Inhaltsverzeichnis

[1. Einleitung 2](#_Toc208402940)

[1.1. Inhalt und Aufbau 2](#_Toc208402941)

[1.2. Ziel der Arbeit 2](#_Toc208402942)

[2. Technische Grundlagen 3](#_Toc208402943)

[2.1. SPI 3](#_Toc208402944)

[2.2. I²C 3](#_Toc208402945)

[2.3. CAN 3](#_Toc208402946)

[2.4. UART 3](#_Toc208402947)

[2.5. Wifi 3](#_Toc208402948)

[2.6. Bluetooth 3](#_Toc208402949)

[3. Systemanalyse und Integration 4](#_Toc208402950)

[3.1. Bestehendes System 4](#_Toc208402951)

[3.2. Anforderungen 8](#_Toc208402952)

[3.3. Integration ins bestehende System 14](#_Toc208402953)

[4. Konzepte 16](#_Toc208402954)

[4.1. Paarvergleiche 16](#_Toc208402955)

[4.2. Nutzwertanalysen 19](#_Toc208402956)

[5. Hardwareentwicklung 20](#_Toc208402957)

[5.1. Schaltungsaufbau 20](#_Toc208402958)

[5.2. PCB-Layout 42](#_Toc208402959)

[6. Software 43](#_Toc208402960)

[6.1. Konzept 43](#_Toc208402961)

[6.2. Aufbau 43](#_Toc208402962)

[6.3. Umsetzung 43](#_Toc208402963)

[7. Test und Inbetriebnahme 44](#_Toc208402964)

[8. Fazit und Ausblick 45](#_Toc208402965)

[8.1. Zusammenfassung der Arbeit 45](#_Toc208402966)

[8.2. Mögliche Weiterentwicklungen 45](#_Toc208402967)

[9. Literaturverzeichnis 46](#_Toc208402968)

[10. Bildverzeichnis 47](#_Toc208402969)

# Einleitung

## Inhalt und Aufbau

## Ziel der Arbeit

# Technische Grundlagen

## SPI

## I²C

## CAN

## UART

## Wifi

## Bluetooth

# Systemanalyse und Integration

## Bestehendes System

Bei dem bestehenden System handelt es sich um ein vollelektrisches Rennsport-Kart für den Einsatz im Jugend Kart Slalom Wettbewerb und Training. Das Kart soll eine Alternative zu vorherrschenden Karts mit Verbrennungsmotor sein und bietet dafür einige Vorteile.

Das Kart ist ausgestattet mit einem kleinen Synchronmotor mit einer Leistung bis 7KW sowie 55 Nm. Diese Leistungsparameter werden dabei auf sehr kleinem Bauraum erzielt, sodass der Motor zentral platziert werden kann, was dem Fahrverhalten sehr zugute kommt. Die Kraftübertragung auf die Hinterachse findet dabei mittels eines Reimens statt, sodass der gesamte Aufbau auch mechanisch simpel bleibt. Angesteuert wird der Motor über CAN, sodass kein Großer Verkabelungsaufwand für Signalleitungen besteht. Versorgt wird das gesamte System von einer 48V Batterie.   
Die Steuerung des Gesamtsystem übernimmt eine Zentrale Steuereinheit (Vehicle Control Unit - VCU). Auf diesem läuft der Code für Leistungsregelung, Kommunikation und Auswertung aller Sensoren. Hierüber ist es möglich das Fahrverhalten des Karts zu steuern über Parameter wie Verteilung des Drehmoments oder die Veränderung von Drehzahlkurven. Hier kommt der große Vorteil des Elektroantriebs zum Tragen, da hierüber verschiedene Leistungsstufen definiert werden können. Wo vorher unterschiedliche Verbrennungsmotoren eingesetzt werden mussten, um unterschiedliche Leistungsklassen zu definieren, ist bei unserem Kart lediglich Softwareänderung notwendig, um jede Leistungsstufe zwischen 3 und 7 KW zu ermöglichen. Über den Code wird zudem sichergestellt, dass über den gesamten Entladezyklus des Akkus dieselbe gleichbleibende Leistung zur Verfügung gestellt wird. Das ist für den Wettbewerb unerlässlich, da hier alle Teilnehmer mit denselben Karts antreten. Die VCU übernimmt zudem sämtliche Kommunikation mit zum Beispiel dem Akku und Auswertung aller Signale wie von den Knöpfen der zentralen Steuereinheit.

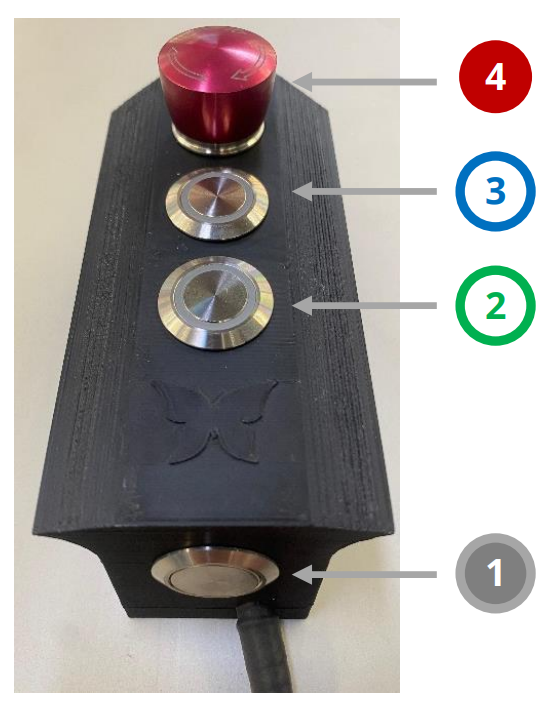
Die Steuerung des Karts erfolgt über eine Zentrale Steuereinheit. Diese besteht aus 3 Knöpfen mit LED-Anzeige und einem Notausknopf. Ein Knopf (Knopf Nr. 1) fungiert als Hauptschalter für das Gesamtsystem. Über ihn wird dem Akku signalisiert den Ausgang aktiv zu schalten und das System mit 48V zu versorgen. Knopf 2 aktiviert den Fahrmodus. Die Aktivierung des Fahrmodus ist dabei an eine Sicherheitsüberprüfung des Systems gekoppelt ohne welche ein Start nicht möglich ist. Dasselbe gilt für den Knopf Nr. 3, jedoch aktiviert dieser den Rückwärtsfahrmodus. Hauptsächlich unterscheiden sich die Knöpfe durch die LED- welche dem Fahrer und Kunden Rückmeldung über den Zustand des Karts geben soll. So zeigen verschiedene Kombinationen aus Leuchtenden und blinkenden LED verschiedene Sati des Karts an wie den Aktivierten Fahrmodus, eine entladene Batterie oder den ausgelösten Notausknopf. Dieser Notausknopf ist das Hauptsicherheitsfeature des Karts. Dabei handelt es sich um einen Normally Closed Druckknopf mit Verriegelung, mit welchem der Fahrer Im Falle eines Notfalls das Kart jederzeit ausschalten kann. Dabei wird nicht nur softwareseitig der Fahrmodus beendet, sondern tatsächlich der Ausgang der Batterie passiv geschalten, sodass das Fahrzeug auch elektrisch in einen sicheren Zustand übergeht, zum Beispiel nach einem Crash, bei dem unter Umständen Kabel gerissen sind oder Kontakte offen liegen. Aus diesem Grund ist der Notaus auch Kabelbruchsicher realisiert, das Bedeutet das sowohl das Drücken des Knopfes als auch eine zerstörte Signalleitung die Aktivierung des Notaus zur Folge hat. Zusätzlich ist diese Steuereinheit mit einem Buzzer ausgestattet, welcher dem Kunden und Fahrer akustische Rückmeldung bei jeder Zustandsänderung des Karts liefert, zum Beispiel bei Aktivierung des Fahrmodus oder einem ausgelösten Notaus.

Abbildung : Steuereinheit SMS Revo SL

Zusätzlich zu der Grundfunktion des Karts zum Einsatz im Slalomsport gibt es ein Erweiterungspaket, die Option 1, welche das Kart je nach Wunsch um verschiedene Funktionen erweitert. Die Hauptfunktionalität ist ein RFID-Reader. Gemeinsam mit dem Kauf und Einbau der Option 1 erhält der Kunde RFID-Chips, auf welchen Unterschiedliche Leistungsmodi gespeichert sind. Durch Auslesen des jeweiligen Chips erhält das Steuergerät einen neuen Leistungswert und schaltet in den entsprechenden Modus um. So ist es für den Kunden möglich die Karts in unterschiedlichen Altersgruppen mit unterschiedlichen Leistungsklassen einzusetzen, ohne dass zwischendurch ein Servicetechniker, das Kart flashen muss oder der Kunde für jede Stufe ein eigenes Kart braucht. Der RFID-Reader besitzt dabei ein eigenes CAN-Interface und schickt die übertragenen Daten direkt per CAN2 an die VCU.   
Eine weitere Funktionalität ist eine Statusleuchte. Diese Statusleuchte wird direkt mit dem Steuergerät verbunden, da diese direkt die LED schaltet. Die LED hat dabei verschiedene Funktionalitäten zu erfüllen. Zum einen gibt sie Rückmeldung über verschiedene Stati des Karts. So kann beispielsweise eine State of Charge Abfrage durchgeführt werden. Die VCU gibt dabei den SOC in Form von Lichtimpulsen zurück. Befindet sich der SOC bei Abfrage zwischen 0% und 25% blinkt die LED einmal bis hin zu viermal, wenn sich der Ladestand zwischen 75% und 100% befindet.  
Des Weiteren ist die Status-LED auch teil eines Sicherheitsfeatures. Die LED beginnt zu blinken, sobald und solange das Kart im Fahrbereiten Zustand ist, also der Startknopf gedrückt wurde, aber das Kart noch nicht in Bewegung ist. Das dient zur Visualisierung für Streckenhelfer und Zuschauer, um zu verhindern, dass Leute verletzt werden, da das Kart unerwartet losfährt. Zu guter Letzt zeigt die LED auch den eingestellten Fahrmodus an, sobald dieser geändert wurde. Das dient dazu, um Sabotage und Betrug im Wettbewerb zu verhindern. Die Änderung soll für jeden so sichtbar wie möglich werden, um zu verhindern, dass einzelne Teilnehmer den Fahrmodus verändern und sich einen Vorteil verschaffen.   
Eine weitere Funktionalität der Option1 ist eine Funkfernbedienung, um einzelne Funktionalitäten von außen Steuern zu können. Dazu gehören die Aktivierung des Notaus und das Starten des Karts, also die Aktivierung des Ready-To-Drive Status. Mittels Tastendrucks auf der Fernbedienung kann der Notaus aktiviert werden, was vor allem bei jüngeren Fahrern zur Sicherheit beiträgt. Das Funksignal unterbricht dabei denselben Signalpfad wie der Knopf am Kart. Ein anderer Knopf kann den Ready-To-Drive Modus aktivieren, hier wird lediglich ein Signal an die VCU gesendet, welches dieselbe Funktionalität wie der Knopf am Kart erzeugt. Der Vorteil liegt darin, dass vor allem jüngere Fahrer unterstützt werden können, sollten sie im Wettbewerb nervös werden oder selbst nicht wissen, wie das Kart gestartet wird Der dritte Knopf an der Fernbedienung sendet ein Signal an die VCU, welches die Abfrage und Anzeige des SOC auslöst, welcher anschließend über die LED dargestellt wird. Die letzte taste wird zur Identifikation des Karts verwendet, zu welchem die Fernbedienung gehört. Hierbei leuchtet die Status-LED auf, solange der Knopf gedrückt bleibt. Alle diese Funktionalitäten werden über ein Zugekauftes Funkmodul realisiert, welches aus einem Funkempfänger und 4 Relais auf einer Platine besteht. Die Funktionalitäten der Relais sind dabei vordefiniert. Kanal 1 verhält sich wie ein Flip-Flop und ändert seinen Zustand bei jedem Tastendruck, speichert also seinen Zustand, weshalb es für die Funktionalität des Notaus genutzt wird. Das Relais unterbricht dabei einen kabelbruchsicheren Signalpfad, welche daraufhin als Notaus detektiert wird. Die Relais für Kanal 2 und 3 speichern ihren Zustand nicht und schalten ihren Zustand lediglich für kurze Zeit, sollte eine Funksignal empfangen worden sein. Sie realisieren deshalb Funktionalitäten, für die nur ein kurzes Signal an das Steuergerät gesendet werden soll. Das Relais zieht dabei lediglich eine Signalleitung auf 12V-Pegel, welches direkt mit der VCU verkabelt ist. Dadurch wird dort ein Signal erkannt. Für Kanal 4 ist ein bistabiles Relais auf dem Modul vorhanden, welches zwei Unterschiedliche Zustände annehmen kann, von denen einer immer aktiv ist. Es Verbindet die Status-LED standartmäßig mit dem LED-Signal, welches von der VCU an das Erweiterungssystem geschickt wird. Für die Dauer des Knopfdrucks schaltet das Relais um und aktiviert ein konstantes Leuchten der Status-LED.

## Anforderungen

Die Aufgabe meines Projekts besteht darin alle Erweiterungsoptionen, welche für das Kart verfügbar sind in einem kompakten und günstigen System zu vereinen. Bisher waren alle diese Komponenten voneinander unabhängige Systeme, welche separat zugekauft werden müssen und durch viel Verkabelungsaufwand verbunden und angebunden werden. Dadurch war Vorbereitung kaum möglich, der Verkabelungsaufwand enorm und unübersichtlich, Fehlersuche erschwert.   
Ziel ist Verkabelungsaufwand zu verringern und damit das System Stabiler, günstiger und weniger Fehleranfällig zu gestalten. Es soll möglich sein, sowohl Kabelbaum und Erweiterungsplatine unabhängig voneinander vorbereiten und testen zu können. Dabei sollen alle bisherigen Funktionen erhalten bleiben. Zusätzlich soll das neue System Schnittstellen schaffen, welche sowohl dem Kunden und Fahrer neue Informationen geben sollen als auch für den Entwicklungs- und Wartungstechniker neue Schnittstellen schaffen, um Probleme schnell und einfacher beheben zu können, sowie Anpassungen und Updates einfacher aufspielen zu können. Neben dem Beibehalten der alten Funktionen soll auch die Kompatibilität zu bereits gebauten und verkauften Karts beibehalten werden, um auch diese nachrüsten zu können, ohne Änderungen am Kabelbaum des Karts vornehmen zu müssen.

Aufgabe der Erweiterung bleibt weiterhin die Funktionalität eines RFID-Readers. Die Ausgelesenen Daten werden dabei sowohl auf der Erweiterungsplatine als auch per CAN in der VCU benötigt. Auf der RFID -Karte müssen dafür verschiedene Daten gespeichert sein. Die RFID-Karten werden weiterhin zur Einstellung des vordefinierten Leistungsmodus verwendet. Hierfür wird auf den Karten die ID des Leistungsmodus gespeichert und per CAN an die VCU übermittelt. Um zu verhindern, dass jeder Kunde die Einstellungen aller Karts verändern und so unter Umständen Wettbewerbe verfälschen kann, wird zusätzlich zum Modus auch eine Kundenspezifische Nummer auf der karte gespeichert und übertragen. Mit dieser Nummer, welche per CAN an das Steuergerät übermittelt werden soll, kann validiert werden, dass die Chips und das Kart zum selben Kunden gehören, sodass jeder Kunde die Chips nur an seine eigenen Karts einsetzen kann. Zusätzlich soll die Kundennummer auch auf der Erweiterungsplatine ausgelesen werden. Eine wichtige Anforderung an den RFID-Reader ist eine hohe Verschlüsselungsstufe. Da die Karts auch im Wettbewerb eingesetzt werden, darf es nicht möglich sein RFID-Chips zu kopieren, zu fälschen oder zu manipulieren.

Mit den bestehenden Funktionen gehört eine Status-Leuchte weiterhin zu den Anforderungen. Diese Statusleuchte muss sowohl von der VCU als auch dem Funkempfänger, welcher auf meiner neuen Platine platziert sein wird, steuerbar sein. Die LED muss in verschiedenen Situationen steuerbar sein:

* Ready-To-Drive-Status:   
  Die LED blinkt kontinuierlich, sobald das Kart in Fahrbereitem Zustand ist, solange das Kart steht. Das Steuersignal dafür sendet die VCU.
* SOC-Abfrage:   
  Die LED blinkt 1- bis 4-mal, je nach aktuellem SOC-Stand. Die Berechnung des SOCs und Übermittlung des LED-Signals übernimmt die VCU
* Identifizierung:  
  Die LED leuchtet kontinuierlich, solange per Funkfernbedienung die Identifizierung des Karts abgefragt wird. Das Signal wird vom Funkempfänger ausgegeben

Um den Aktuellen Verkabelungsaufwand zu reduzieren, soll das Signal nicht mehr als Digitalsignal über eine Leitung direkt vom Steuergerät aus übermittelt werden, sondern als CAN-Message an die Erweiterungsplatine gesendet werden. Um Abwärtskompatibilität beizubehalten, soll zusätzlich die Möglichkeit einer Physischen Übertragung und Auswertung auf der Platine vorgesehen werden.

Eine weitere Funktion, welche beibehalten werden soll, ist die Möglichkeit einige Funktionen des Karts mittels einer Funkfernbedienung zu steuern.   
Dabei soll es Möglich sein folgende vier Funktionen mittels der Fernbedienung steuern zu können:

* Notaus - Funktionalität:  
  Mittels einer Taste muss der Notaus ausgelöst werden, welcher sofort die Stromversorgung für das Kart trennt und es zum Stillstand bringt. Die Funktionalität des Notaus ist dabei bereits vorhanden, die Aufgabe der Option 1 besteht lediglich darin ein Signal 12V Signal auszugeben, welches den Notaus aktivieren kann. Zur Sicherheit muss dieses Signal Kabelbruchsicher realisiert werden und soll nicht Softwaregesteuert durch den Mikrocontroller sein.
* Remote Drive - Funktionalität:

Remote Drive bezeichnet das Signal, welches das Kart in den Fahrbereiten Zustand (Ready to Drive) versetzt, in welchem der Fahrer jederzeit in der Lage ist, loszufahren. Die Aktivierung und damit verbundenen Sicherheitschecks übernimmt vollständig die VCU, Aufgabe des Zusatzsystems ist es lediglich ein Signal an das Steuergerät zu senden. Um Verkabelungsaufwand zu minimieren, soll dieses Signal nicht mehr über eine eigene Leitung im kabelbaum realisiert werden, sondern als Signal auf dem CAN-Bus an die VCU gesendet werden. Um die Möglichkeit zu haben alte Karts nachzurüsten, soll aber zusätzlich die Möglichkeit vorgesehen werden, das Signal auch weiterhin über eine eigene physische Leitung an das Steuergerät zu übermitteln, da die alten Kabelbäume keine CAN-Leitungen zur Option 1 führen.

* SOC-Abfrage:  
  Die dritte Funktionalität, welche per Funkfernbedienung zur Verfügung gestellt werden soll, ist die Abfrage des State of Charge. Mit Drücken des Knopfes an der Fernbedienung soll die Status-LED zwischen ein und vier-mal, abhängig vom aktuellen State of Charge, blinken. Die Abfrage des SOC und Berechnung der Anzahl der LED-Signale übernimmt dabei die VCU. Diese Funktionalität soll weiterhin die VCU übernehmen, obwohl der SOC für andere Funktionalitäten bereits an die VCU übermittelt wird, um die Kompatibilität zu alten Kabelbäumen zu wahren, welche noch Keine CAN-Verbindung zur Option 1 vorgesehen haben. Aus demselben Grund soll das Signal auch weiterhin zur Übertragung über eine eigene Leitung zum Steuergerät vorgesehen werden. Primär soll das Signal aber per CAN übermittelt werden, um die Kabelanzahl zu reduzieren und damit Kosten und Aufwand zu verringern.
* Identifikation:  
  Eine weitere Funktionalität der Funkfernbedienung soll es sein, dass Kart zu identifizieren. Hier soll die Status-LED leuchten, solange die entsprechende Taste auf der Fernbedienung gedrückt gehalten wird.

Ziel soll dabei auch sein die Kosten für die Erweiterung zu minimieren und sich von Zukaufteilen zu lösen. Durch das Zugekaufte Modul waren die Funktionen kaum anpassbar du wenig flexibel. Auch die Notwendigkeit einer eigenen Signalleitung für jedes Signal trug zu viel verkabelungsaufwand und höheren Kosten bei. Das neue System soll dahingehend einige Verbesserungen schaffen. So soll es auch möglich sein, dass die Zuweisung einer Fernbedienung auch nach dem Einbau der Option 1 möglich ist. Bisher musste eine Fernbedienung per Tastendruck auf dem Empfängermodul verbunden werden, bevor das Modul verbaut werden konnte und musste somit für den gesamten Fertigungs- und Verkabelungsprozess bei dem System bleiben. Außerdem war es nur mit großem Aufwand möglich defekte Fernbedienungen zu ersetzen oder Ersatz für verlorene Fernbedienungen zu verbinden. Für das Neue System soll dieser Verbindungsvorgang softwareseitig ausgelöst werden können, um eine Fernbedienung auch nachträglich ohne großen Aufwand verbinden zu können. Trotz allem soll für Testzwecke oder Sonderwünsche weiterhin eine Aktivierung des Verbindungsmodus per Knopfdruck auf der Platine vorgesehen werden.

Neben diesen bereits in alten variante bereitgestellten Funktionen soll die Erweiterungsplatine auch einige neue Funktionalitäten und Schnittstellen zur Verfügung stellen. Eine dieser Funktionen ist eine kabellose Datenverbindung. Dabei soll es möglich sein, sich sowohl mit Mobilen Endgeräten wie Handy oder Tablet, aber auch mit Laptops mit dem Kart zu verbinden, um verschiedene Funktionalitäten dieser Schnittstelle nutzen zu können:

* Livedaten:  
  Die Hauptfunktionalität, welche die kabellose Schnittstelle bieten soll, ist die Zur Verfügungstellung von Livedaten. Sowohl der Kunde als auch ein Wartungstechniker sollen Livedaten vom Kart empfangen können, welche in einer GUI leicht verständlich und übersichtlich aufbereitet werden. Hierfür ist eine zuverlässige und schnelle Verbindung nötig. Zudem müssen die Daten per CAN empfangen und ausgewertet werden.
* Kunden - Einstellungen:  
  Neben dem Einsehen von Livedaten soll es zudem möglich sein einfache Einstellungen, welche keine neuen Funktionalitäten erlauben oder verbieten, für den Kunden selbst vorzunehmen. Hierbei ist es wichtig, dass diese Einstellungen persistent gespeichert werden, damit die Daten auch nach einem Neustart des Karts erhalten bleiben. Zudem muss optisch dargestellt werden können, welche Einstellungen gerade aktiv oder deaktiviert sind.
* Hersteller – Einstellungen:  
  Neben den Einstellungen, die der Kunde selbst an seinem Kart vornehmen kann, soll es zusätzlich einige Einstellungen geben, welche nur der Hersteller vornehmen kann. Hintergrund ist, dass in jedes Kart dieselbe Platine verbaut und identische Software aufgespielt wird. Je nach Bedarf und Wunsch des Kunden werden nun die Funktionalitäten freigeschalten oder gesperrt. Dadurch sollen hohe kosten durch unterschiedliche Systeme und großer Aufwand beim Auseinanderhalten der Karts mit unterschiedlichen Systemen vermieden werden. Zusätzlich ist es dadurch sehr einfach nachträglich weitere Features freizuschalten, ohne Hardware oder Software am Kart verändern zu müssen. Um zu verhindern das Kunden auf Funktionalitäten Zugriff erhalten, welche nicht gekauft wurden müssen die Einstellungsmöglichkeiten Zugriffsgeschützt werden.

Der Verbindungsaufbau soll dabei so einfach wie möglich sein, muss gleichzeitig aber trotzdem sicher genug sein, dass nur der Besitzer des Karts Verbindung aufbauen kann. Für den Einsatz im Wettbewerb muss es außerdem möglich sein Zugriff auf das Netzwerk zu verhindern, da oft ein Kartclub die Karts für die gesamte Veranstaltung zur Verfügung stellt. Hier darf es nicht möglich sein die Einstellungen während der Veranstaltung zu manipulieren oder Livedaten anderer Fahrer auszulesen.

Eine weitere Schnittstelle für Kunde und Techniker wird über ein Display zur Verfügung gestellt. Dieses Display soll mehrere Seiten besitzen, zwischen welchen einfach gewechselt werden kann. Auch hier steht die Darstellung von Livedaten im Vordergrund, welche auch für den Fahrer live während der Fahrt sichtbar sind. Hier sollen vor allem Daten wie Temperatur und Ladestand angezeigt werden. Dabei ist es wichtig die Daten grafisch und Farblich aufzubereiten, um es dem Fahrer einfacher zu machen, die Werte wahrzunehmen. Des Weiteren sollen auch einzelne Funktionen über das Display veränderbar sein. Aufgrund der Fehlenden Eingabemöglichkeiten soll es sich dabei nur um boolesche Variablen handeln, die verändert werden können. Die Einstellungen sollen einfach und übersichtlich sein. Da es sehr ablenkend für vor allem junge Fahrer sein kann, soll es die Funktion geben, dass sich das Display während der Fahrt ausschaltet. Die Daten sollen wieder sichtbar werden, sobald das Kart zum Stillstand kommt. Neben den Livedaten und den Einstellungen sollen hier auch Fehlercodes angezeigt werden, die während dem Initialisierungsprozess aufgetreten sind und indizieren, dass einzelne Funktionalitäten nicht zur Verfügung stehen.

Zusätzlich soll eine Logdateien Funktionalität vorgesehen werden. Das System soll sowohl Fehler speichern, welche im Betrieb der Erweiterungsplatine auftreten, als auch Fehler im Betrieb der VCU speichern, welche per CAN an die VCU übermittelt werden. Die Speicherung erfolgt dabei mit Timestamp und Fehlercode. Der Fehlerspeicher soll dabei einfach auslesbar sein, ohne sich physisch mit dem System verbinden zu müssen.

Grundsätzlich soll es dabei möglich sein alle Funktionen unabhängig voneinander nutzen zu können, sodass sich jeder Kunde ein eigenes Paket aus Leistungen zusammenstellen kann. Damit muss es auch möglich sein alle Funktionalitäten einzeln zu deaktivieren oder freizuschalten.

## Integration ins bestehende System

Einer der wichtigsten Punkte bei der Entwicklung des neuen Systems ist Kompatibilität mit dem Bereits vorhandenen Gesamtsystem. Dazu gehört, dass sich an anderen Systemen so wenig wie möglich ändern soll. Das beinhaltet sowohl mechanische Anpassungen als auch minimale Anpassungen in der Software oder dem Kabelbaum. Ziel bleibt dabei trotzdem weiterhin möglichst viele Verbesserungen in Bezug auf Kosten und Arbeitsaufwand zu erzielen.   
Vor allem mechanisch soll sich nichts am bisherigen Kart ändern, zum einen, da die Option 1 feste Anbindungspunkte am Kart nutzt, welche sich nur mit sehr großem konstruktionsaufwand für das gesamte Kart verändern lassen, und diesen vorhandenen Platz an den Anbindungspunkte bereits vollständig ausnutzt, zum anderen soll das neue System aber auch in bereits gebauten Karts ohne großen Aufwand nachgerüstet werden können. Diese mechanischen Vorgaben definieren damit die Größe und Form der Platine zu einem sehr großen Teil. Auch die Position der Stecker ist durch die Kabeldurchführungen im Gehäuse sehr beschränkt. Für die Erweiterung mit einem Display soll es ein eigenes Gehäuse geben, sodass für das Display das bestehende Gehäuse nicht bearbeitet werden muss sondern lediglich die Kabel herausgeführt werden müssen, wofür die bereits vorhandenen Kabeldurchführungen genutzt werden können.   
Auch für den Kabelbaum sollen möglichst wenig Änderungen nötig werden. Da das System bisher aus mehreren losen verkabelten Komponenten bestand gab es nur einen Stecker mit deutlich weniger Pins als bisher. Dieser 6Pin Stecker wird für die Anbindung des RFID-readers und der Statusleuchte übernommen, da bei diesem Stecker die losen, abisolierten Enden der Kabel einzeln in den Stecker auf der Platine gequetscht werden ohne das es einen Gegenstecker gibt, der vorbereitet werden muss. Das sorgt dafür, dass der RFID-reader und die Statusleuchte weiterhin wie vorher vorbereitet werden können und auch bereits vorbereitete Komponenten weiterverwendet werden können. Für alle Leitungen, die von der VCU zur Option1 geführt werden, wird ein neuer 14-Pin Stecker vorgesehen. Die Steckverbindung besteht dabei aus 2 Bauteilen mit Crimp Kontakten als Pins. Durch die Trennung von externen Bauteilen und Kabelbaumseite kann weiterhin der kabelbaum separat vorbereitet werden ohne auf optimale Komponenten angewiesen zu sein. Die Verwendung eines neuen Steckers wird durch die höhere Anzahl an Kabeln notwendig. Für die Daten, welche auf Display und Webseite angezeigt werden sollen, muss zusätzlich Can1 zur Erweiterungsplatine geführt werden. CAN1 wird zusätzlich auch für die VCU-Befehle benötigt, welche über die Funkverbindung erhalten werden. Für alte Karts, die keine CAN1 Verbindung vorgesehen haben, sind zusätzlich noch die PINs zur Übertragung per Kabel vorgesehen, sodass zumindest die Funktionalität der Funkfernbedienung erhalten bleibt und die alten Karts ohne Anpassungen die Funktionalitäten des bisherigen Erweiterungspakets ohne Einschränkungen nutzen können, sollte aus Reparaturgründen die Option1 ausgetauscht werden müssen. Für Livedaten und Telemetrie müsste der Kabelbaum jedoch nachgerüstet werden.  
Auch beim Steuergerät steht es im Vordergrund, dass alte Systeme weiterhin zuverlässig laufen, sollte das alte System durch eine neue Platine ausgetauscht werden müssen. So soll sowohl die Auswertung von Digitalsignalen auf der Option1 möglich sein, welche von der VCU gesendet werden, aber auch die Option1 sieht Signale weiterhin vor, sodass sie sowohl als CAN-Signal oder über eine eigene Leitung an die VCU gesendet werden können. Auch die Struktur von Signalen soll gleichbleiben, um in der Auswertung keine Anpassungen vornehmen zu müssen. So werden die Daten, welche per RFID-Reader ausgelesen werden, weiterhin im Selben Aufbau und mit denselben IDs auf CAN2 an die VCU gesendet. Einige Änderungen müssen jedoch vorgenommen werden, diese beschränken sich jedoch auf das Senden und Empfangen von CAN-Nachrichten und deren Verarbeitung, sodass lediglich Erweiterungen, aber keine grundsätzlichen Änderungen in der Funktionsweise der Steuergerätsoftware nötig sind.

# Konzepte

Um diese Anforderungen an das neue System möglichst gut zu erfüllen kann es mehrere Möglichkeiten geben, welche jeweils besondere Aspekte eines Problems oder einer Anforderung besonders erfüllen können, oder aber auch auf Ausgewogenheit abzielen, um jede Anforderung möglichst gleich gut abzudecken. Um eine fundierte Entscheidung für eines dieser verschiedenen Konzepte zu treffen, ist es daher nötig, Anforderungen möglichst spezifisch zu definieren und zu gewichten, aber auch Konzepte möglichst detailreich zu beschreiben und Stärken und Schwächen einzelner Möglichkeiten gezielt herauszuarbeiten. Diese Vorarbeit bietet die Möglichkeit auf einer fundierten Entscheidungsgrundlage die Konzepte zu vergleichen und die bestmögliche Lösung zu identifizieren und eine begründete Entscheidung für eine der Umsetzungsmöglichkeiten für das System zu treffen. Für die Durchführung dieser Gewichtungen, Vergleiche und Abwägungen gibt es verschieden Möglichkeiten, welche in den Ingenieurwissenschaften zur Anwendung kommen. Für diese Arbeit wurde sich für Paarvergleiche zur Gewichtung der Anforderungen, sowie für Nutzwertanalysen zum Vergleich der Konzepte auf Grundlage der Gewichtungen der Anforderungen entschieden. Dieses System wurde gewählt, da es sich hierbei um ein sehr einfaches und schnell umsetzbares System handelt, welches sich für kleinere Systeme sehr gut eignet, ohne unübersichtlich zu werden. Zudem bleibt die Entscheidungsfindung hierbei sehr transparent und nachvollziehbar. Im Folgenden sollen die Paarvergleiche und Nutzwertanalysen kurz beschrieben und erklärt werden. Die Möglichen Konzepte sollen zudem erläutert werden, sowie die Entscheidungen, die getroffen wurden, erklärt werden.

## Paarvergleiche

Beim Paarvergleich geht es darum Anforderungen gegeneinander abzuwägen und zu priorisieren. Der Paarvergleich dient dabei dazu dieses Vorgehen zu vereinfachen und transparenter zu machen, indem man jeden Punkt mit jedem anderen einzeln vergleicht. Am Ende ergibt sich eine sehr genaue Staffelung. Dieses Verfahren soll kurz erklärt werden bevor die Paarvergleiche für jeden der vorher definierten Anforderungspunkte durchgeführt und erklärt werden.

Bei den zu vergleichenden Anforderungen handelt es sich nicht um die vorher definierten Anforderungen an das System, sondern viel mehr sollen für diese Anforderungen verschiedene Konzepte diskutiert werden. Daher werden verschiedene Kriterien erstellt, die für diese Konzepte relevant sein können, aufgrund derer am Ende die Wahl für eines der Konzepte in der Nutzwertanalyse fällt. Dieses Vorgehen soll beispielhaft an dem Paarvergleich gezeigt werden.  
Für den Paarvergleich wurde eine Excelliste erstellt, welche die Berechnungen automatisch durchführt und dem Anwender einfach aufzeigt, welche Informationen von Ihm benötigt werden.  
Ein Bild, das Reihe, Diagramm, Screenshot, parallel enthält.

KI-generierte Inhalte können fehlerhaft sein.

Im Bild kann man die grobe Struktur erkennen. In die gelben Felder sind Eingaben vom Benutzer vorgesehen, in den Restlichen Weißen Feldern werden über Formeln die Eingaben erzeugt. Die Rot Umrandeten Felder sind hier zu einfacheren Erklärung hervorgehoben. In diesen Feldern werden die Kriterien eingetragen, welche miteinander verglichen werden müssen. Um die Ergebnisse durch falsche Eingaben nicht zu verfälschen, muss der Nutzer diese Kriterien nur in die Zeilen Eintragen, in die Spaltenüberschriften werden die Kriterien Namen dann gemäß der Reihenfolge ihrer Eintragungen automatisch kopiert. Damit ergibt sich die Tabelle auf deren Grundlage anschließend die Vergleichsergebnisse eingetragen werden können. In der letzten Spalte sollten zudem Kurze Erläuterungen zu den Kriterien erstellt werden, um es dem Leser oder jemandem, der ebenfalls an dem System arbeiten soll möglichst verständlich zu machen, worauf dieses Kriterium abzielt. Eine Beispielhafte Tabelle könnte damit in etwa so aussehen.

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

KI-generierte Inhalte können fehlerhaft sein.

Auf Grundlage dieser Tabelle können nun in die gelben Felder zwischen den Kriterien die Vergleichsergebnisse eingetragen werden. Wie bei einer Matrix werden die Vergleichsergebnisse in das Feld eingetragen, in welchem Zeile und Spalte mit dem jeweiligen Namen zusammenstoßen. Die Gewichtung wird durch die zahlen 0 bis 2 dargestellt, wobei 2 definiert, dass das Kriterium, welches in der Zeile steht, höher zu gewichten ist als das in der der Spalte. Eine 0 hingegen zeigt das Gegenteil an, in diesem Fall wäre die Spalte höher gewichtet als die Zeile. Sollte es Kriterien geben, welche gleich gewichtet werden sollen, kann eine 1 eingetragen werden. Da es jede Kombination aus Kriterien zweimal gibt, da jedes Kriterium als Zeile und Spalte vorkommt, muss vom Nutzer der Tabelle nur die Gewichtung in der oberen Hälfte eingetragen werden. Der Wert der korrespondierenden Zelle wird anhand der Formel

Berechnet, wobei die vom Nutzer eingetragene Gewichtung ist und die Gewichtung der korrespondierenden Zelle. Daraus ergibt sich, dass eine vom Nutzer eingetragene 2 zwangsläufig zu einer 0 in der korrespondierenden Zelle führt, wodurch genau der gewünschte Effekt erzielt wird.

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

KI-generierte Inhalte können fehlerhaft sein.

Aus den Gewichtungswerten wird nun zeilenweise die Summe gebildet. Dadurch, dass der Wert im Falle einer Höhergewichtung der Zeile gegenüber der Spalte am höchsten ist, ergibt sich nun, dass die Spalte mit der höchsten Summe gegenüber allen anderen Spalten am wichtigsten wiegt. Im oberen Beispiel wäre das die Störungssicherheit. Um nun ein mit anderen Paarvergleichen vergleichbares Ergebnis zu erhalten, werden die Werte auf insgesamt 100% normiert. Das erreicht man, indem man die Summen der einzelnen Zeilen durch die Summe aller Werte teilt. Mit diesen Werten kann nun in einer Nutzwertanalyse zur Bewertung der einzelnen Konzepte genutzt werden. Dieses Verfahren soll nun anhand der realen Konzeptentscheidungen angewendet werden.

## Nutzwertanalysen

blablabla

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

KI-generierte Inhalte können fehlerhaft sein.

# Hardwareentwicklung

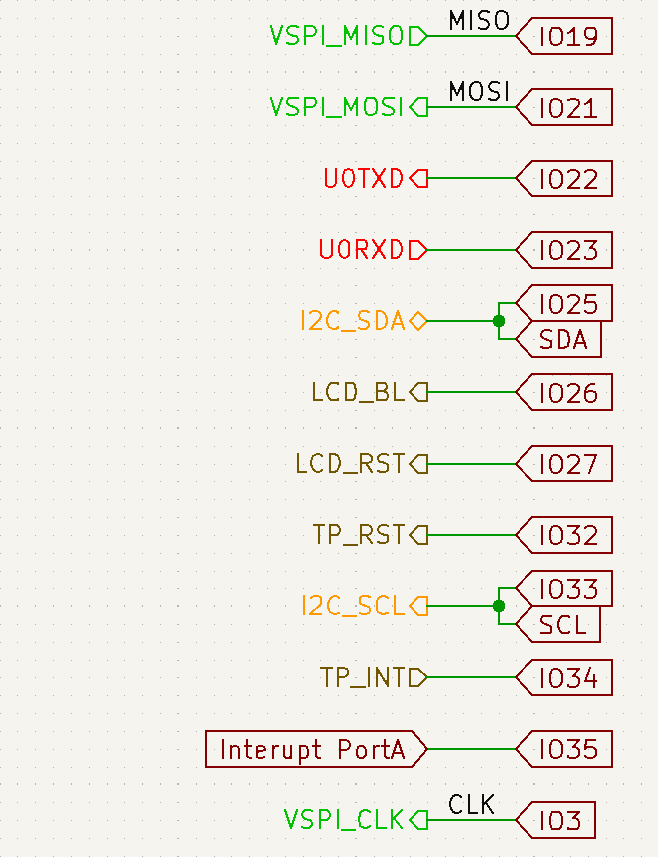
## Schaltungsaufbau

Der erste Entwicklungsschritt für das neue Erweiterungssystem ist der Aufbau geeigneter Schaltungen, welche alle Funktionalitäten abbilden. In diesem Kapitel soll es daher um die Schaltungen, deren Aufbau und Funktionsweiße gehen. Der Schaltungsaufbau ist dabei so organisiert, dass jede Einzelschaltung in einem eigenen Schaltplan realisiert ist und ein Masterblatt diese Systeme alle miteinander verbindet. Dabei wird jede Schaltung einzeln betrachtet und erläutert. Ein Bild, das Text, Diagramm, Plan, parallel enthält.

KI-generierte Inhalte können fehlerhaft sein.   
Mikrocontroller  
Zuallererst soll hierbei der Microcontroller als Zentrales Element der Schaltung betrachtet werden. Dabei kommt ein ESP32-Microcontroller zum Einsatz. Dieser übernimmt sämtliche Rechenoperationen und Kommunikationsaufgaben für das System. Der Mikrocontroller wird dabei mit 3.3V versorgt. Dadurch, dass der 3.3V auch Signalpegel für viele Signale ist, ist es sehr wichtig, ein stabiles und rauschfreies Signal am Microcontroller zu erhalten. Dafür sind die Kondensatoren C801 und C802 zwischen VCC und Ground vorgesehen. Bei ihnen handelt es sich um Abblockkondensatoren, welche Hochfrequente Störungen gegen Ground ableiten und kurzfristige, minimale Spannungseinbrüche abpuffern sollen. Außerdem ist es wichtig sicherzustellen, dass die Stromversorgung beim Einschalten des ESP32 stabil ist, um einen sicheren Boot-Vorgang zu gewährleisten ohne Fehler zu erhalten. Das Einschalten passiert dabei über den EN-Pin des Mikrocontrollers. Hierfür wird ein RC-Glied vor den Enable-Pin geschalten. Das Datenblatt empfiehlt dabei mindestens als Wartezeit, bis die Spannungsversorgung am ESP stabil anliegt, bevor der Chip über EN-Pin aktiviert wird. Die Verzögerung des RC-Glied lässt sich dabei folgendermaßen berechnen

Das Datenblatt empfiehlt für , das entspricht einer Verzögerung von .   
Ein Bild, das Text, Diagramm, Plan, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.  
Zusätzlich beim Bootvorgang zu beachten sind die Strapping Pins des Mikrocontrollers. Bei Strapping-Pins handelt es sich um die Pins, welche den Ablauf des Bootvorgangs definieren. So lässt sich beispielsweise der Boot Mode definieren, ob die Software, welche bereits auf dem Gerät gespeichert ist für den Bootvorgang verwendet wird oder ob extern neue Software aufgespielt werden soll. Die Strapping Pins des ESP32 sind dabei GPIO 0, GPIO 2, GPIO 5, GPIO 12 (MTD1), und GPIO 15 (MTD 0). Für diese Pins ist im Datenblatt eine Standartbelegung definiert, bei welcher der normale Bootvorgang abläuft, wobei der Chip aus seinem eigenen Speicher bootet. Die Pegel werden dabei über Pull-Up und Pull-Down Widerstände erzielt, das sind die Widerstände R801 bis R805. Um diesen Bootvorgang nicht zu beeinflussen, wurde daher bewusst darauf verzichtet, die Strapping Pins für weitere Funktionalitäten zu nutzen. Lediglich die Pins GPIO 0 und 2, welche den Bootmodus definieren, werden noch zusätzlich vom USB-Controller gesteuert, um ein Flashen per USB zu ermöglichen.  
Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.  
Ein großer Vorteil des ESP32 ist, dass beinahe alle Pins für alle Funktionen verwendet werden können. Lediglich die Pins GPIO 34, 35, 36 und 39 sind nur als Input verwendbar und die Pins GPIO 6 bis 11 werden intern verwendet und dürfen nicht verwendet werden. Somit bietet der Chip sehr viele Möglichkeiten für Signale oder Kommunikation. Ein verwendetes Kommunikationsprotokoll ist I²C. Es wird sowohl zur Ansteuerung des Displays als auch für die Datenübertragung an eine GPIO Port Erweiterung verwendet. I²C ist ein Protokoll, welches mit zwei Signalleitungen auskommt. Für Datenübertragung (SDA) ist hierbei Port GPIO 25 vorgesehen, für das Clock-Signal wird Pin GPIO 33 verwendet. Die Auswahl der Pins beruht dabei auf praktischen Gesichtspunkten, so sollen die Ports physisch möglichst nebeneinander liegen, um das spätere Layout zu vereinfachen. Für I²C müssen zudem Pull-Up Widerstände vorgesehen werden, da es sich bei I²C um Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.OpenDrain-System handelt, weswegen die Signale auf einen Stabilen Pegel gezogen werden müssen. Bei Übertragung eines Signals wird die Leitung dann auf Ground Potential gezogen. Bei I²C haben sich 4.7 kΩ als Standartwert für die I²C Kommunikation etabliert.   
Ein weiterer sehr verbreiteter Standard zur Kommunikation mit Peripheriegeräten ist SPI. Bei dem aktuellen System werden damit das Display, ein Flashspeicher, sowie der Controller zur Kommunikation auf CAN-Bus 2 gesteuert. Anders als I²C benötigt SPI mindestens 4 Leitungen. Dazu gehört auch wieder ein Clock-Signal, welches jeden Teilnehmer synchronisiert. Dazu kommen anders als bei I²C aber zwei Datenleitungen. Hier wird unterschieden zwischen gesendeten und empfangenen Nachrichten. Jeder Teilnehmer besitzt ein Master In, Slave Out-Signal (MISO) und ein Master Out, Slave In-Signal (MOSI) über welches die Teilnehmer miteinander kommunizieren können. Hinzu kommt ein ChipSelect Signal für jeden Teilnehmer, der auf dem Bus Kommunizieren möchte. Über dieses Signal muss immer genau ein Teilnehmer vom Master freigegeben werden, welcher dann kommunizieren kann. Auch hier werden die Pins so ausgewählt, dass das Layout möglichst vereinfacht werden kann. Eine weitere verbreitete Kommunikationsart, vor allem zur Kommunikation unter Steuergeräten, ist der CAN-Bus. Der Chip besitzt intern einen Controller für Two-Wire Automotive Interface (TWAI), welcher in der Lage ist eine CAN-Kommunikation aufzubauen. Dabei werden 2 physische Leitungen für CAN-High und CAN-Low benötigt. Diese werden nicht am Mikrocontroller angeschlossen, weil zur Kommunikation noch ein CAN-Transceiver benötigt wird. Dieser erhält über UART seine Daten und Befehle vom TWAI-Controller, dabei kommunizieren die Teilnehmer über zwei Leitungen für erhaltene und gesendet Nachrichten. Daher werden auch zwei Pins am Chip dafür verwendet. Da es sich hierbei um ein asynchrones Protokoll handelt, ist hier kein Clocksignal wie bei SPI oder I²C nötig. Im vorliegenden System kommuniziert darüber sowohl der RFID-Reader als auch die Datenübertragung mit dem Steuergerät findet über CAN-Kommunikation statt. Da der Mikrocontroller selbst leider keine zwei CAN-Busse unterstützt muss für den zweiten Bus ein externer CAN-Controller vorgesehen werden, dieser überträgt seine Daten via SPI. Der Controller für Bus 2 bietet allerdings eine Interrupt-Funktionalität. Bei Empfang einer Nachricht wird über diesen Pin ein Signal ausgegeben, welches am ESP32 als Interrupt detektiert werden kann.   
Die Signale U0TXD und U0RXD gehören ebenfalls zu einer UART-Schnittstelle. Diese wird im vorliegenden Beispiel lediglich zur Kommunikation mit einem extern angeschlossenen USB-Gerät verwendet und soll als Schnittstelle zum flashen eines neuen Programms dienen. Auch hier wird mit zwei Leitungen kommuniziert, wobei jeweils zwischen Receive und Transmit unterschieden wird, also eine Signalleitung, zum Empfangen von Nachrichten und eine zum Senden. Auch hier werden die nebeneinanderliegenden Pins GPIO 22 und 23 genutzt. Die Restlichen Pins werden für die Steuerung des Displays genutzt. Zum einen gibt es einen Pin für die Hintergrundbeleuchtung des Displays. Dafür wird der Pin als Ausgang eines PWM-Generators definiert. Über die Frequenz des Pulsweiten Modulierten Signals wird somit die Helligkeit der Hintergrundbeleuchtung bestimmt. Ein weiterer Pin ist der DC-Pin des Displays. Dieser ist ein standartmäßiger Ausgang für ein Digitalsignal. Dieses Signal wird dafür genutzt, um dem Display mitzuteilen, ob ein Befehl übertragen wird oder ob es gerade Daten zum Darstellen erhält. Die Pins GPIO 26 und 27 werden als Digitaler Ausgang genutzt. Hier werden Signale ausgegeben, welche genutzt werden können, um das Display und den Touch Controller zu resetten. GPIO 34 wird ähnlich wie bei CAN genutzt, um einen Interrupt durch den Touch Controller des Displays auszulösen. Der Pin wird auf High gezogen, wenn der Touch Controller eine Berührung erkennt, und der Pin kann im Chip als Interrupt definiert werden, sodass jede Berührung direkt verarbeitet werden kann. Hierfür wird einer der Pins genutzt, welche nur als Eingang fungieren, da es sich nur um einen Einseitigen Interrupt handelt. Lediglich der Touch Controller darf diesen Pin setzen, niemals der Mikrocontroller, dieser wertet das Signal lediglich aus. Dasselbe gilt auch für den Pin GPIO 35. Über diesen ist der Interrupt-Pin des GPIO-Expanders an den Chip angeschlossen. Dieser erzeugt ein Signal an dem Pin, welcher dann als Interrupt im ESP32 definiert werden kann, sobald sich der Wert an einem der als Input definierten Pins ändert. Somit können auch dort ausgelesene Daten zeitnah vom ESP ausgelesen und verarbeitet werden.   
Die bereits angesprochene GPIO Port Erweiterung hat dabei die Funktion alle übrigen Funktionalitäten und Pins zur Verfügung zu stellen, welche am Mikrocontroller keinen Platz mehr gefunden haben. Die Port Erweiterung stellt allerdings lediglich GPIO-Funktionalität zur Verfügung und kann keine Sonderfunktionalitäten wie SPI oder I²C bereitstellen. Daher sind die Pins am Chip für die Kommunikation verwendet und die meisten GPIO-Funktionalitäten werden über die Erweiterung erzielt. Zudem wurde darauf geachtet keine Zeitkritischen Signale über die Erweiterung zu realisieren, da hier jedes mal per I²C erst Daten abgefragt werden müssen, bevor diese am ESP32 verarbeitet werden können, wodurch die Daten deutlich später zur Verfügung stehen, als wären sie direkt am Chip angeschlossen. Der GPIO-Expander besitzt dabei 2 Register mit jeweils acht Eingängen. Um es übersichtlich zu halten, wurde Register A nur für Eingangssignal und Register B nur für Ausgangssignale verwendet. Das hat außerdem den Vorteil, dass das Signal für Interrupts von Port B ignoriert werden kann, da dort keine Eingehenden Signale erwartet werden, wodurch ein Port mehr für andere Funktionalität direkt am Chip erhalten bleibt. So übernimmt der GPIO-Expander zum Beispiel das Auslesen der vom Funkempfänger gesendeten Signale.   
Ein Bild, das Text, Diagramm, Zahl, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.  
Dazu gehören neben dem Fehlersignal und der Rückgabe des Lernmodus auch die Empfangenen Signale auf Kanal 2, 3 und 4. Durch die Interrupt Funktionalität des Bauteils wird der Mikrocontroller benachrichtigt, sobald sich an diesen Werten etwas ändert. Neben den Signalen liest die Porterweiterung auch das LED-Signal aus, welches von der VCU gesendet wird. Da dieses nur noch in seltenen Fällen verwendet werden wird, nämlich wenn ein altes Kart mit einer neuen Platine aufgerüstet werden soll, hat dieses Signal keine hohe Priorität, weshalb die höhere Bearbeitungszeit, bis es am Mikrocontroller ankommt, vernachlässigbar ist. Gleichzeitig wird über die Erweiterung auch die LED gesteuert. LED\_ON ist dabei das Signal, welches den Transistor für die LED-Schaltet. Auch hier fallen minimal längere Verarbeitungszeiten nicht ins Gewicht. Dasselbe gilt für das Signal Activate\_Lernmodus. Hierrüber wird der Lernmodus des Funkempfängers aktiviert, mit welchem man neue Fernbedienungen einem Funkmodul zuweisen kann. Dieses Signal besteht aus mehreren Impulsen, welche mit Verzögerung gesendet werden, sodass hier genügend Zeit zwischen den Signalen besteht, um einen neuen Impuls über I²C an die Erweiterung zu übermitteln. Zwei weitere Signale, welche über die Port Erweiterung realisiert, werden sind CAN1\_S-Mode und CAN2\_S-Mode. Über diese Signale wird der Silent Mode für die CAN-Transceiver für CAN1 und CAN2 aktiviert. Dieses Signal eignet sich sehr gut, um über die langsamere Expansion realisiert zu werden, da dieses Signal nur sehr selten verändert wird. Bei den Signale TX\_2 und TX\_3 handelt es sich wieder um Signale, welche nur zur Kompatibilität mit alten Systemen vorgesehen sind. TX\_2 realisiert dabei die Aktivierung des Fahrmodus, wenn ein Signal per Funk empfangen wurde, TX\_3 sendet eine SOC-Anfrage zur VCU. Diese Signale kommen aber nur bei Karts alter Generationen zum Einsatz, die keinen CAN-Bus zur Kommunikation zur Option 1 führen, sondern bei denen das Signal noch über eine eigene Signalleitung an die VCU übermittelt wird. Die restlichen beiden Signale sind Steuersignale für einen über SPI gesteuerten RFID-Reader. Diese sind lediglich für testzwecke vorgesehen, um den sehr teuren RFID-Reader, welcher aktuell direkt per CAN angeschlossen ist, zukünftig ersetzen zu können durch ein deutlich günstigeres Bauteil. Deshalb wird hier auch ein Signal wie ein SPI ChipSelect und der Reset für den Controller über den langsamen GPIO-Expander realisiert, obwohl es sich bei Signalen für die SPI-Kommunikation um zeitkritischere Signale handelt. Allerdings werden diese Signale im normalen Betrieb keinerlei Rolle spielen. Um überhaupt Daten übermitteln zu können muss die Adresse des Geräts definiert werden. Das passiert über eine 7 Bit lange Adresse, wobei die letzten Drei Bit über die Eingänge A0 bis A2 definiert werden. Das ist vor allem dann relevant, wenn in einer Schaltung mehrere GPIO-Port Erweiterungen verbaut werden. Im vorliegenden System gibt es keine Überschneidungsgefahr bei den Adressen, weshalb die letzten Adressbits auf 0 0 0 festgelegt werden, definiert durch das Ground-Potenzial. Die Spannungsversorgung erfolgt auch hier mit 3.3V. Auch hier ist ein Abblockkondensator vorgesehen, um eine stabilere Spannungs-versorgung zu gewährleisten. Der Widerstand vor dem Reset Pin sorgt dafür, dass das lowaktive Signal auf einem definierten Pegel liegt und die Port Erweiterung dauerhaft aktiv ist, ohne ungewollt zurückgesetzt zu werden.  
Funkempfänger  
Das Funkmodul, welches die Signale für Aktivierung des Notaus, Aktivierung des Fahrmodus oder Abfrage des State of Charge empfangen wird, wird als Aufsteckplatine realisiert. Dafür wird das Empfangs – und Auswertmodul CX-12 R von SVS verwendet. Dieses ist mit den bisher Ein Bild, das Elektronik, Schaltung, Elektronisches Bauteil, Elektrisches Bauelement enthält.

KI-generierte Inhalte können fehlerhaft sein.verwendeten Funkfernbedienungen kompatibel und bietet alle Funktionalitäten, welche für das System benötigt werden. Das Modul besitzt 16 Pins, welche im Folgenden genauer betrachtet werden sollen, sowie die Beschaltung, welche näher beleuchtet und erklärt wird.

Der erste Pin ist für die Aktivierung eines Sleep Modus vorgesehen. Um den Stromverbrauch zu senken, wird nach einiger zeit eine Taktung aktiv. Diese Taktung tritt 10 Minuten nach Einschalten des Moduls oder dem letzten Datenempfang in Kraft. Ist diese Taktung aktiv, treten allerdings Verzögerungen beim Empfang und dem Verarbeiten der Funksignale auf. Diese Latenz kann bis zu einer Sekunde betragen. Über den Pin lassen sich dafür verschiedene Modi aktivieren. Dabei wird unterschieden, ob die Lötbrücken A und B auf der Platine gesetzt sind:

|  |  |  |
| --- | --- | --- |
|  | **Pin 1 Pegel** | |
| GND | VCC / unverbunden |
| **Lötbrücke B**  gesetzt | Taktung nach 10 Minuten seit Datenempfang | Dauerhaft in Taktung |
| **Lötbrücke B**  nicht gesetzt | Dauerbetrieb ohne Taktung | Taktung nach 10 Minuten seit Datenempfang |

Die Lötbrücke A bewirkt, das Pin 1 dauerhaft auf Ground gezogen wird.   
Die Platine wird standartmäßig ohne gesetzte Lötbrücken geliefert. Um den Arbeitsaufwand zu verringern, soll es nicht nötig sein, jede Platine noch einmal händisch bearbeiten zu müssen, um Lötbrücken zu setzen. Daher soll der Modus nur über den Pegel an Pin 1 bestimmt werden und beide Lötbrücken bleiben ungesetzt. Für das aktuelle System soll der Empfänger dauerhaft aktiv bleiben, somit also Pin 1 auf Groundpotential gezogen werden ohne gesetzte Lötbrücke. Das hängt damit zusammen, dass durch den Notaus auch Sicherheitsrelevante Funktionalitäten über den Funkempfänger realisiert werden, für den 1 Sekunde Latenz zu lange als Verarbeitungszeit sind.

Um Signale empfangen zu können, muss es zuerst möglich sein Fernbedienungen mit dem Empfänger zu verbinden. Dafür bietet das Funkmodul mehrere Modi, welche sich über Pin 2 aktivieren lassen. Die Lernmodi werden dabei durch unterschiedliche Impulse auf Ground Potential aktiviert:

|  |  |
| --- | --- |
| **Modus** | **Aktivierungsmethode** |
| Lernmodus I | GND, 1x kurz (<1s) |
| Lernmodus II | GND, 2x kurz (<1s) |
| Lernmodus III | GND, 3x kurz (<1s) |
| Lernmodus IV | GND, 4x kurz (<1s) |
| Löschmodus I | GND, 1x lang (>3s) |
| Löschmodus II | GND, 2x lang (>3s) |

Wenn ein neuer Sender mit dem Empfängermodul verbunden wird, wird dessen Adresse im Funkempfänger gespeichert. Dieser besitzt bis zu 60 Speicherplätze für Senderadressen. Bei Lernmodus I und III wird eine neue Fernbedienung mit dem Modul verbunden. Hierbei können nun über alle Tasten die unterschiedlichen Funktionalitäten von Kanal 1 bis 4 genutzt werden. Der Unterschied zwischen Modus I und III liegt darin, dass bei Modus I eine Bestätigung an den Sender übermittelt wird und der Nutzer somit an der Fernbedienung eine Rückmeldung über ein eventuelles Fehlschlagen der Verbindung erhält. Mit Modus II und IV lässt sich eine einzelne Taste anlernen. Hier wird nicht die gesamte Fernbedienung mit dem Empfangsmodul gekoppelt, sondern lediglich die jeweilige taste. Auch hier unterscheiden sich die beiden Modi nur in der Rückmeldung an den Sender. Die verbleibenden beiden Modi werden genutzt, um bereits angelernte Fernbedienungen wieder zu lösen. Dabei entfernt Modus I einen Eintrag aus der Liste. Dabei handelt es sich um die Adresse des Senders, von welchem das Modul ein Signal empfängt, solange der Löschmodus aktiv ist. Löschmodus II entfernt hingegen alle gespeicherten Einträge aus der Adressen Speicherliste. Auf dem bisherigen Gesamtmodul wurde diese Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.Funktionalität über einen Knopf auf der Platine realisiert. Das hatte zur Folge, dass eine neue Fernbedienung nur im Ausgebauten Zustand verbunden werden konnte. Mit dem neuen System soll das auch über den Mikrocontroller gesteuert werden können und damit eine Aktivierung über die Netzwerkverbindung oder das Display möglich sein. Nichtsdestotrotz ist weiterhin ein Knopf vorgesehen, um das Funkmodul ohne den Chip testen zu können. Bei dem Knopf handelt sich dabei um einen Drucktaster, welcher den Pin auf Ground zieht. Der Widerstand R412 dient dabei als Pull-Up Widerstand, um bei offenem Schalten einen definierten Spannungspegel am Pin zu garantieren. Der Pin ist zusätzlich mit der GPIO Port Erweiterung am ESP 32 verbunden, über welcher nun durch Schalten des GPIO-Ports gegen Ground ebenfalls der Lernmodus aktiviert werden kann.

Rückmeldung über den Aktivierten Lernmodus gibt Pin 3. Bei dem Pin handelt es sich um einen LED-Ausgang, welcher durch unterschiedliche Frequenz unterschiedliche Modi anzeigt.

|  |  |
| --- | --- |
| **Modus** | **Anzeige** |
| Normaler Betrieb | 1x blinken pro empfangenes Signal |
| Lernmodus I | 1x Blinken pro 2 Sekunden |
| Lernmodus II | 2x Blinken pro 2 Sekunden |
| Lernmodus III | 3x blinken pro 2 Sekunden |
| Lernmodus IV | 4x blinken pro 2 Sekunden |
| Löschmodus I | Blinkt dauerhaft |

Diese Rückmeldung soll für Test- und Debugging Zwecke auch visuell über eine LED dargestellt werden, soll aber auch vom Mikrocontroller ausgewertet werden können, um Rückmeldung per Display oder Netzwerk geben zu können, sobald die Modi über den ESP32 aktiviert wurden. Da es sich bei dem Ausgang um ein Digitalsignal bis maximal 3.3V handelt, kann das Signal einfach an einen der Pins der Porterweiterung des Mikrocontrollers angeschlossen werden. Um einen Stabilen High-Pegel für das Lowaktive Signal zu erzeugen, ist R402 als Pull-Up Widerstand dazwischen geschalten. Die LED wird zwischen den Ausgang des Funkmoduls und ein 3.3V Signal geschalten, da es sich bei dem Signal um ein lowaktives handelt. Als Vorwiderstand dient hier R401. Der Widerstandswert ergibt sich dabei aus dem gewünschten Strom, der durch die LED fließen soll:  
Durch die LED fließt damit ein Strom von 5 mA.

Bei Pin 4 handelt es sich um den ersten digitalen Ausgang des Funkempfängers. Dieser Ausgang schaltet auf High, wenn ein Funksignal empfangen wird, ausgelöst durch den Druck der Taste 4 auf einer der verbundenen Fernbedienungen. Der Ausgang bleibt danach so lange auf 3.3V Spannungspegel bis die Taste losgelassen wird, mindestens aber 1.5 Sekunden. Der Pin wird zur Auswertung direkt mit dem Mikrocontroller verbunden. Zusätzlich wird noch eine LED hinzugefügt, welche den Pegel des Ausgangs anzeigt, um das Debugging zu vereinfachen und die Fehlersuche zu erleichtern. Da es sich hier um ein High-Aktives Signal handelt wird die LED gegen Ground geschalten. Auch hier sollen 5 mA Strom durch die LED fließen, weshalb auch hier wieder 220Ω als Vorwiderstand genutzt werden.   
Um digitale Ausgänge handelt es sich auch bei den Pins 5, 8 und 9 für die Kanäle 3, 2 und 1. Diese verhalten sich genau wie der Digitale Ausgang für Kanal 4 mit der Erweiterung, dass die 3 restlichen Kanäle zusätzlich zum ESP32 auch noch einen Transistor schalten, welcher das Signal zusätzlich bei alten Systemen an die VCU übermitteln soll. Bei dem Signal von Kanal 1 fällt zusätzlich noch die Verbindung zum Mikrocontroller weg, da es sich hier um ein Sicherheitsrelevantes Signal handelt, welches ausfallsicher realisiert werden muss und deshalb nicht softwareseitig verarbeitet werden soll.

Ein weiterer Ausgang für empfangene Signale ist ein Analogausgang auf Pin 6. An diesem Pin wird ein Analogwert zwischen 3.3V und 0V mit 10 Bit Auflösung ausgegeben. Für unser System wird dieser Ausgang jedoch nicht benötigt, weshalb er unverbunden bleibt.

Der nächste Pin wird zur Konfiguration des Moduls benötigt. Das Modul erwartet Signale auf einer Frequenz von 433,62 MHz. Sollte diese Frequenz nicht nutzbar oder gestört sein besteht die Möglichkeit auf eine Alternativfrequenz von 434,22 MHz auszuweichen. Die Auswahl dieser Frequenz wird mit dem Pegel an Pin 7 definiert. Liegt der Pin auf GND, arbeitet das Modul auf Frequenz 2, liegt der PEGEL BEI 3.3V oder wird der Pin nicht beschalten, arbeitet der Empfänger mit Frequenz 1. Da bereits ein ähnliches Modul im Einsatz war, welches mit denselben Frequenzen arbeitete, und dort kaum Probleme auftraten, wird die Frequenz einmalig festgelegt. Zu testzwecken soll trotzdem die Möglichkeit bestehen im fehlerfall auf alternative Frequenzen auszuweichen, weshalb ein Widerstand als Pull-Up auf 3.3V und ein Pull-Down Widerstand gegen Ground vorgesehen sind, wobei der Pull-Down Widerstand standartmäßig unbestückt bleibt.

Die folgenden Pin 10 und 11 definieren die Versorgung und logischen Pegel für den Funkempfänger. Pin 10 ist die Versorgungsspannung, welche auch gleichzeitig den logischen High-Pegel definiert, und zwischen 2.0 und 3.6V liegen muss. Als Versorgungsspannung wird daher 3.3V verwendet. Pin 11 definiert das Groundpotential.

Neben den Eingängen, über welche das Modul Signale empfängt, besitzt das Modul auch mit Pin 12 einen Analogeingang. Dieser wird standartmäßig dafür verwendet, um Batteriespannungswerte zu übermitteln, wenn das Modul nicht mit einer konstanten Spannungsquelle verwendet wird. Das Modul sendet entsprechend des anliegenden Spannungspegels ein Signal, welches von der Funkfernbedienung empfangen wird und mittels einer LED auf der Fernbedienung visualisiert wird.

Dabei gelten folgende Grenzwerte:

|  |  |  |
| --- | --- | --- |
| **Spannung U** | **Rückmeldung** | **Bedeutung** |
| U > 1.2V | LED aus | Ausreichende Versorgungsspannung |
| 1.2V > U > 1.1V | LED blinkt | Abnehmende Versorgungsspannung |
| 1.1V > U | LED blinkt schnell | Zu geringe Versorgungsspannung |

Für das vorliegende System ist die Ausgabe eines Analogen Spannungswerts nicht relevant, weshalb der Pin Dauerhaft auf 3.3V gelegt wird, sodass ein durchgehend ausreichendes Spannungssignal erkannt wird und keine Rückmeldung an die Fernbedienung geschieht.

Zusätzlich zu den LEDs als visuelle Rückgabe gibt es auch die Möglichkeit auditive Rückmeldung vom Modul zu erhalten. Dafür ist am Empfänger der Ausgang für eine Piezoscheibe vorgesehen. Dieser Ausgang gibt vordefinierte Signale bei bestimmten Ereignissen aus, welchem dem Nutzer Rückmeldung über den Status des Systems geben. So gibt die Piezo-Scheibe Tonfolgen aus, wenn die Versorgungsspannung unter einen definierten Schwellwert sinkt. Auch die Aktivierung und Deaktivierung des Lernmodus wird mit je einer individuellen Tonfolge pro Modus ausgegeben. Auch das Einlernen und Löschen eines Senders wird mit je einer Tonfolge quittiert, genauso wie das Löschen der gesamten Speicherliste. Da diese Informationen bereits durch Auswertung der LED-Pins durch den Mikroprozessor erhalten werden und dieser die Informationen auf Display oder Netzwerk visuell zur Verfügung stellt, besteht für dieses System kein Bedarf diese Informationen zusätzlich noch über Tonfolgen auszugeben. Daher ist der Ausgang unbeschalten.

Ein weiterer LED-Ausgang, welcher auch vom Mikrocontroller ausgewertet wird, ist Pin 14. Hier handelt sich um eine visuelle Ausgabe von aufgetretenen Fehlern. Diese Fehler werden auch hier durch unterschiedlich viele Lichtimpulse zurückgegeben. Dabei wird unterschieden zwischen normalem Empfangsbetrieb und Aktiviertem Lösch-/Lernmodus:

|  |  |  |
| --- | --- | --- |
| **Normaler Empfangs-betrieb** | **Fehler** | **Darstellung** |
| Senderversorgungsspannung gering | Blinkt 1x |
| Senderversorgungsspannung kritisch | Blinkt 3x |

|  |  |  |
| --- | --- | --- |
| **Lern- / Lösch-modus** | **Fehler** | **Darstellung** |
| Eintrag konnte nicht gelöscht werden | Blinkt 2x |
| Speicherliste ist voll | Blinkt 3x |
| Sender wurde bereits eingelernt | Blinkt 2x |

Zu Test und Debugging Zwecken ist auch hier weiterhin eine LED zur visuellen Rückmeldung vorgesehen, um die Fehler auch ohne Mikrocontrollerauswertung schnell und einfach identifizieren zu können. Auch hier wird ein gewünschter Strom von 5 mA vorgegeben, sodass sich auch hierwieder ein Vorwieder stand von berechnen lässt. Auch hier handelt es sich wieder um ein Lowaktives Signal, sodass die LED zwischen 3.3V und Pin geschalten werden muss, um bei Aktivierung des Signalpins zu leuchten. Zusätzlich wird der Pin noch mit einem der Eingänge an der GPIO-Porterweiterung des Mikroprozessors verbunden, um die Fehler am ESP32 auswerten und für den Kunden darstellen zu können. Dafür ist auch der Widerstand R407 als Pull-Up vorgesehen, um bei dem Lowaktiven Signal einen definierten High-Pegel zu erzeugen.

Die Pins 15 und 16 sind nun noch als Anschlüsse für eine Antenne vorgesehen. Standartmäßig wird das Modul mit einer einfachen Drahtantenne ausgestattet. Dafür ist Pin 16 vorgesehen. Für die Verwendung einer abgesetzten Antenne lässt sich aber auch ein Koaxialkabel mit einer Impedanz von 50Ω anschließen, in diesem Fall wird Pin 15 für den Anschluss der Schirmung und Pin 16 für den Anschluss des Innenleiters verwendet. Für die Verwendung einer Drahtantenne gibt das Datenblatt eine Länge von 173mm vor. Diese berechnet sich für eine Monopolantenne nach folgender Formel:

VCU-Signale

Ein Bild, das Text, Diagramm, Schrift, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.Wie bereits weiter oben beschrieben sollen die ersten drei Kanäle des Funkmoduls auch ein Signal an die VCU senden. Zu diesem Zweck gibt es drei Transistorschaltungen auf der Platine, welche jeweils ein Signal erzeugen, welches anschließend über eine eigene Signalleitung an die VCU übertragen wird. Diese Signale sind allerdings nur noch aus Kompatibilitätsgründen für ältere Systeme vorgesehen, welche noch keine CAN-Verbindung zur Option1 vorgesehen haben. Einzige Ausnahme bildet der Notaus, welcher über die Funkfernbedienung aktiviert werden kann. Dieser muss aus Sicherheitsgründen Ausfallsicher realisiert werden, weshalb er nicht über den Mikrocontroller geführt werden darf. Dieses Signal ist weiterhin als eigene Leitung von der Option 1 zur VCU vorgesehen. Die Schaltung ist dafür ebenfalls drahtbruchsicher realisiert. Das bedeutet, dass der Notaus sowohl bei einem erkannten Funksignal als auch einer defekten Leitung als aktiviert erkannt wird, sodass ein Schaden am Kabel nicht zu einem Ausfall des Sicherheitssystems führt. Das wird erreicht, indem als Zustand „Notaus aktiviert“ der Stromlose zustand definiert wird. Im Idle Zustand fließt somit dauerhaft ein geringer Strom. Ein Abfall dieses Stromflusses führt zur Erkennung eines aktivierten Notaus. In der Schaltung wird dies durch einen Transistor als Low-Side Schalter realisiert. Das Signal Notaus\_VCU, welches durch über den Stecker zur VCU geführt wird, wird durch den Widerstand R902 dauerhaft auf 12V gezogen. Somit wird im Zustand „Notaus nicht aktiv“ dauerhaft ein 12V Signal erkannt. Liegt nun am Eingang TX\_1, welcher direkt mit dem Pin am Funkempfänger Modul verbunden ist, ein logischer High-Pegel an, Schaltet der Transistor und zieht den Ausgang gegen Ground. Am Steuergerät wird nun Kein Pegel mehr erkannt, was zum Auslösen der Notausprozedur führt. Das ist dasselbe Verhalten, als würde das Kabel beschädigt und den Kontakt öffnen, obwohl der Transistor nicht durchgeschalten ist. Zur Einfacheren Fehlersuche und zu Debugging Zwecken wurde zusätzlich noch eine LED hinzugefügt. Die LED soll leuchten sobald der Notaus durch den Transistor aktiviert wurde, der Ausgang also auf Groundpotential liegt. Die LED wird deshalb zwischen 12V und Ausgang geschalten. Zusätzlich wird ein Vorwiderstand hinzugefügt, um den Strom durch die LED auf 5mA zu begrenzen.   
Durch Verwendung und Verfügbarkeit stehen nur Widerstände der E12 Reihe zur Auswahl. Dort ist der nächste Widerstandswert 1.8 kΩ.  
Damit ergibt sich für den Strom

Bei dem Transistor handelt es sich um einen N-kanal MosFET, welcher als Schalter eingesetzt wird. Der Transistor schaltet wenn , wobei die Threshold Spannung bei 1V liegt. Dadurch, dass Source dauerhaft auf Ground liegt, gilt immer , damit gilt immer

somit schaltet der Transistor wenn gilt

Diese Bedingung ist mit TX\_1 auf 3.3V erfüllt, sodass der Ausgang auf Ground gezogen wird, sobald ein Signal auf Kanal 1 vom Funkempfänger erkannt wird. Der Gatewiderstand R901 ist dabei vorhanden, um die Gateströme zu begrenzen und das Funkmodul zu schützen.

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.Sehr ähnlich zu der Transistorschaltung für das Notaussignal sind auch die Schaltungen für das RemoteDrive-Signal und die SOC-Abfrage. Anders als das Notaussignal müssen diese Signale aber nicht drahtbruchsicher realisiert werden. Stattdessen liegt der Ausgang RemoteDrive\_VCU standartmäßig auf Ground und wird beim Schalten des MosFET auf 12V gezogen. Der Widerstand R1002 dient hier also als Pull-Down Widerstand. Am Schalten des MosFET hat sich im Vergleich zur Schaltung für das Notaus Signal nichts verändert. Dadurch das der Transistor gedreht ist und Source weiterhin auf Ground liegt, schaltet der Transistor, sobald die 3.3V als High-Pegel am Gate des Transistors anliegen. Auch hier soll eine LED hinzugefügt werden, welche optisch den Pegel des Ausgangs anzeigen soll. Diese wurde hier ebenfalls so eingebaut, das sie leuchtet, sobald der Transistor geschalten hat. In diesem Fall liegt der Ausgang auf 12V, sodass die LED zwischen Ground und Ausgang platziert werden muss. Auch wurden als Vorwiderstand 1.8kΩ gewählt, da sich an den Bauteilen zur Notaus-Schaltung nichts verändert hat. Unterschiedlich zur Notaus-Schaltung ist jedoch, dass das Eingangssignal TX\_2 nicht direkt von dem Funkempfänger kommt, sondern vorher vom Mikrocontroller ausgelesen wurde, welche jetzt den Transistor steuert. Das hat den Zweck, dass über Software steuerbar ist, ob die Signale am Ausgang gesetzt werden sollen, da diese nur als Backup für alte Systeme vorgesehen sind. Für die Schaltung zur Erzeugung des State of Charge Signals gelten dieselben Aussagen, da die Schaltungen zur Erzeugung von RemoteDrive-Signal und SOC-Abfrage identisch sind.

CAN-Transceiver

Statt die Signale jeweils über ein eines Datenkabel an die VCU zu übermitteln, sollen sie zukünftig als CAN-Signal übermittelt werden. Dafür benötigt der Mikroprozessor jedoch Zusatzbeschaltungen. Zwar besitzt der ESP32 intern einen TWAI-Controller, für die Kommunikation auf dem CAN-Bus ist jedoch zusätzlich ein CAN-Transceiver notwendig, welcher als Interface zwischen dem CAN-Controller und dem physikalischen CAN-Bus fungiert. Dafür wird hier der TJA1051T-3 als Bauteil eingesetzt. Der Can-Transceiver übernimmt dabei die Umsetzung der vom Controller erzeugten CAN-Frames auf physikalische Spannungspegel auf dem Bus. Andersherum wandelt er genauso erhaltene Buspegel in empfangene Nachrichten um, sodass die Frames schließlich vom CAN-Controller ausgewertet werden können. Damit das Funktioniert muss der Transceiver jedoch auch richtig beschalten werden. Das Bauteil erwartet eine Versorgungsspannung zwischen 4.5V und 5.5V, wodurch sich ein Spannungspegel von 5V hervorragend anbietet. Da das restliche System jedoch auf einem Logischen Spannungslevel von 3.3V arbeitet wird hier die Bauteilvariante „T-3“ eingesetzt. Diese besitzt einen weiteren Spannungseingang , mit welchem der die Versorgungsspannung für den I/O Level Adapter festgelegt wird, also die Definition des Logikpegels für die Eingangssignale. Dieser wird auf 3.3V gelegt und definiert für den CAN-Transceiver somit denselben Logikpegel wie für die restliche Schaltung. Die beiden Kondensatoren C1702 und C1703 sind hierbei wieder als Abblockkondensatoren vorgesehen. Sie sollen Spannungsspitzen abblocken und so das Bauteil schützen und gleichzeitig kurzfristige Spannungseinbrüche abdämpfen. Ein weiter Kontrolleingang ist der Pin S. Über diesen kann der Silent Mode für den CAN-Transceiver aktiviert werden. Im Silent-Mode ist der Sender deaktiviert, die Bus-Pins gehen über in einen rezessiven Zustand. Alle anderen IC-Funktionen, einschließlich des Empfängers, arbeiten weiterhin wie im Normal-Mode. Der Silent-Mode kann verwendet werden, um zu verhindern, dass ein fehlerhafter CAN-Controller Ein Bild, das Text, Diagramm, Screenshot, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.die gesamte Netzwerkkommunikation stört.   
Die restlichen 4 Pins dienen der tatsächlichen Datenübertragung. Bei den Pins TXD und RXD handelt es sich um die serielle Kommunikation mit dem CAN-Controller. Von diesem erhält der Transceiver fertige CAN-Frames und sendet seinerseits wieder empfangene Bussignale, welche er in einen CAN-Frame umgewandelt hat. Die tatsächliche Buskommunikation findet über die Pins CANH und CANL statt. Diese bilden die physikalischen CAN High und CAN Low Leitungen ab. Der Transceiver wandelt die Nachrichten um und sendet diese als dominante und rezessive Pegel auf den Bus. Zwischen CANH und CANL findet sich noch ein Widerstand. Dieser dient als Terminierungswiderstand für den Bus. Er ist an beiden Enden des Busses notwendig, um Störungen zu minimieren und Reflexionen zu vermeiden. Im CAN-Protokoll ist die Impedanz des Busses auf 120Ω definiert, dadurch ergibt sich auch der Widerstandswert. Durch die Verwendung von 2 CAN-Bussen für Daten und den RFID-Reader findet sich diese Schaltung zweimal auf der Platine. Der ESP32 besitzt jedoch nur einen CAN-Controller, weshalb die Kommunikation auf dem zweiten CAN-Bus durch einen Externen Controller realisiert werden muss. Dafür wird die oben gezeigte und erklärte Schaltung um einen CAN-Controller erweitert. Als CAN-Controller kommt hierbei der MCP2515-I/ST zum Einsatz. Dieser bietet die selbe Funktionalität wie der interne TWAI-Controller, mit dem Unterschied, dass die Kommunikation über SPI stattfindet. Die Daten, aus denen der Controller einen CAN-Frame bauen soll, werden dabei per SPI übermittelt und der Controller übernimmt dieselben Aufgaben wie der TWAI-Controller. Dazu gehören das Erstellen der CAN-Frames aus den übermittelten Daten, ID und den berechneten Kontrollfeldern wie CRC oder DLC und die relevanten Bits wie Acknowledge oder End of Frame. Zusätzlich übernimmt der Controller Aufgaben wie Arbitrierung und die damit verbundenen Steuerung, wann die Daten tatsächlich auf den Bus gesendet werden können. Auch die Auswertung der empfangenen Frames übernimmt der Controller. Diese Daten werden wiederum an den Microcontroller übermittelt, welcher die Abarbeitung der Daten im Programmcode durchführen kann. Die SPI-Kommunikation findet dabei über die Pins SO, SI und SCK statt. Bei SCK handelt es sich um das Clocksignal, welches grundlegend für die Synchrone Kommunikation zwischen Master und Slave ist. SO ist das „Slave Out“ Signal, welches im restlichen System als MISO verwendet wird. Über dieses Signal kommuniziert der Slave und sendet seine Nachrichten und Daten an den ESP32. Über den Pin SI kommuniziert der ESP32 mit dem CAN-Controller. Hier empfängt der Slave seine Daten, welche ihm vom Master über die MOSI-Signalleitung übermittelt werden. Damit die Kommunikation zustande kommt muss vorher der CAN-Controller über den ChipSelect ausgewählt werden. Das passiert über den CS-Pin am CAN-Controller. Wird dieser gegen Ground gezogen, wird der Controller informiert, dass er nun mit dem Master kommunizieren soll. Zusätzlich bietet der SPI-Controller im CAN-Controller eine Interuptfunktionalität über den Pin . Über diesen Pin wird ein Signal an den ESP32 gesendet, sobald eine Nachricht empfangen wurde oder ein Fehler aufgetreten ist. Dieser Pin kann im Mikrocontroller als Auslöser für Ein Bild, das Text, Diagramm, Zahl, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.einen Interrupt definiert werden, sodass eingehende Nachrichten Zeitnah bearbeitet werden können. Versorgt wird der Controller mit 3.3V. Auch hier wird mit C306 ein Abblockkondensator zum Schutz des Bauteils und der Sicherstellung der Spannungsversorgung hinzugefügt. Der Reset-Pin wird mit einem Pull-Up Widerstand auf High gezogen, sodass er bei jedem Neustart automatisch aktiviert und bei Shutdown resettet wird.   
Neben dem Interrupt Pin vom SPI-Controller gibt es auch Interrupt Pins, welche direkt vom CAN-Controller selbst gesteuert werden. So indizieren die Pins RX0BF und RX1BF eine empfangene Nachricht im Receive Buffer RXB0 oder RXB1. Auch diese können am ESP32 als Interruptquelle definiert werden. Da bereits die Interruptfunktionalität über SPI genutzt wird, wird die direkte Interrupt Möglichkeit aus dem Buffer des Controllers nicht benötigt und die Pins bleiben unbeschalten. Dasselbe gilt für die Pins TX0RTS, TX1RTS und TX2RTS. Diese Signale bieten die Möglichkeit ein Request To Send Signal direkt an den Buffer im Controller zu senden, woraufhin der Controller seine im Buffer gespeicherten Frames an den Transceiver übermitteln würde. Diese Funktionalität wird ebenfalls über Befehle via SPI realisiert, weswegen die Pins nicht benötigt werden. Als nächstes sollen die Funktionen der Pins OSC1 und OSC2 betrachtet werden. Das Datenblatt empfiehlt für eine Stabile Kommunikation einen externen Oszillator als System Clock für den CAN-Controller. OSC1 ist dabei der Input für den Timer, OSC2 der Output, welcher zum Oszillator zurückgeführt wird. Der ausgewählte Oszillator arbeitet dabei auf einer Frequenz von 8 MHz. Das Datenblatt gibt dabei für den Oszillator die Beschaltung vor mit welcher das Signal möglichst stabil wird und die Start-Up Zeit, welche vom Taktgeber abhängt, möglichst gering bleibt. Die empfohlenen Werte für einen 8MHz Oszillator sind dabei mit 22pF angeben, woraus sich auch die Kondensatoren in der Schaltung ergeben. Das Clock Signal kann zusätzlich über den Ausgang CLKOUTSOF abgegriffen werden. Dieser Ausgang stellt ein Clocksignal zur Verfügung, welches mittels eines Prescalers verändert werden kann, um als Software Timer für genutzt werden zu können. Die Pins TX und RX übernehmen bei dem Controller nun die tatsächliche Kommunikation mit dem Transceiver. Hier werden die tatsächlichen Frames an den CAN-transceiver übermittelt, welcher anschließend die Kommunikation auf dem CAN-Bus übernimmt. TX dient dabei dazu die Frames an den Transceiver zu übermitteln, RX empfängt die Daten des Transceivers.

LED-Treiber

Ein Bild, das Text, Screenshot, Schrift, Diagramm enthält.

KI-generierte Inhalte können fehlerhaft sein.Da die Pins des Mikrocontrollers nicht genug Strom liefern und das LED-Modul einen Spannungspegel von 12 V benötigt, wird eine Transistorschaltung als LED-Treiber notwendig. Der Treiber wird hier als Low-Side Switch realisiert, welcher den negativen Anschluss der LEDs gegen Ground verbindet. Als Schalter wird hier ein MosFET eingesetzt, welcher vom ESP32 angesteuert wird, wenn die LED leuchten soll. Die positive Seite der LEDs liegt dabei dauerhaft auf 12V, die Negative Seite der LEDs wird auf Ground gezogen, sobald der Transistor durchschaltet. Der Gatewiderstand ist vorhanden, um den Pin am Mikrocontroller vor Stromspitzen zu schützen. Auch hier gilt dieselbe Rechnung wie bei den Transistorschaltungen für die VCU-Signale. Dadurch, dass Source dauerhaft auf Ground liegt, schaltet der Transistor sobald gilt. Das ist der Fall, sobald der Pin am ESP32 das Gate auf 3.3V zieht.

LED-Signal Auswertung

Ein Bild, das Text, Schrift, Reihe, Diagramm enthält.

KI-generierte Inhalte können fehlerhaft sein.Neben den Signalen, welche für veraltete Systeme noch per Signalleitung an die VCU übermittelt werden, übermitteln diese Karts auch die LED-Signale von der VCU noch mittels einer eigenen Signalleitung. Diese Signale müssen ausgewertet werden, um an die LED-Treiber Schaltung weitergegeben zu werden. Da es sich um ein digitales 12V Signal handelt, reicht es das Signal auf maximal 3.3V zu normieren, sodass das Signal direkt an einen der Pins des ESP32 angeschlossen werden kann. Das Signal vom Steuergerät kommt über den Eingang VCU\_LED direkt vom Stecker. Anschließend wird das Signal von 12V mittels eines Spannungsteilers auf maximal 3.3V am Pin normiert. Die Spannung am Ausgang LED\_Signal, welcher mit dem Pin der GPIO-Erweiterung vom Mikrocontroller verbunden ist, lässt sich folgendermaßen berechnen:  
Daraus ergeben sich die Formel für

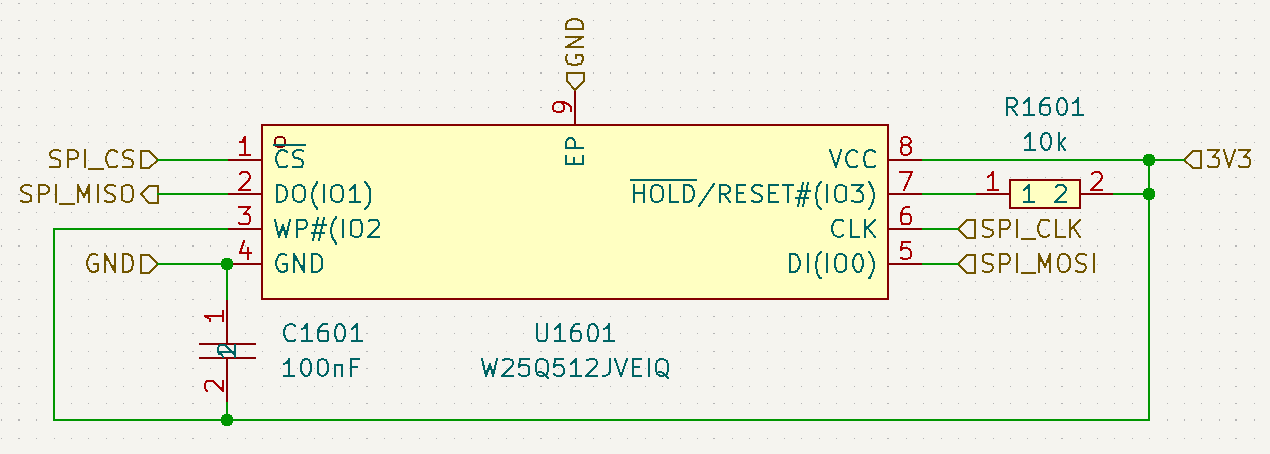
Für ergibt sich damit für

Durch die E-Reihe ergibt sich damit der nächste Wert als 27kΩ. Mit 27kΩ beträgt die maximale Spannung am ESP32-Pin

Damit erkennt der ESP32 an seinem Pin 0V, wenn kein Signal gesendet wird und 3.24V als High Pegel.  
Zum Schutz des Mikrocontrollers ist zusätzlich noch eine Zenerdiode zwischen den Pin und Ground platziert. Die Zenerdiode hat eine Durchbruchspannung bei 3.3V. Das bedeutet, sobald die Spannung am Mikrocontroller 3.3V überschreitet, wird die Diode leitend und hält den Pin bei 3.3V und schützt den Controller so vor Schäden durch Überspannung.

SPI-Flash

Um die Funktionalität eines persistenten Speichers für Logdaten und Einstellungen zu erreichen, ist auf der Platine ein Flash Speicher vorgesehen. Dieser Speicher ist per SPI-Kommunikation adressierbar und beschreibbar. Der Speicher ist dabei persistent, das bedeutet das er seinen Inhalt behält, selbst wenn er nicht mehr bestromt wird. Die Daten sind auch beim nächsten Start-up erhalten. Die Größe des Speichers beträgt 4 MB, für Speicherung von Textdateien und einzelnen Integerwerten ist das völlig ausreichend.   
Der Speicher wird dafür mit 3.3V versorgt. Wie bei allen anderen Bauteilen ist auch hier der Kondensator C1601 als Abblockkondensator vorgesehen. Zusätzlich zum Groundpin GND besitzt der SPI-Flash noch ein großes Lötpad EP unter dem Bauteil, welches ebenfalls auf Ground gezogen wird.



Die SPI-Kommunikation findet bei dem Speicher über die Pins DO und DI statt. DO steht dabei für „Data Out“ und wird daher mit dem MISO-Signal verbunden, dass der Speicher darüber Daten an den Mikroprozessor senden kann. Der Daten empfang über die MOSI-Signalleitung findet am „Data In“ (DI) Pin statt. Hierüber empfängt der Flash befehle und Daten, welche er persistent speichern soll. Als Taktgeber für die SPI-Kommunikation dient der CLK-Pin. Hier erhält der Chip das Clock-Signal vom SPI-Bus. Das letzte SPI-Signal ist der Chip Select für den Speicher. Über diesen wird dem Speicher mitgeteilt, dass nun mit ihm kommuniziert werden soll. Zusätzlich zu den Pins für die SPI-Kommunikation besitzt der Chip noch zwei weitere externe Steuersignale. Eines davon ist Pin 7 für die HOLD-Funktionalität. Der HOLD-Pin ermöglicht es, das Gerät anzuhalten, während es aktiv ausgewählt ist. Wenn HOLD auf Low gesetzt wird, während Chip Select auf Low ist, dass Gerät also gezielt angesprochen wird, wird der Data Out Pin deaktiviert und Signale an den DI- und CLK-Pins werden ignoriert. Sobald der Pegel am HOLD-Pin wieder auf High gesetzt wird, geht der Chip wieder in normalen Betriebsmodus über und nimmt seine Kommunikation wieder auf. Das ist vor allem dann nützlich, wenn sich mehrere Geräte dieselben SPI-Signale teilen, zum Beispiel bei mehreren parallelen Speicherchips. Da dies hier nicht der Fall ist, wird diese Funktionalität nicht benötigt. Weil es sich bei dem Signal um ein lowaktives handelt, wird der Pin mittels eine s Pull-Up Widerstands dauerhaft auf 3.3V gezogen, um ungewollte Störungen durch einen undefinierten Pegel am Pin zu vermeiden. Ein weiteres Steuersignal ist WriteProtect (WP) an Pin 3. Über dieses Signal lässt sich das Schreiben in das Statusregister des Chips verhindern. Zusammen mit Steuerbits im Chip lässt sich damit der gesamte Speicher oder einzelne Bereiche gezielt vor Überschreiben schützen. Da es für das System keine besonders geschützten Speicherbereiche benötigt, wird auch diese Funktionalität nicht genutzt. Der Pin wird dafür dauerhaft auf High gezogen, da es sich auch hier um ein lowaktives Signal handelt.

Power Supply

Wichtig für das System ist die Erzeugung der verschiedenen Spannungslevel. Die Versorgung der Platine findet mit den 48V direkt aus der Batterie des Karts statt. Mit diesem Pegel kann jedoch kein einziges Bauteil auf der Platine arbeiten, dies benötigen 12V, 5V und hauptsächlich 3.3V. Alle diese Spannungspegel müssen daher auf der Platine selbst erzeugt werden. Besonderer Fokus liegt dabei darauf möglichst stabile Versorgungspannungen zu erzeugen, um vor allem Probleme in der Kommunikation zwischen den Komponenten zu verhindern. Auch die Auswahl der Spannungswandler ist wichtig, da diese nur einen bestimmten Strom liefern können. Der Stromverbrauch ist aber bereits durch die Bauteile definiert. Zudem muss darauf geachtet werden Spannungswandler nicht unnötig zu kaskadieren. Durch Kaskadierungen sinkt der Wirkungsgrad der Schaltungen enorm, wodurch die gleichzeitig die Verlustleistung zunimmt. Das hat deutlich mehr Abwärme zur Folge welche Probleme auf der Platine schaffen kann. Als erstes soll daher der Strom bestimmt werden, den die jeweiligen Spannungswandler treiben können müssen. Dazu werden die Maximalströme der Bauteile aus dem Datenblatt betrachtet, um auch den Worst Case abdecken zu können.

|  |  |  |
| --- | --- | --- |
| **Bauteil** | **Versorgungsspannung** | **Stromverbrauch** |
| CAN-Controller | 3.3V | 10 mA |
| Funkreceiver | 3.3V | 18.2 mA |
| GPIO Port Erweiterung | 3.3V | 125 mA |
| Mikrocontroller | 3.3V | 250 mA |
| USB to Serial Wandler | 3.3V | 12 mA |
| Display | 3.3V | 33 mA |
| SPI-Flash | 3.3V | 40 mA |
| RFID-Reader | 3.3V | 26 mA |
| **Gesamt** |  | **515 mA** |

Alle Bauteile, welche mit 3.3V versorgt werden, benötigen im Worst Case 515 mA. Um auf bereits vorhandene Lagerbestände zurückzugreifen und nicht viele verschiedene Bauteile beschaffen zu müssen, sollen möglichst Bauteile eingesetzt werden, welche bereits in anderen Systemen in der Firma zum Einsatz kamen. Für einen Spannungswandler zu 3.3V gibt es daher zwei mögliche Optionen:

1. **Bauteil P7803-500:**

Bei dem Bauteil handelt es sich um einen DCDC-Wandler, welcher direkt aus dem 48V Eingangssignal ein 3.3V Signal erzeugt. Das hätte den Vorteil, dass unnötige Kaskadierungen vermieden werden, wodurch der Wirkungsgrad steigt. Außerdem ist das Spannungssignal stabiler, da nicht die Störeinflüsse mehrerer Spannungswandler weitergegeben werden können. Jedoch ist dieses Bauteil nicht in der Lage den benötigten Strom zu liefern. Der Ausgangsstrom beträgt maximal 500mA, sodass das Bauteil den Maximalstrom nicht liefern kann, was auf Dauer zur Zerstörung des Moduls oder der Schaltung führt.

1. **Bauteil AN\_SY8089A:**bei dem Bauteil handelt es sich um einenSpannungswandler, welcher die 3.3V aus einem 5V Eingangssignal generiert. Dieses Bauteil hat den Vorteil, dass die Ausgangsspannung meist stabiler und weniger Störanfällig ist, je kleiner der Spannungsunterschied zwischen Eingang und Ausgang ist. Nachteile sind die durch die Kaskadierung von 48->5V und von 5V -> 3.3V bedingten höheren Leistungsverluste und Wärmeabgabe. Jedoch ist dieses Bauteil in der Lage bis zu 2A kontinuierlich an Strom zu liefern. Deshalb wird dieses Bauteil in der Schaltung verwendet.

|  |  |  |
| --- | --- | --- |
| **Bauteil** | **Versorgungsspannung** | **Stromverbrauch** |
| CAN-Transceiver 1 | 5V | 50mA |
| CAN-Transceiver 2 | 5V | 50mA |
| 3.3V Spannungswandler | 5V | 365mA |
| **Gesamt** |  | **465mA** |

Alle Bauteile, die mit 5V versorgt werden, benötigen im Worst Case 470 mA. Der Stromverbrauch des 3.3V Spannungswandlers berechnet sich dabei aus dem Worst Case Stromverbrauch der 3.3V Versorgungsspannung.  
Der Wirkungsgrad η ist dem Datenblatt bei einem Laststrom von 515mA entnommen. Damit ergibt sich ein maximaler Eingangsstrom von 365 mA für den maximalen Stromverbrauch aller 3.3V.   
Auch für die Spannungserzeugung des 5V Signals gibt es wieder unterschiedliche Bauteile, welche zur Auswahl stehen.

1. **N7805-500**

Bei diesem Bauteil handelt es sich um einen DCDC, welcher aus 12V ein 5V Signal erzeugt. Die Verwendung dieses Bauteils würde bedeuten, das zuerst aus 48V ein 12V Signal generiert werden muss, aus welchem dann wiederum ein 5V Signal erzeugt wird. Dieses würde wiederum Grundlage zur Generierung des 3.3V Signals sein. Diese Kaskadierung birgt viele Nachteile und sollte deshalb vermieden werden.

1. **P7805-500**Dieses Bauteil erzeugt die 5V direkt aus dem 48V Eingangssignal. Dadurch wird die Kaskadierung aufgebrochen. Zudem ist der Ausgangsstrom mit 500mA ausreichend für die Versorgung aller Bauteile mit 5V Spannungspegel. Die Wahl fällt daher auf dieses Bauteil.

|  |  |  |
| --- | --- | --- |
| **Bauteil** | **Versorgungsspannung** | **Stromverbrauch** |
| RFID-Reader | 12V | 120 mA |
| Status-LED | 12V | 100 mA |
| **Gesamt** |  | **220 mA** |

Alle Bauteile, die mit 5V versorgt werden, benötigen im Worst Case 470 mA.   
Für den 48V zu 12V DCDC steht nur ein Bauteil zur Verfügung, welches in den Lagerbeständen vorrätig ist. Dieses liefert einen Ausgangsstrom von Damit ist die Stromstärke deutlich ausreichend, um alle Bauteile zu versorgen, welche mit der 12V Spannungsversorgung gespeist werden. Hier kann man Zusätzlich noch erkennen, warum die Wahl für den 5V DCDC auf das Bauteil fällt, welches direkt 48V als Eingangsspannung nutzt. Würde zusätzlich noch der 12V->5V DCDC versorgt werden müssen, betrüge die mindestens benötigt Stromstärke

Das würde den Ausgangsstrom des DCDC um 80 mA überschreiten.  
Als zusätzlichen Schutz für die Schaltung ist zusätzlich noch eine Sicherung hinter dem 12V Pfad eingebaut. Diese Sicherung bleibt leitend bis 750 mA und löst bei 1.5A garantiert aus. Diese Sicherung ist dafür da im Falle eines Bauteilschadens die folgenden Bauteile für dem hohen Kurzschlussstrom des DCDC zu schützen. Für den Eingang wird ebenfalls eine Sicherung vorgesehen, zur Auslegung dieser Sicherung muss jedoch der Eingangstrom bekannt sein. Dieser setzt sich zusammen aus dem Eingangsstrom der beiden DCDC-bauteile für 12V und 5V.

Die ausgewählte Sicherung bleibt leitend bis zu einem Strom von 300mA. Garantiert auslösen wird sie lt. Datenblatt ab einem Strom von 650 mA. Die Sicherung eignet sich damit sehr gut, um als Schutz für die Schaltung bei Kurzschlüssen zu fungieren.

Ein Bild, das Text, Diagramm, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Zusätzlich wird als Schutzbauteil noch eine Schottky Diode hinzugefügt. Diese hat die Aufgabe die Schaltung bei Verpolung zu schützen. Wird die Schottky Diode in Sperrrichtung betrieben, wird sie zerstört, wodurch keine Spannung mehr anliegt, welche die restliche Schaltung beschädigen kann. Die Kondensatoren, welche an den Ein und Ausgängen der Spannungswandler platziert sind, dienen der Glättung der Spannungspegel und sollen Spannungsspitzen dämpfen, um Schäden am Bauteil zu verhindern. Die Werte der Kondensatoren sind durch das Datenblatt festgelegt. Zur einfacheren Fehlersuche wird dem Ausgang noch eine LED hinzugefügt. Diese soll leuchten, wenn der Spannungswandler einen Spannungspegel an seinem Ausgang bereitstellt. Der Vorwiderstand berechnet sich dabei entsprechend dieser Formel:

Gemäß der verwendeten e-12 Widerstandsreihe ist der nächstmögliche Wert 1.8kΩ.   
Die Schaltung für den Spannungswandler von 48V zu 5V verhält sich genau wie die vorherige Schaltung, mit der Ausnahme, dass sich die Werte des einen Kondensators ändern, welche im Datenblatt spezifiziert werden. Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.  
Auch die Parameter der Sicherung bleiben die selben, nur das dieses bauteil für den Einsatz bis 6V zugelassen ist, wärend die vorherige Sicherung im Spannungsausgang bis 13.6V eingesetzt werden durfte. Die Ströme, bei welchen die Sicherung auslöst, bleiben identisch. Zusätzlich ändert sich der Wert des Vorwiderstands der LED.

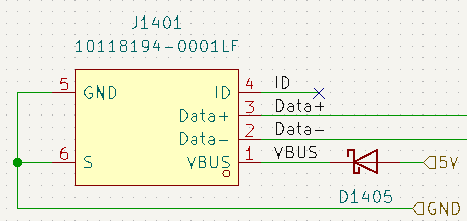
Als letzte Schaltung muss nun noch der Spannungsregler für das 3.3V Signal betrachtet werden. Dieser erzeugt aus dem 5V Eingangssignal ein 3.3V Ausgangssignal. Dieser Spannungsregler eignet sich für mehrere Ausgangsspannungen im Bereich von 1.2V bis 3.3V. Deshalb muss der Chip korrekt beschalten werden, um den korrekten Output zu erzeugen. Die Pins IN und En sind dabei die, welche das Bauteil versorgen. In ist der Spannungseingang, in unserem Fall, dass 5V Signal. Dieses wird auch mit dem EN-Pin verbunden. Dieser aktiviert das Bauteil. Die Festlegung der Ausgangsspannung wird über die Pins FB und LX erreicht. Über FB wird eine Spannung ausgegeben, wobei über den Spannungsteiler die gewünschte Ausgangsspannung erzeugt wird. Über eine Spule wird dieser Spannungswert wieder zum Bauteil zurückgeführt, welches nun auf diese Spannung programmiert wird. Ein Bild, das Text, Diagramm, Screenshot, Reihe enthält.

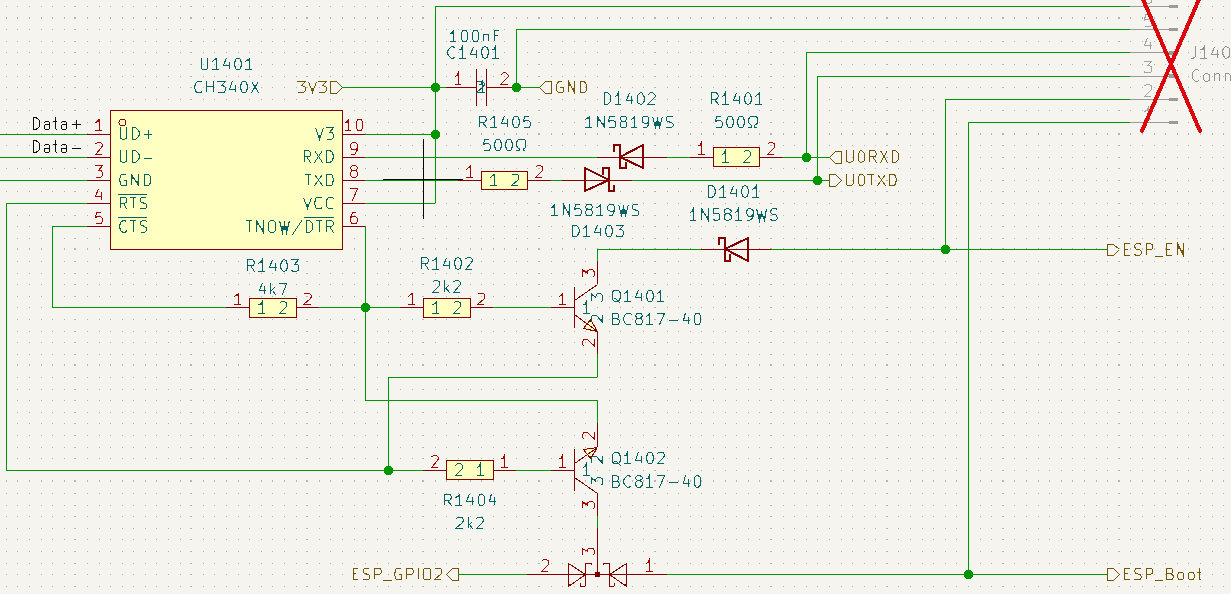
KI-generierte Inhalte können fehlerhaft sein.Die Ausgangsspannung lässt sich dabei mit der Formel

berechnen. Damit gilt für :

Mit den Werten und ergibt sich eine Abweichung von 0.7% für die Spannung.  
Sowohl die Bauteilwerte der Spule als auch die der Kondensatoren sind wiederum vom Datenblatt vorgegeben. Auch hier soll wieder eine LED vorgesehen werden, welche einen Spannungspegel am Ausgang visuell anzeigt. Für den Vorwiderstand lässt sich folgender Wert berechnen:

USB-Schaltung

Um diese ganzen Funktionalitäten nutzen zu können bedarf es verschiedener Schnittstellen. Eine dieser Schnittstellen eine Micro-USB Buchse, über welche es möglich sein soll, den Chip mit neuer Software zu bespielen. Da der ESp32 mit USB nicht direkt arbeiten kann, wird noch ein Wandler benötigt, welcher die USB-Kommunikation in eine UART-Kommunikation umwandelt, über welche der Mikrocontroller kommunizieren kann. Die USB-Buchse besitzt dabei 6 Pins. Diese Pins übertragen alle Signale der Physischen USB-Kommunikation. Die Pins GND und S liegen dabei auf Groundpotential. Bei Pin S handelt es sich um den Schirm das Kabels, welches die Signale vor Störungen schützen soll. Hier ist es wichtig, den Schirm auf dasselbe Potenzial wie den Systemground zu legen, da sonst Störungen unter Umständen sogar verstärkt werden können. Über VBUS lässt die Busspannung abgreifen. Diese beträgt bei USB 5V, da der Host immer die Spannungsversorgung für alle Devices übernimmt. Die Spannung wird hier abgegriffen und auf den 5V Versorgungsspannungspfad geführt, um über USB die Platine bestromen zu können. Normale USB-Ports eines PCs können bis zu maximal 500mA an Strom liefern, das ist vollkommen ausreichend, um die CAN-Transceiver und den Spannungswandler auf 3.3V zu betreiben, da der Spannungswandler von 48V auf 5V auch nicht mehr als 500mA liefern kann. So muss die Platine für Testzwecke nicht immer an eine Externe Spannungsquelle angeschlossen sein, welche die 48V Eingangsspannung liefern kann, sondern kann auch mit einem normalen USB-Netzteil betrieben werden oder beim Flashen direkt vom PC aus. Zwar fallen dabei die 12V Funktionalitäten aus, da diese nicht mit Spannung versorgt werden können, dazu gehören jedoch nur der RFID-Reader, die LED und die VCU-Signale, welche für einen grundlegenden Systemtest vernachlässigbar sind, da es auf der Platine dafür optische Indikation über LEDs gibt, ob die Schaltungen funktionieren. Die Schaltung wird dabei von einer TVS-Diode geschützt. Diese Verhindert Spannungsspitzen, welche zum Beispiel ESD-bedingt auftreten können und schützt so die gesamte Schaltung vor Zerstörung. Die tatsächliche Datenübertragung der Bits findet über die Pins Data+ und Data- statt. Die Bits werden dabei über Zwei Signalpegel übertragen, wobei D+ und D- immer 0V oder 3.3V führen. Für den Ruhezustand liegt D+ auf 3.3V, D- auf 0V. Wird nun eine logische 1 übertragen, bleiben die Pegel unverändert bei einer logischen 0 tauschen die beiden Leitungen ihren Zustand. Dieser Wechsel findet bei jeder übertragenen logischen 0 statt. Die Auswertung der Pegelveränderungen übernimmt dabei der USB to Serial Wandler. Im vorli9egenden System übernimmt das der CH340X. Der USB-Wandler hat dabei 2 Pins für Spannungsversorgung. Über VCC wird der Chip mit Spannung versorgt und über V3 wird die Spannung definiert, mit welcher der Chip versorgt wird. Im Vorliegenden System soll der Chip mit 3.3V versorgt werden, da auch alle restlichen Komponenten auf 3.3V arbeiten. Damit das Bauteil korrekt arbeitet muss dafür der Pin V3 ebenfalls auf 3.3V gezogen werden. Würde der Chip mit 5V versorgt, würde der Pin V3 über einen Abblockkondensator gegen Ground verbunden werden. Die beiden Pins DU+ und DU- sind die beiden Pins, welche mit der USB-Buchse verbunden sind. Über diese erhält der Wandler die Bits, welche über die USB-Leitung übermittelt werden und anschließend umgewandelt werden, um über eine UART-Schnittstelle an den ESP32 gesendet zu werden. Die Übermittlung an den ESP32 geschieht über die Pins RXD und TXD. Über diese beiden Pins findet die asynchrone UART-Kommunikation statt. Die beiden Widerstände sind dabei zum Schutz der Bauteile vorgesehen. So sollen diese im Falle eines Kurzschlusses den Strom begrenzen und so den ESP32 und den USB-Wandler schützen. Gleichzeitig dienen die Widerstände zur Dämpfung von Signalflanken, wodurch Störungen vermindert werden soll. Das ist vor allem bei langen Leitungen relevant. Bei den beiden Dioden handelt es sich um TVS-Dioden. Diese Schützt sowohl den Chip als auch den Mikrocontroller vor Überspannung, indem die Diode bei Überspannung durchbricht und die Spannungen auf einen Maximalwert begrenzt. Die TVS-Diode hat dabei gegenüber einer Zenerdiode den Vorteil, dass sie deutlich schneller arbeitet und damit kürzere und höhere Spannungsspitzen abfangen kann. Diese treten vor allem durch ESD-Probleme auf, welche über das Kabel auftreten können.



Damit die Software, welche über die USB-Schnittstelle übermittelt wird, auch als neue Software für den Mikrocontroller genutzt wird, muss dieser in den Entsprechenden Bootmodus gebracht werden. Dafür müssen GPIO 0 und 2 auf den korrekten Pegel gezogen werden. Die dafür vorgegebenen Pegel sind Beide Pins gegen Ground zu ziehen. Das wird über die Pins RTS und DTR erreicht. Standartmäßig erfüllt der Pin 6 jedoch nicht die Funktionalität von DTR, sondern TNOW. Dieses Signal ist ein Indikator für eine aktuell ausgeführte Übertragung von Daten über die USB-Schnittstelle. Über einen externen Widerstand lässt sich die Funktion des Pins ändern, sodass er nun die Funktion eines DTR-Pins erfüllt. Dieser Pin zeigt den Zustand „Data Terminal Ready“ an und ist lowaktiv. Dafür muss der TNOW-Pin mittels eines 4.7kΩ zu CTS verbunden werden. CTS ist dabei das Signal für “Clear To Send”. Das weitere Signal, welches zur korrekten Aktivierung des Boot-Modus benötigt wird, ist das Signal RTS. Dieses Signal ist ein Ausgang, welcher als Lowaktives Signal ein „Request To Send“ anzeigt. Um einen korrekten Bootvorgang zu gewährleisten sind verschiedene Phasen nötig, damit die Software auch als neues Programm auf den Mikrocontroller aufgespielt wird. Zuerst muss der Enable-Pin am ESP32 auf Ground gezogen werden, um den Reset einzuleiten. Beim darauffolgenden Start-Up müssen die GPIO-Pins 0 und 2 auf den korrekten Pegeln liegen, damit der korrekte Bootmodus ausgeführt wird. Dieser Ablauf wird über die Schaltung mit den beiden Transistoren automatisch erzeugt, ohne dass ein Button auf der Platine benötigt wird, mit welchem händisch ein Bootmodus aktiviert werden muss, wie bei einigen Entwicklungsboards der Fall ist. Die Transistorschaltung ist dabei dem ESP Devkit V4 entnommen, welches mit demselben Chip arbeitet. Durch diese Schaltung werden abhängig DTR und RTS unterschiedliche Pegel an den Pins GPIO 0 und EN erzeugt:

|  |  |  |  |
| --- | --- | --- | --- |
| **DTR** | **RTS** | **EN** | **GPIO 0 / GPIO 2** |
| 1 | 1 | 1 | 1 / 0 |
| 0 | 0 | 1 | 1 / 0 |
| 1 | 0 | 0 | 1 / 0 |
| 0 | 1 | 1 | 0 / 0 |

Die beiden Transistoren schalten jeweils, wenn die Spannung zwischen Basis und Emitter ist. Das ist der Fall, wenn DTR und RTS unterschiedliche Signalpegel aufweisen. Sind die beiden Pegel an DTR und RTS identisch, dann schalten die Transistoren nicht. In diesem Fall haben die Pins GPIO 0 / 2 und EN die Pegel, welche über die Widerstände am Mikrocontroller definiert sind. Das ist für GPIO 0 High wie für Enable, GPIO 2 wird über einen Pull-Down Widerstand standartmäßig auf Ground gezogen. Wenn DTR auf High liegt und RTS auf Low schaltet der Transistor Q1401. Er zieht damit den Enable-Pin am Mikrocontroller auf den Pegel von RTS, also gegen Ground und löst damit den Reset des Microcontrollers aus. GPIO 0 liegt weiterhin auf High durch den Pull-Up Widerstand am Mikrocontroller. Sind die Pegel vertauscht, liegt also DTR auf LOW und RTS auf High schaltet der Transistor Q1402. Dadurch wird GPIO 0 auf den Pegel von DTR gezogen, welcher zu diesem Zeitpunkt auf Ground liegt. Der Enable-Pin des Mikrocontrollers bleibt dabei auf High durch den Pull-Up Widerstand am Mikrocontroller. Das ist damit der Bootmodus, indem der ESP32 mit den per USB übermittelten Daten startet. Beim Empfang von Daten via USB erzeugt der USB To Serial-Wandler somit automatisch einen Reset am Controller, aktiviert daraufhin den Bootmodus und kehrt anschließend wieder in den normalen Boot- und Betriebsmodus zurück. Auch hier wurden zum Schutz der Bauteile TVS Dioden vor den Pins des Mikrocontrollers vorgesehen, um Zerstörung durch ESD bedingte Spannungsspitzen zu verhindern. Zusätzlich wurde noch ein Programming Pinheader vorgesehen. Über diesen können sowohl Daten übertragen als auch die Pins in den korrekten Bootmodus versetzt werden. Dieser hat den Vorteil, dass im Falle einer defekten oder Fehlerhaften Schaltung trotzdem noch ein Flashen über die Verwendung eines Devkits oder eines ähnlichen Hilfsmittels möglich ist.

Stecker

## PCB-Layout

# Software

## Konzept

## Aufbau

## Umsetzung

# Test und Inbetriebnahme

# Fazit und Ausblick

## Zusammenfassung der Arbeit

## Mögliche Weiterentwicklungen

# Literaturverzeichnis

# Bildverzeichnis