteur**
EWD elektrodavos AG, Davos
- Installationen von Neu- und Umbauten sowohl von Wohn- und Bürogebäuden

SPRACHEN

Deutsch: ■■■■■■

Englisch: ■■■■■□

REFERENZEN

Gebe ich sehr gerne auf
Anfrage bekannt.

AUSBILDUNG

09/2017 – **Dipl. Techniker HF Elektrotechnik**
10/2020 TEKO Luzern

- Umfassende Kenntnisse in Elektrotechnik, Elektronik, Digitaltechnik, Steuerungs- und Regeltechnik sowie Robotik

08/2005 – **Lehre Elektromonteur**
07/2009 EWD elektrodavos AG, Davos

- Erwerb branchenspezifischer Kenntnisse

BESONDERE KENNTNISSE

EDV

- Microsoft Office
- TinLine und AutoCAD Zeichnungsprogramme
- Grundlagen der Programmierung in Java und Python
- Steiger NPK Kalkulation

Militär

Oberleutnant der Schweizer Armee,
seit 2020 nicht mehr dienstpflchtig.
7 Jahre Zugführer Gebirgsinfanterie.
3 Jahre Stv. Kp Kdt Gebirgsinfanterie.

Stärken

- Zuverlässige und saubere Arbeitsweise
- Team- und Mitarbeiterführung
- Hohes Verantwortungsbewusstsein
- Kunden- und zielorientierte Arbeitsweise

INTERESSEN&FREIZEIT

Hobbys

Fussball, Wandern, Lesen, Wintersport

1 Projektinitialisierung

1.1 Zielsetzung

Es soll ein Fahrzeug, welches ein Microcomputer/Entwicklerboard hat, mithilfe von Muskelspannungen oder Körperbewegungen gesteuert werden. Die Geschwindigkeit und die Richtung vom Fahrzeug soll gesteuert werden, ohne eine Tastatur oder ein Joystick bedienen zu müssen, sondern nur mit der Anspannung/Entspannung der Muskeln oder einer Bewegung der Muskeln.

Dies soll mit EMG-Sensoren erreicht werden. Die bei Muskel- oder Körperbewegung den Aktionsstrom messen und so das Modellfahrzeug über einen Microcomputer steuern.

1.2 Was gibt es schon


Auf dem Markt gibt es schon EMG-Sensoren, diese werden vorwiegend im medizinischen Bereich verwendet. Ein Modellfahrzeug, das mit einem Microcomputer gesteuert ist, ist eine beliebte und gängige Anwendung für Microcomputer. Was es aber noch nicht gibt sind Modellfahrzeuge, die über EMG-Sensoren gesteuert werden. In dieser Hinsicht unterscheidet sich dieses Vorhaben von Produkten die es auf dem Markt gibt.

1.3 Fachexperte

Fachexperte:

Mario Bubalovic
Leihgasse 12b
6340 Baar

 Mariobubalovic@outlook.com

 +41 (0)76 543 80 80

Beruf:

Softwareingenieur
Resource and Development
Siemens Schweiz AG

1.4 Projektauftrag Diplomarbeit

Projekttitel:	EMG-Fahrzeugsteuerung
----------------------	------------------------------

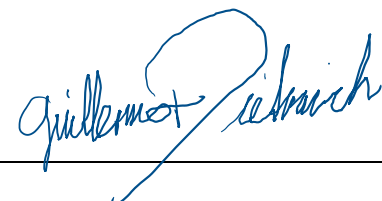
Projektauftraggeber:	TEKO Schweizerische Fachschule Luzern
Projektleiter:	Herr Guillermo Dietrich

Projektdaten			
Start:	17.08.2020	Ende:	12.10.2020

Projektbeschreibung	
Ausgangslage / Projektbegründung:	Als Abschluss zum Studium Elektrotechniker wird eine Diplomarbeit gefordert, an einer selbstdefinierten Problemstellung das aufgebaute Wissen praxisorientiert anzuwenden. Dabei geht es im Wesentlichen um die konsequente Zielorientierung, die Anwendung der Phasen des Projektmanagements auf der Grundlage des vermittelten Unterrichtsstoffes an der HF sowie der konsequenten Umsetzung der definierten Rahmenbedingungen.
Sinn und Zweck / Nutzen:	<ul style="list-style-type: none"> – Mit der Diplomarbeit wird gezeigt, dass eine Aufgabenstellung innerhalb einer vorgegebenen Zeitspanne selbstständig, umfassend und zweckmässig erstellt werden kann sowie einer sauberen Dokumentation und Präsentation. – Die Anwendung der Phasen des Projektmanagements.
Projektrichtziel:	Modellautosteuerung mit EMG-Sensoren inklusiv komplettem Entwicklungsbericht.

Endergebnisse	Erfolgskriterien
<p>1. Je ein elektronisches Exemplar an das TEKO-Sekretariat via Upload Extranet und an den betreuenden Dozenten Jörg Schenker. Sowie die Onlinepublikation der Arbeit auf die von der TEKO zur Verfügung gestellten Plattform. Die Schaltung und einem kompletten Entwicklungsbericht nach den Phasen des Projektmanagements ist als PDF-Datei bis am 12.10.2020 um 17:00 abgegeben.</p> <p>1.1 Eine Beschreibung der Ausgangslage ist erstellt</p> <p>1.2 Eine detaillierter Projektstruktur- und Ablaufplan liegt vor.</p> <p>1.3 Eine Reflexion und Erkenntnisse zur Projektarbeit sind erstellt.</p>	<p>1. Die Entwicklungsdokumentation wird durch den betreuenden Dozenten sowie den Experten als vollständig befunden.</p> <p>1.1 Die Ausgangslage ist so beschrieben, dass sie auch für eine Person mit geringen Fachkenntnissen in der Elektrotechnik verständlich ist.</p> <p>1.2 Anhand des Struktur- und Ablaufplans können die anfallenden Arbeiten überwacht werden. Es sollen keine zu grossen Abweichungen von mehreren Wochen geben.</p> <p>1.3 Es können für zukünftige Projekte Massnahmen abgeleitet werden.</p>


<p>2. Eine Steuerung mit folgenden Eigenschaften der Arbeit «EMG-Fahrzeugsteuerung» ist entwickelt, aufgebaut und getestet.</p> <p>2.1 Ein Modellfahrzeug das mit einem Mikrocomputer gesteuert ist. Der Microcomputer kann mittels EMG Sensoren gesteuert werden.</p> <p>2.2 Während der Fahrt kann mittels An- und Entspannung der Muskeln oder bewegen von Körperteilen das Fahrzeug nach links oder rechts gesteuert werden, sowie die Geschwindigkeit reguliert werden.</p>	<p>2. Die Steuerung funktioniert gemäss definierten Testszenarien oder nach allfälligen dokumentierten Abweichungen.</p> <p>2.1 Der Bediener kann mittels EMG-Sensoren auf das Fahrzeug zugreifen, das Fahrzeug kann steuern und verschiedene Geschwindigkeiten fahren.</p> <p>2.2 Während dem Fahren des Modellfahrzeugs kann die Richtung und das Tempo mit den Sensoren, die am Körper befestigt sind gesteuert werden.</p>
<p>3 Die Projektergebnisse sind während einer 15-minütigen Schlusspräsentation vorzustellen.</p>	<p>3 Die erstellte Steuerung ist funktionstüchtig und kann präsentiert werden.</p>

<p>Zielgenehmigung:</p>	<p>Die Ziele werden bewilligt.</p> <p>Datum: 2.08.2020</p> <p>Unterschrift Auftraggeber: </p>

Projekttyp:	
<input type="checkbox"/> Routineprojekt <input checked="" type="checkbox"/> komplexes Standardprojekt <input type="checkbox"/> Potenzial- / Innovationsprojekt <input type="checkbox"/> Pionierprojekt	Begründung: Weil die Schaltung von Grund auf erstellt werden muss und eine Funktion nicht gewährleistet ist.

Projektorganisation:		
Organisationstyp:	Stablinien - Projektorganisation	
Projektmitarbeiter:	Herr Dietrich Guillermo	100%
Steering Committee:	<input type="checkbox"/> Ja <input checked="" type="checkbox"/> Nein	
Sonstige Beteiligte:		

Projektplanung	
Projektphasen / Meilensteine:	1. Projektinitialisierung 2. Projektplanung 3. Projektrealisierung 4. Projektabschluss

Projektentscheid:	<input checked="" type="checkbox"/> Das Projekt wird bewilligt. <input type="checkbox"/> Das Projekt wird abgelehnt. Begründung: Datum: 2.08.2020 Unterschrift Auftraggeber: 
--------------------------	---

Tabellenverzeichnis 1

Projektauftrag

1.5 Zielscheibe

1. Je ein elektronisches Exemplar an das TEKO-Sekretariat via Upload Extranet und an den betreuenden Dozenten Jörg Schenker. Sowie die Onlinepublikation der Arbeit auf die von der TEKO zur Verfügung gestellten Plattform. Die Schaltung und einem kompletten Entwicklungsbericht nach den Phasen des Projektmanagements ist als PDF-Datei bis am 12.10.2020 um 17:00 abgegeben.
 - 1.1 Eine Beschreibung der Ausgangslage ist erstellt
 - 1.2 Eine detaillierter Projektstruktur- und Ablaufplan liegt vor.
 - 1.3 Eine Reflexion und Erkenntnisse zu der Projektarbeit ist erstellt.
2. Eine Steuerung mit folgenden Eigenschaften der Arbeit «EMG-Fahrzeugsteuerung» ist entwickelt, aufgebaut und getestet.
 - 2.1 Ein Modellfahrzeug das mit einem Mikrocomputer gesteuert ist. Der Microcomputer kann mittels EMG Sensoren gesteuert werden.
 - 2.2 Während der Fahrt kann mittels An- und Entspannung der Muskeln oder bewegen von Körperteilen das Fahrzeug nach links oder rechts gesteuert werden, sowie die Geschwindigkeit reguliert werden.
- 3 Die Projektergebnisse sind während einer 15-minütigen Schlusspräsentation vorzustellen.

Guillermo Dietrich

Horwerstrasse 34

6005 Luzern

Endergebnisse

Kunde

Sinn und Zweck

Erfolgskriterien

- Mit der Diplomarbeit wird gezeigt, dass eine Aufgabenstellung innerhalb einer vorgegebenen Zeitspanne selbstständig, umfassend und zweckmässig erstellt werden kann sowie einer sauberen Dokumentation und Präsentation.
- Die Anwendung der Phasen des Projektmanagements.

1. Die Entwicklungsdokumentation wird durch den betreuenden Dozenten sowie dem Experten als vollständig befunden.
 - 1.1 Die Ausgangslage ist so beschrieben, dass sie auch einer Person mit geringen Fachkenntnissen in der Elektrotechnik verständlich ist.
 - 1.2 Anhand des Struktur- und Ablaufplans können die anfallenden Arbeiten überwacht werden. Es sollen keine zu grossen Abweichungen von mehreren Wochen geben.
 - 1.3 Es können für zukünftige Projekte Massnahmen abgeleitet werden.
- 2 Die Steuerung funktioniert gemäss definierten Testszenarien oder nach allfälligen dokumentierten Abweichungen.
 - 2.1 Der Bediener kann mittels EMG-Sensoren auf das Fahrzeug zugreifen, das Fahrzeug kann steuern und verschiedene Geschwindigkeiten fahren.
 - 2.2 Während dem Fahren des Modellfahrzeugs kann die Richtung und das Tempo mit den Sensoren, die am Körper befestigt sind gesteuert werden.
- 3 Die erstellte Steuerung ist funktionstüchtig und kann präsentiert werden.

1.6 Erstellen der Teilaufgaben

Teilaufgabe 1:

Signal der EMG – Sensoren empfangen und in einen brauchbaren Code umwandeln. Mit den nötigen Reglern (z.B PID-Regler), damit das Signal weiterverwendet werden kann.



Abbildung 1

Teilaufgabe 2:

Mit einem Microcomputer das Signal verarbeiten und in einem geeigneten Format als Steuerung umwandeln.



Abbildung 2

Teilaufgabe 3:

Ein Fahrzeug mit einem Microcomputer, welches mit Batterie betrieben ist, erstellen (ohne Netzeinspeisung).

Die Achsenmotoren ansteuern.

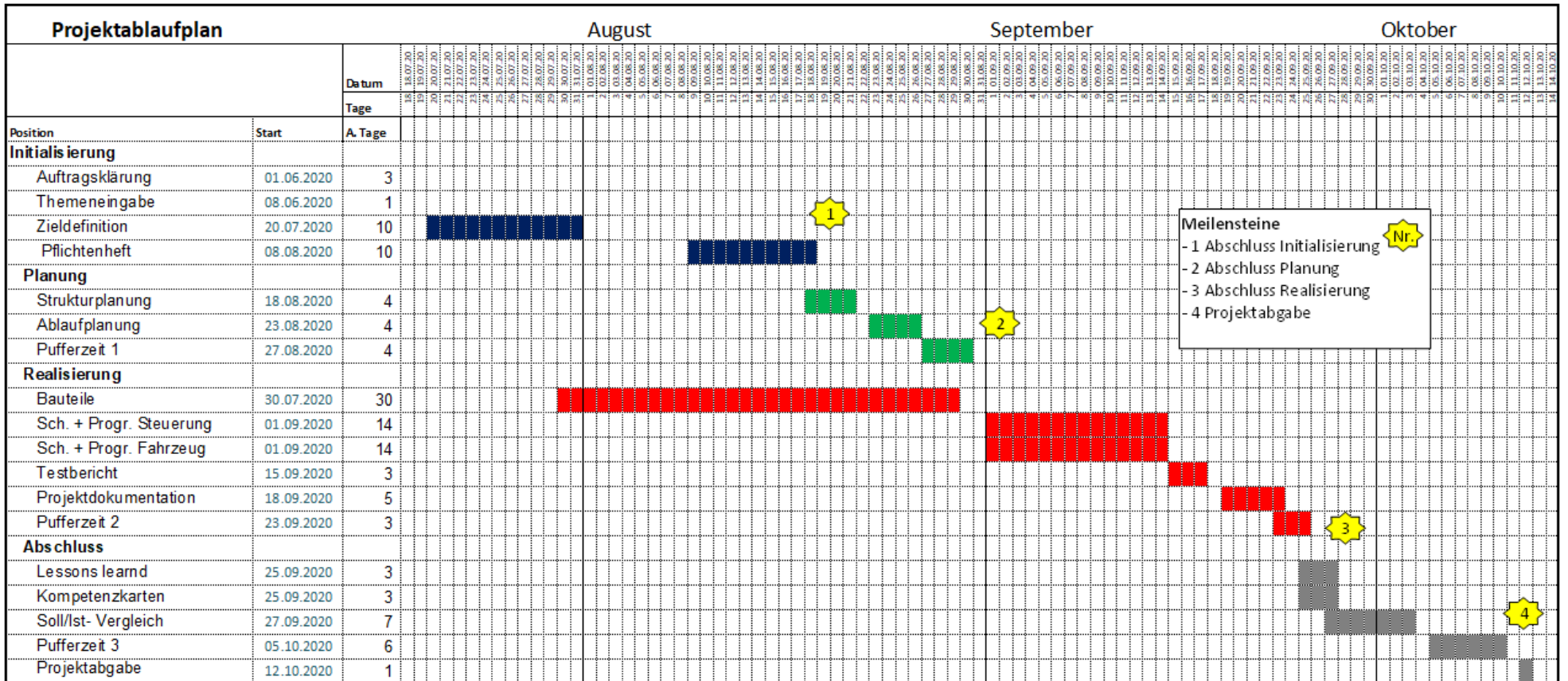
Eine Drehbewegung ansteuern (z.B Servo).



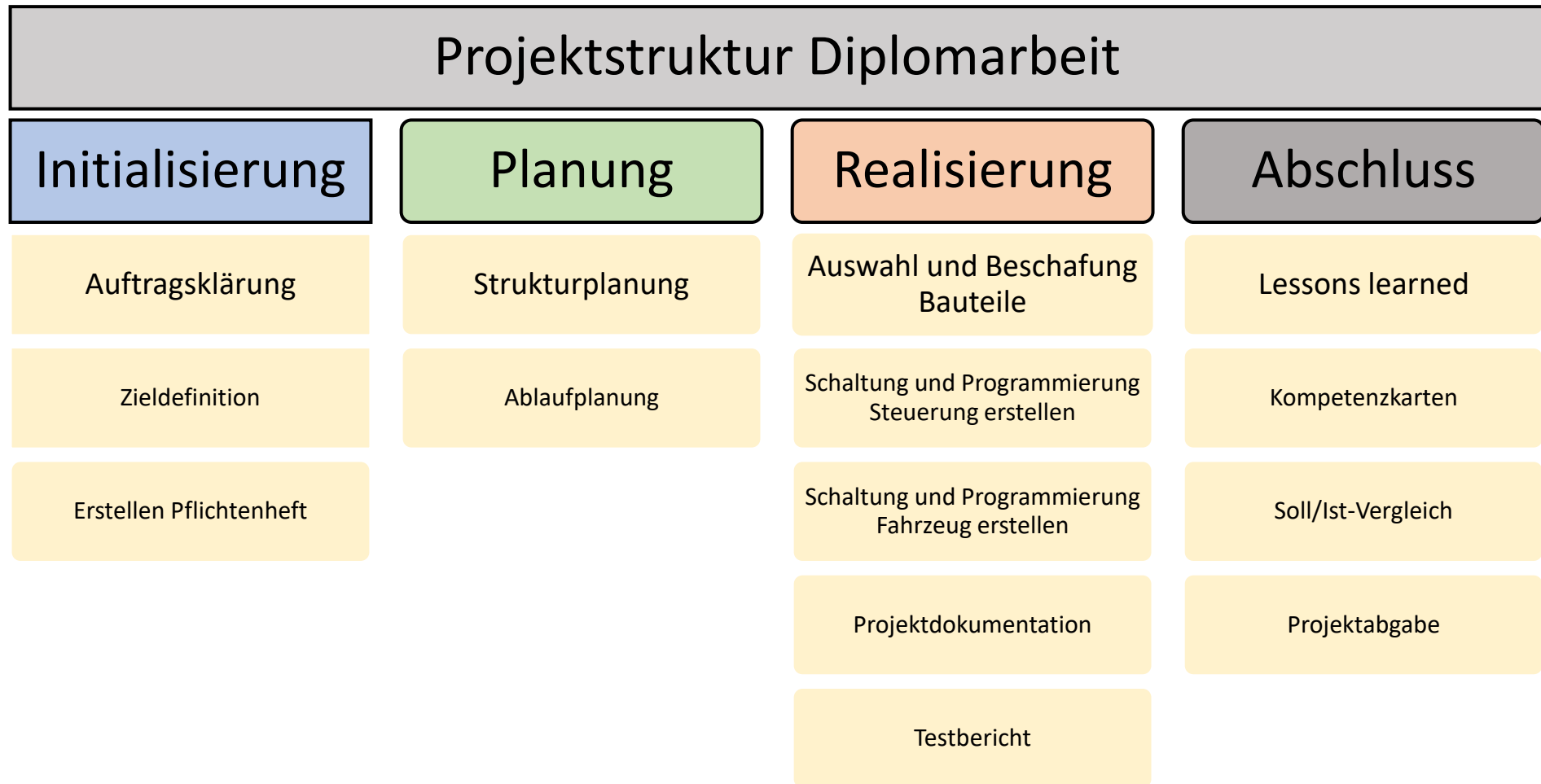
Abbildung 3

2 Projektplanung

2.1 Projektablaufplan



2.2 Projektstruktur Diplomarbeit



3 Projektrealisierung

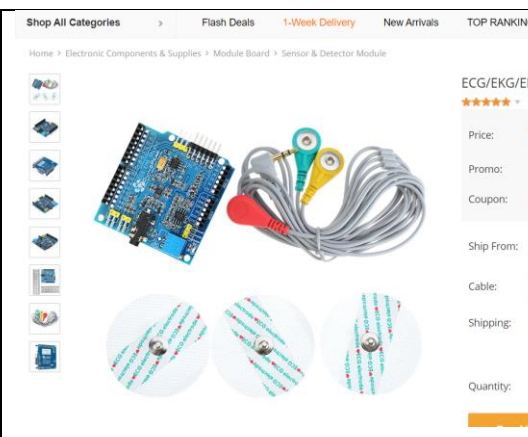
3.1 Lösungsvarianten

In einem ersten Schritt geht es darum Lösungen für die drei definierten Teilaufgaben zu finden.

Lösungsfindung Teilaufgabe 1

Es wurden verschieden EMG Sensoren im Internet angeschaut und recherchiert, um die beste Lösung für die Teilaufgabe 1 zu finden, wird eine Nutzwertanalyse erstellt. Neben dem Preis werden folgende Kriterien beurteilt: Bekanntheit/Qualifikation, Handhabung Programmierung, Erhältlichkeit, Kompatibel mit Microcomputer und die Qualität.

Variante A

<p>Anbieter: Banggood</p> <p>Bezeichnung: ECG/EKG/EMG Shield Heart Rate Muscle Sensor Meter with Cable and Electrodes - With cable</p> <p>Internetseite: https://www.banggood.com/ECG-or-EKG-or-EMG-Shield-Heart-Rate-Muscle-Sensor-Meter-with-Cable-and-Electrodes-p-1660705.html?gmcCountry=CH&currency=CHF&createTmp=1&utm_source=googleshopping&utm_medium=cpc_bgs&utm_content=xibei&utm_campaign=xibei-ssc-ch-en-all-0228&ID=558312&cur_warehouse=CN</p>	
--	---

Tabellenverzeichnis 5 Teilaufgabe 1 Variante A

Variante B

<p>Anbieter: reichelt elektronik</p> <p>Bezeichnung: GRV EMG DETECTOR Arduino - Grove EMG-Detektor</p> <p>Internetseite: https://www.reichelt.com/ch/de/arduino-grove-emg-detektor-grv-emg-detektor-p191295.html?PROVID=2808&gclid=EAIaIQobChMI49ep6rWq7AIVSrDtCh3SiQdqEAQYBSABEgIUy_D_BwE</p>	
--	--

Tabellenverzeichnis 6 Teilaufgabe 1 Variante B

Variante C

Anbieter: Play Zone

Bezeichnung: MyoWar Muskel Sensor

Internetseite: https://www.play-zone.ch/de/myoware-muscle-sensor.html?gclid=EAlaIQobChMI8f2U3cyq7AIVjOmyCh31PwaEEAAYASAAEgLE2fD_BwE



Tabellenverzeichnis 7

Teilaufgabe 1 Variante C

Variante D

Anbieter: Wish

Bezeichnung: Muskelsignalsensor Emg Sensor
Controller erkennt Muskelaktivität

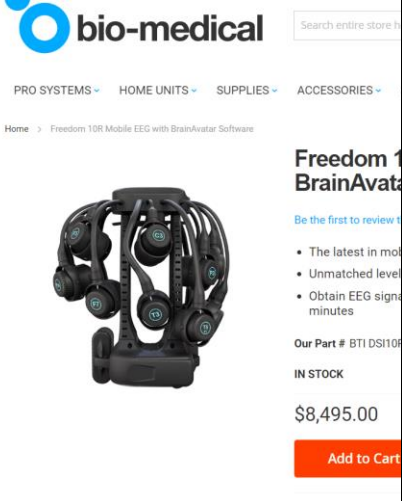
Internetseite: [https://www.wish.com/product/5e8dd25b49adae7f3b0a8a32?from_ad=goog_shopping& display_country_code=CH& force_currency_code=CHF& pid=googleadwords_int&c=%7BcampaignId%7D&ad c id=5e8dd25b49adae7f3b0a8a32&ad](https://www.wish.com/product/5e8dd25b49adae7f3b0a8a32?from_ad=goog_shopping&display_country_code=CH&force_currency_code=CHF&pid=googleadwords_int&c=%7BcampaignId%7D&ad_c id=5e8dd25b49adae7f3b0a8a32&ad)



Tabellenverzeichnis 8

Teilaufgabe 1 Variante D

Variante E

<p>Anbieter: Bio Medical</p> <p>Bezeichnung: Freedom 10R Mobile EEG with BrainAvatar</p> <p>Internetseite: https://bio-medical.com/freedom-10r-mobile-eeg-with-brainavatar-software.html</p>	
--	---

Tabellenverzeichnis 9

Teilaufgabe 1 Variante E

PROJEKT: EMG Steuerung

Objekt **EMG - Fahrzeugsteuerung**
 Aufgabe **Teilaufgabe 1 / EMG Sensoren**
 Verfahrensart **Nutzwertanalyse**

Varianten	Angebot	Anbieter	CHF	X	100
Banggood	A		26.00		100 Punkte
reichelt elektronik	B		33.00		79 Punkte
Play Zone	C		56.00		46 Punkte
Wish	D		66.00		39 Punkte
Bio Medical	E		8'495.00		0 Punkte

Bewertung		A	B	C	D	E
1. Leistung	max 100 Punkte					
Die Leistungskriterien werden <u>objektbezogen</u> beurteilt und bewertet:						
- Bekanntheit/Qualifikation	max. 20 Punkte	6	15	20	4	15
- Handhabung Programmierung	max. 20 Punkte	5	7	18	5	10
- Erhältlichkeit	max. 20 Punkte	3	15	19	3	5
- Kompatibel mit Microcomputer	max. 20 Punkte	10	10	18	7	0
- Qualität	max. 20 Punkte	3	5	15	2	20
	Zwischentotal	27	52	90	21	50
2. Preis	max 120 Punkte	100	79	46	39	0
3. Total wirtschaftlich günstigstes Preis-/Leistungsverhältnis	max 156 Punkte	127	131	136	60	50
4. Rangordnung der Angebote / Zuschlag:	<u>Angebot C</u>	3.	2.	1.	4.	5.

Ort / Datum: Luzern, 2.10.2020

Unterschrift

Abbildung 4 Teilaufgabe 1 Nutzwertanalyse

Mann hat sich für die Variante C, MyoWar Muskel Sensor System entschieden. Obwohl es nicht das preiswerteste Produkt ist. Die zusätzlichen Leistungskriterien, die in die Nutzwertanalyse einfließen, geben den Ausschlag.

Lösungsfindung Teilaufgabe 2

Es wurden verschiedene Microcomputer/ Entwicklungsboards im Internet angeschaut und recherchiert, um die beste Lösung für die Teilaufgabe 2 zu finden, wird eine Nutzwertanalyse erstellt. Neben dem Preis werden folgende Kriterien beurteilt: Programmierbarkeit, Analoge Pins, Speicherplatz, Drahtlosverbindungen und der CPU Kern sowie der CPU Takt [MHz].

Variante A

<p>Anbieter: Espressif</p> <p>Bezeichnung: NodeMcu ESP 32</p> <p>Internetseite: https://www.espressif.com/en/products/socs/esp32</p>	 A black NodeMcu ESP 32 development board with a large ESP32 chip in the center, various pins, and a USB port.
---	--


Tabellenverzeichnis 10 Teilaufgabe 2 Variante A

Variante B

<p>Anbieter: Arduino</p> <p>Bezeichnung: Arduino Uno</p> <p>Internetseite: https://www.arduino.cc/</p>	 A blue Arduino Uno development board with a USB Type-B port, a DC power jack, and various pins.
--	--

Tabellenverzeichnis 11 Teilaufgabe 2 Variante B

Variante C

<p>Anbieter: Raspberry Pi Foundation</p> <p>Bezeichnung: RASP PI 4 B 2GB Raspberry Pi 4 B, 4x 1,5 GHz, 2 GB RAM, WLAN, BT</p> <p>Internetseite: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/</p>	 A green Raspberry Pi 4 Model B single-board computer with various ports and components.
---	--

Tabellenverzeichnis 12 Teilaufgabe 2 Variante C

Variante D

Anbieter: Arduino

Bezeichnung: Arduino Uno WIFI Rev 2

Internetseite: <https://www.arduino.cc/>



Tabellenverzeichnis 13 Teilaufgabe 2 Variante D

Variante E

Anbieter: ASUS

Bezeichnung: Tinker Board S (RK3288)

Internetseite: <https://www.asus.com/ch-de/Single-Board-Computer/Tinker-Board/>



Tabellenverzeichnis 14 Teilaufgabe 2 Variante E

PROJEKT: EMG Steuerung

Objekt **EMG - Fahrzeugsteuerung**

Aufgabe **Teilaufgabe 2 / Microcomputer**

Verfahrensart **Nutzwertanalyse**

Varianten	Angebot Anbieter	CHF	X	100	
NodeMcu Esp 32	A	10.00		100	Punkte
Arduino Uno	B	24.00		42	Punkte
Rasperi Pi 4	C	49.00		20	Punkte
Arduino Uno WIFI Rev 2	E	52.00		19	Punkte
ASUS Tinker Board S	E	95.00		11	Punkte

Bewertung	A	B	C	D	E
1. Leistung max 100 Punkte					
Die Leistungskriterien werden <u>objektbezogen</u> beurteilt und bewertet:					
- Programmierung <i>max. 20 Punkte</i>	15	15	10	15	10
- Analoge Pins (Für EMG Sensoren) <i>max. 20 Punkte</i>	10	10	0	10	10
- Memory / Arbeitsspeicher <i>max. 20 Punkte</i>	10	5	15	5	20
- Drahtlosverbindung <i>max. 20 Punkte</i>	10	0	10	10	10
- CPU Kerne / CPU Takt [MHz] <i>max. 20 Punkte</i>	10	5	15	5	20
Zwischentotal	55	35	50	45	70
2. Preis max 120 Punkte	100	42	20	19	11
3. Total wirtschaftlich günstigstes Preis-/Leistungsverhältnis max 156 Punkte	155	77	70	64	81
4. Rangordnung der Angebote / Zuschlag: <u>Angebot A</u>	1.	3.	4.	5.	2.

Ort / Datum: Luzern, 2.10.2020

Unterschrift 

Abbildung 5 Teilaufgabe 2 Nutzwertanalyse

Das NodeMcu von Espressif ist das preiswerteste und anhand der Nutzwertanalyse das Produkt, das eingesetzt wird.

Lösungsfindung Teilaufgabe 3

Die Lösungsfindung zum Fahrzeug wurde mittels Mindmap realisiert. Alle wichtigen Entscheidungen sind darauf erfasst. Alle Bauteile, die eingesetzt werden sollen und die Idee, die dahintersteht.

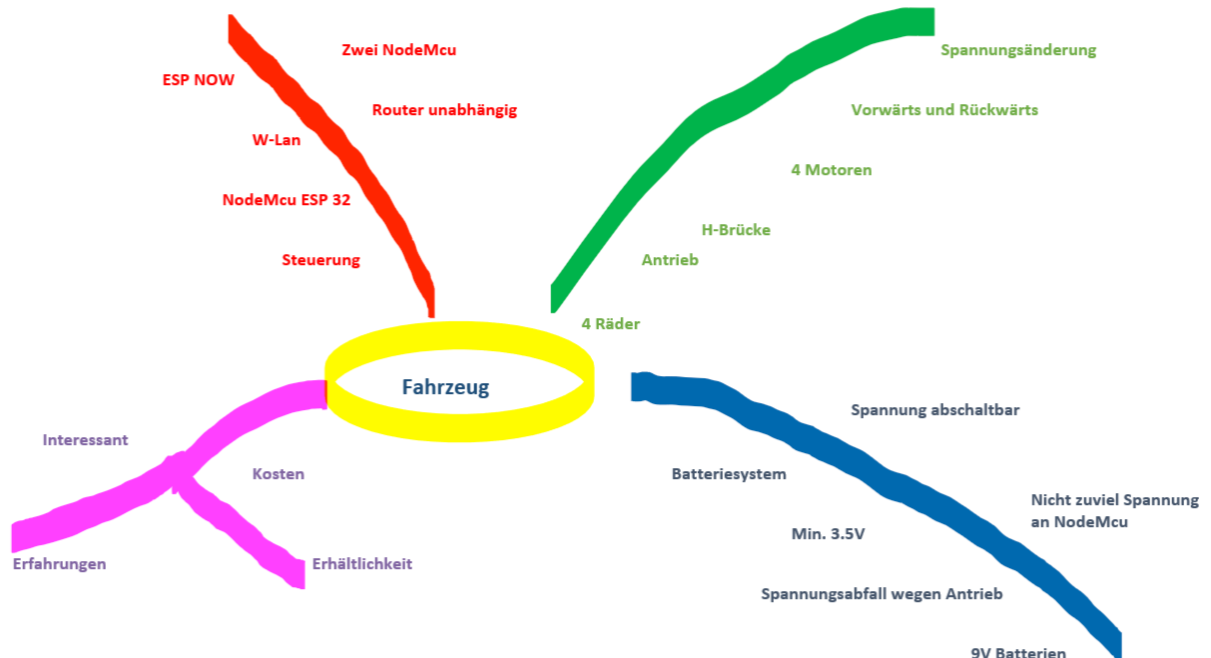


Abbildung 6 Teilaufgabe 3 Mindmap

Als Grundsatz sollen preiswerte Bauteile eingesetzt werden. Es wurde entschieden 4 Motoren einzusetzen, für jedes Rad ein Motor. Die Fahrtrichtung soll über verschiedene Spannungen und Drehrichtungen der Motoren realisiert werden. Die Spannungsversorgung für die Motoren soll über eine 9V Batterie erfolgen. Damit das NodeMcu nicht zu viel Spannung hat und zur Hilfe der Motorenansteuerung soll eine H-Brücke eingebaut werden, die einen 5V Ausgang hat.

3.2 Dokumentation

Die Dokumentation ist in zwei Teile gegliedert. Der einte Teil ist die Steuerung und der andere das Fahrzeug.

Steuerung

Verdrahtungsschema: Auf dem Verdrahtungsschema sind alle Verdrahtungen ersichtlich bis auf den linken und rechten Oberarm.

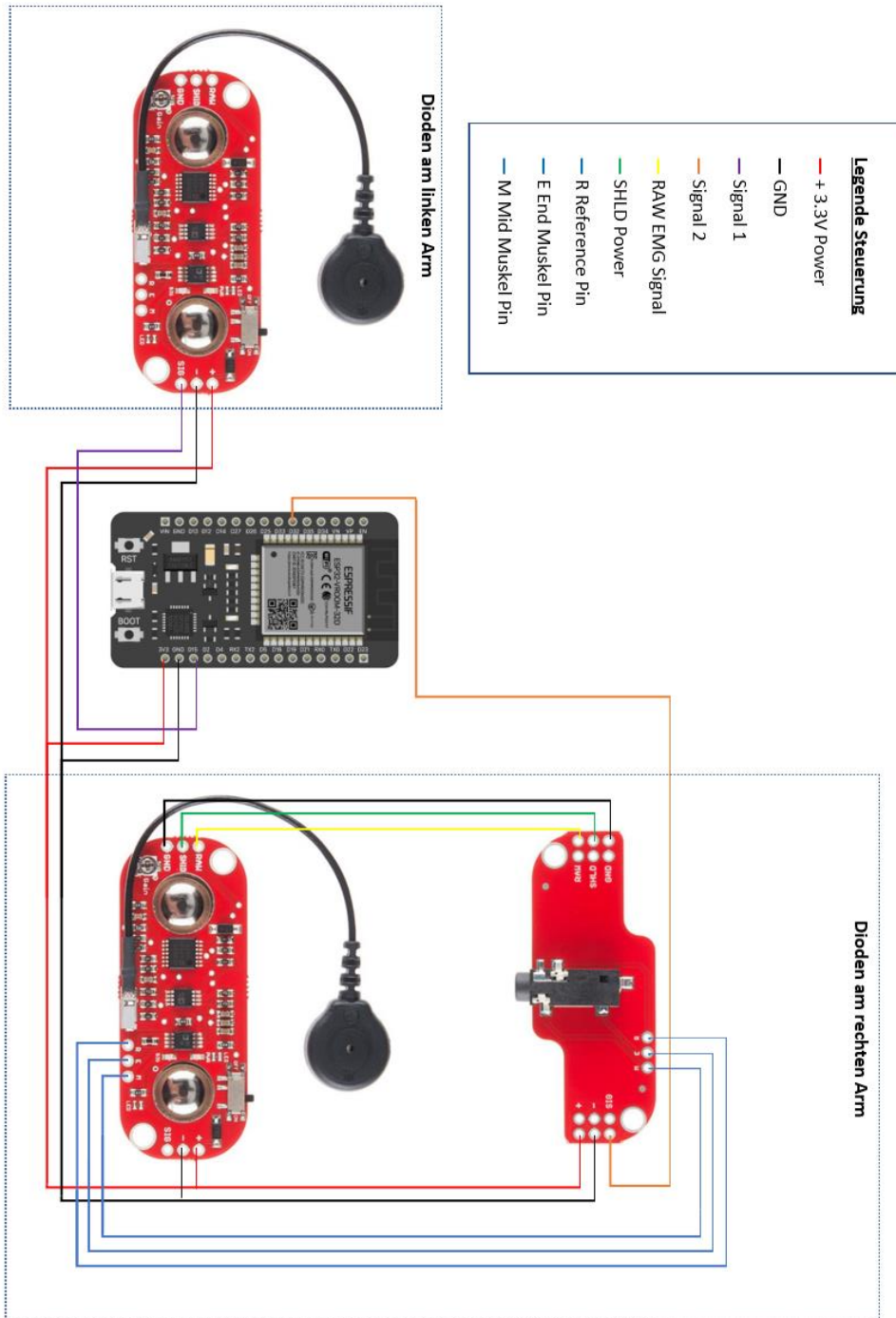


Abbildung 7 Verdrahtungsschema Steuerung

Foto der Dioden auf der Haut: Auf der Aufnahme ist ersichtlich, wie die Dioden auf den Oberarm aufgesetzt werden. Es braucht drei Dioden für die Gewinnung eines Signals. Dioden E(rot) und M(blau) auf dem Muskel und die Diode R(schwarz) als Referenz neben dem Muskel.

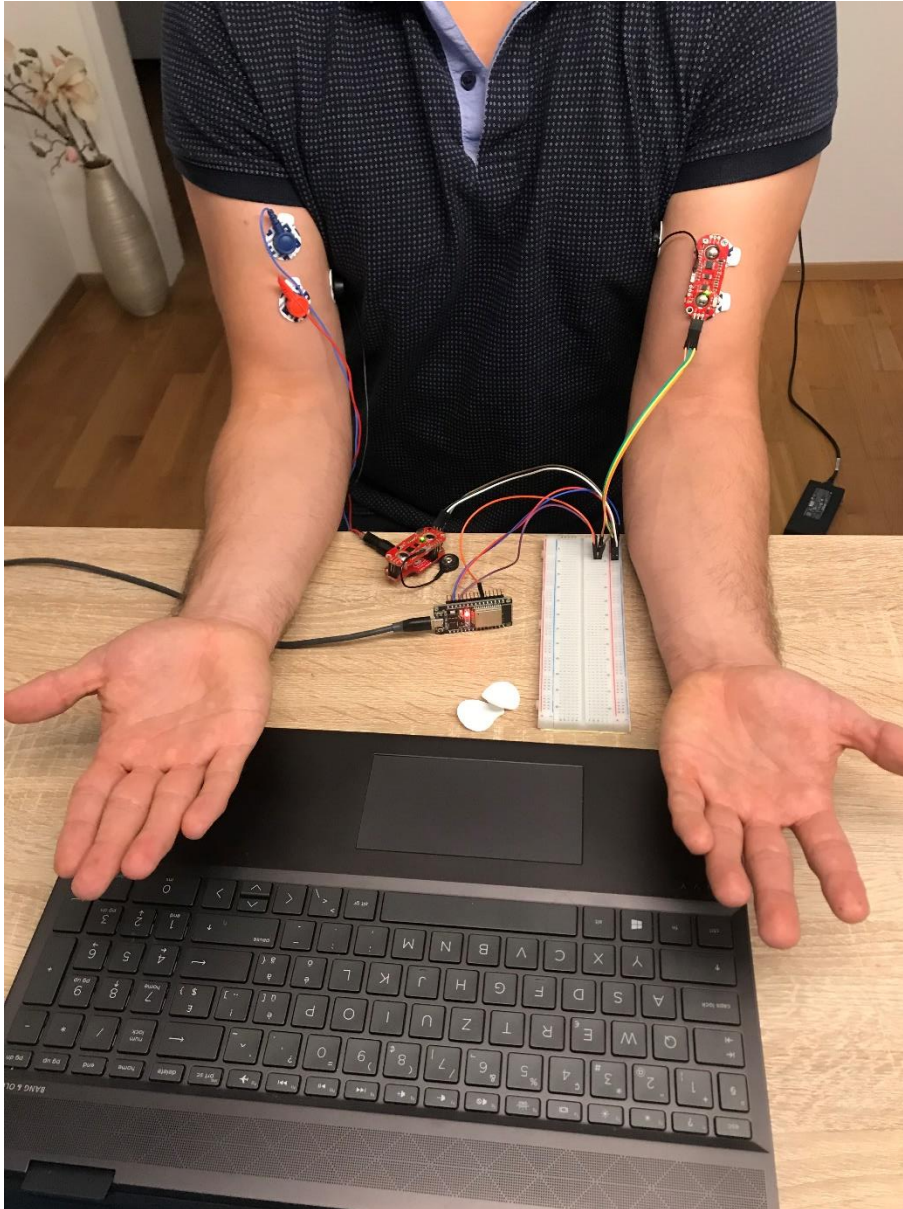


Abbildung 8 Foto Dioden Steuerung

Fahrzeug

Verdrahtungsschema Fahrzeug: Auf dem Verdrahtungsschema vom Fahrzeug ist ersichtlich, wie das NodeMcu die H Brücke und die vier Antriebe miteinander verdrahtet sind.

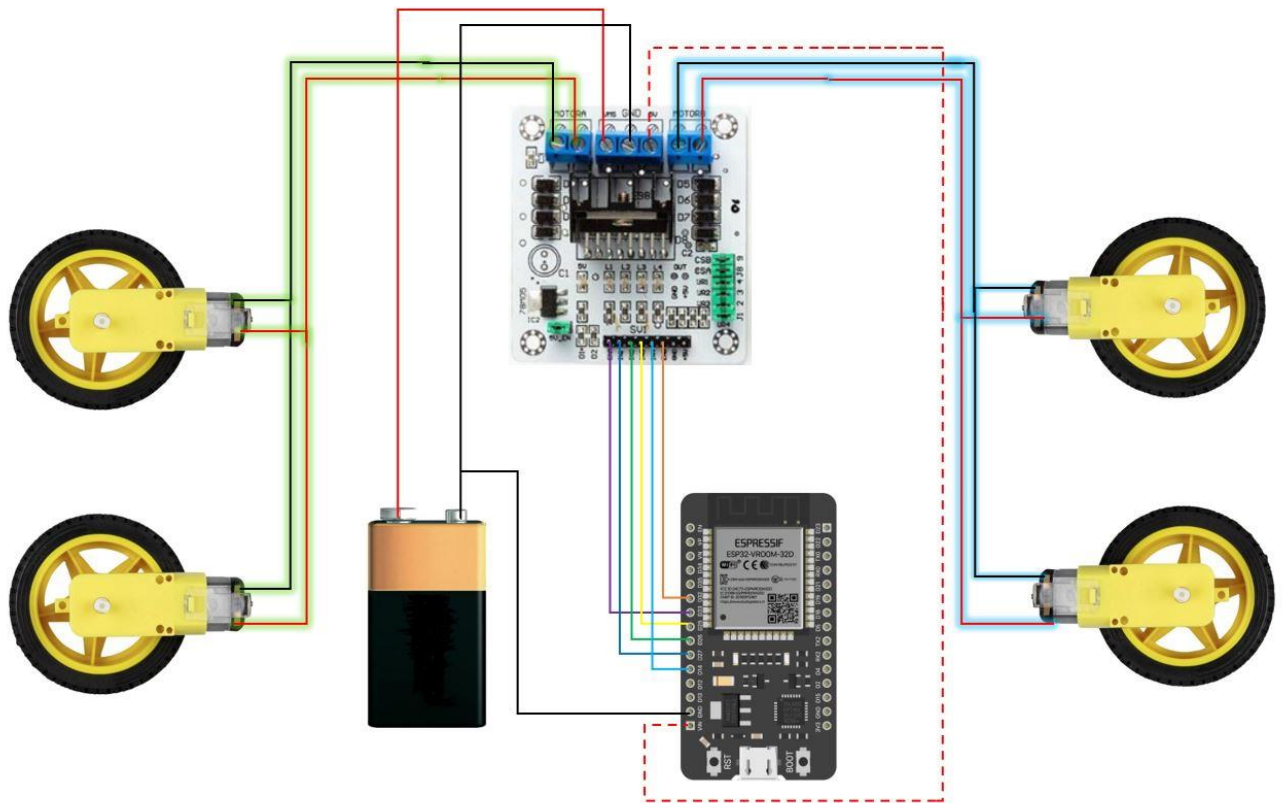


Abbildung 9 Verdrahtungsschema Fahrzeug

Legende Fahrzeug	
— + 9V Power	— PMW Signal 1
— + 5V Power	— PMW Signal 2
— GND	— IN1 Steuerung
■ Motoren A Ausgang 1	— IN2 Steuerung
■ Motoren A Ausgang 2	— IN3 Steuerung
■ Motoren A Ausgang 1	— IN4 Steuerung
■ Motoren A Ausgang 2	

Abbildung 10 Legende Fahrzeug

Foto Fahrzeug: Auf dem Foto sieht man die Anordnung auf dem Fahrzeugrahmen.

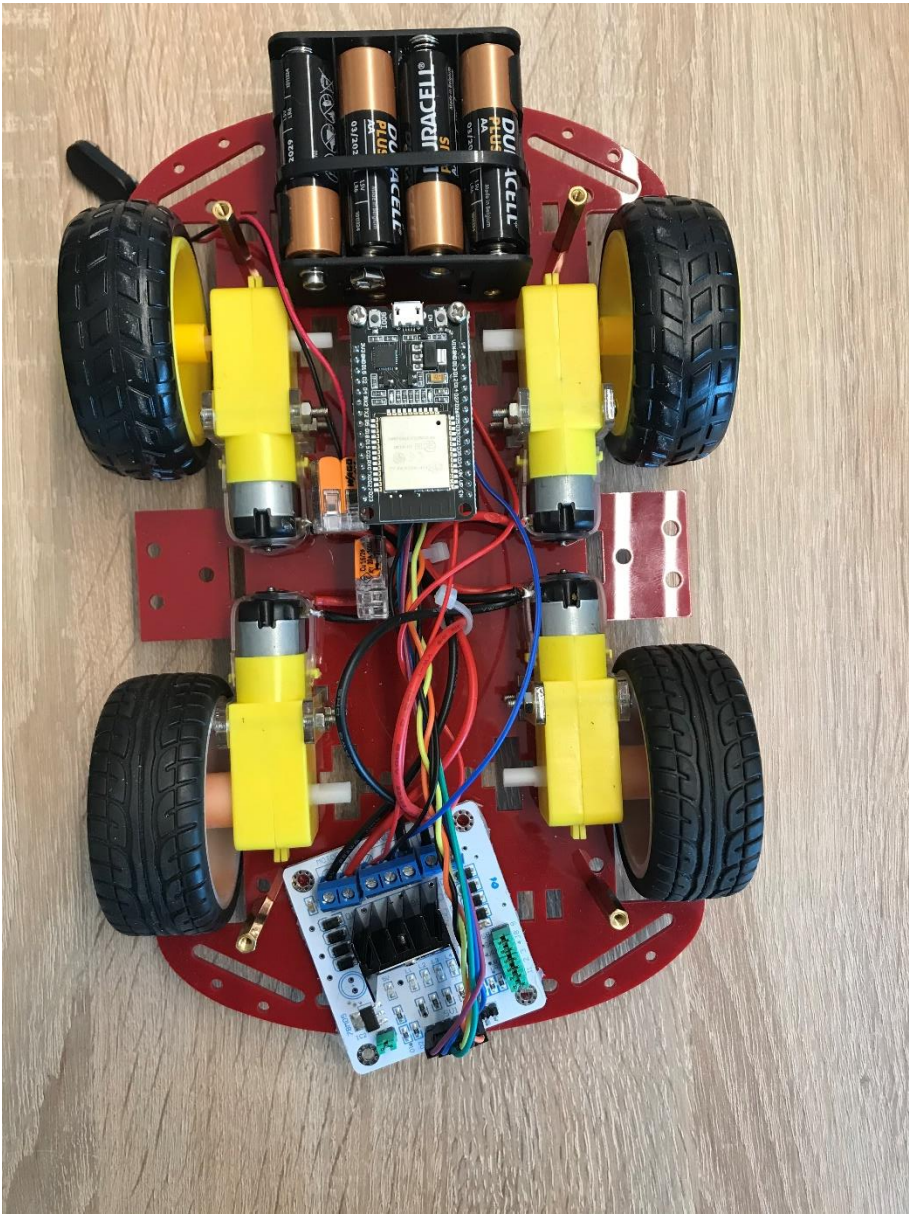


Abbildung 11 Foto Fahrzeug

3.3 Programmierung

Damit die Programmierung der Steuerung nachvollziehbar und leichter verständlich ist, sind zuerst die Ideen und Grundlagen aufgeführt.

Grundlage Kommunikation

Die Datenübertragung vom einen ESP32 zum anderen wird mit ESP-NOW realisiert. ESP-NOW kann mit Arduino IDE programmiert werden. Es ist ein kabelloses Kommunikationsprotokoll welches kleine Pakete überträgt. So können die Geräte ohne Internet oder Router selbstständig miteinander kommunizieren. Für dieses Projekt ist eine One-Way Kommunikation, vom ESP32 bei den EMG - Sensoren zum ESP32 beim Fahrzeug ausreichend.



Abbildung 12 ESP-NOW

Um via ESP-NOW zu kommunizieren muss die MAC Adresse von den ESP 32 bekannt sein. Mit folgender Eingabe links, erhalte ich die MAC Adresse vom ESP32 am Fahrzeug über die Serial Monitor Ausgabe rechts.

```
1 // MAC Adresse von einem ESP32 |
2
3 #include "WiFi.h"
4
5 void setup() {
6   Serial.begin(115200);
7   WiFi.mode(WIFI_MODE_STA);
8   Serial.println(WiFi.macAddress());
9 }
10
11 void loop() {
12
13 }
```

Abbildung 13 Code Mac Adresse

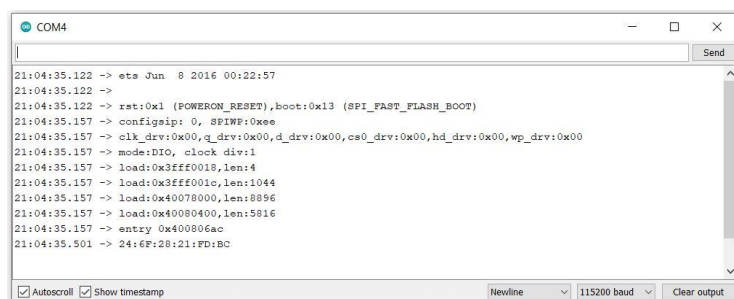


Abbildung 14 Code Ausgabe Serial Monitor

Grundlage Steuerung

Die Steuerung ist zweidimensional. Wie auf dem Bild unten ersichtlich ist, sollen die Signale der EMG-Sensoren in zwei Integer Werte umgewandelt werden. Integer Werte 1, 2 oder 3 für die X-Achse und Integer Werte 1, 2 oder 3 für die Y-Achse. Bei entspanntem Muskel Int 1, beim Heben vom Unterarm Int 2, und beim Anspannen Int 3. Auf dem Bild unten sieht man wie sich die Fahrtrichtung vom Fahrzeug mit dem X und Y Wert ändert.

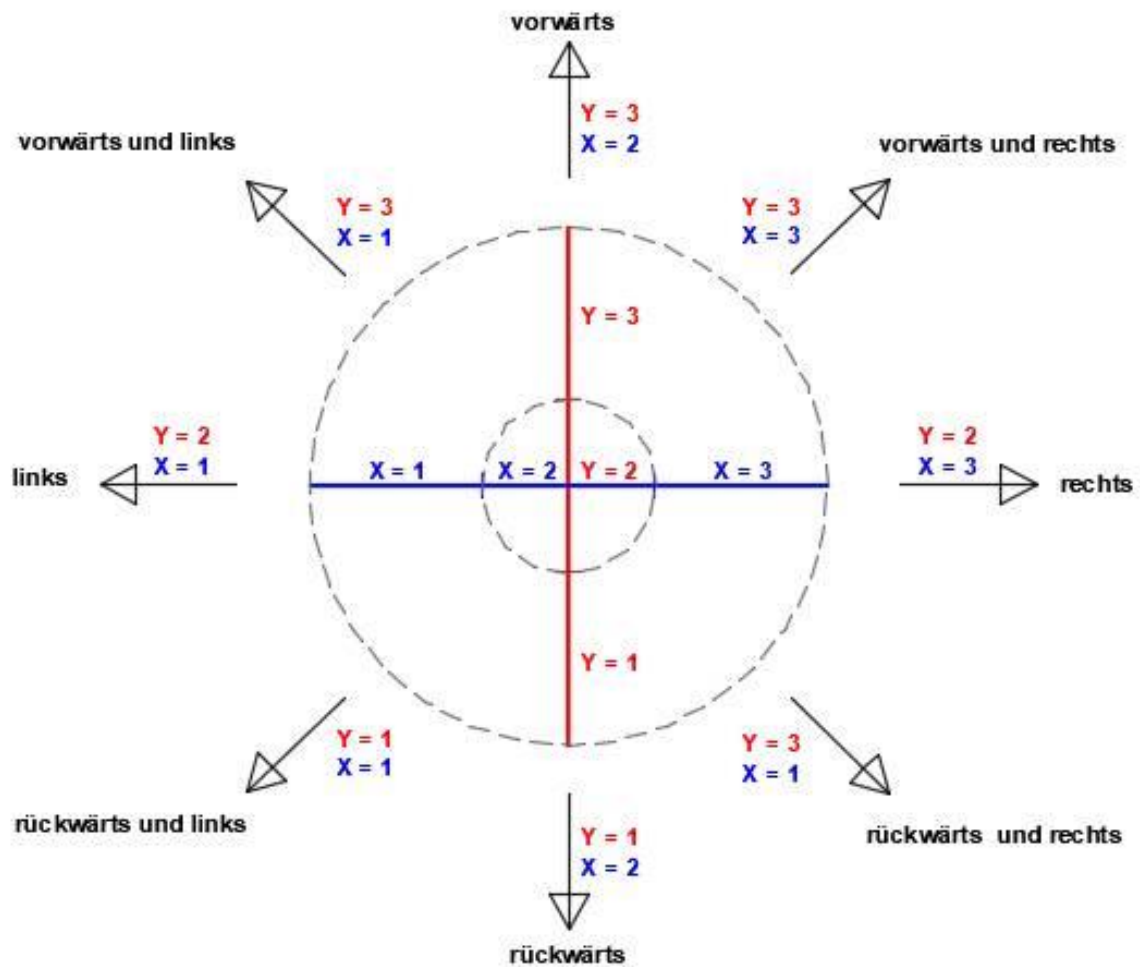


Abbildung 15 Grundlage Steuerung

Zustandsdiagramm

Im Zustandsdiagramm sieht man alle möglichen Zustände und wie diese erreicht werden können.

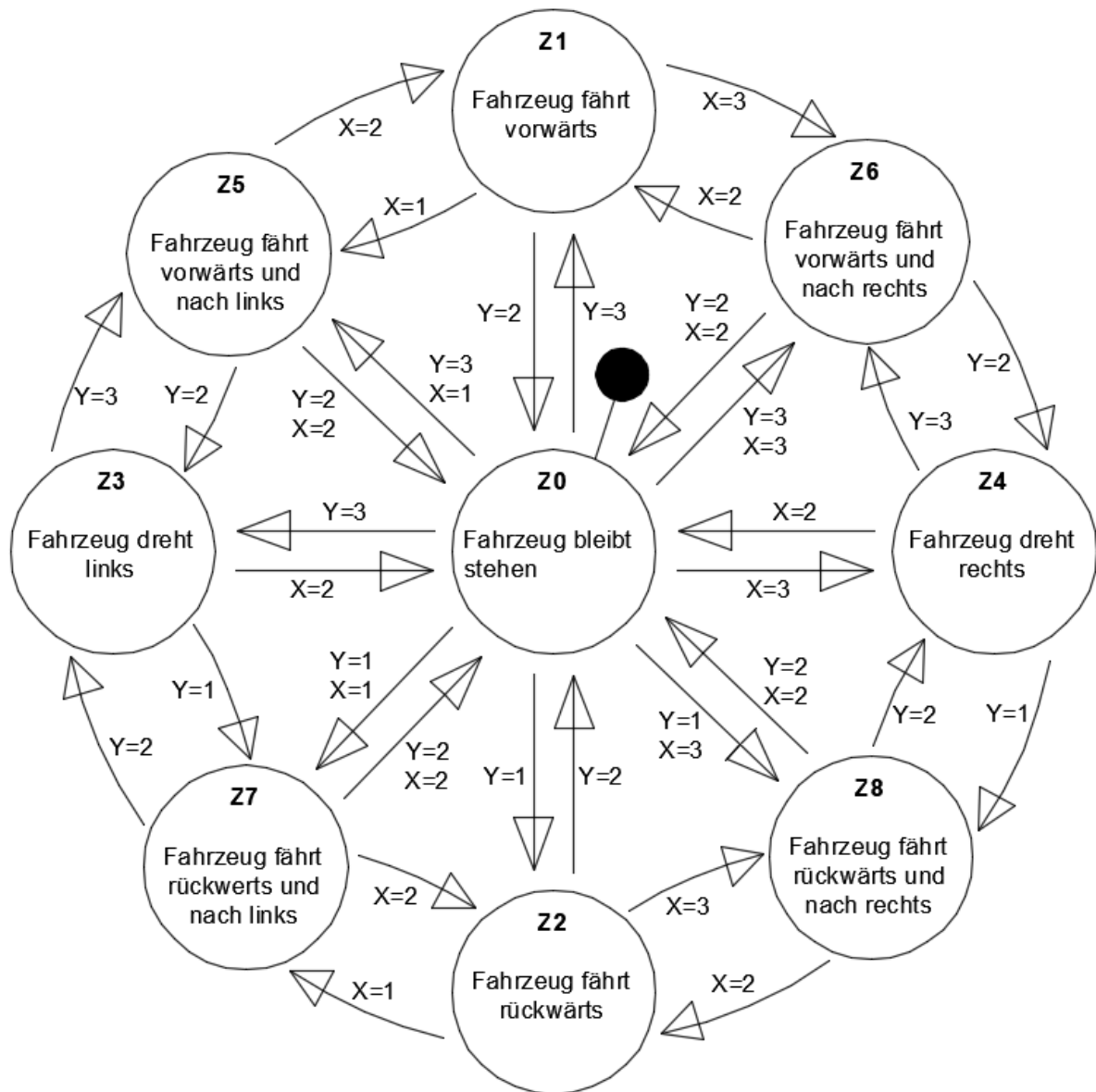


Abbildung 16 Zustandsdiagramm

Programmcode ESP32 der Steuerung

Der Code ist hier für ein besseres Verständnis im Einzelnen erklärt. Am Schluss findet sich der ganze Programmiercode der Steuerung. Zuerst werden die nötigen Bibliotheken für das Programm geladen.

```
9 //Die nötigen Bibliotheken für den Datentransfer einfügen
10 #include <esp_now.h>
11 #include <WiFi.h>
12 |
```

Abbildung 17 Code Steuerung Bibliotheken

Die nötigen globalen Variablen werden definiert. Einerseits die «X» und «Y» Integer für die Steuerung und zum anderen die float Variablen «voltage1» und «voltage2» zum Messen der Muskelspannung.

```
14 // Die Variablen die gesendet werden sollen
15 int x =2; //linker Arm
16 int y =2; //rechter Arm
17 // Die Variablen der EMG Sensoren
18 float voltage1 = 0;
19 float voltage2 = 0;
--
```

Abbildung 18 Code Steuerung Variablen

Damit die Daten dem ESP32 auf dem Fahrzeug übermittelt werden können, wird die MAC Adresse in eine uint8_t Tabelle mit dem Namen «broadcastAdress» gespeichert.

```
22 // Die MAC Adresse des Empfängers NODEMCU ESP32 auf dem Auto
23 uint8_t broadcastAddress[] = {0x24, 0x6F, 0x28, 0x21, 0xFD, 0xBC};
24
25 // Variable to store if sending data was successful
26 String success;
```

Abbildung 19 Code Steuerung MAC Adresse

Eine Struktur mit den Daten, die gesendet werden sollen wird erstellt. Diese Struktur heisst «meineDaten». Die Variablen, die gesendet werden, sind ein char mit dem Namen «a» und zwei integer mit dem Namen «xsenden» und «ysenden».

```

28 //Die Struktur als Behälter um die Daten zu senden
29 //Die Struktur muss beim Empfänger gleich sein
30 typedef struct struct_message {
31     char a[32];
32     int xsenden;
33     int ysenden;
34 } struct_message;
35
36 // Kreiert eine Strukturierte Nachricht mit dem Namen "meineDaten"
37 struct_message meineDaten;

```

Abbildung 20 Code Steuerung Struktur

Eine Funktion wird initialisiert die abfragt ob die «meinDaten» übertragen wurden, der auf Zeile 26 initialisierte String wird in «Übertragung erfolgreich :)» oder «Übertragung gescheitert : (» umgewandelt.

```

39 // Definition der Rückmeldung ob die Nachricht gesendet wurde oder nicht
40 void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
41     Serial.print("\r\nLast Packet Send Status:\t");
42     Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "Delivery Fail");
43     if (status ==0){
44         success = "Delivery Success :)";
45     }
46     else{
47         success = "Delivery Fail :(";
48     }
49 }

```

Abbildung 21 Code Steuerung Rückmeldung

Die Datenrate pro Sekunde für die serielle Datenübertragung wird auf 115200 gesetzt. Eine Wi-Fi Station wird initialisiert, die initialisierten Klassen in der Zeile 9 und 10 machen es möglich. Falls die Initialisierung nicht erfolgreich war, wird eine Fehlermeldung ausgegeben.

```

54 void setup() {
55     // initialisiere der Geschwindigkeit
56     Serial.begin(115200);
57
58     // Setzen der Wi-Fi Funktion
59     WiFi.mode(WIFI_STA);
60
61     // Inizialisierung ESP-NOW
62     if (esp_now_init() != ESP_OK) {
63         Serial.println("Error initializing ESP-NOW");
64         return;
65     }

```

Abbildung 22 Code Steuerung Datenrate

Nachdem die Initialisierung vom ESP-NOW erfolgreich war, wird die Funktion von der Zeile 39 «OnDataSent» in das Register eingetragen. Danach wird die Verbindung auf des ESP 32 vom Fahrzeug aufgebaut, das auch über ESP-NOW verfügt, damit die Daten diesem gesendet werden können. Auch hier ist eine Fehlermeldung eingebaut.

```
67 // Nach dem erfolgreichen initialisieren von ESP-NOW wird die Rückmeldungsfunktion registriert
68 esp_now_register_send_cb(OnDataSent);
69
70 // registrieren der verbindung der zwei ESP32
71 esp_now_peer_info_t peerInfo;
72 memcpy(peerInfo.peer_addr, broadcastAddress, 6);
73 peerInfo.channel = 0;
74 peerInfo.encrypt = false;
75
76 // hinzufügen der Verbindung
77 if (esp_now_add_peer(&peerInfo) != ESP_OK) {
78     Serial.println("Failed to add peer");
79     return;
80 }
```

Abbildung 23 Code Steuerung Verbinden

Im void loop() werden zuerst den zwei Sensor Eingängen aus der Zeile 18 und 19 die Pins 15 und 32 zugeteilt, sowie das sie analoge Eingänge sind. Anschliessend werden dies zwei Signale den Sensorwerten «sensorValue1» und «sensorValue2» übergeben.

```
86 void loop() {
87
88     // lesen der zwei input Variablen
89     int sensorValue1 = analogRead(32);
90     int sensorValue2 = analogRead(15);
91     // Konvertieren des analogen Eingangs
92     float voltage1 = sensorValue1 * (5.0 / 500);
93     float voltage2 = sensorValue2 * (5.0 / 500);
94     // Ausgabe der Werte:
95     // Serial.println(voltage1);
96     // Serial.println("voltage1");
97     // Serial.print(",");
98     // Serial.println(voltage2);
99     //Serial.println("voltage2");
}
```

Abbildung 24 Code Steuerung Signalverarbeitung

In den Zeilen 94 bis 99 werden die zwei werte «voltage1» und «voltage2» optisch dargestellt, unten ist ersichtlich, wie sie am Serial Monitor ausgegeben werden. Sie schwanken von 0 bis 40.

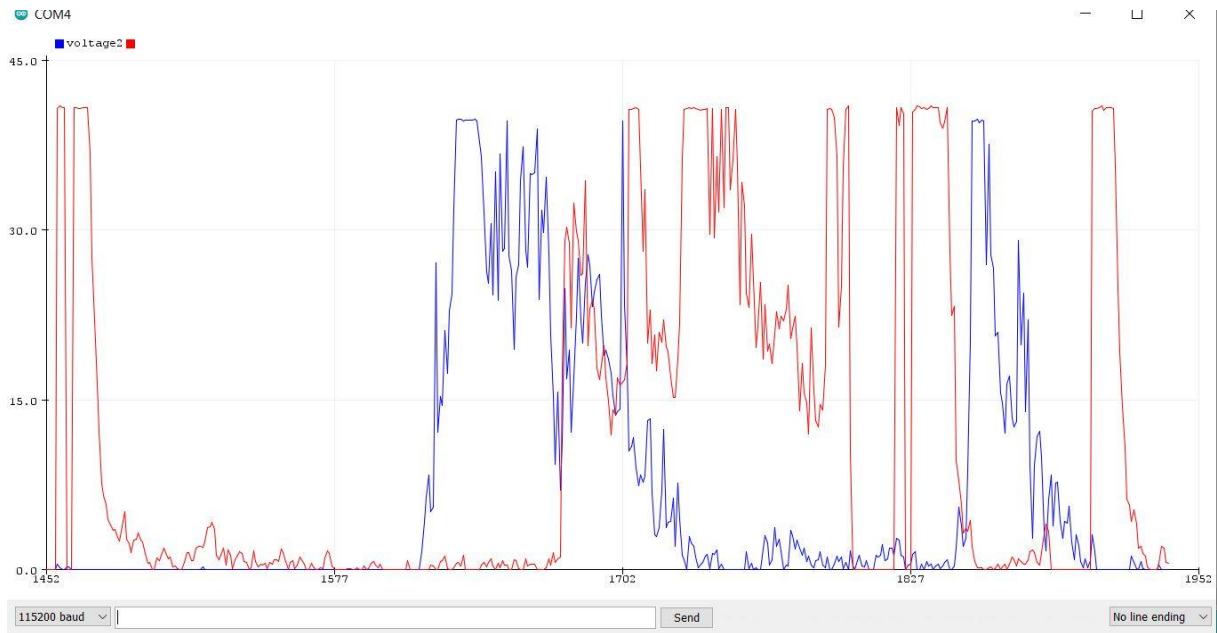


Abbildung 25 Code Steuerung Signalausgabe

Die «X» und «Y» Werte werden kaskadiert geändert. Möglicher Wert ist 1, 2 oder 3.

- Wenn der «voltage1» Wert grösser als 25 ist, wird der «X» Wert auf 3 gesetzt.
- Wenn der «voltage1» Wert dann wieder unter 20 sinkt, wird der «X» Wert wieder auf 2 gesetzt.
- Wenn der «voltage1» Wert unter 2 sinkt wird der «X» Wert auf 1 gesetzt.
- Wenn er wieder grösser als 4 wird, wird der «X» Wert wieder auf 2 gesetzt.

Zwischen «voltage2» und «Y» verhält es sich gleich wie bei «voltage1» und «X».

```
102▢   if (voltage1 > 25 && x == 0){
103     x = 1;
104 }
105▢   else if (voltage1 < 2 && x == 2){
106     x = 1;
107     }
108▢ else if (voltage1 < 20 && x == 3){
109     x = 2;
110     }
111▢   else if (voltage1 > 4 && x == 1){
112     x = 2;
113     }
114
115▢   if (voltage2 > 25 && y == 2){
116     y = 3;
117 }
118▢   else if (voltage2 < 2 && y == 2){
119     y = 1;
120     }
121▢ else if (voltage2 < 20 && y == 3){
122     y = 2;
123     }
124▢   else if (voltage2 > 4 && y == 1){
125     y = 2;
126     }
```

Abbildung 26 Code Steuerung Werte verarbeiten

Die «X» und «Y» Werte werden in der Datei «meineDaten» gesetzt. Danach werden diese der MAC Adresse, die wir in der Zeile 23 eingegeben haben, gesendet. Wir erhalten eine Nachricht ob das Senden erfolgreich war oder nicht.

Das ist der ganze Code der auf dem ESP 32 der Steuerung programmiert wurde. Mit den übermittelten der «X» und «Y» Werte auf den ESP 32 vom Fahrzeug, kann der ESP 32 vom Fahrzeug die Antriebe steuern.

```

135     // Setzen der Werte um zu senden
136     strcpy(meineDaten.a, "THIS IS A CHAR");
137     meineDaten.xsenden = x;
138     meineDaten.ysenden = y;
139
140
141     // Senden der Nachricht
142     esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &meineDaten, sizeof(meineDaten));
143
144     // Kontrolle ob die Nachricht gesendet wurde
145     if (result == ESP_OK) {
146         Serial.println("Sent with success");
147     }
148     else {
149         Serial.println("Error sending the data");

```

Abbildung 27 Code Steuerung Signal senden

Hier ist der ganze Programmiercode der Steuerung ersichtlich.

```

//Die nötigen Bibliotheken für den Datentransfer einfügen
#include <esp_now.h>
#include <WiFi.h>

// Die Variablen die gesendet werden sollen
int x = 2; //linker Arm
int y = 2; //rechter Arm
// Die Variablen der EMG Sensoren
float voltage1 = 0;
float voltage2 = 0;

// Die MAC Adresse des Empfängers NODEMCU ESP32 auf dem Auto
uint8_t broadcastAddress[] = {0x24, 0x6F, 0x28, 0x21, 0xFD, 0xBC};

// Variable to store if sending data was successful
String success;

//Die Struktur als Behälter um die Daten zu senden
//Die Struktur muss beim Empfänger gleich sein
typedef struct struct_message {
    char a[32];
    int xsenden;
    int ysenden;
} struct_message;

// Kreiert eine strukturierte Nachricht mit dem Namen "meineDaten"
struct_message meineDaten;

// Definition der Rückmeldung ob die Nachricht gesendet wurde oder nicht
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
    Serial.print("\r\nLast Packet Send Status:\t");
    Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "Delivery Fail");
    if (status == 0) {
        success = "Delivery Success :)";

```

```

}
else {
    success = "Delivery Fail :(";
}
}

// the setup routine runs once when you press reset:

void setup() {
    // inicialisiere der Geschwindigkeit
    Serial.begin(115200);

    // Setzen der Wi-Fi Funktion
    WiFi.mode(WIFI_STA);

    // Inizialisierung ESP-NOW
    if (esp_now_init() != ESP_OK) {
        Serial.println("Error initializing ESP-NOW");
        return;
    }

    // Nach dem erfolgreichen Initialisieren von ESP-NOW wird die Rückmeldungsfunktion
    registriert
    esp_now_register_send_cb(OnDataSent);

    // registrieren der Verbindung der zwei ESP32
    esp_now_peer_info_t peerInfo;
    memcpy(peerInfo.peer_addr, broadcastAddress, 6);
    peerInfo.channel = 0;
    peerInfo.encrypt = false;

    // hinzufügen der Verbindung
    if (esp_now_add_peer(&peerInfo) != ESP_OK) {
        Serial.println("Failed to add peer");
        return;
    }
}

// the loop routine runs over and over again forever:
void loop() {

    // lesen der zwei input Variablen
    int sensorValue1 = analogRead(32);
    int sensorValue2 = analogRead(15);
    // Konvertieren des analogen Eingangs
    float voltage1 = sensorValue1 * (5.0 / 500);
    float voltage2 = sensorValue2 * (5.0 / 500);
    // Ausgabe der Werte:
    // Serial.println(voltage1);
    // Serial.println("voltage1");
}

```

```

// Serial.print(",");
// Serial.println(voltage2);
//Serial.println("voltage2");

if (voltage1 > 25 && x == 0) {
  x = 1;
}
else if (voltage1 < 2 && x == 2) {
  x = 1;
}
else if (voltage1 < 20 && x == 3) {
  x = 2;
}
else if (voltage1 > 4 && x == 1) {
  x = 2;
}

if (voltage2 > 25 && y == 2) {
  y = 3;
}
else if (voltage2 < 2 && y == 2) {
  y = 1;
}
else if (voltage2 < 20 && y == 3) {
  y = 2;
}
else if (voltage2 > 4 && y == 1) {
  y = 2;
}

// Serial.println("x");
//Serial.println(x);
// Serial.print(",");
// Serial.println("y");
// Serial.println(y);

// Setzen der Werte um zu senden
strcpy(meineDaten.a, "THIS IS A CHAR");
meineDaten.xsenden = x;
meineDaten.ysenden = y;

// Senden der Nachricht
esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &meineDaten,
sizeof(meineDaten));

// Kontrolle ob die Nachricht gesendet wurde
if (result == ESP_OK) {
  Serial.println("Sent with success");
}
else {

```

```
Serial.println("Error sending the data");

//Warten einer halben Sekunde für den nächsten loop
delay (100);
}
}
```

Tabellenverzeichnis 15 Ganzer Code Steuerung

Der Code von ESP32 auf dem Fahrzeug.

Zuerst wird der Programmiercode im Detail erklärt, anschliessend ist der ganze Code zusammenhängend ersichtlich. Zuerst werden die nötigen Bibliotheken für das Programm geladen.

```
~
7  #include <esp_now.h>
8  #include <WiFi.h>
```

Abbildung 28 Code Fahrzeug Bibliotheken

Eine Struktur mit den Daten, die empfangen werden sollen wird erstellt. Diese Struktur heisst «meineDaten». Die Variablen, die empfangen werden, sind ein char mit dem Namen «a» und zwei integer mit dem Namen «xsenden» und «ysenden». Die Struktur sowie die Daten sind identisch mit der die auf den ESP32 der Steuerung initialisiert wurde.

```
12 // Erzeugen einer Struktur um Daten zu empfangen
13 typedef struct struct_message {
14     char a[32];
15     int xsenden;
16     int ysenden;
17 } struct_message;
18
19
20 // Erstellen einer strukturierten Nachricht namens "meineDaten"
21 struct_message meineDaten;
22
```

Abbildung 29 Code Fahrzeug erstellen Struktur

Die zwei globalen Integer Variablen «X» sowie «Y» werden erstellt.

```
23 int x = 0;
24 int y = 0;
```

Abbildung 30 Code Fahrzeug globale Variablen

Eine Rücksendefunktion namens «OnDataRecv» wird erstellt. Diese wird ausgeführt, wenn eine Nachricht über ENP-NOW eintrifft. Wir kopieren den Inhalt der empfangenen Nachricht von «incomingData» in «meineDaten».

```
26 // Rücksendefunktion wenn Daten erhalten
27 void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len) {
28     memcpy(&meineDaten, incomingData, sizeof(meineDaten));
29     Serial.print("Bytes received: ");
30     Serial.println(len);
31     x = meineDaten.xsenden;
32     y = meineDaten.ysenden;
33     Serial.println(meineDaten.xsenden);
34     Serial.println(meineDaten.ysenden);
35 }
```

Abbildung 31 Code Fahrzeug Funktion Daten erhalten

Die globalen integer Variablen für die zwei Motoren auf der Seite A und der zwei Motoren auf der Seite B werden erzeugt. Die Pinbelegung wird auch zugeteilt.

```
37
38
39 // IO Mapping belegung der Node MCU pins
40
41 // Motoransteuerung A
42 int motor1Pin1 = 27;
43 int motor1Pin2 = 26;
44 int enable1Pin = 33;
45 // Motoransteuerung B
46 int motor2Pin1 = 25;
47 int motor2Pin2 = 14;
48 int enable2Pin = 32;
```

Abbildung 32 Code Fahrzeug Ausgänge

Die PWM Einstellungen werden vorgenommen. Die Motoren drehen ab einem «dutyCycle» von 90. Es wird mit «pmwChannel1» und «dutyCycle1» für die linke Fahrzeugseite gearbeitet. «pmwChannel2» sowie «dutyCycle2» sind für die rechte Fahrzeugseite.

```

62 // Setzen der Pin funktionstypen
63 pinMode(motor1Pin1, OUTPUT);
64 pinMode(motor1Pin2, OUTPUT);
65 //pinMode(enable1Pin, OUTPUT);
66 //pinMode(enable2Pin, OUTPUT);
67 pinMode(motor2Pin1, OUTPUT);
68 pinMode(motor2Pin2, OUTPUT);
69 delay(100);
70 // Festlegung der Art der PWM Ausgänge
71 ledcSetup(pwmChannel1, freq, resolution);
72 ledcSetup(pwmChannel2, freq, resolution);
73
74 // Übergabe der PWM einstellungen an den node mcu pin
75 ledcAttachPin(enable1Pin, pwmChannel1);
76 ledcAttachPin(enable2Pin, pwmChannel2);

```

Abbildung 33 Code Fahrzeug PWM Signal

Die Datenrate pro Sekunde für die serielle Datenübertragung wird auf 115200 gesetzt. Eine Wi-Fi Station wird initialisiert für die Datenübertragung.

```

78 Serial.begin(115200);
79
80
81 // Set device as a Wi-Fi Station
82 WiFi.mode(WIFI_STA);
83

```

Abbildung 34 Code Fahrzeug Datenübertragungsrate

Die Initialisierung von ESP-NOW wird durchgeführt. Falls diese nicht funktioniert, wird eine Meldung über den Serial Monitor ausgegeben. Ein Register für die Rücksendefunktion wird erstellt. Diese wird ausgeführt wenn die Daten empfangen werden. Für diesen Fall registrieren wir die Funktion «OnDataRecv» die in der Zeile 26 erstellt wurde.

```

84 // Init ESP-NOW
85 if (esp_now_init() != ESP_OK) {
86     Serial.println("Error initializing ESP-NOW");
87     return;
88 }
89
90 // Register for a callback function that will be called when data is received
91 esp_now_register_recv_cb(OnDataRecv);
92

```

Abbildung 35 Code Fahrzeug Initialisierung ESP-NOW

Wie in der Abbildung 36 dargestellt ist, wird die Fahrtrichtung mit dem «X» und «Y» Wert gesteuert. In der if-Schleife wenn «x» = 1 ist und «y» = 3 passiert folgendes: Das «dutyCycle1» Signal wird auf das Maximum gestellt, die zwei Motoren auf der rechten Seite erhalten somit die Maximalspannung. -> Damit drehen die rechten Räder vom Fahrzeug so schnell wie möglich. Das «dutyCycle2» Signal ist auf ca. 40% gestellt, die zwei Motoren auf der linken Seite erhalten somit etwa 40% der Maximalspannung. -> Damit drehen die linken Räder langsamer als die Räder auf der rechten Seite. Das Fahrzeug fährt vorwärts und nach links.

```
 99   if (x == 1 && y == 3)
100     //vorwärts fahren und links drehen
101   {
102     dutyCycle1 = 255;
103     dutyCycle2 = 100;
104     ledcWrite(pwmChannel1, dutyCycle1);
105     ledcWrite(pwmChannel2, dutyCycle2);
106     digitalWrite(motor1Pin1, LOW);
107     digitalWrite(motor1Pin2, HIGH);
108     digitalWrite(motor2Pin1, LOW);
109     digitalWrite(motor2Pin2, HIGH);
110     Serial.print("Up");
111     delay(20);
112   }
```

Abbildung 36 Code Fahrzeug vorwärts links

In der if-Schleife wenn «x» = 3 ist und «y» = 3 passiert folgendes: Das «dutyCycle2» Signal wird auf das Maximum gestellt, die zwei Motoren auf der linken Seite erhalten somit die Maximalspannung. -> Damit drehen die linken Räder vom Fahrzeug so schnell wie möglich. Das «dutyCycle1» Signal ist auf ca. 40% gestellt, die zwei Motoren auf der rechten Seite erhalten somit etwa 40% der Maximalspannung. -> Damit drehen die Räder rechts langsamer als die Räder auf der linken Seite. Das Fahrzeug fährt vorwärts und nach rechts.

```
115   if(x==3 && y==3)
116     // vorwärts fahren und rechts drehen
117   {
118     dutyCycle1 = 100;
119     dutyCycle2 = 255;
120     ledcWrite(pwmChannel1, dutyCycle1);
121     ledcWrite(pwmChannel2, dutyCycle2);
122     digitalWrite(motor1Pin1, LOW);
123     digitalWrite(motor1Pin2, HIGH);
124     digitalWrite(motor2Pin1, LOW);
125     digitalWrite(motor2Pin2, HIGH);
126     delay(20);
127   }
```

Abbildung 37 Code Fahrzeug vorwärts rechts

In der if-Schleife wenn «x» = 3 ist und «y» = 1 passiert folgendes: Das «dutyCycle1» Signal wird auf das Maximum gestellt, die zwei Motoren auf der linken Seite erhalten somit die Maximalspannung. - > Damit drehen die linken Räder vom Fahrzeug so schnell wie möglich. Das «dutyCycle2» Signal ist auf ca. 40% gestellt, die zwei Motoren auf der rechten Seite erhalten somit etwa 40% der Maximalspannung. -> Damit drehen die Räder links langsamer als die Räder auf der rechten Seite. An den «motor1Pin1» und der «motor1Pin2» wird die Spannung gewechselt, «HIGH» und «LOW», damit drehen die Räder rückwärts. Das Fahrzeug fährt rückwärts und nach rechts.

```

129   if (x == 3 && y == 1)
130       // rückwärts fahren und rechts drehen
131   {
132       dutyCycle1 = 255;
133       dutyCycle2 = 100;
134       ledcWrite(pwmChannel1, dutyCycle1);
135       ledcWrite(pwmChannel2, dutyCycle2);
136       digitalWrite(motor1Pin1, HIGH);
137       digitalWrite(motor1Pin2, LOW);
138       digitalWrite(motor2Pin1, HIGH);
139       digitalWrite(motor2Pin2, LOW);
140       delay(20);
141   }

```

Abbildung 38 Code Fahrzeug rückwärts rechts

In der if-Schleife wenn «x» = 1 ist und «y» = 1 passiert folgendes: Das «dutyCycle2» Signal wird auf das Maximum gestellt, die zwei Motoren auf der linken Seite erhalten somit die Maximalspannung. - > Damit drehen die linken Räder vom Fahrzeug so schnell wie möglich. Das «dutyCycle1» Signal ist auf ca. 40% gestellt, die zwei Motoren auf der rechten Seite erhalten somit etwa 40% der Maximalspannung. -> Damit drehen die Räder links langsamer als die Räder auf der rechten Seite. An den «motor1Pin1» und der «motor1Pin2» wird die Spannung gewechselt, «HIGH» und «LOW», damit drehen die Räder rückwärts. Das Fahrzeug fährt rückwärts und nach links.

```

143   if (x == 1 && y == 1)
144       //rückwärts fahren und links drehen
145   {
146       dutyCycle1 = 100;
147       dutyCycle2 = 255;
148       ledcWrite(pwmChannel1, dutyCycle1);
149       ledcWrite(pwmChannel2, dutyCycle2);
150       digitalWrite(motor1Pin1, HIGH);
151       digitalWrite(motor1Pin2, LOW);
152       digitalWrite(motor2Pin1, HIGH);
153       digitalWrite(motor2Pin2, LOW);
154       delay(20);
155   }

```

Abbildung 39 Code Fahrzeug rückwärts links

In der if-Schleife wenn «x» = 2 ist und «y» = 3 passiert folgendes: Das «dutyCycle2» und «dutyCycle1» Signal wird auf das Maximum gestellt, alle Motoren erhalten somit die Maximalspannung. -> Damit fährt das Fahrzeug geradeaus vorwärts.

```
157   if (x == 2 && y == 3)
158       // Auto fährt geradeaus vorwärts
159   {
160       dutyCycle1 = 255;
161       dutyCycle2 = 255;
162       ledcWrite(pwmChannel1, dutyCycle1);
163       ledcWrite(pwmChannel2, dutyCycle2);
164       digitalWrite(motor1Pin1, LOW);
165       digitalWrite(motor1Pin2, HIGH);
166       digitalWrite(motor2Pin1, LOW);
167       digitalWrite(motor2Pin2, HIGH);
168       delay(20);
169   }
```

Abbildung 40 Code Fahrzeug vorwärts

In der if-Schleife wenn «x» = 2 ist und «y» = 1 passiert folgendes: Das «dutyCycle2» und «dutyCycle1» Signal wird auf das Maximum gestellt, alle Motoren erhalten somit die Maximalspannung. Die Polarität der Spannung wird gewechselt. -> Damit fährt das Fahrzeug geradeaus rückwärts.

```
171   if(x==2 && y==1)
172       // Auto fährt gerade rückwärts
173   {
174       dutyCycle1 = 255;
175       dutyCycle2 = 255;
176       ledcWrite(pwmChannel1, dutyCycle1);
177       ledcWrite(pwmChannel2, dutyCycle2);
178       digitalWrite(motor1Pin1, HIGH);
179       digitalWrite(motor1Pin2, LOW);
180       digitalWrite(motor2Pin1, HIGH);
181       digitalWrite(motor2Pin2, LOW);
182       delay(20);
183   }
```

Abbildung 41 Code Fahrzeug rückwärts

In der if-Schleife wenn «x» = 1 ist und «y» = 2 ist, passiert folgendes: Das «dutyCycle1» und das «dutyCycle2» Signal werden auf das Maximum gestellt, die vier Motoren erhalten die Maximalspannung. -> Der Pluspol «HIGH» für die zwei Motoren auf der linken Seite vom Fahrzeug wird entgegengesetzt zu der rechten Seite geschaltet. Damit drehen die zwei Räder auf der rechten Seite nach vorne und die auf der linken Seite rückwärts. Das Fahrzeug dreht somit auf der Stelle nach links.

```
185   if (x == 1 && y == 2)
186   {
187       // Auto dreht im Stillstand drht im Kreis nach links
188       dutyCycle1 = 255;
189       dutyCycle2 = 255;
190       ledcWrite(pwmChannel1, dutyCycle1);
191       ledcWrite(pwmChannel2, dutyCycle2);
192       digitalWrite(motor1Pin1, LOW);
193       digitalWrite(motor1Pin2, HIGH);
194       digitalWrite(motor2Pin1, HIGH);
195       digitalWrite(motor2Pin2, LOW);
196       delay(20);
197   }
```

Abbildung 42 Code Fahrzeug links

In der if-Schleife wenn «x» = 3 ist und «y» = 2 ist, passiert folgendes: Das «dutyCycle1» und das «dutyCycle2» Signal werden auf das Maximum gestellt, die vier Motoren erhalten die Maximalspannung. -> Der Pluspol «HIGH» für die zwei Motoren auf der linken Seite vom Fahrzeug wird entgegengesetzt zu der rechten Seite geschaltet. Damit drehen die zwei Räder auf der rechten Seite nach rückwärts und die zwei Räder auf der linken Seite vorwärts. Das Fahrzeug dreht somit auf der Stelle nach rechts.

```
199   if (x == 3 && y == 2) {
200       // Auto dreht im Stillstand, dreht sich im Kreis nach rechts
201       dutyCycle1 = 255;
202       dutyCycle2 = 255;
203       ledcWrite(pwmChannel1, dutyCycle1);
204       ledcWrite(pwmChannel2, dutyCycle2);
205       digitalWrite(motor1Pin1, HIGH);
206       digitalWrite(motor1Pin2, LOW);
207       digitalWrite(motor2Pin1, LOW);
208       digitalWrite(motor2Pin2, HIGH);
209       delay(20);
210   }
```

Abbildung 43 Code Fahrzeug rechts

In der if-Schleife wenn «x» = 3 ist und «y» = 2 ist, passiert folgendes: Alle Motoren Pin Ausgänge werden auf «LOW» gestellt, das heisst die Motoren haben keine Spannung und das Fahrzeug bewegt sich nicht.

```
212 | if(x==2 && y==2) {
213 |     // Auto befindet sich im Stillstand
214 |     digitalWrite(motor1Pin1, LOW);
215 |     digitalWrite(motor1Pin2, LOW);
216 |     digitalWrite(motor2Pin1, LOW);
217 |     digitalWrite(motor2Pin2, LOW);
218 |     delay(20);
219 | }
    |
```

Abbildung 44 Code Fahrzeug Stillstand

Hier ist der ganze Programmiercode vom Fahrzeug ersichtlich.

```
/* Arduino Diplomarbeit
   von Guillermo Dietrich
   Stand 21.09.2020

*/

#include <esp_now.h>
#include <WiFi.h>

// Erzeugen einer Struktur, um Daten zu empfangen
typedef struct struct_message {
    char a[32];
    int xsenden;
    int ysenden;
} struct_message;

// Erstellen einer strukturierten Nachricht namens "meineDaten"
struct_message meineDaten;

int x = 0;
int y = 0;

// Rücksendefunktion, wenn Daten erhalten
void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len) {
    memcpy(&meineDaten, incomingData, sizeof(meineDaten));
    Serial.print("Bytes received: ");
    Serial.println(len);
    x = meineDaten.xsenden;
    y = meineDaten.ysenden;
    Serial.println(meineDaten.xsenden);
    Serial.println(meineDaten.ysenden);
}
```

```

// IO Mapping Belegung der Node MCU pins

// Motoransteuerung A
int motor1Pin1 = 27;
int motor1Pin2 = 26;
int enable1Pin = 33;
// Motoransteuerung B
int motor2Pin1 = 25;
int motor2Pin2 = 14;
int enable2Pin = 32;

// PWM Einstellungen
const int freq = 30000;
const int pwmChannel1 = 0;
const int pwmChannel2 = 1;
const int resolution = 8;
int dutyCycle1 = 200; // Dreh geschwindigkeit des motors A
int dutyCycle2 = 200; // Dreh geschwindigkeit des motors B

void setup() {

// Setzen der Pin Funktionstypen
pinMode(motor1Pin1, OUTPUT);
pinMode(motor1Pin2, OUTPUT);
//pinMode(enable1Pin, OUTPUT);
//pinMode(enable2Pin, OUTPUT);
pinMode(motor2Pin1, OUTPUT);
pinMode(motor2Pin2, OUTPUT);
delay(100);
// Festlegung der Art der PWM Ausgänge
ledcSetup(pwmChannel1, freq, resolution);
ledcSetup(pwmChannel2, freq, resolution);

// Übergabe der PWM Einstellungen an den node mcu pin
ledcAttachPin(enable1Pin, pwmChannel1);
ledcAttachPin(enable2Pin, pwmChannel2);

Serial.begin(115200);

// Set device as a Wi-Fi Station
WiFi.mode(WIFI_STA);

// Init ESP-NOW
if (esp_now_init() != ESP_OK) {
Serial.println("Error initializing ESP-NOW");
return;
}
}

```



```

// Register for a callback function that will be called when data is received
esp_now_register_recv_cb(OnDataRecv);

}

void loop() {

if (x == 1 && y == 3)
  //vorwärts fahren und links drehen
  {
  dutyCycle1 = 255;
  dutyCycle2 = 100;
  ledcWrite(pwmChannel1, dutyCycle1);
  ledcWrite(pwmChannel2, dutyCycle2);
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, HIGH);
  Serial.print("Up");
  delay(20);
  }

if (x == 3 && y == 3)
  // vorwärts fahren und rechts drehen
  {
  dutyCycle1 = 100;
  dutyCycle2 = 255;
  ledcWrite(pwmChannel1, dutyCycle1);
  ledcWrite(pwmChannel2, dutyCycle2);
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, HIGH);
  delay(20);
  }

if (x == 3 && y == 1)
  // rückwärts fahren und rechts drehen
  {
  dutyCycle1 = 255;
  dutyCycle2 = 100;
  ledcWrite(pwmChannel1, dutyCycle1);
  ledcWrite(pwmChannel2, dutyCycle2);
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, HIGH);
  digitalWrite(motor2Pin2, LOW);
  delay(20);
  }
}

```

```

if (x == 1 && y == 1)
  //rückwärts fahren und links drehen
  {
  dutyCycle1 = 100;
  dutyCycle2 = 255;
  ledcWrite(pwmChannel1, dutyCycle1);
  ledcWrite(pwmChannel2, dutyCycle2);
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, HIGH);
  digitalWrite(motor2Pin2, LOW);
  delay(20);
  }

```

```

if (x == 2 && y == 3)
  // Auto fährt geradeaus vorwärts
  {
  dutyCycle1 = 255;
  dutyCycle2 = 255;
  ledcWrite(pwmChannel1, dutyCycle1);
  ledcWrite(pwmChannel2, dutyCycle2);
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, HIGH);
  delay(20);
  }

```

```

if (x == 2 && y == 1)
  // Auto fährt gerade rückwärts
  {
  dutyCycle1 = 255;
  dutyCycle2 = 255;
  ledcWrite(pwmChannel1, dutyCycle1);
  ledcWrite(pwmChannel2, dutyCycle2);
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, HIGH);
  digitalWrite(motor2Pin2, LOW);
  delay(20);
  }

```

```

if (x == 1 && y == 2)
  {
  // Auto dreht im Stillstand links drehen
  dutyCycle1 = 255;
  dutyCycle2 = 255;
  ledcWrite(pwmChannel1, dutyCycle1);
  ledcWrite(pwmChannel2, dutyCycle2);
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor2Pin1, HIGH);
  }

```

```

digitalWrite(motor2Pin2, LOW);
delay(20);
}

if (x == 3 && y == 2) {
  // Auto dreht im Stillstand rechts
  dutyCycle1 = 255;
  dutyCycle2 = 255;
  ledcWrite(pwmChannel1, dutyCycle1);
  ledcWrite(pwmChannel2, dutyCycle2);
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, HIGH);
  delay(20);
}

if (x == 2 && y == 2) {
  // Auto befindet sich im Stillstand
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, LOW);
  delay(20);
}
}

```

3.4 Testbericht

Im Testbericht geht es darum festzustellen, ob das Programm ordnungsgemäss läuft und die Vorgaben erreicht wurden. Die zwei aufgetauchten Fehler konnten behoben werden.

Testbericht vom 07.10.2020					
Test	Was wird getestet?	Was wird erwartet?	Was ist passiert?	Status	Massnahmen/Ursachen
1	Laden Programmiercode der Steuerung	Der Code wird ohne Fehleranzeige kompiliert und geladen	Der Code lädt auf den ESP 32 von der Steuerung ohne Fehleranzeige		
2	Laden Programmiercode des Fahrzeugs	Der Code wird ohne Fehleranzeige kompiliert und geladen	Der Code lädt auf den ESP 8266 geladen werden. Nicht kompatibel mit meiner Bitrate des Laptops		Das ESP 8266 wurde durch ein ESP32 ersetzt -> Die Steuerung konnte geladen werden
3	Die Übertragung der EMG Sensoren wird getestet	Die EMG Sensoren reagieren auf die unterschiedlichen Muskelanspannungen	Der Funktionsgraph der EMG Sensoren reagiert richtig		
4	Der Analogwert der EMG Sensoren wird in ein Digitalwert umgewandelt	Es wird ein Integer Wert zwischen 1 und 3 ausgegeben.	Der Integer Wert wechselt zwischen 1 und 3		
5	Die Übertragung auf das Fahrzeug funktioniert	Der X und Y Wert für die Steuerung kommt beim ESP32 vom Fahrzeug an.	Die Übertragung der X und Y Werte kommen an		
6	Der linke Arm wird voll angespannt der rechte Arm hat	Das Fahrzeug fährt vorwärts gemäss Signal der EMG Sensoren	Das Fahrzeug stellt ab. Der Batterieabfall war zu gross, sodass das		Die Batterie durch eine leistungsfähigere ersetzt. -> jetzt funktioniert

	eine mittlere Anspannung		ESP32 nicht mehr genug Spannung hatte.		
7	Der linke Arm wird voll entspannt der rechte Arm hat eine mittlere Anspannung	Das Fahrzeug fährt rückwärts gemäss Signal der EMG Sensoren	Das Fahrzeug fährt rückwärts		
8	Der linke Arm hat eine mittlere Anspannung der rechte Arm wird entspannt	Das Fahrzeug dreht sich nach links gemäss Signal der EMG Sensoren	Das Fahrzeug dreht sich nach links		
9	Der linke Arm hat eine mittlere Anspannung der rechte Arm hat eine volle Anspannung	Das Fahrzeug dreht sich nach rechts gemäss Signal der EMG Sensoren	Das Fahrzeug dreht sich nach rechts		
10	Der linke Arm wird voll angespannt und der rechte Arm wird entspannt	Das Fahrzeug fährt nach vorne und macht eine Kurve nach links gemäss Angaben der EMG Sensoren	Das Fahrzeug fährt nach vorne und macht eine Kurve nach links		
11	Der linke Arm und der rechte Arm werden voll angespannt	Das Fahrzeug fährt nach vorne und macht eine Kurve nach rechts gemäss EMG Sensoren	Das Fahrzeug fährt nach vorne und macht eine Kurve nach rechts		
12	Der linke Arm wird entspannt und der rechte Arm wird auch entspannt	Das Fahrzeug fährt rückwärts und macht eine Kurve nach links gemäss Angaben der EMG Sensoren	Das Fahrzeug fährt rückwärts und macht eine Kurve nach links		
13	Der linke Arm wird entspannt und der rechte Arm wird voll angespannt	Das Fahrzeug fährt rückwärts und macht eine Kurve nach rechts gemäss Signal der EMG Sensoren	Das Fahrzeug fährt rückwärts und macht eine Kurve nach rechts		
14	Der rechte und der linke Arm haben eine	Das Fahrzeug bleibt stehen	Das Fahrzeug bleibt stehen		

	mittlere Anspannung				
--	------------------------	--	--	--	--

Tabellenverzeichnis 17

Testbericht

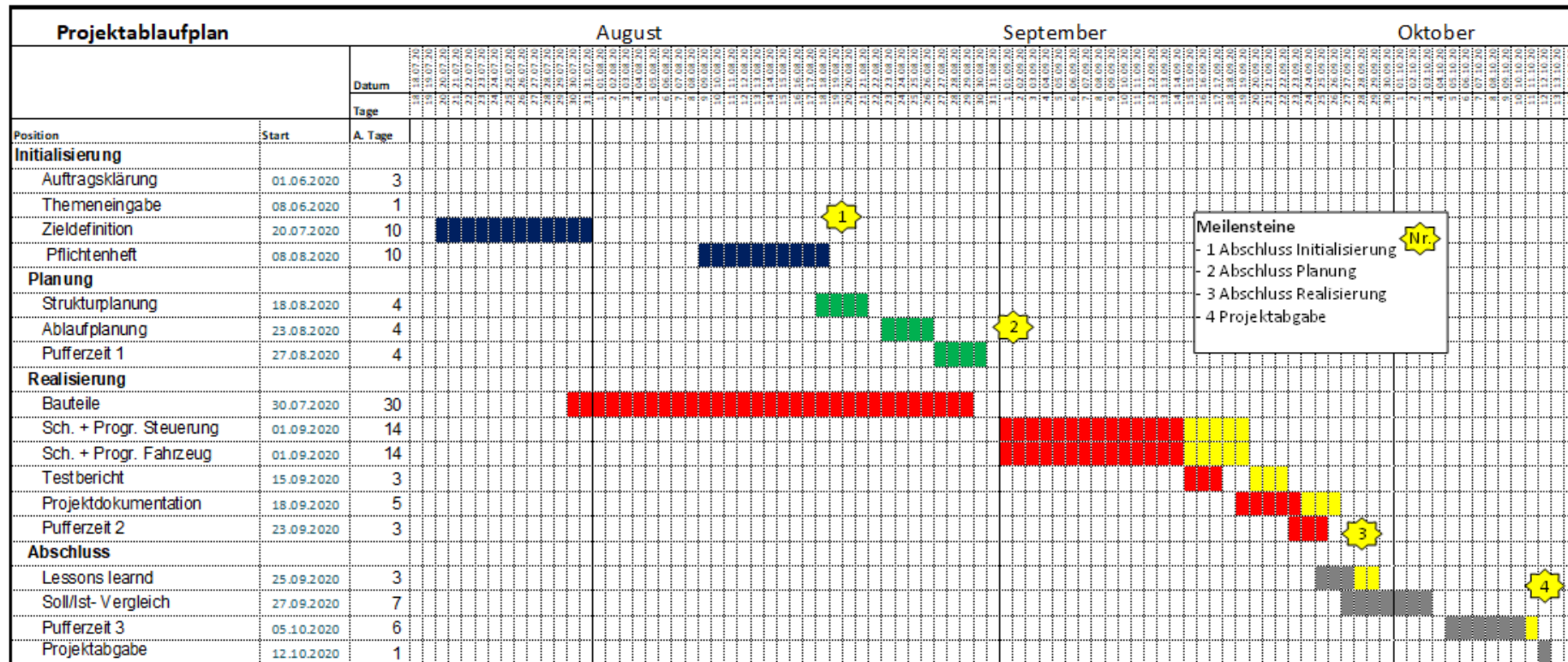
4 Projektabschluss

4.1 Soll / Ist Vergleich

Am Schluss des Projektes geht es darum den Stand des Projektes mit den gegebenen Vorgaben zu vergleichen und Abweichungen festzustellen.

Soll / Ist Vergleich Ablaufplan

Als erstes kontrollieren wir die Ablaufplanung. An den **gelb** markierten Tagen ist ersichtlich wo die Diplomarbeit in Verzug wahr.



Soll / Ist Vergleich anhand der Zielscheibe

- ✓ 1. Je ein elektronisches Exemplar an das TEKO-Sekretariat via Upload Extranet und an den betreuenden Dozenten Jörg Schenker. Sowie die Onlinepublikation der Arbeit auf die von der TEKO zur Verfügung gestellten Plattform. Die Schaltung und einem kompletten Entwicklungsbericht nach den Phasen des Projektmanagements ist als PDF-Datei bis am 12.10.2020 um 17:00 abgegeben.
 - 1.1 Eine Beschreibung der Ausgangslage ist erstellt
 - 1.2 Eine detaillierter Projektstruktur- und Ablaufplan liegt vor.
 - 1.3 Eine Reflexion und Erkenntnisse zu der Projektarbeit ist erstellt.
- ✓ 2. Eine Steuerung mit folgenden Eigenschaften der Arbeit «EMG-Fahrzeugsteuerung» ist entwickelt, aufgebaut und getestet.
 - 2.1 Ein Modellfahrzeug das mit einem Mikrocomputer gesteuert ist. Der Microcomputer kann mittels EMG Sensoren gesteuert werden.
 - 2.2 Während der Fahrt kann mittels An- und Entspannung der Muskeln oder bewegen von Körperteilen das Fahrzeug nach links oder rechts gesteuert werden, sowie die Geschwindigkeit reguliert werden.
- 3 Die Projektergebnisse sind während einer 15-minütigen Schlusspräsentation vorzustellen.

Guillermo Dietrich
 Horwerstrasse 34
 6005 Luzern

Endergebnisse

Kunde

Sinn und Zweck

Erfolgskriterien



- Mit der Diplomarbeit wird gezeigt, dass eine Aufgabenstellung innerhalb einer vorgegebenen Zeitspanne selbstständig, umfassend und zweckmässig erstellt werden kann sowie einer sauberen Dokumentation und Präsentation.
- Die Anwendung der Phasen des Projektmanagements.











- 1. Die Entwicklungsdokumentation wird durch den betreuenden Dozenten sowie dem Experten als vollständig befunden.
 - 1.1 Die Ausgangslage ist so beschrieben, dass sie auch einer Person mit geringen Fachkenntnissen in der Elektrotechnik verständlich ist.
 - 1.2 Anhand des Struktur- und Ablaufplans können die anfallenden Arbeiten überwacht werden. Es sollen keine zu grossen Abweichungen von mehreren Wochen geben.
 - 1.3 Es können für zukünftige Projekte Massnahmen abgeleitet werden.
- 2 Die Steuerung funktioniert gemäss definierten Testszenarien oder nach allfälligen dokumentierten Abweichungen.
 - 2.1 Der Bediener kann mittels EMG-Sensoren auf das Fahrzeug zugreifen, das Fahrzeug kann steuern und verschiedene Geschwindigkeiten fahren.
 - 2.2 Während dem Fahren des Modellfahrzeugs kann die Richtung und das Tempo mit den Sensoren, die am Körper befestigt sind gesteuert werden.
- 3 Die erstellte Steuerung ist funktionstüchtig und kann präsentiert werden.



Projekt: Diplomarbeit // EMG-Fahrzeugsteuerung

Stautsbericht: 1 vom 06.09.2020

Projektleiter Guillermo Dietrich	Projektziele Allgemein: Fahrzeugsteuerung mit EMG-Sensoren KW 36: Zusammenbauen der Hardware & Programmierung Fahrzeug.	Verteiler <ul style="list-style-type: none"> Herr Jörg Schenker Herr Mario Bubalovic
--	--	---

Gesamt- beurteilung	Projektverlauf  Tendenz 	Projektklima  Tendenz 	Termine  Tendenz 	Risiken  Tendenz 	Ressourcen  Tendenz 
--------------------------------	---	---	---	--	---

Aktueller Projektstand

- Durch Schwierigkeiten der Programmierung eines ESP8266 wurde ca. 5 Tage verloren. Mann ist aber noch im Zeitplan. Deshalb aber die Beurteilung bei den Terminen auf Gelb. Die Risiken und Ressourcen konnten aber mit einer Änderung der Microcomputer auf ESP32 das mit Adruino oder Phytion beschrieben werden kann.

Was läuft gut?

- Der Zusammenbau des Modelfahrzeug ist fertig. Jetzt kann man sich u die Steuerung kümmern.

Was läuft nicht gut?











- Die Ansteuerung der Fahrzeugmotoren mit dem ESP8266 funktionierte nicht gut. Nach ca. 10 Stunden wurde entschlossen die Motoren mit einem ESP32 mit einer H-Brücke anzusteuern.

Geplante nächste Schritte / getroffene Massnahmen

- Die Steuerung der Fahrzeugmotoren mit dem ESP8266 und einem Motor-Schild an der stelle wird ein ESP32 mit einer H-Brücke L298N angesteuert.
- Als nächster Schritt werden die Signale der Sensoren untersucht und versucht in brauchbare Werte umzuwandeln.

Projekt-Statusbericht; Stefan Thöni, Josef Räber

Projektleiter Guillermo Dietrich	Projektziele Allgemein: Fahrzeugsteuerung mit EMG-Sensoren KW 36: Steuerung Fahrzeug und EMG-Sensoren	Verteiler <ul style="list-style-type: none"> • Herr Jörg Schenker • Herr Mario <u>Bubalovic</u>
--	--	--











Gesamt-<u>beurteilung</u>	Projektverlauf  Tendenz 	Projektklima  Tendenz 	Termine  Tendenz 	Risiken  Tendenz 	Ressourcen  Tendenz 
----------------------------------	---	---	--	--	---

Aktueller Projektstand <ul style="list-style-type: none"> • Die Sensoren wurden mit einem ESP32 verkabelt und getestet. • Das Signal wurde optimiert für die Weiterverarbeitung. Dem Fahrzeug wurden alle Fahrtrichtungen programmiert.
--

Was läuft gut? <ul style="list-style-type: none"> • Signalübertragung der EMG-Sensoren. Was läuft nicht gut? <ul style="list-style-type: none"> • Das Fahrzeug soll kabellos fahren, deshalb sollen die zwei ESP32 ohne Internet oder Router miteinander kommunizieren können. Ist komplexer als gedacht.

Geplante nächste Schritte / getroffene Massnahmen <ul style="list-style-type: none"> • ESP32 Akku betrieb, für das Fahrzeug erstellen und auch ESP32 Akku betrieb für die Sensoren, Die Last muss zwingend vom ESP getrennt werden, das es keinen Leistungszusammenbruch gibt beim Starten der 4Elektromotoren des Fahrzeugs.











Projektleiter Guillermo Dietrich	Projektziele Allgemein: Fahrzeugsteuerung mit EMG-Sensoren KW 36: Steuerung Fahrzeug und EMG-Sensoren	Verteiler <ul style="list-style-type: none"> • Herr Jörg Schenker • Herr Mario Bubalovic
--	--	---

Gesamt- beurteilung	Projektverlauf 	Projektklima 	Termine 	Risiken 	Ressourcen 
Tendenz					

Aktueller Projektstand <ul style="list-style-type: none"> • Das Fahrzeug kann gesteuert werden. • Die Sensoren funktionieren, das Signal kann in das Programm eingebettet werden. 	Was läuft gut? <ul style="list-style-type: none"> • Das Fahrzeug fährt Kabellos im Batteriebetrieb. Was läuft nicht gut? <ul style="list-style-type: none"> • Mit der Dokumentation habe ich wegen den ständigen Projektänderungen noch nicht angefangen.
--	---

Geplante nächste Schritte / getroffene Massnahmen <ul style="list-style-type: none"> • Verbesserungen einiger Abstimmungen in der Fahrt des Modelfahrzeuges. • Dokumentation erstellen, alle Projektänderungen aufzeigen.
--

Projektleiter Guillermo Dietrich	Projektziele Allgemein: Fahrzeugsteuerung mit EMG-Sensoren KW 37: Dokumentation der Steuerung	Verteiler <ul style="list-style-type: none"> • Herr Jörg Schenker • Herr Mario Bubalovic
--	--	---

Gesamt- beurteilung	Projektverlauf 	Projektklima 	Termine 	Risiken 	Ressourcen 
Tendenz					

Aktueller Projektstand <ul style="list-style-type: none"> • Die Dokumentation wird erstellt • Am Anfang vom erstellen des Testberichts

Was läuft gut? <ul style="list-style-type: none"> • Die Steuerung funktioniert, die Dokumentation ist Abgeschlossen Was läuft nicht gut? <ul style="list-style-type: none"> • Das Projekt ist nicht im angestrebten Terminplan, es kann aber rechtzeitig fertig gestellt werden.
--

Geplante nächste Schritte / getroffene Massnahmen <ul style="list-style-type: none"> • Testen ob alles einwandfrei funktioniert, erstellen eines Testberichts. • Strukturieren und Gliedern der Diplomarbeit.
--

<p>Projektleiter Guillermo Dietrich</p>	<p>Projektziele Allgemein: Fahrzeugsteuerung mit EMG-Sensoren KW 38: Lessons learned und Soll/Ist-Vergleich</p>	<p>Verteiler</p> <ul style="list-style-type: none"> • Herr Jörg Schenker • Herr Mario Bubalovic
--	--	--

<p>Gesamt- beurteilung</p>	<p>Projektverlauf</p> <p>■ □ □</p>	<p>Projektklima</p> <p>■ □ □</p>	<p>Termine</p> <p>■ □ □</p>	<p>Risiken</p> <p>■ □ □</p>	<p>Ressourcen</p> <p>■ □ □</p>
<p>Tendenz</p>	<p>↗</p>	<p>↗</p>	<p>↗</p>	<p>↗</p>	<p>↗</p>

<p>Aktueller Projektstand</p> <ul style="list-style-type: none"> • Leassons learned und soll Ist Vergleich ist Abgeschlossen. 	<p>Was läuft gut?</p> <ul style="list-style-type: none"> • Die Arbeiten sind so weit abgeschlossen <p>Was läuft nicht gut?</p> <ul style="list-style-type: none"> • Der Projektverlauf ist auf guten Weg
---	--

<p>Geplante nächste Schritte / getroffene Massnahmen</p> <ul style="list-style-type: none"> • Arbeiten an der Formatierung / Darstellung der Arbeit • Arbeit durchlesen lassen für die Verständlichkeit und Rechtschreibfehler

4.2 Lessons Learned

Was habe ich unterschätzt

Der Programmiereteil für eine Projektarbeit habe ich unterschätzt. Es wurde viel komplexer als vor Beginn der Arbeit von mir erwartet.

Was würde ich anders machen

Das Risiko, das die Diplomarbeit nicht verwirklicht werden kann mit den EMG-Sensoren war gross, so ein Risiko würde ich nicht mehr eingehen, sondern die Arbeit so anlegen das ich sicher bin das es ein Erfolg wird.

Was habe ich gut gemacht

Ein Frühzeitiges auseinandersetzen mit dem Thema der Fahrzeugsteuerung und der EMG-Sensoren ermöglichte es die Projektziele so umsetzen zu können, wie sie definiert wurden. Sich Zeit für Abklärungen zu nehmen hat sich bei der Realisierung gelohnt.

Welche Erkenntnisse konnte ich gewinnen

Einschlägig ist die Erkenntnis, dass es eine klare Struktur für ein Projekt braucht. Ab einer gewissen Grösse ist es nicht möglich ein Projekt, ohne eine Projektstruktur einem Projektablaufplan oder einem klar definiertem Projektumfang zu erstellen. Desto grösser oder komplexer ein Projekt wird desto wichtiger ist es mit Projektmanagement Instrumenten zu arbeiten. Damit man erfolgreich sein kann.

Eine andere Erkenntnis ist das Arbeiten mit Leistungsstarken Motoren oder anderen Leistungsstarken Komponenten darauf geachtet werden muss das diese unbedingt getrennt von der Steuerung versorgt werden müssen. Auch schon im Leistungsbereich eines Modellfahrzeugs.

Des Weiteren konnte ich im Bereich der Motorensteuerung mittels PWM Signal Erfahrungen sammeln. Im Bereich der EMG-Sensoren konnte ich Wissen aufbauen und mich entwickeln. Das Programmieren von Entwicklerboards ESP32 konnte ich festigen, die Programmiersprache C+ ist mir mittlerweile geläufig.

Was nehme ich in das nächste Projekt mit

Es muss immer eine Zeitreserve vorhanden sein, denn es gibt immer ein Projektteil, bei dem es zu Schwierigkeiten kommt.

Die Anwendung von Projektmanagement ist der Schlüssel zum Erfolg.

Wenn man Zeit in der Vorbereitung und Planung investiert, kann man in der Realisierung umso mehr Zeit einsparen.

Abbildungsverzeichnis

Abbildung 1	12
Abbildung 2	12
Abbildung 3	12
Abbildung 4 Teilaufgabe 1 Nutzwertanalyse.....	18
Abbildung 5 Teilaufgabe 2 Nutzwertanalyse.....	21
Abbildung 6 Teilaufgabe 3 Mindmap.....	22
Abbildung 7 Verdrahtungsschema Steuerung.....	23
Abbildung 8 Foto Dioden Steuerung	24
Abbildung 9 Verdrahtungsschema Fahrzeug	25
Abbildung 10 Legende Fahrzeug.....	25
Abbildung 11 Foto Fahrzeug.....	26
Abbildung 12 ESP-NOW	27
Abbildung 13 Code Mac Adresse Abbildung 14 Code Ausgabe Serial Monitor.....	27
Abbildung 15 Grundlage Steuerung	28
Abbildung 16 Zustandsdiagramm	29
Abbildung 17 Code Steuerung Bibliotheken.....	30
Abbildung 18 Code Steuerung Variablen.....	30
Abbildung 19 Code Steuerung MAC Adresse.....	30
Abbildung 20 Code Steuerung Struktur.....	31
Abbildung 21 Code Steuerung Rückmeldung	31
Abbildung 22 Code Steuerung Datenrate.....	31
Abbildung 23 Code Steuerung Verbinden	32
Abbildung 24 Code Steuerung Signalverarbeitung.....	32
Abbildung 25 Code Steuerung Signalausgabe	33
Abbildung 26 Code Steuerung Werte verarbeiten	34
Abbildung 27 Code Steuerung Signal senden	35
Abbildung 28 Code Fahrzeug Bibliotheken.....	39
Abbildung 29 Code Fahrzeug erstellen Struktur.....	39
Abbildung 30 Code Fahrzeug globale Variablen.....	39
Abbildung 31 Code Fahrzeug Funktion Daten erhalten.....	40
Abbildung 32 Code Fahrzeug Ausgänge	40
Abbildung 33 Code Fahrzeug PMW Signal.....	41
Abbildung 34 Code Fahrzeug Datenübertragungsrate	41
Abbildung 35 Code Fahrzeug Initialisierung ESP-NOW	41
Abbildung 36 Code Fahrzeug vorwärts links.....	42
Abbildung 37 Code Fahrzeug vorwärts rechts.....	42
Abbildung 38 Code Fahrzeug rückwärts rechts	43
Abbildung 39 Code Fahrzeug rückwärts links	43
Abbildung 40 Code Fahrzeug vorwärts.....	44
Abbildung 41 Code Fahrzeug rückwärts	44
Abbildung 42 Code Fahrzeug links.....	45
Abbildung 43 Code Fahrzeug rechts	45
Abbildung 44 Code Fahrzeug Stilstand	46

Tabellenverzeichnis

Tabellenverzeichnis 1	Projektauftrag.....	10
Tabellenverzeichnis 2	Zielscheibe	11
Tabellenverzeichnis 3	Projektablaufplan	13
Tabellenverzeichnis 4	Projektstrukturplan	14
Tabellenverzeichnis 5	Teilaufgabe 1 Variante A	15
Tabellenverzeichnis 6	Teilaufgabe 1 Variante B.....	15
Tabellenverzeichnis 7	Teilaufgabe 1 Variante C.....	16
Tabellenverzeichnis 8	Teilaufgabe 1 Variante D	16
Tabellenverzeichnis 9	Teilaufgabe 1 Variante E.....	17
Tabellenverzeichnis 10	Teilaufgabe 2 Variante A	19
Tabellenverzeichnis 11	Teilaufgabe 2 Variante B.....	19
Tabellenverzeichnis 12	Teilaufgabe 2 Variante C.....	19
Tabellenverzeichnis 13	Teilaufgabe 2 Variante D	20
Tabellenverzeichnis 14	Teilaufgabe 2 Variante E.....	20
Tabellenverzeichnis 15	Ganzer Code Steuerung.....	38
Tabellenverzeichnis 16	Programmiercode Fahrzeug komplett	50
Tabellenverzeichnis 17	Testbericht.....	53
Tabellenverzeichnis 18	Ablaufplan Soll / Ist Vergleich.....	54
Tabellenverzeichnis 19	Zielscheibe Soll / Ist Vergleich	55
Tabellenverzeichnis 20	Statusbericht 1.....	56

Literaturverzeichnis

Wikipedia	https://de.wikipedia.org/wiki/Elektromyografie
Espressif	https://www.espressif.com/en/products/software/esp-now/overview
Github	https://github.com/topics/esp-now
Funduino	https://fanduino.de/anleitung