

BACHELORTHESIS

Janik Alexander Naumann
[REDACTED]

Bau und Konzeption eines Zugleit- und Sicherungssystems für autonomen Fahrbetrieb

BETREUUNG

Prof. Dr. Paweł Buczek (Erstprüfer)
Prof. Dr. Marc Hensel (Zweitprüfer)

ABGABE

23.12.2024

FAKULTÄT TECHNIK UND INFORMATIK

Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science
Department of Information and Electrical Engineering

Inhalt

1	Kurzbeschreibung	4
2	Einleitung	5
2.1	Einführung in die Thematik	5
2.2	Zugleit- und Sicherungstechnik (Stand der Technik)	5
2.2.1	Gleisfreimeldung	5
	Visuelle Freimeldung	5
	Achszähler	5
	Gleisstromkreise	6
2.2.2	Fahrweg und Fahrstraße	6
2.2.3	Stellwerke	7
	Mechanische Stellwerke	7
	Relaisstellwerke	7
	Elektronische Stellwerke (ESTW)	8
2.2.4	Zugbeeinflussung	8
	Punktförmige Zugbeeinflussung (PZB)	8
	Linienförmige Zugbeeinflussung (LZB)	8
	European Train Control System (ETCS)	8
2.3	Zielsetzung und Versuchsaufbau	9
3	Umsetzung	10
3.1	Strecke	10
3.2	Zugsteuerung (Hauptmodul)	12
3.2.1	Das Märklin-Motorola Format	12
	Adresse	13
	Sonderfunktion	14
	Geschwindigkeit und Richtung / Funktionen F1...F4	14
	Datenpakete	16
3.2.2	Endstufe	17
3.2.3	Zusatzbeschaltung	19
	Galvanische Trennung des Controllers	19
	Mikrocontroller	21
	Pegelwandlung	22
	Spannungsversorgung	22

3.2.4	Platinenlayout.....	23
3.3	Elektronisches Stellwerk (ESTW).....	24
3.3.1	Weichensteuerung und Rückmeldung	24
3.3.2	Signale.....	26
3.3.3	Gleisstromabschaltung	27
3.3.4	Gleisrückmeldung	28
3.3.5	Platinenlayout.....	29
3.4	Kommunikation.....	30
3.4.1	Stellwerk-Bus	30
3.4.2	Hauptmodul und Steuersoftware.....	32
3.5	Hauptmodul-Software	33
3.5.1	Library <code>motorola()</code>	34
3.5.2	Library <code>estw</code>	35
3.5.3	Library <code>i2c</code>	36
3.5.4	Library <code>uart</code>	36
3.6	Stellwerk-Software.....	37
3.6.1	Library <code>track</code>	37
3.6.2	Library <code>i2c</code>	38
3.7	Steuersoftware („Der Fahrdienstleiter“)	38
3.7.1	Backend: Bahnbetriebssoftware	39
	Kommunikation.....	39
	Bahnbetriebsteuerung	39
3.7.2	Frontend: Benutzeroberfläche (GUI).....	40
	Automatikbetrieb.....	40
	Handbetrieb	41
3.7.3	Kooperation von Frontend und Backend	42
3.7.4	Fahrstraßenbildung und Zugleitung	43
	Berechnung des Fahrwegs <code>createRoute()</code>	43
	Generierung der Fahrstraße <code>lockRoute()</code>	43
	Überwachung laufender Fahrten <code>locateTrains()</code>	44
	Ausführung der Fahrten und Auflösung der Fahrstraßen <code>endRoutes()</code>	44
4	Ergebnisse.....	45
4.1	Streckennetz	45

4.2	Bahnbetrieb	46
4.3	Transfer auf die Realität.....	46
4.4	Stabilität des Systems	47
4.5	Sicherheitslücken	47
4.6	Aussicht auf Erweiterungen	48
5	Fazit.....	49
6	Verzeichnisse	51
6.1	Quellenverzeichnis.....	51
6.2	Literaturverzeichnis	52
6.3	Abbildungs- und Tabellenverzeichnis	53
7	Anhang.....	55
7.1	Schaltplan Hauptmodul	55
7.2	Schaltplan Stellwerkmodul	56
7.3	Programmfluss-Diagramm Hauptmodul.....	57
7.4	Programmfluss-Diagramm Stellwerkmodul.....	58
7.5	Programmfluss-Diagramm Fahrstraßenbildung	59

1 Kurzbeschreibung

Der Schienenverkehr besitzt in der heutigen Zeit einen hohen Stellenwert. Aufgrund der physikalischen Gegebenheiten (feste Spurführung, lange Bremswege) müssen Zugfahrten besonders gesichert sein, um schwere Unfälle zu verhindern. Der Begriff der Leit- und Sicherungstechnik umfasst Ansätze zum Planen, Überwachen und Sichern von Zugfahrten. Die hierzu in Deutschland eingesetzte Technik gilt als sicher, jedoch treten aufgrund des nicht einheitlichen Modernisierungsstandes häufig Teilsystemausfälle auf, die den Betriebsablauf stören. Diese Arbeit untersucht, welche Einrichtungen zum sicheren Durchführen von Zugfahrten notwendig sind und präsentiert ein Konzept für ein einheitliches, robustes Zugleit- und Sicherungssystem.

Ein physikalisches Modell wird hierzu mit einer Signalanlage, einer Weichensteuer- und Rückmeldeanlage, einer Gleisfreimeldeanlage und einem Zugbeeinflussungssystem ausgestattet. Diese Teilsysteme werden von Stellwerken auf ESTW-Basis verwaltet und überwacht, die über ein Bussystem durch eine Betriebszentrale ferngesteuert werden. Diese führt die Fahrwegplanung und Sicherung durch und überwacht alle durchgeführten Zugfahrten. Das entstandene System wird im Anschluss als Basis für die Implementierung eines autonomen Fahrbetriebs verwendet.

In modern times, Rail transportation has a high importance. Due to physical conditions, railway operations have to be secured to prevent accidents. The „train control and safety technology“ includes techniques for planning, monitoring and securing railway operations. The technology used for this purpose in Germany is considered safe, however, subsystems failures are caused by the inconsistent state of modernization. In this thesis, the necessary elements for safe railway operations are investigated. Furthermore a concept for a unified, robust train control and safety system is presented.

To achieve this, a physical model is equipped with a signaling system, a turnout control and monitoring system, a track occupancy detection and an automatic train protection. These subsystems are managed by signal boxes based on electronic interlocking technology, which are remotely controlled by an operation center. This center handles route planning and securing while monitoring all train operations. The resulting system is later used as a basis for the implementation of autonomous train operation.

2 Einleitung

2.1 Einführung in die Thematik

Die Bahn zählt zu den sichersten Verkehrsmitteln. Dies liegt primär an der durch die Schienen gegebene Spurführung, durch welche ein Abkommen eines Fahrzeugs vom Fahrweg im Regelfall nicht möglich ist. Weiterhin werden verschiedene Verfahren zur Sicherung des Schienenverkehrs angewandt, um Zusammenstöße, Entgleisungen und andere Havariefälle zu verhindern [1]. In diesem Abschnitt werden einige dieser Verfahren für das bessere Verständnis der im Abschnitt 3 dokumentierten Vorgehensweisen erklärt.

2.2 Zugleit- und Sicherungstechnik (Stand der Technik)

Grundlage für die sichere Durchführung von Zugfahrten ist die Kenntnis über die bestehenden Gefährdungen. Neben der bereits erwähnten Spurführung ist die geringe Haftreibung maßgebende Eigenschaft des Eisenbahnbetriebs. Die Spurführung birgt in Fällen der Überschreitung physikalischer Grenzen die Gefahr von Entgleisungen [1]. Gleisbögen und Weichen sind Bereiche, in denen besondere Vorsicht geboten ist. Die geringe Haftreibung geht mit langen Bremswegen einher, die meist die Sichtweite des Fahrzeugführers übersteigen. Zusammenstöße können demnach nur verhindert werden, wenn Fahrten entsprechend langsam durchgeführt oder Verfahren zur Freimeldung von Fahrwegen angewandt werden.

In Deutschland werden aufgrund des nicht einheitlichen Modernisierungsstandes des Schienennetzes verschiedene Verfahren und Technologien zur Zugsicherung eingesetzt. Während beispielsweise in den meisten Regionen bereits elektrische Signale (Lichtsignale) eingesetzt werden, sind vereinzelt noch mechanisch gesteuerte Formsignale im Einsatz.

2.2.1 Gleisfreimeldung

In der frühen Geschichte des Schienenverkehrs wurden Zugfahrten im zeitlichen Abstand durchgeführt. Beispielsweise alle fünf Minuten durfte ein Zug in eine Richtung abfahren. Da diese Vorgehensweise in Fällen, in denen etwa ein vorausgefahrener Zug auf der Strecke liegenblieb, bestand große Unfallgefahr. Aus diesem Grund wurde später das **Fahren im festen Raumabstand** eingeführt [2]. Hierbei wird einem Zug lediglich dann die Abfahrt gestattet, wenn der vorausliegende Streckenabschnitt frei von Hindernissen ist. Diese **Gleisfreimeldung** kann auf verschiedene Weisen geschehen, die nachfolgend näher erklärt sind.

Visuelle Freimeldung

Bei der visuellen Gleisfreimeldung wird ein Zug beim Fahren in und aus dem zu überwachenden Abschnitt beobachtet und auf Vollständigkeit (Zugschlussignal) überprüft. Dieses Verfahren wird heute nur noch in Ausnahmefällen (u.a. bei Stellwerkstörungen) angewandt.

Achszähler

Die Gleisfreimeldung kann auch durch Zählung von Achsen erfolgen. Hierbei werden am Anfang und Ende des Abschnitts Achszähler installiert. Beim Ausfahren eines Zuges aus dem Abschnitt muss die Anzahl der gezählten (ausgezählten) Achsen mit der Zahl der zuvor eingefahrenen (eingezählten) Achsen übereinstimmen.

Die in Deutschland zumeist verwendeten Achszähler sind sogenannte induktive Schienenkontakte. Dabei befindet sich an der Außenseite einer Schiene ein Sender eines magnetischen Wechselfeldes und an der Innenseite ein Empfänger. Eine diese Stelle passierende Achse wird vom Empfänger durch die Änderung des Magnetfeldes detektiert. In der Regel befinden sich zwei Achszähler direkt hintereinander, um die Fahrtrichtung des Fahrzeugs zu erkennen.

Achszähler sind relativ zuverlässig und werden in Deutschland häufig eingesetzt, jedoch sind sie im europäischen Raum teils in Bezug auf die Zuverlässigkeit umstritten [1].

Gleisstromkreise

Bei der Freimeldung durch Gleisstromkreise bilden die Schienen elektrische Leiter. Ein Sender am einen Ende des Abschnitts lässt einen Strom über die Gleise fließen, den der Empfänger am anderen Ende detektiert. Überfährt ein Zug die Schienen, wird dieser Stromkreis über die Achsen kurzgeschlossen (Ruhestromprinzip). Hiermit kann zuverlässig erkannt werden, ob sich ein Fahrzeug im zu überwachenden Bereich befindet.

Gleisstromkreise haben den Vorteil, dass sie den gesamten Gleisabschnitt überwachen. Nach einer Störung muss der Abschnitt nicht visuell freigemeldet werden, wie es bei durch Achszähler überwachten Abschnitten der Fall ist. Nachteilig ist, dass Gleisabschnitte elektrisch voneinander getrennt werden müssen. Das Installieren der benötigten Isolierstöße ist relativ aufwändig [1].

2.2.2 Fahrweg und Fahrstraße

Ist ein Zug bereit zur Abfahrt, benötigt der Fahrzeugführer zum Starten der Fahrt die Erlaubnis des Fahrdienstleiters. Die Erlaubnis wird durch Signale, einen schriftlichen oder mündlichen Befehl erteilt. Zuvor stellt der Fahrdienstleiter sicher, dass die Zugfahrt ohne Gefährdungen durchgeführt werden kann. Dazu muss der geplante Fahrweg des Zuges unbelegt und frei von Hindernissen sein. Jedes **Fahrwegelement** (Weichen, Bahnübergänge, ...) muss sich in der richtigen Position befinden, bevor eine **Fahrstraße** festgelegt werden kann. Eine Fahrstraße ist der gesicherte Fahrweg, der für eine Zugfahrt festgelegt wird und bis zum Ende der Fahrt von keinem anderen Fahrzeug befahren werden darf. Fahrwegelemente, die einer verschlossenen Fahrstraße angehören, dürfen nicht nachträglich verstellt werden. Die Fahrstraße wird aufgelöst, wenn der Zug in den letzten Streckenabschnitt (Zielgleis) eingefahren ist. Für den Fall, dass ein Zug auf dem Zielgleis nicht zum Stehen gebracht wird, wird der Folgeabschnitt zur Fahrstraße dazugezählt (Durchrutschweg). Beim Befahren dieses Abschnitts greifen die in 2.2.4 beschriebenen Zugbeeinflussungsverfahren ein, um die **feindliche Fahrt** zu beenden [3].

Auf vielbefahrenen Strecken werden häufig Teilfahrstraßen eingesetzt, welche sich dadurch auszeichnen, dass Abschnitte, die vom Zug verlassen wurden, bereits in weitere Fahrwege eingeplant werden können. Bei unveränderter Strecke lassen sich somit höhere Verkehrsaufkommen bewältigen. Teilfahrstraßen werden in dieser Arbeit nicht näher behandelt.

2.2.3 Stellwerke

Stellwerke sind wichtige Einrichtungen in der Eisenbahninfrastruktur, die für die Generierung und Sicherung von Fahrwegen zuständig sind. Weichen, Signale und Gleisfreimeldesysteme sind den Stellwerken untergeordnete Elemente, die von diesen gesteuert bzw. ausgewertet werden. Aufgrund des technischen Fortschritts wurden mit der Zeit verschiedene Arten von Stellwerken entwickelt, die im Folgenden beschrieben sind.

Mechanische Stellwerke

Zum Zeitpunkt der Entwicklung der ersten Stellwerke existierten keine Ansätze zur elektrischen Informationsverarbeitung oder Steuerung von Stellgliedern. Die entstandenen Stellwerke arbeiteten rein mechanisch. Die Steuerung von Weichen und Signalen erfolgte über Seilzüge, die im Stellwerk mit Hebeln zur Handbetätigung verbunden waren. Hiervon leitet sich der Begriff „Hebelstellwerk“ ab. Die Gleisfreimeldung erfolgte visuell. Ein eingestellter Fahrweg konnte durch ein Verschlussystem festgelegt (verschlossen) werden, wobei die zugehörigen Hebel in ihrer Position festgesetzt wurden. Eine wichtige Grundlage war bereits damals das Prinzip der Signalabhängigkeit. Dieses besagt, dass ein Signal sich erst dann auf Fahrt stellen lassen darf, wenn alle Fahrwegelemente der zugehörigen Fahrstraße verschlossen sind [4].

Mechanische Stellwerke sind vereinzelt noch heute im Einsatz. Meist sind sie mit elektrischen Zusatzeinrichtungen erweitert.

Relaisstellwerke

Durch den Fortschritt in der elektronischen Informationsverarbeitung konnten später Relaisstellwerke errichtet werden, die zur Verarbeitung von Informationen Relais verwenden und auf Mechanik verzichten. Diese Stellwerke verfügen meist über eine Gleisfreimeldeanlage. Das Einstellen von Fahrwegen erfolgt über einen Stelltisch, dessen Stellelemente (i.d.R. elektrische Tasten) in ihrer Anordnung mit dem physikalischen Streckenaufbau übereinstimmen.

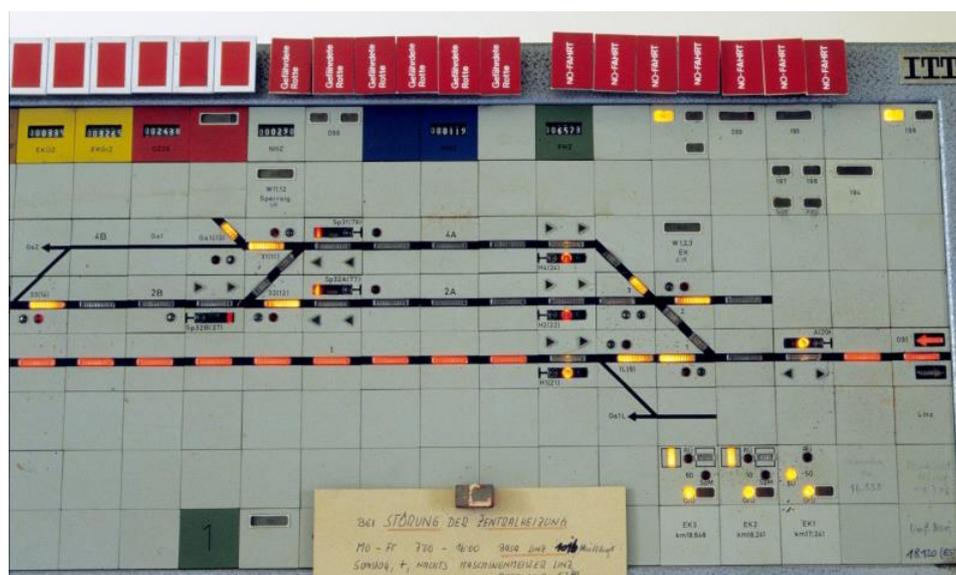


Abbildung 1 – Stelltisch eines Gleisbildstellwerks [5]

Weil die Bedienoberfläche eine Nachbildung der Strecke ist, gehört diese Stellwerksform zu den Gleisbildstellwerken. Eine Vielzahl heute in Betrieb befindlicher Stellwerke entsprechen in Ihrer Bauform dem Relaisstellwerk.

Elektronische Stellwerke (ESTW)

Elektronische Stellwerke entsprechen funktional den Relaisstellwerken, wobei die Informationsverarbeitung auf Mikrocontroller umgelagert ist. Heute neu errichtete Stellwerke entsprechen dem Typ ESTW oder DSTW (Digitale Stellwerke).

2.2.4 Zugbeeinflussung

Die Beeinflussung von Schienenfahrzeugen ist für die Sicherheit des Schienenverkehrs von hoher Relevanz. In der Regel zeigen Signale die sichere Befahrbarkeit eines Streckenabschnitts an. Der Fahrzeugführer muss diese Signale wahrnehmen und bei Halt-Befehl das Fahrzeug zum Stehen bringen. Tut er dies nicht, so geht von dieser **feindlichen Zugfahrt** eine Gefahr für andere Fahrten aus. Gleich verhält es sich, wenn die Zugfahrt mit überhöhter Geschwindigkeit stattfindet. In diesen Fällen kommt die Zugbeeinflussung zur Anwendung, die in Gefahrensituationen am Fahrzeug eine Zwangsbremung bewirkt.

Punktförmige Zugbeeinflussung (PZB)

Durch die punktförmige Zugbeeinflussung besteht die Möglichkeit, ein Fahrzeug beim Vorbeifahren an ausgewählten Punkten der Strecke zu beeinflussen. Dazu sind Fahrzeug und Strecke mit Elektromagnete ausgestattet. Der Fahrzeugmagnet erzeugt dauerhaft ein elektromagnetisches Feld mit den Frequenzen 500 Hz, 1000 Hz, 2000 Hz. Der Gleismagnet bildet einen auf eine dieser Frequenzen abgestimmten Schwingkreis. Wenn sich Fahrzeug- und Gleismagnet nah beieinander befinden, findet eine induktive Kopplung statt, die von der Fahrzeugelektronik erkannt wird. Auf diese Weise kann eine Zwangsbremung eingeleitet werden, wenn ein Fahrzeug ein Halt zeigendes Signal überfährt oder auf eine Reduzierung der zulässigen Höchstgeschwindigkeit nicht reagiert.

Das aktuell in Deutschland eingesetzte System PZB 90 ist eine Weiterentwicklung des Indusi-Systems aus den 1930er Jahren [1].

Linienförmige Zugbeeinflussung (LZB)

Bei Zugfahrten mit erzielten Geschwindigkeiten oberhalb von 160 km/h kommt die PZB nicht zum Einsatz. Stattdessen befinden sich auf Schnellfahrstrecken Kabellinienleiter, die schleifenförmig im Gleisbett verlegt sind. Eine Schleife deckt jeweils 300 m der Strecke ab. Den Linienleiter umgibt ein elektromagnetisches Feld, das von einer Fahrzeugantenne empfangen wird. Das Fahrzeug ist damit durchgehend mit der LZB-Zentrale verbunden, die für die Fahrwegsicherung sorgt und Geschwindigkeitsvorgaben mitteilt [1].

Das System wird in Deutschland noch häufig verwendet, jedoch wird es von ETCS schrittweise abgelöst.

European Train Control System (ETCS)

Das ETCS schafft ein einheitliches Zugsicherungssystem für europäische Länder. Dies ermöglicht grenzüberschreitende Zugfahrten ohne die Notwendigkeit eines Lokwechsels. Die Kommunikation zwischen Strecken- und Fahrzeugausrüstung kann punktförmig oder

linienförmig erfolgen. Die punktförmige Datenübertragung erfolgt über sogenannte Eurobalisen. Das Fahrzeug sendet dauerhaft ein 27 MHz Energiesignal aus, welches die Balise beim Überfahren induktiv mit Energie versorgt. Während dieser Zeit sendet diese ein Datenpaket an das Fahrzeug.

Der ETCS-Standard definiert mehrere Betriebsmodi, die im folgender Tabelle beschrieben sind.

Level 0	Zugbeeinflussung ist nicht aktiv. Das ETCS-System erfasst lediglich die momentane Fahrzeuggeschwindigkeit.
Level STM	Das Fahrzeug verwendet das nationale Zugbeeinflussungssystem (z.B. PZB in Deutschland)
Level 1	ETCS ist aktiv. Nur Kommunikation (punkt- oder linienförmig) von Strecke zu Fahrzeug möglich. Signale nicht weiter notwendig. Gleisfreimeldung weiterhin nach nationalen Standards durchgeführt.
Level 2	Bidirektionale, kontinuierliche Kommunikation findet statt. Dazu wird das Mobilfunknetz (GSM-R) verwendet. Die Balisen dienen dabei nur als Festdatenbalisen (wie „Kilometersteine“) zur Ortung des Fahrzeugs. Herkömmliche Gleisfreimeldung weiter erforderlich.
Level 3	Züge prüfen eigenständig ihre Vollständigkeit (Integritätsprüfung). Damit ist die herkömmliche Gleisfreimeldung nicht mehr erforderlich.

Tabelle 1 – Liste verfügbarer ETCS-Betriebsmodi

Das ETCS-System befindet sich derzeit im Ausbau und soll in Deutschland zukünftig die Systeme PZB und LZB ablösen. Es bietet die Grundlage für einen autonomen Bahnbetrieb [6].

2.3 Zielsetzung und Versuchsaufbau

Wie bereits beschrieben besteht das deutsche Schienennetz aus vielen verschiedenen technischen Einrichtungen, die zum Teil dem Stand der Technik entsprechen, während andere veraltet sind.

Ziel dieser Arbeit ist es, ein Schienennetz zu entwerfen, das über ein robustes Zugleit- und Sicherungssystem verfügt, jedoch mit möglichst einfacher und einheitlicher Ausrüstung auskommt. Konkret soll eine Steuer- und Überwachungseinrichtung für Fahrwegelemente, eine Signalisierungsanlage, eine Gleisfreimeldeanlage und ein Zugbeeinflussungssystem entstehen. Im weiteren Verlauf soll ein autonomer Betrieb realisiert werden.

Bei allen Versuchen wird sich stets am Stand der Technik und an den Praktiken im deutschen Schienenverkehr orientiert.

Für das Durchführen der Versuche ist zunächst das Errichten eines Schienennetzes erforderlich, an welchem alle Versuche durchgeführt werden und die Streckenausrüstung installiert werden kann. Als Schienen dienen dabei handelsübliche C-Gleise der Firma Märklin, die der Nenngröße H0 (Maßstab 1:87) entsprechen. Konkret werden folgende Produkte verwendet.

Kategorie	Nennmaße	Artikelnr.	Bemerkung
Geraden	L=171,7 mm	24172	kurz
	L=188,3 mm	24188	lang
Gleisbögen	$\alpha=24,3^\circ$ r=437,5 mm	24224	
	$\alpha=30,0^\circ$ r=360,0 mm	24130	inneres Gleis
	$\alpha=30,0^\circ$ r=437,5 mm	24230	äußeres Gleis
Weichen	Überlagerung von 24188, 24224	24611	links
		24612	rechts
Weichenantrieb		74491	in Verbindung mit 24611, 24612

Tabelle 2 – Verwendete Produkte der Firma Märklin

Die Spurweite beträgt 16,5 mm und der Gleisabstand ist 77,5 mm (Abstand der Mittelleiter).

Die elektrische Versorgung der Schienenfahrzeuge erfolgt über ein Dreileiter-System. Dabei dienen die zwei Schienen als Neutralleiter, während die Versorgungsspannung über einen Mittelleiter bereitgestellt wird. Die Fahrzeuge stellen den elektrischen Kontakt zum Mittelleiter über einen Schleifer her. Die Steuerung der Züge erfolgt digital über das Märklin-Motorola Format (siehe 3.2.1).

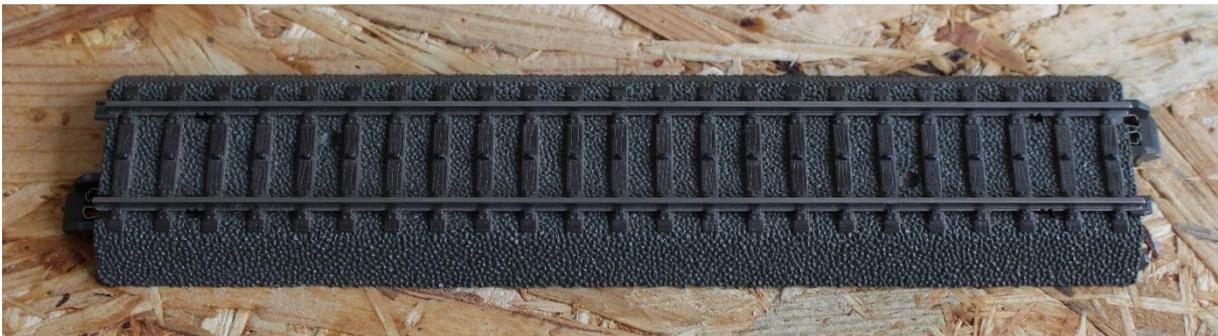


Abbildung 2 – C-Gleis mit Mittelleiter

3 Umsetzung

Alle in diesem Abschnitt thematisierten Softwares liegen als Quellcode wie auch als kompilierte Binärdateien dieser Arbeit bei und können beim Erstprüfer eingesehen werden.

3.1 Strecke

Die Schienen werden auf einer zuvor aufgestellten Holzplatte mit einer Fläche von 300x125 cm verlegt. Die Platte bietet gute Möglichkeiten, Kabel unauffällig und geordnet zu verlegen und Vorrichtungen für Versuche zu befestigen.

Das erbaute Schienennetz zeichnet sich durch eine elliptische Form aus. Die zwei parallel angeordneten Bahnhöfe sind an beiden Enden durch lange Gleisbögen verbunden. Während das Netz grundsätzlich zweigleisig ausgebaut ist, gibt es einen Streckenabschnitt, auf dem nur ein Gleis verfügbar ist. An dieser Engstelle kann später die Stabilität des autonomen Betriebs – insbesondere in Hinsicht auf die Gleisfreimeldung – getestet werden.

Das Schienennetz ist auf der folgenden Seite schematisch dargestellt.

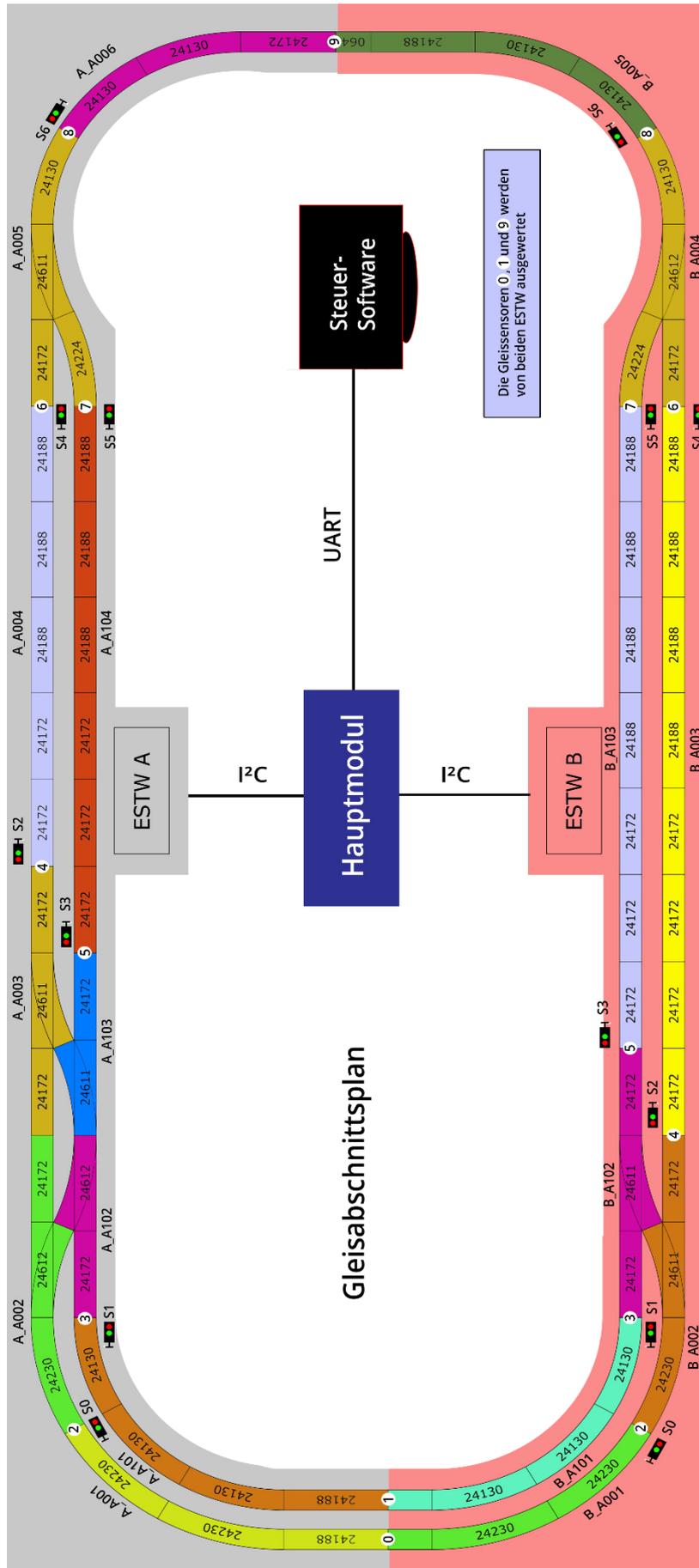


Abbildung 3 – Gleisabschnittsplan (Gesamtkonzept)

3.2 Zugsteuerung (Hauptmodul)

In den 1990er Jahren brachte Märklin die Produktlinie „Märklin Digital“ auf den Markt. Zuvor hatten Modellbahn-Anwender über einen regelbaren Transformator die Spannung zwischen dem Mittelleiter und den Außenleitern reguliert. Die Fahrzeuge leiteten die Spannung direkt an den Fahrmotor weiter und fuhren mit einer zur eingestellten Spannung proportionalen Geschwindigkeit. Richtungswechsel wurden über einen kurzen Spannungsimpuls realisiert, der die Fahrspannung übertraf und somit einen Umschaltmechanismus im Fahrzeug ansteuerte, der den Motor umpolte. Nachteil dieser Ansteuerung war, dass Fahrzeuge nicht individuell gesteuert werden konnten.

Das neue System zeichnete sich dadurch aus, dass der Transformator durch ein digitales Fahrpult (Handelsname Control 80) ausgetauscht wurde, das ein sogenanntes Energie-Daten-Signal generierte. Mit einem Decoder-Modul auf digitale Modellbahnanlagen umgerüstete Fahrzeuge konnten so angesteuert und einzeln in ihrer Fahrgeschwindigkeit reguliert werden. Weiterhin gab es die Möglichkeit, pro Fahrzeug eine fahrtrichtungsabhängige Sonderfunktion (meist Beleuchtung) zu steuern. Am Fahrpult konnten über ein Tastenfeld bis zu 80 Decoder adressiert und gesteuert werden. Nach wenigen Jahren wurde ein anderes Fahrpult (Control Unit 6021) verfügbar, mit welchem vier Zusatzfunktionen je Decoder gesteuert werden konnten.

Diese sehr alte Variante der digitalen Zugsteuerung ist nicht sehr komplex konzipiert, funktioniert zuverlässig und wird auch heute von den meisten Märklin-Lokomotiven unterstützt. Daher wird sich für die Implementierung des sogenannten Märklin-Motorola Formats entschieden. Der gesamte Prozess ist im Folgenden beschrieben.

3.2.1 Das Märklin-Motorola Format

Der Name „Märklin-Motorola“ stammt aus der Zeit, in der die digitale Modellbahnsteuerung neu war. Märklin vermarktete in dieser Zeit Fahrpulte und Decoder mit integrierten Schaltkreisen des Herstellers Motorola. Das Format gilt heute als veraltet, ist in modernen Handsteuergeräten wie der „Märklin Mobile Station“ aus Gründen der Abwärtskompatibilität aber noch unter der Bezeichnung „fx“ vorhanden. Im Folgenden wird der Begriff Motorola-Format verwendet.

Es gibt zwei Varianten des Motorola-Formats. Die erste Variante (Motorola I) wurde mit den ersten Märklin Digital Fahrpulten vertrieben. Es gab keine Unterstützung für die Zusatzfunktionen F1...F4, die Jahre später mit dem Format Motorola II eingeführt wurden. Die beiden Varianten unterscheiden sich in der Zusammensetzung des sogenannten S Bits. Die Erklärung hierzu ist in Abschnitt 3.2.1.1 zu finden.

Wie bereits erwähnt werden die Fahrzeuge über ein Energie-Daten-Signal gesteuert. Genau betrachtet handelt es sich hierbei um ein pulsweitenmoduliertes Signal mit zwei erlaubten Pulsweiten. Eine Periode dauert $208 \mu\text{s}$. Das Signal beginnt immer bei $+U_V$ und wechselt nach entweder $\frac{1}{8}T$ ($26 \mu\text{s}$) oder nach $\frac{7}{8}T$ ($182 \mu\text{s}$) den Pegel zu $-U_V$. Die Versorgungsspannung U_V liegt dabei im Intervall $[16, 24] \text{ V}$. Ein kurzer High-Pegel ist als logische 0 definiert, während der lange High-Pegel eine 1 kodiert.

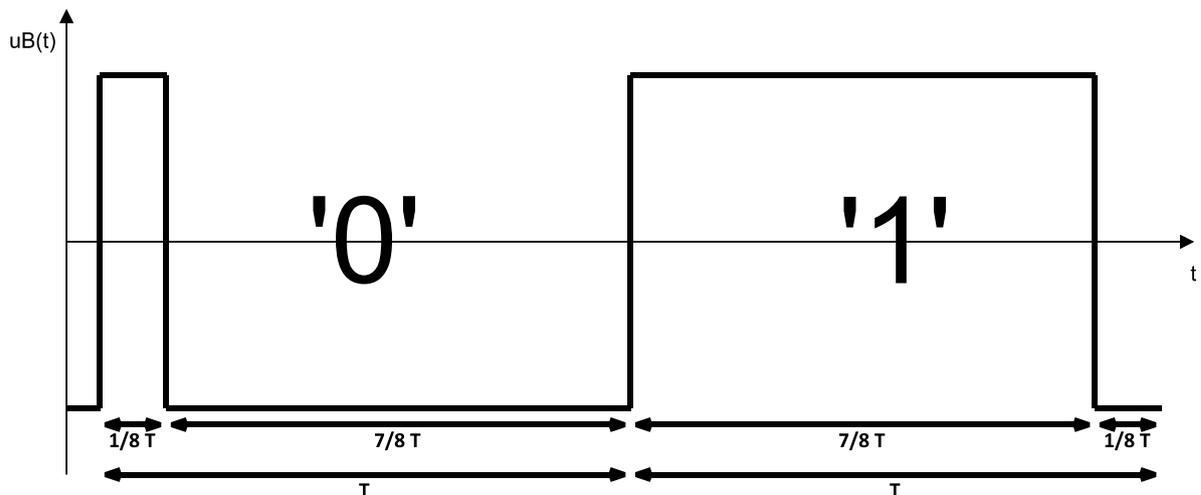


Abbildung 4 – Bitfolge "01" im Motorola-Format

3.2.1.1 Protokoll

Die Datenübertragung beim Motorola-Format erfolgt unidirektional vom Fahrpult zu den Fahrzeugdecodern. Ein Datenpaket ist wie folgend dargestellt aufgebaut.

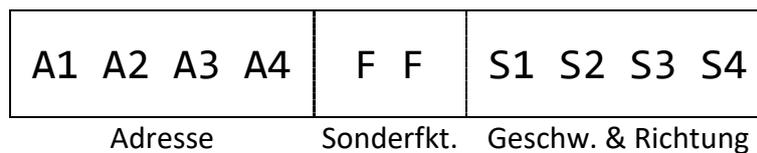


Abbildung 5 – Aufbau eines Motorola-Datenpakets [7]

Adresse

Die Symbole A1...A4 bilden das Adressfeld. Es handelt sich hierbei um vier Trits, also acht Bits, die wie folgend bestimmt werden.

Die Adresse des anzusteuernenden Decoders wird zunächst in die ternäre Form umgerechnet. Das Verfahren gleicht prinzipiell der Umrechnung von Dezimal in Binär, jedoch wird die Basis 3 verwendet. Auf den Dezimalwert wird so oft eine ganzzahlige Teilung mit Rest durch die Basis 3 angewendet, bis das Ergebnis 0 beträgt.

Ein Beispiel demonstriert die Umrechnung für die Adresse 34:

34: 3 = 1	R: 1	→	<table style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">A1</td> <td style="padding: 2px 10px;">A2</td> <td style="padding: 2px 10px;">A3</td> <td style="padding: 2px 10px;">A4</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">1</td> </tr> </table>	A1	A2	A3	A4	1	2	0	1
A1	A2			A3	A4						
1	2			0	1						
11: 3 = 3	R: 2										
3: 3 = 1	R: 0										
1: 3 = 0	R: 1		(open)								

Tabelle 3 – Rechenbeispiel zu den Adressbits

Die bestimmten Trits werden dann in die acht Adressbits umgeformt. Dabei wird eine ternäre '0' zu "00", eine ternäre '1' zu "11" und eine ternäre '2' zu "10". Das Symbol "01" ist im Adressfeld unzulässig. Die gesendete binäre Bitfolge im Beispiel lautet demnach: "11 10 00 11".

Die Adresse muss im Intervall $[0, 80]$ liegen. Dabei ist 0 die Nulladresse, die explizit keinen Decoder anspricht. Zudem ist die Adresse 80 als 0 definiert, während die Nulladresse den Zahlenwert 80 zugewiesen bekommt [7].

Sonderfunktion

Die fahrtrichtungsabhängige Sonderfunktion (F0) wird durch die FF Bits ein ("11") und ausgeschaltet ("00"). Die Symbole "01" und "10" sind unzulässig. In der Regel steuert die Sonderfunktion das Spitzensignal des Fahrzeugs.

Geschwindigkeit und Richtung / Funktionen F1...F4

Die Funktion der Symbole S1...S4 unterscheidet sich je nach gesendetem Datenpaket und Variante des Motorola-Formats.

Bei der Variante Motorola I sind die Symbole S1 S2 S3 S4 den Bits AA BB CC DD zugeordnet. D C B A sind die binärkodierte Dezimalwerte der Fahrstufen. Dabei steht Fahrstufe 0 für Stillstand (Stop) und 1 für die Richtungsumkehr. Den Fahrstufen 1...14 sind die Zahlenwerte 2...15 zugewiesen.

Folgende Übersicht verdeutlicht die Zusammensetzung.

				Fahrstufe	D	C	B	A
				Stop	0	0	0	0
				Umkehr	0	0	0	1
S1 S2 S3 S4 AA BB CC DD	→	1		0	0	1	0	
		2		0	0	1	1	
		13		1	1	1	0	
		14		1	1	1	1	

Tabelle 4 – Kodierung der Fahrstufen beim Motorola-Format [7]

Beispiel: Fahrstufe 7

$$7 + 1 = 8 \xrightarrow{\text{binär}} \begin{matrix} D & C & B & A \\ 1 & 0 & 0 & 0 \end{matrix} \xrightarrow{\text{invers}} \begin{matrix} S1 & S2 & S3 & S4 \\ AA & BB & CC & DD \\ \boxed{00} & \boxed{00} & \boxed{00} & \boxed{11} \end{matrix}$$

Tabelle 5 – Rechenbeispiel zu den S-Bits

Diese Art der Angabe der Fahrstufe hat den Nachteil, dass sie relativ zur eingestellten Fahrtrichtung ist. Im Fall, dass ein Decoder den einmalig versendeten Richtungsumkehr-Befehl nicht erhält (z.B. bei kurzzeitigen Kontaktschwierigkeiten am Schleifer oder wenn das Fahrzeug sich gerade nicht auf den Schienen befand), fährt das Fahrzeug in die falsche Richtung. Außerdem kann das Fahrpult nicht erkennen, ob das Fahrzeug auf Vorwärts- oder Rückwärtsfahrt eingestellt ist.

Die zweite Variante (Motorola II) führt die absolute Angabe von Fahrstufen ein, die obig beschriebene Probleme eliminiert. Zudem wird die Möglichkeit der Steuerung der vier Zusatzfunktionen F1...F4 hinzugefügt.

Die Symbole S1 S2 S3 S4 werden den Bits AE BF CG DH zugeordnet. Die Bits A B C D werden weiterhin so bestimmt, wie zuvor beschrieben. Die restlichen Bits ergeben sich nach folgender Tabelle. Das Protokoll ist vollständig abwärtskompatibel zu Motorola I.

S1 S2 S3 S4 AE BF CG DH				→	Fahrstufen		E F G H				Kategorie
							E	F	G	H	
					-14 bis	-7	1	0	1	0	„rückwärts schnell“
					-6 bis	-0	1	0	1	1	„rückwärts langsam“
					+0 bis	+6	0	1	0	1	„vorwärts langsam“
					+7 bis	+14	0	1	0	0	„vorwärts schnell“

Tabelle 6 – Fahrstufen-Kategorien bei Motorola II [7]

Das erhöhte Risiko für Übertragungsfehler durch die verlorene Redundanz (ABCD werden nicht mehr doppelt gesendet) wird durch die eingeführten Kategorien etwas verringert, da bei auftretenden Bitfehlern beispielweise zwar die Fahrstufe falsch übertragen werden kann, große Abweichungen jedoch durch Überprüfung der mitgesendeten Kategorie auffallen. Da die Fahrstufen-Datenpakete periodisch versendet werden, stellen Bitfehler, die die empfangene Fahrstufe zu einer in der gleichen Kategorie befindlichen verändern, kein größeres Problem dar (siehe folgender Abschnitt).

Die oben beschriebene Veränderung bringt mehrere Vorteile. Zunächst wird die Fahrstufe nun absolut angegeben. Rückwärtsfahrt wird durch eine der beiden Kategorien „rückwärts schnell“ oder „rückwärts langsam“ erwirkt. Ein Decoder, der bei Richtungswechsel nicht empfangsbereit war, erkennt auch nachträglich die eingestellte Fahrtrichtung. Das Fahrpult bestimmt zudem jetzt die reale Fahrtrichtung. Der Richtungsumkehr-Befehl aus dem alten Motorola Format wird bei Motorola II nicht benötigt, jedoch wird er zum Erhalt der Kompatibilität weiterhin versendet. Neuere Decoder reagieren hierauf nicht.

Letztlich bieten die EFGH Bits die Möglichkeit, weitere Informationen zu übertragen. Diese Option wird genutzt, um die Sonderfunktionen F1...F4 zu schalten. Hierzu werden die Kategorien um vier Codes erweitert.

Funktion	E	F	G	H
F1	1	1	0	f
F2	0	0	1	f
F3	0	1	1	f
F4	1	1	1	f

Tabelle 7 – Kategorien zur Funktionssteuerung [7]

Die Variable f ist durch den gewünschten Zustand der Funktion ('0': aus, '1': ein) zu ersetzen. Die Fahrstufen-Bits ABCD bleiben beim Steuern von Funktionen unverändert.

Die Schaltkommandos für die Funktionen können in Verbindung mit bestimmten Fahrstufen dazu führen, dass gültige Datenpakete aus dem Motorola I-Format entstehen. Diese Sonderfälle müssen abgefangen werden, um unkoordiniertes Verhalten der älteren Decoder zu verhindern. Die Fälle sind unterhalb aufgelistet.

Datenpaket			E F G H (regulär)	E F G H (modifiziert)
FS=2	&	F1 aus	1 1 0 0	1 0 1 0
FS=10	&	F1 an	1 1 0 1	0 1 0 1
FS=3	&	F2 aus	0 0 1 0	1 0 1 0
FS=11	&	F2 an	0 0 1 1	0 1 0 1
FS=5	&	F3 aus	0 1 1 0	1 0 1 0
FS=13	&	F3 an	0 1 1 1	0 1 0 1
FS=6	&	F4 aus	1 1 1 0	1 0 1 0
FS=14	&	F4 an	1 1 1 1	0 1 0 1

Tabelle 8 – Sonderfälle des Motorola-Formats zwecks Abwärtskompatibilität [7]

Beispiel: Sonderfall FS=10 & F1 an

	A B C D	E F G H		A E B F C G D H
regulär	1 1 0 1	1 1 0 1	→	1 1 1 1 0 0 1 1
modifiziert		0 1 0 1	→	1 0 1 1 0 0 1 1

Tabelle 9 – Rechenbeispiel zu den Sonderfällen

Die reguläre Bitfolge würde einen gültigen Motorola I-Befehl ergeben. Daher werden die EFGH Bits durch eine vordefinierte Bitfolge ersetzt.

Datenpakete

Nachdem der Aufbau eines Datenpakets vollständig erklärt ist, ist zu definieren, wann und wie häufig es ausgesendet werden soll. Jeder mit dem Motorola-Format kompatible Decoder erwartet grundsätzlich zwei identische und an ihn adressierte Datenpakete in kurzem zeitlichen Abstand, um die darin enthaltene Anweisung zu akzeptieren und umzusetzen. Es werden aus diesem Grund immer zwei gleiche Pakete hintereinander ausgesendet. Dieses Muster wird im Folgenden als Doppelpaket bezeichnet. Da bereits ein Bitfehler in einem Doppelpaket dazu führt, dass der Decoder die Daten nicht akzeptiert, werden immer zwei Doppelpakete, also vier reguläre Pakete, nacheinander versendet. Die Sendepause zwischen zwei einzelnen Paketen dauert $T_1 = 1,25$ ms. Zwischen zwei Doppelpaketen vergeht $T_2 = 4$ ms [7].

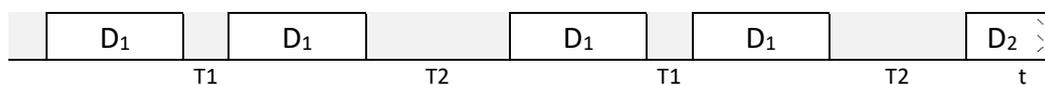


Abbildung 6 – Sendereihenfolge der Datenpakete beim Motorola-Format

Die Sendereihenfolge der verschiedenen Datenpakete ist im Folgenden aufgelistet. Datenpakete werden erst dann periodisch an einen Decoder gesendet, wenn die zugehörige Adresse erstmals am Fahrpult eingegeben wurde. Bis zu diesem Zeitpunkt werden Pakete mit der Nulladresse ausgesendet, damit alle angeschlossenen Verbraucher bereits mit Wechselspannung versorgt werden. Ist der erste Decoder angemeldet, werden abwechselnd je zwei Doppelpakete mit Geschwindigkeits- und Richtungsinformation bzw. Geschwindigkeits- und einer Funktions-Information gesendet. Die Richtung wird also bei jedem zweiten Sendevorgang übertragen, während alle Funktionen erst nach acht Vorgängen übermittelt werden.

Anz.	Datenpaket
4x	Geschwindigkeit & Richtung
4x	Geschwindigkeit & F1
4x	Geschwindigkeit & Richtung
4x	Geschwindigkeit & F2
4x	Geschwindigkeit & Richtung
4x	Geschwindigkeit & F3
4x	Geschwindigkeit & Richtung
4x	Geschwindigkeit & F4

Wiederholung

Tabelle 10 – Sendereihenfolge der Motorola-Datenpakete (1 Decoder angemeldet) [7]

Sind mehrere Decoder angemeldet, werden die Datenpakete an die verschiedenen Adressen in Schichten übertragen, sodass jeder Decoder in möglichst kleinen Zeitabständen Geschwindigkeits- und Richtungsinformationen erhält.

Anz.	Datenpaket	Decoder
4x	Geschwindigkeit & Richtung	Decoder A
4x	Geschwindigkeit & F1	Decoder A
4x	Geschwindigkeit & Richtung	Decoder B
4x	Geschwindigkeit & F1	Decoder B
4x	Geschwindigkeit & Richtung	Decoder A
4x	Geschwindigkeit & F2	Decoder A
4x	Geschwindigkeit & Richtung	Decoder B
4x	Geschwindigkeit & F2	Decoder B

U.S.W.

Tabelle 11 – Sendereihenfolge der Motorola-Datenpakete (2 Decoder angemeldet) [7]

3.2.2 Endstufe

Zweck des Energie-Daten-Signals ist es, digitale Datensignale gemeinsam mit elektrischer Energie über nur zwei Leiter zu übertragen. Da bei den Fahrzeugen insbesondere durch den Fahrmotor eine hohe Stromaufnahme zu verzeichnen ist, ist eine Endstufe notwendig, um den Ausgang des das Signal erzeugenden Schaltkreises nicht zu überlasten.

Anforderung an die Endstufe ist, dass sie aus einer Eingangsspannung mit geringen Spannungspegeln (in diesem Fall CMOS 3,3 V) eine belastbare Ausgangsspannung generiert. Ströme von 3-5 A sind auf kleineren Modellbahnanlagen üblich und sollten für die Endstufe kein Problem darstellen. Die Ausgangsspannung muss zwischen der positiven und der negativen Betriebsspannung U_V umgeschaltet werden können. Ferner sollen Schaltzeiten, Verluste und Signalverzerrungen möglichst niedrig sein.

MOSFETs sind für die Anwendung in dieser Endstufe gut geeignet, da sie große Ströme schalten können und zudem geringe Schaltverzögerungen (i.d.R. wenige Nanosekunden) haben. Zudem sind sie leicht zu verschalten, in vielen Varianten erhältlich und kostengünstig.

Zwei MOSFETs können zu einer Halbbrücke verschaltet werden, womit das Ausgangssignal (Drain-Spannung U_{D2} des unteren MOSFET) zwischen zwei Spannungen (U_{D1} und U_{S2})

umgeschaltet werden kann. Die Schaltung wird in Motortreibern oder Wechselrichtern verwendet.

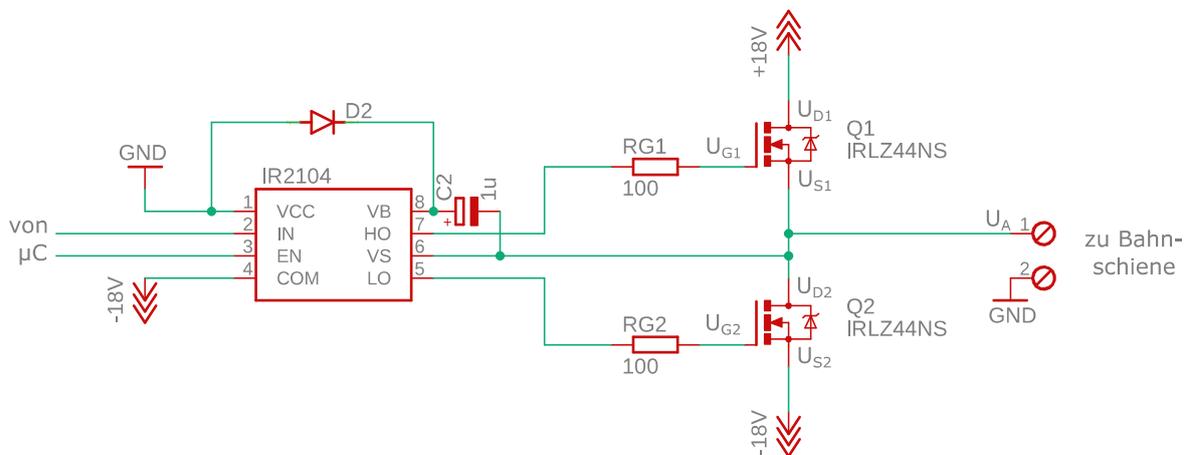


Abbildung 7 – Motorola-Endstufe: Halbbrücke mit Treiber (alle Spannungen beziehen sich auf GND als Bezugspotential)

Die verwendeten n-Kanal MOSFETS des Herstellers Infineon mit der Bezeichnung IRLZ44N eignen sich für Ströme bis zu 47 A. Der geringe Einschaltwiderstand $R_{DS,on} = 0,022 \Omega$ verringert die Abwärme und begünstigt den Wirkungsgrad. Die Schaltverzögerungen befinden sich im unteren zweistelligen Nanosekunden-Bereich [8]. In Anbetracht der kleinsten zu erwartenden Schaltperiode $T_{min} = \frac{1}{8}T = \frac{208 \mu s}{8} = 26 \mu s$ fallen diese nicht stark ins Gewicht. Insbesondere in Bezug auf die im Folgenden diskutierte Treiberstufe sind keine Probleme durch zu lange Schaltverzögerungen zu erwarten.

Die Halbbrücke ist in der rechten Bildhälfte der Abbildung 7 erkennbar. Die Ausgangsspannung kann durch gezieltes Umpolen der Gates umgeschaltet werden. Konkret ergeben sich folgende Schaltmöglichkeiten.

U_{G1}	U_{G2}	U_A
LOW	LOW	Hochohmig
LOW	HIGH	$-U_V$
HIGH	LOW	$+U_V$
HIGH	HIGH	Kurzschluss

Tabelle 12 – Schaltzustände der Halbbrückenschaltung

Werden beide Gates gleichzeitig angesteuert (HIGH), entsteht ein Kurzschluss, der zu Bauteilzerstörungen führt. Dieser Fall ist unbedingt zu vermeiden. Da sich aufgrund der Schaltverzögerungen der MOSFETs bei gleichzeitiger Umschaltung beider Gates auf den jeweils anderen Pegel kurze Querströme ergeben können, sollten immer erst beide Gates eine Weile LOW gehalten werden, bevor wieder ein HIGH-Pegel auf eines der Gates ausgegeben wird. Hierzu dienen Halbbrückentreiber, die als integrierte Schaltkreise erhältlich sind. Diese bringen in die Ansteuerung der MOSFET-Gates eine Totzeit ein, die Querströme verhindert. Die Totzeit beträgt bei dem verwendeten IR2104 Halbbrückentreiber-IC von Infineon $t_{tot} = 520 \text{ ns}$. Während dieser Zeit ignoriert der IC das Eingangssignal und hält beide Gates LOW.

Der IC löst gleichzeitig ein weiteres Problem der Halbbrücke: Die MOSFETs leiten dann, wenn die Gate-Source-Spannung U_{GS} oberhalb der Schwellenspannung $U_{GS0} = 2\text{ V}$ liegt. Da sich jedoch das Source-Potential von Q1 mit der Ausgangsspannung ändert, würde beim Einschalten von Q1 U_{GS} sofort wieder unter U_{GS0} fallen. Hierfür wird der Treiber-IC mit einem externen Kondensator beschaltet, der mit der Kathode auf dem Potential der Ausgangsspannung liegt. Über eine Diode wird der Kondensator von der Versorgungsspannung V_{CC} geladen, wenn Q1 nicht leitet. Wird Q1 angesteuert, erhöht sich die Ausgangsspannung und damit das Potential an der Kathode des Kondensators. Dadurch wird auch das Anoden-Potential um den gleichen Wert angehoben. Der IC kann damit U_{GS} für ausreichende Zeit auf einem konstanten Wert halten. Die ausgegebene Gate-Spannung des Low side-FET (Q2) kann nach dem Einschalten konstant bleiben, da hier U_S konstant ist.

Da die Versorgungsspannung V_{CC} des Treiber-IC 20 V nicht übersteigen sollte, wird der COM Kontakt an $-U_V$ angeschlossen, während V_{CC} mit der Masse verbunden ist. COM kann nicht an die Masse angeschlossen werden, da die Gate-Spannung für Q2 dann im LOW-Zustand 0 V betragen würde, wodurch Q2 dauerhaft leiten würde ($U_{GS2} = 0\text{ V} - (-18\text{ V}) = 18\text{ V}$).

Der Treiber-IC wird vom Mikrocontroller am IN Kontakt mit dem Motorola-Signal angesteuert. Das belastbare Energie-Daten-Signal U_B wird generiert, sobald der Enable (EN) Kontakt HIGH ist. Ist EN LOW, ist der Zustand von U_B hochohmig. Der Controller muss sich auf demselben Bezugspotential befinden, wie der Treiber-IC.

3.2.3 Zusatzbeschaltung

Nachdem die Grundlagen der Zugsteuerung diskutiert wurden, soll nun eine Platine entworfen werden, die die in Absatz 3.2.2 beschriebene Grundschaltung mit benötigten Zusatzschaltungen vereint. Die Platine erhält den Namen „Hauptmodul“, da sie im weiteren Projektverlauf neben der Generierung des Energie-Daten-Signals auch als zentrale Kommunikationsschnittstelle dient. Die notwendigen Erweiterungen der bereits bekannten Halbbrückenschaltung werden im folgenden Teil beschrieben.

Galvanische Trennung des Controllers

Dass der Massekontakt des Mikrocontrollers auf $-U_V$ liegt, kann in Verbindung mit anderen Teilsystemen zu Problemen führen, wenn diese ein anderes Bezugspotential verwenden. Daher wird eine galvanische Trennung der beiden Steuerleitungen vom Controller zum Treiber-IC vorgenommen, sodass der Controller die reguläre Masse verwenden kann. Hierbei kommen Optokoppler zum Einsatz, die gleichzeitig eine Pegelumsetzung ermöglichen. Die Pegel des Controllers (0 V LOW, $3,3\text{ V}$ HIGH) werden in zum IC kompatible Pegel ($-U_V$ LOW, $\text{min. } -U_V + 3\text{ V}$ HIGH) umgesetzt.

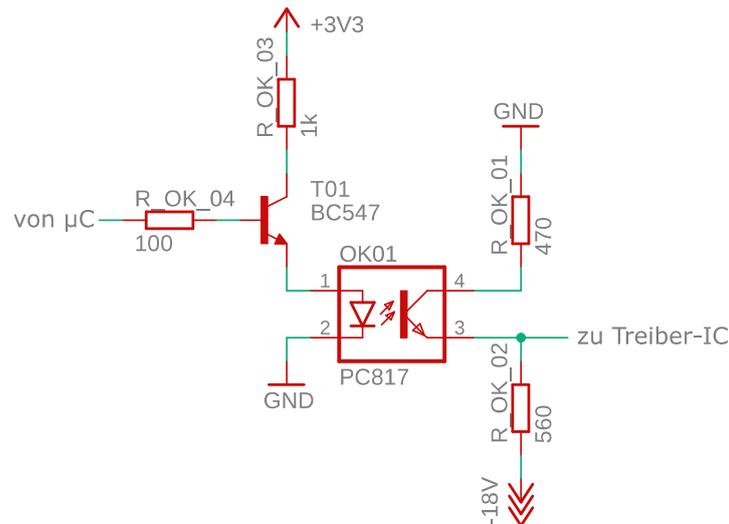


Abbildung 8 – Motorola-Endstufe: Galvanische Trennung zwischen Controller und Treiber-IC mittels Optokoppler

In der gezeigten Schaltung bewirkt ein HIGH Pegel des Controllers die Ansteuerung des npn-Bipolartransistors T01. Dieser lässt einen konstanten Strom durch die Leuchtdiode im Inneren des Optokopplers (Sharp PC817) fließen. Mit $U_F = 1,2 \text{ V}$ gilt (ohne Einberechnung des Basisstroms) für den Diodenstrom $I_F = \frac{3,3 \text{ V} - 1,2 \text{ V}}{1 \text{ k}\Omega} = 2,1 \text{ mA}$ [9]. Auf der galvanisch getrennten Seite lässt der interne Fototransistor einen Strom von GND über die Widerstände R_{OK01} und R_{OK02} fließen. Der Abgriff des Treiber-Eingangs erfolgt am Emitter, wodurch das Signal nicht invertiert wird. Ungeachtet des Transistors bilden die beiden Widerstände einen unbelasteten Spannungsteiler (sehr hoher Eingangswiderstand des IC). Hierdurch wird der HIGH Pegel auf etwa $-\frac{1}{2}U_V$ herabgesetzt. Mit $U_V = 18 \text{ V}$ ergibt sich genauer:

$$U_2 = U_V \cdot \frac{R_2}{R_1 + R_2} = -18 \text{ V} \cdot \frac{560 \Omega}{470 \Omega + 560 \Omega} \approx -9,79 \text{ V}$$

Der IC erkennt HIGH Pegel zuverlässig oberhalb einer Eingangsspannung von 3 V gegenüber der Masse (hier also -15 V). Die Halbierung der Spannung ist daher nicht problematisch. Die symmetrische Verteilung der beiden Widerstände um den Fototransistor bewirkt, dass beide Pegelwechsel etwa gleich schnell ausgeführt werden.

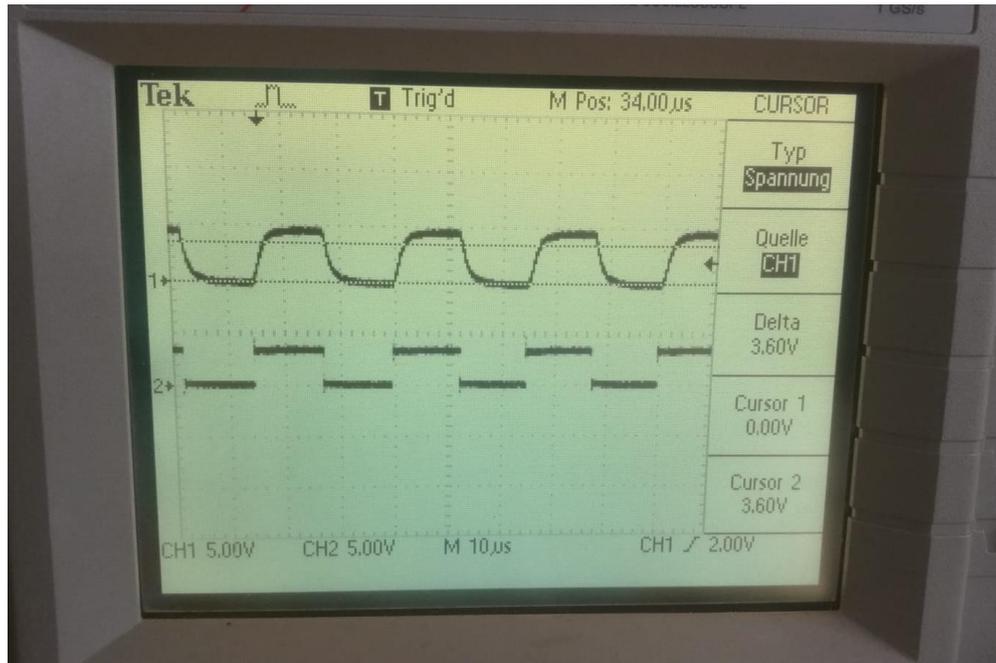


Abbildung 9 – Optokoppler Eingangssignal (CH 2) und Ausgangssignal (CH1)

Mikrocontroller

Als Mikrocontroller für die Kommunikation und die Erzeugung des Motorola-Signals kommt ein Texas Instruments TM4C123GH6PM auf dem TM4C123GXL Evaluationsboard zum Einsatz. Der Controller bietet u.a. folgende für diese Anwendung relevante Eigenschaften/Funktionen.

- ARM Cortex-M4F 32 Bit Mikroprozessor
(Maximale Taktrate: 80 MHz)
- 12 Timer-Module
(6x 16/32 Bit und 6x 32/64 Bit)
- 2 PWM-Module
(je 4 Generator-Blöcke, insg. 16 Outputs)
- 8 UART-Module
- 4 I²C-Module
- 6 GPIO-Blöcke
(je 8 IO-Pins, insg. 48 Pins)

Tabelle 13 – Features des TM4C123G Mikrocontrollers [10]

Das Evaluationsboard besitzt einen weiteren Chip, der das sogenannte In-Circuit Debug Interface (ICDI) bereitstellt. Dieses bietet die Möglichkeit, den Hauptchip via USB zu programmieren, debuggen und eine der UART Schnittstellen anzusprechen. Dies ist sehr hilfreich beim Verfassen und Testen des Quellcodes, beim Einrichten einer Kommunikation mit einer Desktop-Software und macht ein externes Gerät zum Flashen des Programmspeichers überflüssig. Das Hauptmodul integriert zwei Pinheader, auf welche das Evaluationsboard aufgesteckt werden kann. Hierdurch können Platine und Board separat getestet werden.

Pegelwandlung

Der verwendete Mikrocontroller arbeitet mit CMOS 3,3 V Logikpegeln. Der in 3.4.1 vorgestellte Datenbus soll aus Gründen der Kompatibilität mit den Stellwerk-Controllern (siehe 3.3) und zur Verbesserung des Störabstandes (Signal-Rausch-Abstand) mit TTL 5 V Pegeln betrieben werden. Hierzu sind den zugehörigen Pins des Hauptmodul-Controllers Pegelwandler nachzuschalten. Diese müssen bidirektional arbeiten und sollten möglichst schnell und effizient sein.

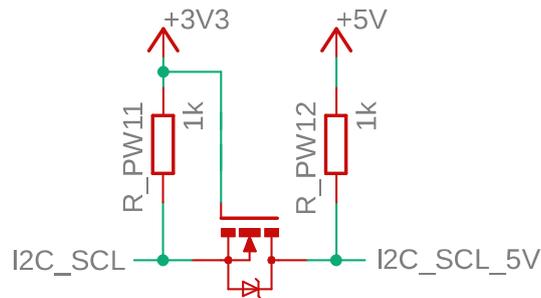


Abbildung 10 – Hauptmodul: Pegelwandler 3,3 V ↔ 5 V für I²C Kommunikation

Die Funktionsweise des gezeigten Pegelwandlers zeigt sich durch Betrachtung von vier Zuständen.

Eingang	Pegel	Beschreibung
I2C_SCL	LOW	$U_{GS} = 3,3 V$, MOSFET schaltet durch und zieht Ausgang auf LOW
I2C_SCL	HIGH	$U_{GS} = 0 V$, MOSFET sperrt und Ausgang wird über R_{PW12} auf HIGH gezogen
I2C_SCL_5V	LOW	Bodydiode leitet und zieht Ausgang auf (fast) LOW (Vorwärtsspannung der Diode), MOSFET wird leitend ($U_{GS} = 3,3 V$) und zieht Ausgang auf LOW
I2C_SCL_5V	HIGH	Ausgang wird über R_{PW11} auf HIGH gezogen, MOSFET sperrt

Tabelle 14 – Zustände des bidirektionalen Pegelwandlers

Die bei diesem Pegelwandler verwendeten Widerstände ersetzen gleichzeitig die bei einem I²C-Bus nötigen Pullup-Widerstände, die die Open-Drain-Pins der kommunizierenden Chips auf Versorgungspotential ziehen. Es werden 2N7002 n-Kanal Kleinsignal-FETs in SMD-Bauweise von Infineon verwendet.

Spannungsversorgung

Zur VCC-Spannungsversorgung des Controllers, der Pegelwandler und eventuell benötigter weiterer Verbraucher wird eine stabile 5 V Spannungsversorgung benötigt. Es kommt daher der Schaltregler LM2678-5.0 von Texas Instruments zum Einsatz. Die Verschaltung ist an das im Datenblatt gezeigte Beispiel angelehnt [11].

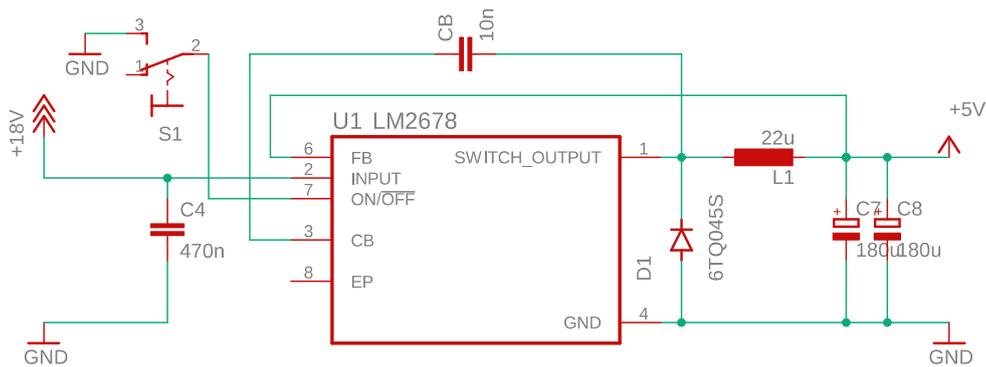


Abbildung 11 – Hauptmodul: 5 V Spannungsversorgung mittels Schaltregler

Der Ausgang ist mit bis zu 5 A belastbar. Der interne MOSFET beaufschlagt die Induktivität L_1 mit einem 260 kHz PWM-Signal [11]. Während der HIGH Phase baut sich in der Spule ein Magnetfeld auf. Durch den anschließenden Zusammenbruch des Magnetfelds erzeugt die Spule einen hohen Strom in Richtung des Ausgangs (Selbstinduktion). Die Schottky-Diode D_1 lässt diesen Strom fließen und schützt damit den Ausgang des Reglers. Durch den Feedback-Pin (FB) passt der Regler den Tastgrad des PWM-Signals an den Belastungszustand des Ausgangs an. Der Kondensator C_B ermöglicht schnelles Umladen des Gates des internen MOSFETs.

Das Evaluationsboard wird über den VBUS-Pin mit VCC verbunden. Die 3,3 V Betriebsspannung erzeugt es über einen internen Linearspannungsregler.

3.2.4 Platinenlayout

Beim Entwurf des Platinenlayouts wird besonders auf ausreichende Kühlung der Leistungsbauteile achtgegeben. Daher wird überwiegend SMD-Bestückung vorgesehen. Je nach Anforderung werden $\frac{1}{8}$ W Widerstände in 0805 Bauweise oder $\frac{2}{3}$ W Widerstände in 1206 Bauweise eingeplant. Die Leistungstransistoren (MOSFETS), der Schaltregler und die zugehörige Diode sind als D²PAK (Double Decawatt Package, auch TO-263) Variante ausgeführt. Die MOSFETS für die Pegelwandlung haben SOT-23 Gehäuse.

Das Layout sieht an der oberen linken Ecke einen Schalter vor, der den Pegel am ENABLE-Pin des Schaltreglers steuert und damit die VCC-Spannung ein und ausschalten kann. Weiterhin gibt es einige Steckverbinder. Zur Kommunikation mit bis zu vier Stellwerken (siehe Abschnitt 3.3) sind dem I²C-Pegelwandler vier Steckverbinder nachgeschaltet, die an die Daten und Taktleitung (SDA, SCL) angebunden sind. An der linken unteren Ecke befinden sich die Versorgungs-Anschlüsse für die Stellwerke. Die vierpoligen Stecker sind belegt mit U_V , U_B und GND (2x). Über den einzelnen Stecker am linken Platinenrand wird das Hauptmodul mit Spannung versorgt ($+U_V$, $-U_V$ und 2x GND).

Die Platine ist 150x100 mm groß und besteht aus FR4-Platinenmaterial.

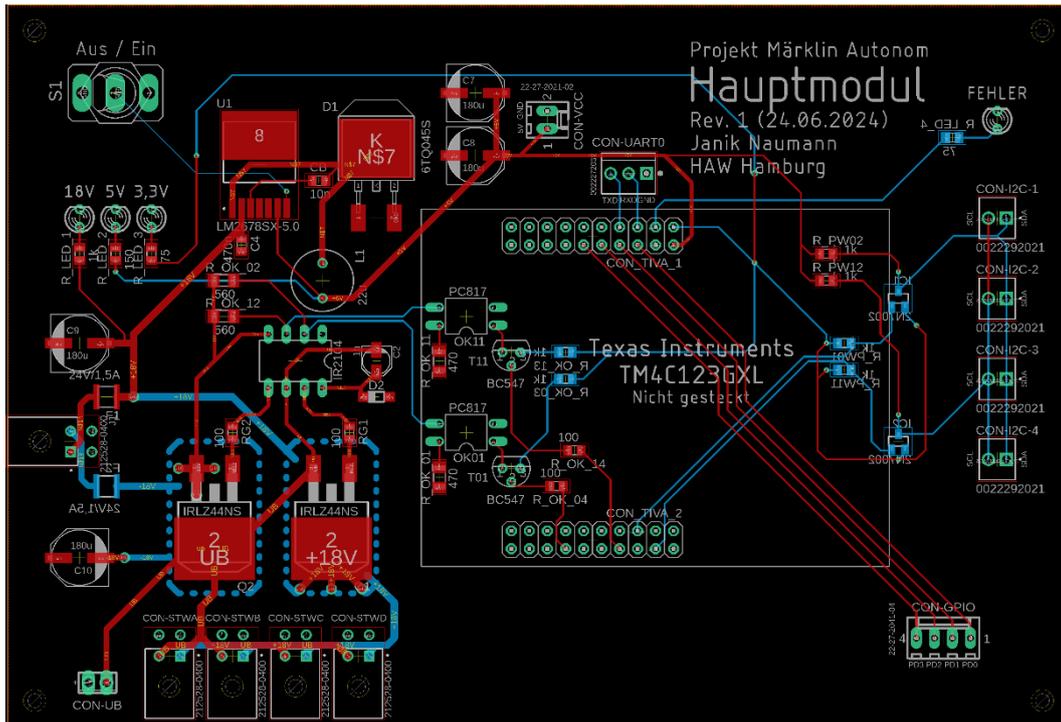


Abbildung 12 – Platinenlayout Hauptmodul

Das Bestücken der Platine mit den SMD-Komponenten wird mittels einer Heißluft-Lötstation in Verbindung mit Lötpaste durchgeführt.

3.3 Elektronisches Stellwerk (ESTW)

In dem zu entwerfenden Zugleit- und Sicherungssystem werden ausschließlich Stellwerke des Typs ESTW (siehe 2.2.3) eingesetzt, die in der prinzipiellen Funktionsweise dem Stand der Technik entsprechen. Jedes Stellwerk besteht aus einer Platine (Stellwerkmodul), die jeweils mehrere der in den folgenden Abschnitten vorgestellten Grundschaltungen vereint.

3.3.1 Weichensteuerung und Rückmeldung

Elementar für jedes Stellwerk ist die Steuerung von Weichen. Für die Ansteuerung von Modellbahn-Weichen, die mit den sogenannten Magnetartikeln ausgestattet sind, ist zunächst ein Verständnis für die Funktion wichtig.

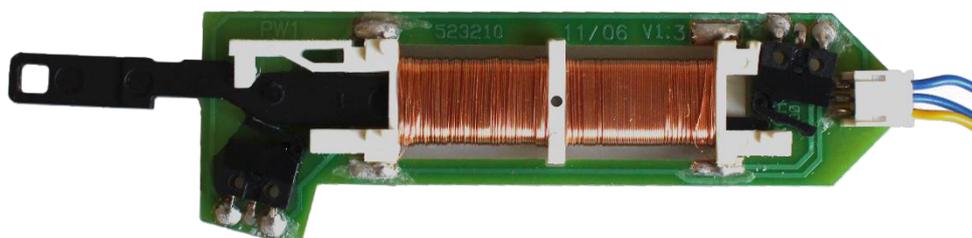


Abbildung 13 – Märklin Weichenantrieb ohne Gehäuse (Artikelnr. 74491)

Der Antrieb besteht aus zwei elektrisch und räumlich voneinander getrennten Spulen, die ein Kunststoffgehäuse umschließen. Im Inneren des Gehäuses befindet sich ein Schieber, an dem ein ferromagnetisches Metallstück befestigt ist. Durch abwechselndes Beaufschlagen der beiden Spulen mit Spannung wird jeweils ein Magnetfeld erzeugt, das den metallischen Teil des Schiebers anzieht. So wird der Schieber im Gehäuse hin und her

bewegt. Das auf einer Seite herausragende Stück des Schiebers wird mit dem Schaltmechanismus der Weiche verbunden werden. So kann die Weiche mittels elektrischer Ansteuerung gestellt werden.

Die Ansteuerung des Stellantriebs erfolgt über drei Leitungen. Die erste Leitung (gelbes Kabel) versorgt beide Spulen dauerhaft mit Versorgungsspannung oder Masse (je nach gewählter Verschaltung). Durch Verbinden einer der beiden blauen Leitungen mit dem anderen Potential wird der Stromkreis einer Spule geschlossen und die Weiche umgestellt. Zwei Endschalter, die vom Schieber betätigt werden, verhindern, dass die Spulen zu lange angesteuert werden, was zu Bauteilerstörungen führen kann. Diese Schalter verleihen der Weiche implizit eine Rückmeldefähigkeit. Da jeweils eine Steuerleitung mit einem intern geschlossenen Stromkreis und die andere mit einem offenen Stromkreis verbunden ist, lässt sich durch eine separate Elektronik ein Signal erzeugen, das den Ist-Zustand der Weiche anzeigt.

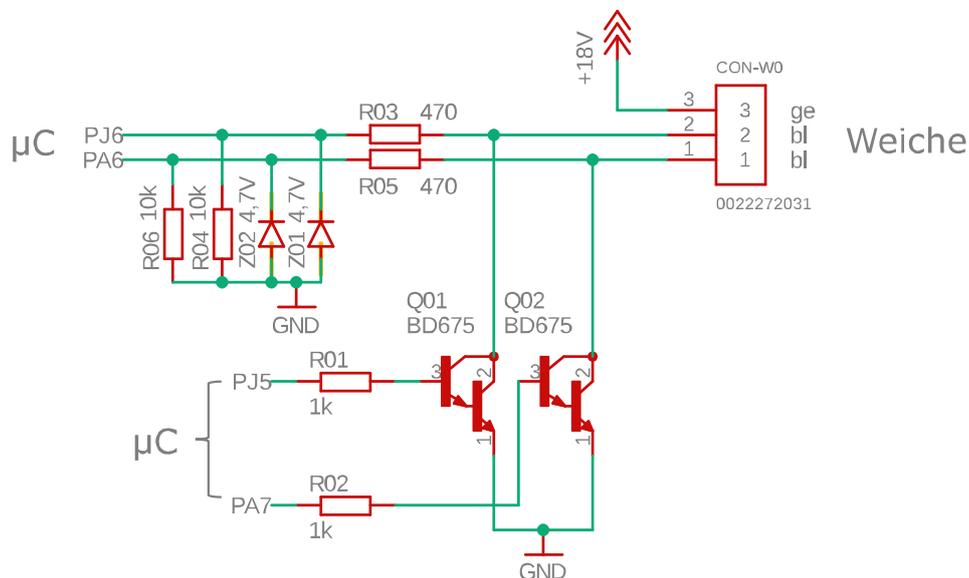


Abbildung 14 – Stellwerk-Grundschtung: Weichensteuerung und Rückmeldung

Die gezeigte Schaltung ist mit vier Pins des Stellwerk-Controllers verbunden. Über die zwei unteren Leitungen (PJ5, PA7) steuert der Controller die Halbleiterschalter (hier Darlington-Transistoren) Q01 und Q02 an. Diese verbinden die jeweilige Steuerleitung mit Masse. Da die Versorgungsleitung auf $+U_V$ liegt, wird, sofern der Stromkreis über die Schalter im Stellantrieb geschlossen ist, die Weiche umgestellt. Wird derzeit kein Transistor angesteuert, so generiert der obere Schaltungsteil zwei digitale Zustandsbits. Dazu verringern die Widerstände R03 und R05 den Strom in den Schaltungsteil und die Zenerdioden Z01 und Z02 leiten Spannungen oberhalb von $4,7\text{ V}$ gegen Masse ab, da auf den Steuerleitungen Spannungen bis zu 24 V anliegen, die einen Mikrocontroller zerstören würden. Die Widerstände R06 und R04 ziehen die Spannung auf Masse, wenn der Stromkreis der jeweiligen Steuerleitung offen ist. Liegt eine Steuerleitung auf U_V , wird ein TTL-Kompatibles HIGH Signal ausgegeben. An den Ausgängen PJ6 und PA6 kann daher von einem Controller der Ist-Zustand der Weiche ermittelt werden. Konkret existieren vier Fälle.

PJ6	PA6	Beschreibung
LOW	HIGH	Weiche in Endlage, kann in andere Position gebracht werden, wenn HIGH Signal auf PA7
HIGH	LOW	Weiche in Endlage, kann in andere Position gebracht werden, wenn HIGH Signal auf PJ5
LOW	LOW	Störung: Weiche oder Verdrahtung fehlerhaft
HIGH	HIGH	Störung: Weiche nicht in Endlage

Tabelle 15 – Belegung der Weichen-Zustandsbits

3.3.2 Signale

Die Signalanlage hat bei der realen Eisenbahn einen sehr hohen Stellenwert, da sie den Fahrzeugführern wichtige Informationen liefert und Befehle erteilt. In diesem Projekt sind Signale jedoch nicht direkt sicherheitsrelevant, da alle Züge zentral gesteuert werden und es daher keinen Fahrzeugführer gibt, der auf sie reagieren könnte. Daher werden die Signale hier stark simplifiziert und als einfache Rot/Grün-Leuchtmelder ausgeführt. Es werden ausschließlich Hauptsignale verbaut.

Um die Signale, die aus zwei LEDs bestehen, über den Mikrocontroller anzusteuern, werden jeweils zwei Halbleiterschalter (hier n-Kanal Kleinsignal-MOSFETs 2N7002) und Vorwiderstände benötigt. Die FETs werden als Low-side-switches verwendet. Die LEDs sind mit der Anode mit VCC verbunden. Die Widerstände R81 und R82 begrenzen den Strom durch die LEDs. Bei einer Vorwärtsspannung $U_F = 1,8 V$ beträgt der Diodenstrom $I_D = \frac{5 V - 1,8 V}{470 \Omega} \approx 6,8 mA$. Die LEDs mit dem Durchmesser von 3 mm leuchten bei dieser Stromstärke ausreichend hell.

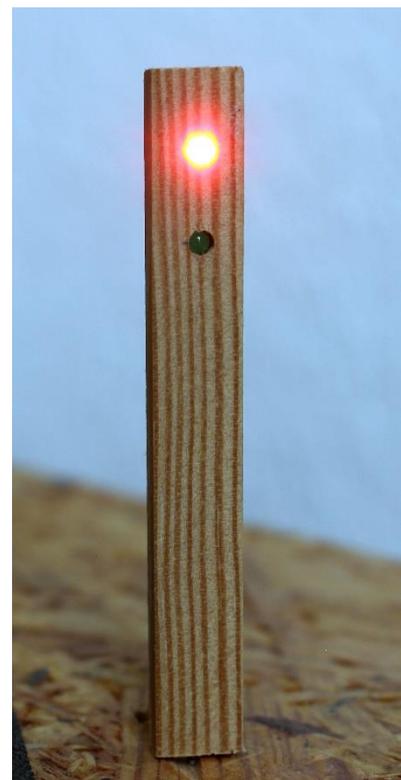


Abbildung 15 – Halt zeigendes Signal

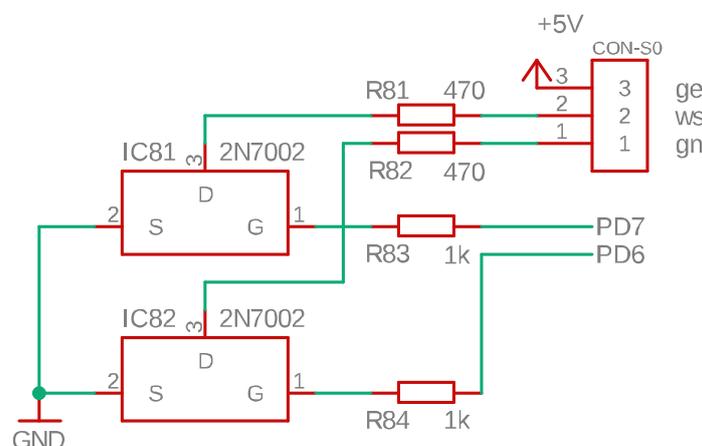


Abbildung 16 – Stellwerk-Grundschtung: Signalsteuerung

An die gezeigte Schaltung wird das Signal über drei Leitungen (eine Versorgungs- und zwei Steuerleitungen) angebunden. Ein HIGH Pegel an PD7 bzw. PD6 bewirkt ein aufleuchten der roten bzw. grünen LED. Die Nutzung der Möglichkeit, beide LEDs gleichzeitig ein oder auszuschalten, ist zunächst nicht vorgesehen, kann jedoch für das Anzeigen von Störungen oder speziellen Befehlen verwendet werden.

3.3.3 Gleisstromabschaltung

Wie bereits zuvor erwähnt wurde, können Signale in diesem Projekt den Bahnverkehr nicht beeinflussen. Ferner gibt es bislang keine Möglichkeit, fahrende Züge im Gefahrenfall durch eine Zugbeeinflussung zum Stehen zu bringen. Versuche der Einrichtung einer punktförmigen Zugbeeinflussung wurden eingestellt, da sich aufgrund der sehr engen Bauformen der Fahrzeuge eine Unterbringung der benötigten Fahrzeugausrüstungen als nicht umsetzbar erwies. Außerdem würde ein beeinflusstes Fahrzeug den Fahrbetrieb nicht ohne Eingreifen von außen wiederaufnehmen können.

Ersatzweise soll die Strecke mit einer Zugbeeinflussung ausgerüstet werden, die sicher ist und keine Fahrzeugausrüstung benötigt. Als eine diese Anforderungen erfüllende und einfach einzurichtende Methode erweist sich die Gleisstromabschaltung. Über ein Signalrelais kann in kritischen Situationen der Mittelleiter der Schienen im betroffenen Gleisabschnitt spannungsfrei geschaltet werden. Die folgende Grundsaltung zeigt, wie eine Gleisstromabschaltung realisiert werden kann.

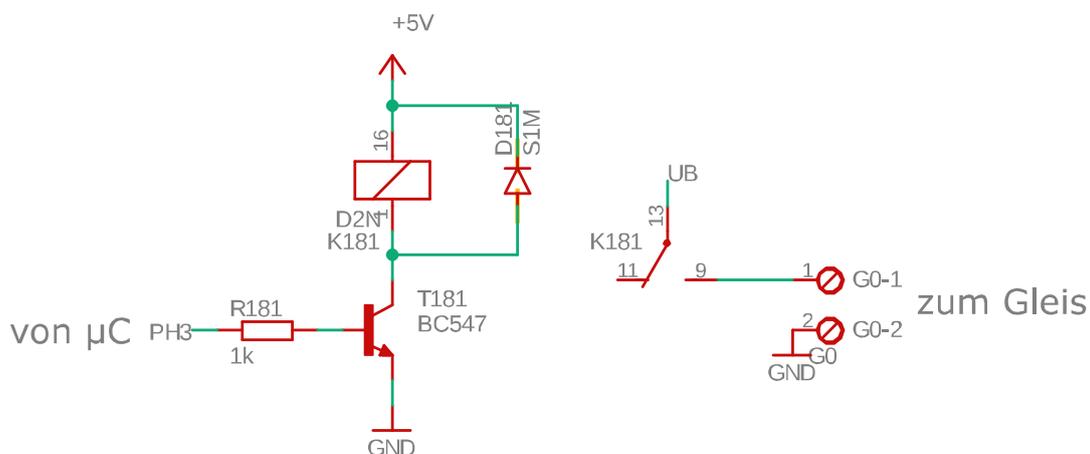


Abbildung 17 – Stellwerk-Grundsaltung: Gleisstromabschaltung

Damit die Abschaltung funktioniert, müssen die Mittelleiter an den einzelnen Gleisabschnittsgrenzen unterbrochen sein. Alle Stellwerke werden vom Hauptmodul über den Versorgungsstecker mit dem Gleisstrom (Energie-Daten-Signal) versorgt. Jeder Abschnitt erhält eine eigene Einspeisung des Gleisstroms, die durch ein Signalrelais unterbrochen werden kann. Die Magnetspulen der Relais werden vom Stellwerk-Controller über einen npn-Bipolartransistor in Emitterschaltung ein und ausgeschaltet. Die Diode D181 (DIOTEC S1M) dient als Freilaufdiode für das Relais. Die verwendeten Relais des Typs TE Connectivity Axicom D2N arbeiten mit einer Steuerspannung $U_L = 5\text{ V}$ und erlauben einen Arbeitsstrom von $I_R = 3\text{ A}$ [12]. Anhand der im Datenblatt angegebenen Spulen-Leistung

$P_L = 150 \text{ mW}$ und des ohmschen Widerstands $R_L = 167 \Omega$ lässt sich der Spulenstrom berechnen.

$$P_L = R_L \cdot I_L^2 \Leftrightarrow I_L = \sqrt{\frac{P_L}{R_L}} = \sqrt{\frac{150 \cdot 10^{-3} \text{ W}}{167 \Omega}} \approx 30 \text{ mA}$$

Der Bipolartransistor des Typs BC547 B ist für einen Kollektorstrom $I_{C,\text{max}} = 100 \text{ mA}$ ausgelegt [13]. Der Basiswiderstand $R_B = 1 \text{ k}\Omega$ erwirkt einen Basisstrom

$$I_B = \frac{U_B}{R_B} = \frac{5 \text{ V} - 0,7 \text{ V}}{1 \text{ k}\Omega} = 4,3 \text{ mA}$$

Damit befindet sich der Transistor im Sättigungsbetrieb. Bei einer Verstärkung $B = 300$ beträgt der maximale Kollektorstrom

$$I_C = I_B \cdot B = 4,3 \text{ mA} \cdot 300 = 1,29 \text{ A}$$

3.3.4 Gleisrückmeldung

Für den autonomen Fahrbetrieb ist eine Erfassung der Position jedes Zuges von elementarer Relevanz. Wie im realen Vorbild ist das Streckennetz daher in Gleisabschnitte eingeteilt, die über die im Folgenden vorgestellte Methode als frei oder besetzt gemeldet werden.

Induktive Schienenkontakte nach dem realen Vorbild lassen sich an den Schienen einer Modellbahn nicht installieren, da diese sehr klein sind. Umfangreiche Versuche der Implementierung von Gleisstromkreisen und Gleisstrom-Achszählern wurden abgebrochen, weil sie nicht zuverlässig funktionierten. Aufgrund des geringen Gewichts der Wagons besteht oft nicht der nötige Anpressdruck der metallischen Achsen an das Gleis. Geringe Verunreinigungen der Schienen führen daher zu Kontaktproblemen. Zudem gibt es Wagen mit Achsen aus Kunststoff, die zu fälschlichen Gleisfreimeldungen führen könnten.



Abbildung 18 – Gleisstrom-Achszähler: Messung zweier Drehgestelle (je 2 Achsen); Messung des 2. DG fehlerhaft durch verschmutztes Rad

Der Ansatz, das Signal elektronisch oder softwaregesteuert aufzubereiten, ist nur schwer umsetzbar, da die Fahrzeuge nicht immer eine konstante Fahrtgeschwindigkeit haben und dies bei der Aufbereitung zu falschen Ergebnissen führen kann. Aus den genannten Gründen wird sich für eine Alternative entschieden, die auf einem zuvor an der HAW durchgeführten Projekt basiert [14].

Durch die Installation von Hall-Sensoren im Gleisbett und das Anbringen von Magneten unter den Fahrzeugen lassen sich zuverlässige Gleisfreimeldungen realisieren. Bei den Sensoren handelt es sich um Hall-Sensoren des Typs Infineon TLE 4905L mit digitalem Ausgangssignal. Als Fahrzeugausrüstung kommen kleine Neodymmagnete mit einem Durchmesser von 4 mm zum Einsatz. Die Sensoren passen genau zwischen zwei Schienenschwellen und ragen in der Höhe nicht über die Kontakte des Mittelleiters hinaus. Am Schleifer der Fahrzeuge ist also beim Überfahren des Sensors nicht mit Kontaktproblemen zu rechnen.



Abbildung 19 – Hall-Gleissensor

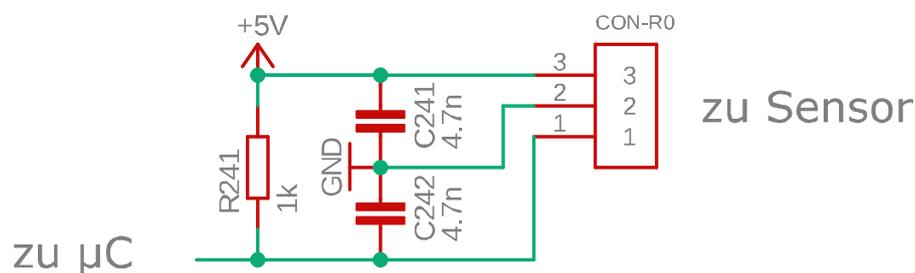


Abbildung 20 – Stellwerk-Grundschialtung: Hall-Sensor Beibeschialtung

Die dargestellte Grundschialtung beinhaltet einen Pullup-Widerstand, der den Open-drain-Ausgang des im Sensor integrierten MOSFETs auf VCC zieht, wenn dieser hochohmig ist. Wird ein Magnet detektiert, verbindet der FET den Pin des Mikrocontrollers mit Masse. Der Kondensator C241 gleicht Schwankungen der Versorgungsspannung aus, während C242 Glitches auf der Signalleitung gegen Masse abführt [15].

Die somit entstandene Freimeldung ist mit der in der Realität eingesetzten Achszählung vergleichbar, wobei hier nicht die Anzahl der Achsen, sondern die Anzahl der verbauten Magnete gezählt wird. Die Positionierung im Gleisbett erinnert optisch zudem an die beim ETCS verwendeten Eurobalisen.

3.3.5 Platinenlayout

Wie auch beim Layout des Hauptmoduls wird das Stellwerkmodul zu einem Großteil mit SMD Komponenten bestückt. Ebenso ist jedes Stellwerkmodul mit einer 5 V Spannungsversorgung ausgestattet. Die Zenerdioden für die Weichenrückmeldung haben ein SOD-80 (minimelf) Gehäuse, die Freilaufdioden für die Gleisstromrelais sind in DO-214AC Bauform ausgeführt.

Ein Modul sieht folgende Anzahl der jeweiligen Grundschaltungen vor:

- 8x Weichensteuerung und Rückmeldung
- 9x Signalsteuerung
- 6x Gleisstromabschaltung
- 15x Gleisrückmeldung

Tabelle 16 – Quantität der Grundschaltungen des Stellwerkmoduls

Jede Ausführung einer Grundschaltung wird als Schaltungssektion bezeichnet. Die Bestückung erfolgt je nach Anforderung. Es können beispielsweise von den acht möglichen Weichensteuerungs-Sektionen nur drei mit Bauteilen bestückt werden.

Die Steuerung aller Sektionen erfolgt über den Mikrocontroller, einen Atmel ATMEGA 2560 in TQFP-100 Bauform. Die 86 verfügbaren IO-Pins reichen aus, um alle Sektionen vollständig steuern zu können [16]. Mit der maximalen Taktrate von 20 MHz lassen sich die notwendigen, nicht sehr rechenintensiven Operationen schnell ausführen. Die Struktur und Funktion der Stellwerk-Software ist in 3.6 beschrieben.

Die Platine ist 250x120 mm groß und besteht aus FR4-Platinenmaterial.

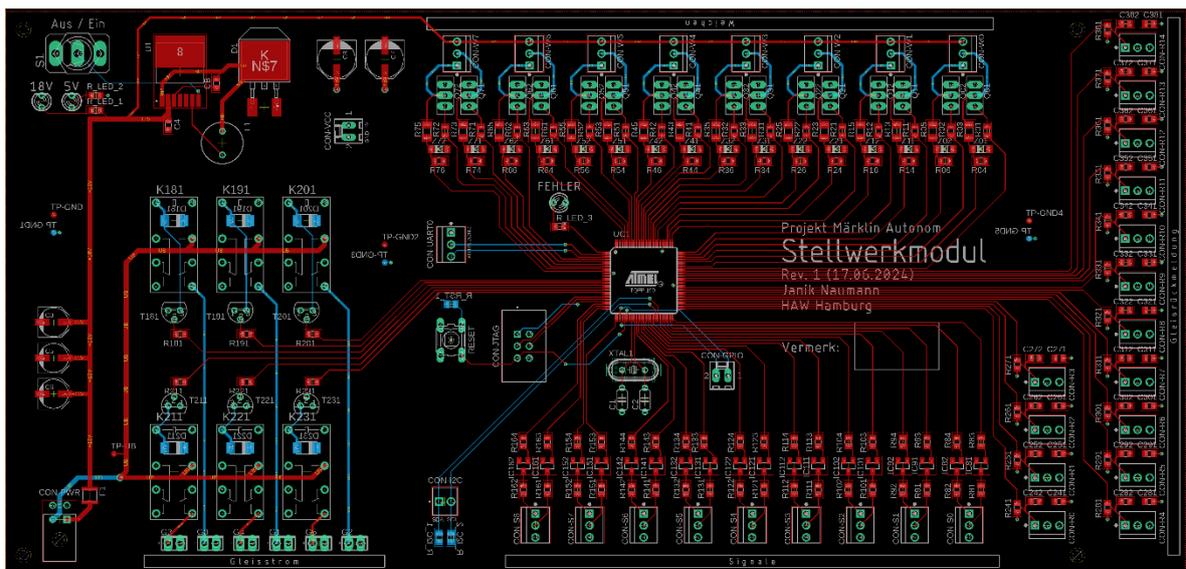


Abbildung 21 – Platinenlayout Stellwerkmodul

3.4 Kommunikation

3.4.1 Stellwerk-Bus

Der Stellwerk-Bus dient dem Datentransfer zwischen dem Hauptmodul und den Stellwerkmodulen. Das Hauptmodul sendet den Stellwerken Befehle/Anforderungen (Requests), worauf die Stellwerke mit kurzen Erfolgs-/Misserfolgsmeldungen antworten. Die Antwort (Answer) kann je nach Request auch Nutzdaten beinhalten.

Für die Erfüllung der genannten Anforderungen eignet sich ein I²C-Bus, da er nach dem Master-Slave-Prinzip arbeitet und die Standardvariante bis zu 127 Teilnehmer (Slaves) unterstützt. I²C wird in der Regel nur bei geringen Leitungslängen verwendet (wenige Zentimeter). Oft werden bei langen Leitungen (> 1m) Range-Extender eingesetzt, die die Spannungspegel auf dem Leitungstück anheben oder eine Impedanzwandlung vornehmen.

Grundsätzlich sind mit niedrigeren Datenübertragungsraten höhere Leitungslängen ohne Range-Extender implementierbar. Da die zwei Stellwerke in diesem Fall relativ nah am Hauptmodul installiert sind, ist ein Betrieb bei der ausreichenden Standardbitrate von 100 kBit/s ohne Zusatzbeschaltung möglich. Für den Fall einer Erweiterung des Busses um weitere Teilnehmer sind die Leitungslängen jedoch von höherer Relevanz. In dieser Implementierung soll der I²C-Bus mit TTL Spannungspegeln arbeiten. Der Controller des Hauptmoduls und die Stellwerk-Controller bieten eine Hardware-Schnittstelle für I²C, was die Implementierung des Stellwerk-Busses in Software stark vereinfacht.

Typische Requests ordnen das Stellen einer Weiche, das Umschalten eines Signals oder das Schalten eines Gleisstromrelais an. Weitere Requests fordern Statusinformationen an, beispielsweise über die Anzahl der zuletzt an einem Messpunkt gezählten Fahrzeugmagnete. Ein Request beginnt immer mit dem Request-Byte. Danach können mehrere Request-Parameter folgen. Beim Versand eines Requests befindet sich das Hauptmodul im Master Transmitter Mode, das Stellwerk agiert als Slave Receiver. Die folgende Tabelle listet alle Requests auf, die von den Stellwerken unterstützt werden.

	Request-Byte ¹	Parameter	Beschreibung
	REQ_NONE		Kein Request, STW reagiert bei Empfang mit Ping-Antwort
Setter	REQ_SET_TURNOUT	T P [F]	Soll-Pos. von Weiche T in Pos. P ändern (P=0: PLUS, P= 1: MINUS) Wenn F gesendet und TRUE, Stellung erzwingen (Störung wird dabei zurückgesetzt)
	REQ_SET_SIGNAL	I T	Signal I in Stellung T bringen (T=0: grün, sonst: rot)
	REQ_SET_SECTION	A T	Gleisstrom für Abschnitt A schalten (T=0: ein, sonst: aus)
	REQ_CLR_SENSOR	Z	Magnetzähler Z auf 0 setzen
Getter	REQ_GET_TURNOUTS		Fordert alle Weichen-Zustände an ²
	REQ_GET_SIGNALS		Fordert alle Signal-Zustände an
	REQ_GET_SECTIONS		Fordert alle Gleisstrom-Relais-Zustände an
	REQ_GET_SENSORS		Fordert alle Magnet-Zählerstände an

Tabelle 17 – Liste aller Stellwerk-Bus-Requests

Ein Stellwerk generiert als Reaktion auf jeden Request eine Antwort mit dem Status-Byte (ANS_OK oder ANS_NOK), gefolgt vom zuvor empfangenen Request-Byte, auf das sich die nachfolgenden Daten beziehen. Wird das empfangene Request-Byte nicht unterstützt, wird stattdessen REQ_NONE als zweites Antwort-Byte festgelegt. Abhängig vom Status-Byte und dem Request-Byte folgt dann eine definierte Anzahl an Nutzdaten-Bytes. Das Hauptmodul

¹ Die Zahlenwerte der Requests sind im Quellcode des Hauptmoduls und der Stellwerke in der Headerdatei requests.h definiert.

² Je Weiche werden vier Zustandsbits ausgegeben, die Kodierung ist im Quellcode des Stellwerk-Codes angegeben (track.h, Variable turnoutStates; track.c, Funktion getTurnouts())

kann die Answer empfangen, indem es als Master Receiver das Stellwerk in den Slave Transmitter Mode versetzt [17].

Die Hauptmodul-Software sieht vor, alle vier GET-Requests in regelmäßigen Abständen an die einzelnen Stellwerke zu senden. Somit sind alle Ist-Daten der Stellwerke auf dem Hauptmodul zwischengespeichert, wodurch die Desktop-Steuersoftware diese schneller abrufen kann. In Fällen, in denen die Daten zwingend aktuell sein müssen, kann die Steuersoftware einen Refresh-Request an das Hauptmodul senden, worauf dieses die nächste Aktualisierung der Daten des angegebenen Stellwerks zeitlich vorzieht. Das Hauptmodul zählt zudem Kommunikationsfehler mit den Stellwerken und versieht diese bei Erreichen eines Schwellenwertes mit einem Error-Flag.

3.4.2 Hauptmodul und Steuersoftware

Das Hauptmodul dient als zentrale Kommunikationsschnittstelle und vermittelt Daten zwischen der Steuersoftware und den einzelnen Stellwerken. Der Computer, welcher die Steuersoftware ausführt, ist über USB mit dem Hauptmodul-Controller verbunden. Das Evaluationsboard, auf dem sich der Controller befindet, ermöglicht es, über diese Verbindung eine UART-Kommunikation zu etablieren. Über den zugehörigen Treiber wird der Stellwerk-Controller für die Steuersoftware als COM Port (Communication port) zugänglich gemacht. Durch zielgerichtete Betriebssystemaufrufe kommt seitens der Steuersoftware die Kommunikation zustande. Das Betriebssystem liefert dabei einen Sende- und Empfangspuffer, der die Kommunikation flüssiger und ohne lange Verzögerungen in der Programmausführung ablaufen lässt. Der Controller antwortet dem Computer auf dieselbe Weise, wie er regulären UART-kompatiblen Geräten antworten würde.

Die Kommunikation erfolgt in Datenpaketen und ähnelt dabei der Kommunikation beim Stellwerk-Bus. Es gibt folgende Requests, die von der Steuersoftware an das Hauptmodul gesendet werden.

	Request-Byte	Parameter	Beschreibung
Hauptmodul-Steuerbefehle	'M' (0x4D)		Kein Request, Hauptmodul reagiert bei Empfang mit Ping-Answer 'O' (0x4F)
	REQ_ADD_TRAIN	T	Fahrzeug mit Adresse T anmelden
	REQ_SET_SPEED	T S	Geschw. des Fahrzeugs T auf S setzen
	REQ_SET_FUNCTIONS	T F	Funktionen des Fahrzeugs T auf F setzen (Bitfolge F (MSB→LSB): 0 0 0 F4 F3 F2 F1 FF)
	REQ_GO		Endstufe für Bahnstrom einschalten
	REQ_STOP		Alle Fahrzeuge sofort anhalten
	REQ_EVAC		Endstufe für Bahnstrom abschalten (Schienen sofort spannungsfreischnalten)
	REQ_FORCE_REFRESH	S	Datensatz des STW (Adresse S) sofort aktualisieren

STW-Setter	REQ_SET_TURNOUT	S T P [F]	Direkte Weitergabe des Requests an Stellwerk S (s.o.)
	REQ_SET_SIGNAL	S I T	Direkte Weitergabe des Requests an Stellwerk S (s.o.)
	REQ_SET_SECTION	S A T	Direkte Weitergabe des Requests an Stellwerk S (s.o.)
	REQ_CLR_SENSOR	S Z	Direkte Weitergabe des Requests an Stellwerk S (s.o.)
STW-Getter	REQ_GET_TURNOUTS	S	Fordert alle Weichen-Zustände an (aus lokalem Datensatz des STW S)
	REQ_GET_SIGNALS	S	Fordert alle Signal-Zustände an (aus lokalem Datensatz des STW S)
	REQ_GET_SECTIONS	S	Fordert alle Gleisstr.-Relais-Zustände an (aus lokalem Datensatz des STW S)
	REQ_GET_SENSORS	S	Fordert alle Magnet-Zählwerte an (aus lokalem Datensatz des STW S)

Tabelle 18 – Liste aller UART-Requests

Die in der Tabelle als STW-Setter markierten Befehle werden sofort an das jeweilig betroffene Stellwerk weitergesendet. Die STW-Getter beziehen die periodisch aktualisierten Stellwerk-Datensätze des Hauptmoduls. Über REQ_FORCE_REFRESH kann die Steuerungssoftware den Datensatz aktualisieren lassen, bevor sie die Daten bezieht. Dies ist nützlich in Fällen, in denen die Daten dringend aktuell (nicht älter als 250 ms) sein müssen.

3.5 Hauptmodul-Software

Die Struktur der Hauptmodul-Software ist modular. Dadurch ist es möglich, einzelne Komponenten ohne größere Anpassungen für weitere Anwendungen einzusetzen. Zusätzlich vereinfacht die Modularität die Programmierung, da jede Komponente problemlos einzeln getestet, debuggt oder variiert werden kann, ohne den restlichen Teil des Programms zu verändern.

Die Software ist in C verfasst und besteht aus vier Libraries. Unterhalb befindet sich ein vereinfachtes Diagramm des Programmaufbaus. In der Übersicht werden einige Variablen und Funktionen nicht dargestellt, da diese für die Betrachtung der Library von außerhalb nicht von Relevanz sind.

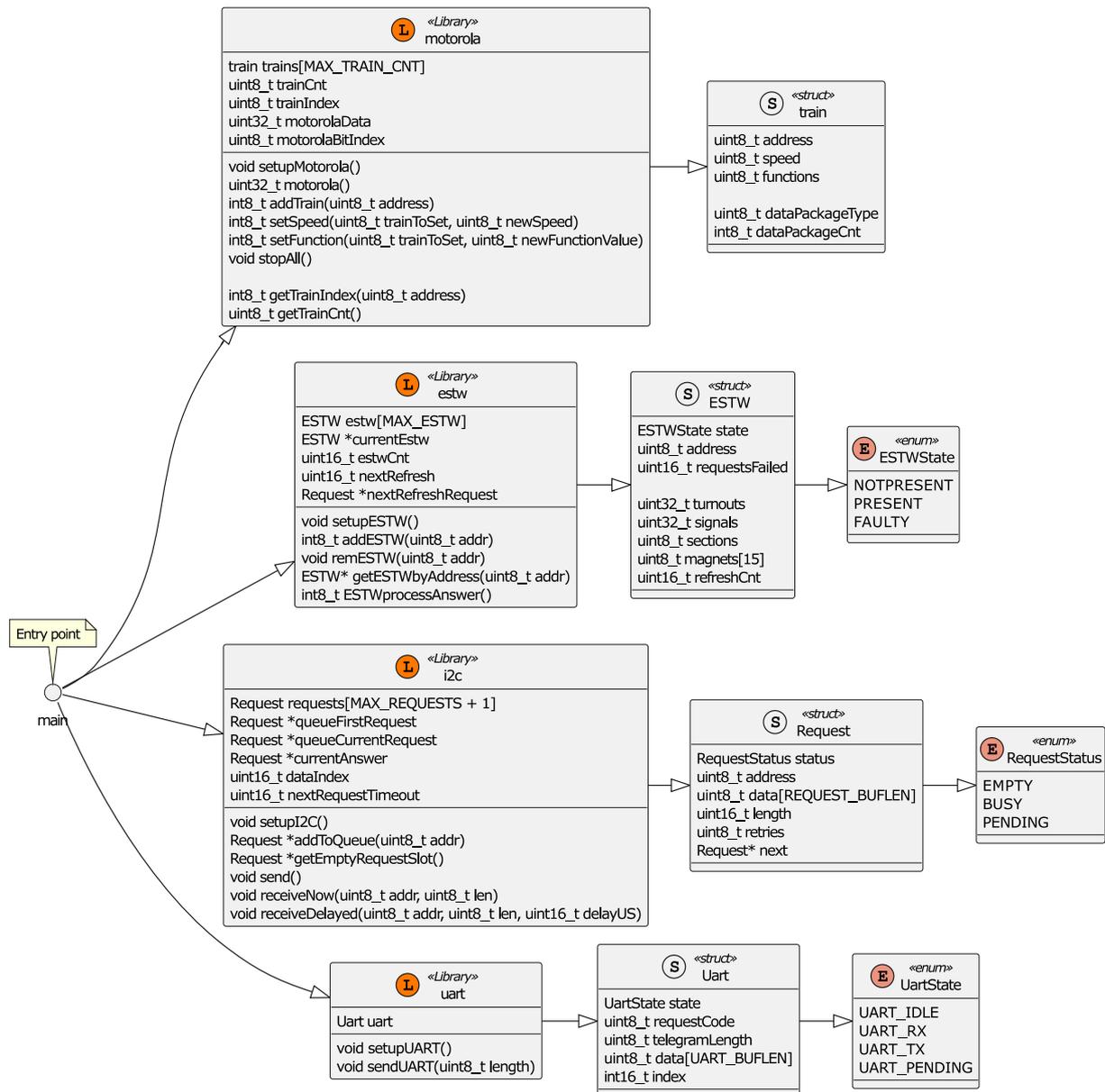


Abbildung 22 – Struktur der Hauptmodul-Software

Die relevanten Teile der einzelnen Software-Komponenten werden im Folgenden grob erklärt. Detaillierte Informationen sind stets dem beiliegenden Quellcode zu entnehmen.

3.5.1 Library `motorola()`

Die Library `motorola` dient der Generierung und Aussendung der Motorola-Bitfolgen an die Endstufe. Die Funktion `motorola()` gibt bei der Ausführung immer das nächste auszusendende Datenpaket (18 Bit) in einem `uint32` zurück. Die Bitfolge beginnt am MSB (Most significant bit). Die unteren 8 Bit beinhalten eine Information über die nach der Aussendung einzuhaltende Sendepause (siehe Abbildung 6). Beim Aufruf von `setupMotorola()` wird u.a. ein PWM-Modul initialisiert, welches periodisch `motorola()` aufruft und die Bits danach aussendet.

Das PWM-Modul basiert auf einem Timer, der in dieser Konfiguration im Count-down-mode betrieben wird. Als Startwert der Abwärtszählung dient der LOAD-Value, der Ausgangspegel

ist am Anfang der Periode immer HIGH. Erreicht der Zählwert den Compare-value (CMP), wird der Ausgang für den Rest der Periode LOW geschaltet. Der LOAD-Value bestimmt also die Dauer eines Bits (208 μ s), während der CMP-Value die Länge des HIGH Pulses bestimmt (26 μ s für logische '0', 182 μ s für logische '1'). Die Werte lassen sich folgendermaßen bestimmen [10].

Für den Prozessortakt $f_{CPU} = 80 \text{ MHz}$ und die Bit-Periode $T_{Motorola} = 208 \mu\text{s}$ gilt:

$$LOAD = \frac{T_{Motorola}}{\frac{1}{f_{CPU}}} = \frac{208 \mu\text{s}}{12,5 \text{ ns}} = 16640$$

$$CMP_0 = \frac{26 \mu\text{s}}{12,5 \text{ ns}} = 2080$$

$$CMP_1 = \frac{182 \mu\text{s}}{12,5 \text{ ns}} = 14560$$

Es muss demnach lediglich der CMP-Value zwischen den beiden berechneten Werten CMP_0 und CMP_1 umgeschaltet werden, um die Motorola-Datenpakete korrekt auszusenden.

Über die Funktion `addTrain()` lässt sich ein Decoder zum Fahrzeug-Pool `trains` hinzufügen. `trains` ist dabei ein strukturierter Datentyp (struct), der Informationen über die Decoder-Adresse, Fahrtgeschwindigkeit und Funktions-Zustände beinhaltet. Zwei weitere Variablen dienen als Zustands- und Datenspeicher für die Berechnung der Bitfolgen. Mit `setSpeed()` oder `setFunction()` lassen sich Fahrtgeschwindigkeit und Funktions-Zustände eines Fahrzeugs ändern. Zu beachten ist bei der Angabe der Geschwindigkeit, dass die Variable `speed` im struct `train` nicht nach dem Zweierkomplement interpretiert werden darf. Zwar gibt das MSB das Vorzeichen an, jedoch sind die unteren sieben Bits immer als die eines positiven Werts zu interpretieren. Diese Maßnahme ist notwendig, da es beim Motorola-Format „zwei Nullen“ gibt (+0 und -0, das Vorzeichen gibt die eingestellte Fahrtrichtung an).

Letztlich bietet `stopAll()` in Gefahrensituationen die Möglichkeit, alle Fahrzeuge sofort zum Stehen zu bringen. Diese Funktion wird bei Empfang des Requests `REQ_STOP` aufgerufen.

3.5.2 Library `estw`

In der Library `estw` werden alle Stellwerke der Bahnanlage verwaltet. Diese liegen dabei als struct-Element vor. Jedes Stellwerk besitzt einen Zustand, welcher angibt, ob das Stellwerk ordnungsgemäß funktioniert oder fehlerhaft ist. Weiterhin werden die Zustandsdaten über die Weichen, Signale, Gleisstrom-Relais und Magnetähler im struct zwischengespeichert. Ein Zähler `refreshCnt` zeigt der Steuersoftware, ob die Daten kontinuierlich aktualisiert werden.

Für die Aktualisierung der Stellwerk-Datensätze nutzt die Library einen Timer, der unter Verwendung der Variable `nextRequest` als Zwischenspeicher für den nächsten zu sendenden Request alle Datensätze regelmäßig aktualisiert. In dieser Anwendung wird alle 50 ms ein neuer Request zur Stellwerk-Bus-Queue hinzugefügt (siehe folgender Abschnitt).

Neue Stellwerke können zur Laufzeit über die Funktionen `addESTW()` und `remESTW()` hinzugefügt, bzw. entfernt werden. Die Funktion `ESTWprocessAnswer()` verarbeitet Answers, die zuvor von Stellwerken empfangen wurden und wird vom Hauptausführungszweig aufgerufen.

3.5.3 Library `i2c`

Die Library `i2c` steuert den Stellwerk-Bus in der Rolle des Master Transmitter, bzw. Master Receiver. Das Array aus Elementen des Typs `struct Request` bietet die Möglichkeit, an Stellwerke zu sendende Requests in einer Warteschlange zu verwahren, bis die tatsächliche Aussendung stattfinden kann. Mit der `addToQueue()` Funktion wird ein neuer Request in einen leeren Slot des Request-Pools `requests` eingesetzt. Der Slot beinhaltet die Slave-Adresse, die zu sendenden Bytes und einen Pointer auf den nächsten Request in der Queue (verkettete Liste). Ein Counter `retries` zählt fehlgeschlagene Übertragungen. Nach Erreichen eines Schwellenwertes wird ein Request aus der Queue entfernt und das Stellwerk als fehlerhaft markiert. Gleich verhält es sich, wenn Übertragungsfehler beim Empfang einer Answer auftreten, oder wenn die Answer eine ungültige Struktur aufweist.

Die `send()` Funktion startet die Interrupt-gesteuerte Übertragung des nächsten Requests in der Queue. Standardmäßig erfolgt die Ausführung im `main-loop` der Software. Die Funktion `receiveDelayed()` wird ausgeführt von der I²C-ISR (Interrupt-Service-Routine), nachdem eine Übertragung eines Requests abgeschlossen ist. Die Funktion nutzt einen Timer, um einen asynchron laufenden Countdown abzuwarten, bevor der Empfang der Answer auf den letzten Request gestartet wird. Die kurze Zeitverzögerung (15 μ s) gibt dem Stellwerk Zeit für die Generierung der Answer. Auf Verwendung von längeren Clock-Stretching-Phasen seitens der Slaves gemäß dem I²C-Standard wird in dieser Implementierung verzichtet, um die Stabilität des Busses und der Stellwerk-Software sicherzustellen.

3.5.4 Library `uart`

Die Library `uart` steuert die Kommunikation mit der Steuersoftware. Beim Aufruf von `setupUART()` wird der UART-Interrupt so eingerichtet, dass er auf den Empfang neuer Bytes und das erfolgte Senden von Bytes reagiert. Aus dem Hauptausführungszweig können Datenpakete versendet werden, indem die Funktion `sendUart()` aufgerufen wird. Diese initiiert den Sendevorgang, indem sie das erste zu sendende Byte aus dem `struct uart` in das UART-Datenregister schreibt. Nachdem dieses erste Byte gesendet wurde, wird erstmals die Interrupt-Service-Routine ausgeführt, welche dann nach und nach die Folgebytes sendet. Beim Empfang wird bereits das erste Byte in der ISR empfangen und in den Struct geschrieben. Nach Empfang des ersten Bytes erfolgt eine Auswertung des Request-Bytes, aus der hervorgeht, ob noch weitere Bytes folgen werden. Beispielsweise besteht der Request `REQ_STOP` aus nur einem Byte. Der Empfang kann daher direkt nach diesem Byte eingestellt werden.

Die Status-Variable `uart.state` zeigt stets an, ob auf den Struct gerade zugegriffen wird, ob er unverarbeitete Daten beinhaltet oder ob er derzeit leer ist. Nach dem Empfang von Requests lautet der Status `UART_PENDING`. Der Hauptausführungszweig fragt den Status

periodisch ab und ruft in diesem Fall `UARTprocessRequest()` auf. Die Bearbeitung des Request und die Generierung der Answer findet hierin statt.

Ein Programmfluss-Diagramm zur Visualisierung obiger Beschreibung befindet sich im Anhang.

3.6 Stellwerk-Software

Aufgabe der Stellwerk-Software ist es, die zugehörigen Sensoren und Aktoren zu steuern, auszuwerten und zu überwachen. Die Software übernimmt hardwarenahe und nicht sonderlich rechenintensive Aufgaben. Trotzdem wird bei der Programmierung neben Störunanfälligkeit besonders auf Effizienz geachtet. Als Programmiersprache wird C verwendet. Es folgt die grafische Darstellung der Programmstruktur.

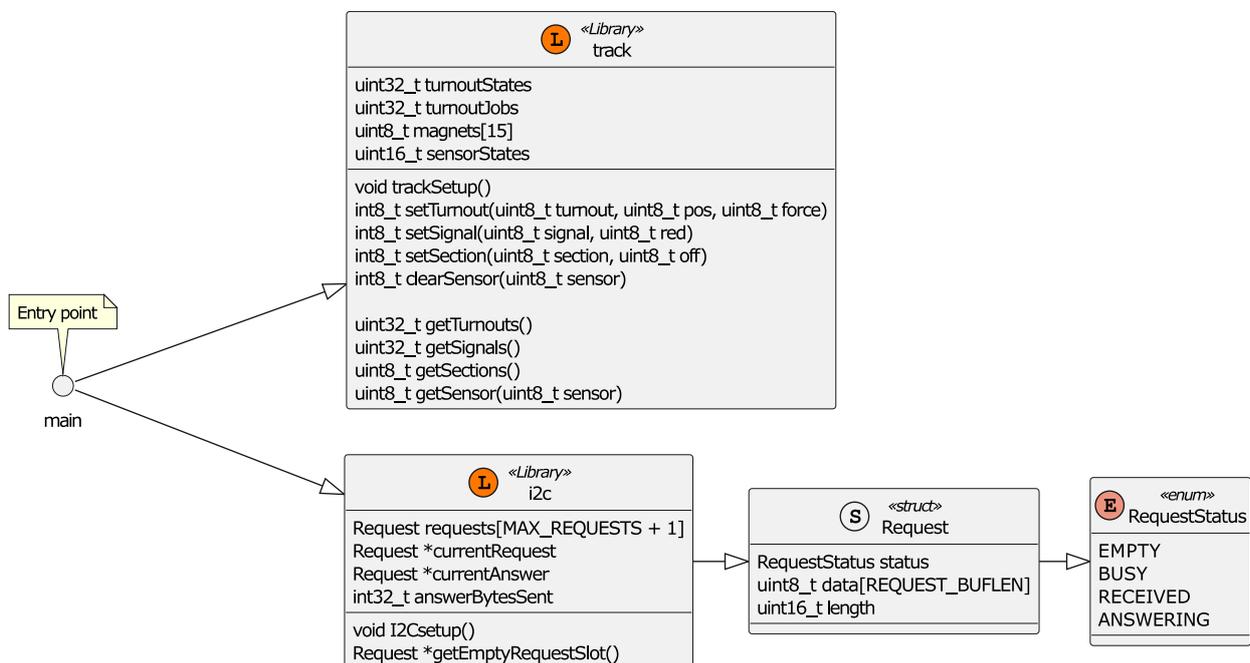


Abbildung 23 – Struktur der Stellwerk-Software

Die Software besteht im Wesentlichen aus zwei Komponenten.

3.6.1 Library track

Die Library `track` verwaltet alle Sensoren und Aktoren in der Zuständigkeit des Stellwerks. Für Weichen, Signale und Gleisstrom-Relais gibt es jeweils einen Setter, der jeweils einen Aktor steuert. Die zugehörigen Getter geben eine Bitfolge zurück, die den Zustand aller Aktoren des Typs beschreibt.³ Während beim Aufruf der Setter für Signale (`setSignal()`) oder Gleisstrom-Relais (`setSection()`) eine sofortige Schaltung des zugehörigen GPIO-Pins erfolgt, ändert ein Aufruf von `setTurnout()` zunächst lediglich die Soll-Position der Weiche. Eine Timer-Interrupt-Routine kontrolliert periodisch, ob sich alle Weichen in der Soll-Position befinden. Ist eine Weiche verstellt, wird die Stellung auf Basis eines Zustandsautomaten (FSM) initiiert.⁴ Hierdurch werden auch Weichen gestellt, deren Soll-

³ Die Belegung der einzelnen Bits ist im Quellcode genau beschrieben.

⁴ Das genaue Verfahren zur Weichenstellung ist im Quellcode des Stellwerks beschrieben (`track.c`, ISR Weichen-Timer)

Zustand sich nicht verändert hat (z.B. eine von Hand verstellte oder defekte Weiche). Bei der Stellung wird über die Rückmeldekontakte stets überprüft, ob die Weiche in die gewünschte Endlage gebracht wurde. Ist dies nicht der Fall, so wird ein Zweitversuch unternommen. Ist dieser Versuch ebenfalls erfolglos, wird die Weiche als fehlerhaft markiert.

Zusätzlich existieren die Funktionen `getSensor()` und `clearSensor()`. Hiermit lässt sich der Magnet-Zählerstand für eine Messstelle auslesen bzw. zurücksetzen. Das Erfassen der Sensorzustände erfolgt Timer-gesteuert und durch Verwendung eines Zwischenspeichers für die vorigen Zustände (ausgelöst/nicht ausgelöst) jedes Sensor-Pins (`sensorStates`). Die Anzahl der gezählten Magneten wird je Sensor in einem `uint8` gespeichert. Es können daher maximal 255 Magneten gezählt werden, wobei dieser Maximalwert auch als Fehlermeldung verwendet wird. Ein Zählerstand mit einem Wert über 254 sollte daher ignoriert werden.

3.6.2 Library `i2c`

Die Library `i2c` steuert die Interaktion mit dem Stellwerkbus bzw. die Kommunikation mit dem Hauptmodul. Der Aufbau gleicht prinzipiell der gleichnamigen Library aus dem Hauptmodul-Code. Jedoch unterscheiden sich die Libraries in den controllerspezifischen Operationen (Registerzugriffe, Interrupt, etc.) und der Rolle, die der Controller im Stellwerk-Bus einnimmt (Master/Slave). Empfangene Requests im Request-Pool werden in der I²C-ISR mit dem Status `RECEIVED` versehen und daraufhin im Hauptausführungszweig ausgewertet. Dabei wird immer eine Answer generiert. Vom Hauptmodul nicht angeforderte Answers werden beim Bearbeiten des nächsten Requests überschrieben.

Ein Programmfluss-Diagramm zur Visualisierung obiger Beschreibung befindet sich im Anhang.

3.7 Steuersoftware („Der Fahrdienstleiter“)

Die Steuersoftware ist ein komplexes Programm, dessen Aufgabe es ist, den Verkehr auf der Bahnanlage zu steuern, zu überwachen und zu sichern. Da es sich um eine Desktop-Anwendung handelt, besteht Zugriff auf stärker belastbare Ressourcen und Parallelisierung ist möglich. Im Vergleich zu den Controller-Softwares besteht demnach die Möglichkeit, Berechnungen deutlich schneller durchzuführen, ohne dabei den Programmfluss an anderen Zweigen zu verlangsamen. Die Berechnung der Fahrwege und das Umwandeln dieser zu Fahrstraßen geschieht daher nicht in der Stellwerk-Software, sondern zentral für alle Stellwerke in der Steuersoftware. Auch in der Realität kommen ESTW vor, die von sogenannten Betriebszentralen aus ferngesteuert werden [1]. Diese Software kann als Nachbildung einer solchen Betriebszentrale angesehen werden.

Da Stellwerke in der Realität unter dem Kommando des Fahrdienstleiters stehen, erhält die im Folgenden beschriebene Software den Namen „Der Fahrdienstleiter“. Die Software soll effizient arbeiten, jedoch auch übersichtlich und wartbar sein. Die Sprache C++ vereint die von C gewohnte Hardwarenähe mit der Möglichkeit der Abstraktion durch Objektorientierung. Unter Verwendung der Windows-Programmierschnittstelle (Windows API) lassen sich mit C++ Benutzeroberflächen erstellen, die eine Interaktion des Nutzers mit

der Logik des autonomen Betriebs ermöglichen. Die Software wird in zwei grundlegende Einheiten unterteilt.

3.7.1 Backend: Bahnbetriebssoftware

Das Backend besteht aus mehreren Klassen, deren Zweck im Folgenden beschrieben ist. Detaillierte Informationen sind stets im Quellcode zu finden. Zunächst ist eine weitere Kategorisierung des Programms vorzunehmen.

Kommunikation

Die Klasse `CCommunication` steuert die Kommunikation zwischen dem Hauptmodul und der Steuersoftware. Wie in 3.4.2 beschrieben stellt die Treibersoftware des Hauptmodul-Evaluationsboards einen COM-Port bereit, über welchen Zugriff auf die UART-Schnittstelle des Mikrocontrollers gewährt wird. Unter Verwendung verschiedener Dateizugriffsfunktionen und spezieller COM-Port-Systemaufrufe aus der Windows API kann somit eine Kommunikation der Software mit dem Hauptmodul erfolgen. Das Protokoll ist in 3.4.2 beschrieben. Versendete Requests entsprechen in der Software jeweils einer Instanz der Klasse `CRequest`. Zu sendende Requests werden in einer priorisierten Warteschlange (`requestsPending`) gesammelt. Ein Request verfügt gemäß der Definition des Protokolls über einen `requestCode` und einen `payload`-Vektor (Nutzdaten). Zusätzlich hat jeder Request einen Zustand und eine Priorität, die bestimmt, an welcher Stelle er in die Warteschlange eingefügt wird. Es besteht weiterhin die Möglichkeit, dem Request eine Callback-Funktion zuzuweisen, die nach dem Empfang der Antwort mit den empfangenen Daten aufgerufen werden soll.

Durch einen Aufruf der Methode `connect()` der Klasse `CCommunication` kann nach der Instanziierung die Kommunikation gestartet werden. Dabei werden mehrere interne Methoden aufgerufen, die unter Verwendung von Systemaufrufen zunächst versuchen, den COM-Port des Mikrocontrollers zu ermitteln und diesen anschließend zu öffnen. Dabei werden Kommunikations- und Zeit-Parameter konfiguriert. Im Erfolgsfall besteht anschließend die Möglichkeit, die ersten Requests zu senden und Answers zu empfangen. Das Senden und Empfangen ist auf einen Timer-Thread ausgelagert. Der Zugriff auf einige Klassenattribute ist hierdurch nur unter Verwendung von Mutexen fehlerfrei realisierbar.

Bahnbetriebsteuerung

Der Bahnbetrieb wird primär in der Klasse `CRailway` gesteuert. Diese verfügt über Vektoren aus Instanzen der Klassen `CTrain`, `CSection`, `CSignalBox`, `CStation` und `CRoute`. Diese Klassen bilden die auf der physischen Strecke zu steuernden oder auszuwertenden Elemente ab und sind im Folgenden beschrieben.

Instanzen von `CSection` bilden die einzelnen Streckenabschnitte ab, die in einer doppelt verketteten Liste gespeichert sind. Jeder Abschnitt besitzt eine ID (gemäß dem Gleisabschnittsplan), Vektoren zugehöriger Signale, Weichen, Sensoren und Relais und können drei Zustände annehmen. Ist der Zählwert für im Abschnitt befindliche Magnete ungleich Null, so ist der Abschnitt belegt. Ist das Attribut `blocked` gesetzt, ist der Abschnitt reserviert (während der Berechnung eines Fahrwegs). Treffen beide Bedingungen nicht zu, ist der Abschnitt frei.

Instanzen von `CTrain` stellen die zu steuernden Schienenfahrzeuge dar. Jeder Zug hat eine ID (i.d.R. die Baureihenbezeichnung der Lokomotive) und eine Adresse zur Ansteuerung per Motorola-Protokoll (siehe 3.2.1). Über Methodenaufrufe lassen sich Geschwindigkeit und Funktionen steuern. Weitere Methoden und Attribute dienen zur Erfassung der Position und zur Berechnung des Fahrwegs bzw. der Fahrstraße. Das Attribut `magnetCnt` definiert die Anzahl der am Fahrzeug angebrachten Magnete, `waitTime` ist ein Abwärtszähler für die Realisierung einer Zeitverzögerung zwischen Ankunft und Abfahrt an einem Bahnhof. Der Vektor `position` enthält Zeiger auf die `CSections`, in denen sich der Zug befindet. Die Länge dieses Vektors variiert während der Fahrt zwischen 1 und 3. Das Symbol `circulation` gibt an, ob der Zug im Uhrzeigersinn oder entgegen diesem fahren soll. Der Vektor `stations` enthält Zeiger auf alle `CStations`, an denen der Zug halten soll. Der nächste Halt wird in `nextStop` gespeichert.

Instanzen von `CSignalBox` sind die Stellwerke, mit denen Weichen, Signale, Gleisstromrelais geschaltet und Magnetzählerstände ermittelt werden können. Die Statusdaten werden durch Timer-gesteuerte Aufrufe von `updateValues()` über `CCommunication` vom Hauptmodul angefordert. Nach Empfang der Daten werden diese über die Callback-Funktion `setData()` an das Stellwerk zurückgegeben.

Instanzen von `CStation` gruppieren `CSections` zu einem Vektor `tracks`, der die Gleise definiert, an denen Züge halten sollen.

Instanzen von `CRoute` werden immer dann generiert, wenn ein Zug abfahrbereit ist. Zu diesem Zeitpunkt handelt es sich um Objekt, das einen Fahrweg und Fahrwegelemente (Weichen und Signale) vereint. Im weiteren Verlauf der Fahrstraßenbildung (siehe 3.7.4) wird das Objekt zu einer Fahrstraße umgewandelt.

3.7.2 Frontend: Benutzeroberfläche (GUI)

Das Frontend dient zur Interaktion des Anwenders mit dem Backend. Es ist zwischen zwei Betriebsmodi zu unterscheiden.

Automatikbetrieb

Die Benutzeroberfläche verfügt über einen Streckenmonitor, auf dem das gesamte Schienennetz abgebildet ist. Die farbliche Markierung einzelner Streckenabschnitte lassen den Nutzer nachverfolgen, welche Aktionen derzeit von der autonomen Logik ausgeführt werden. Eine Linie entspricht jeweils einem Schienenstück. Da die Streckenabschnitte immer aus mehreren Schienenstücken bestehen, haben die zugehörigen Linien auch dieselbe Farbe. Ist ein Abschnitt schwarz dargestellt, ist er frei. Orangefarbene Linien markieren Abschnitte, die zu einer verschlossenen Fahrstraße gehören. Wird ein Abschnitt gerade befahren, nimmt er rote Farbe an.

Unter dem Streckenmonitor gibt es ein Statusfenster „Aktuelle Meldungen“, worin alle relevanten, von der autonomen Logik oder von Kommunikationsinstanz gemeldeten Meldungen gespeichert sind. Zwischen dem Zeitstempel und dem Meldungstext ist ein Symbol eingeblendet, das die Kategorie der Meldung signalisiert. Informationsmeldungen erhalten eine Pinnadel, Warnungen sind mit einem Warndreieck und kritische, den Verkehr gefährdende Ereignisse werden mit einer erhobenen Hand markiert.

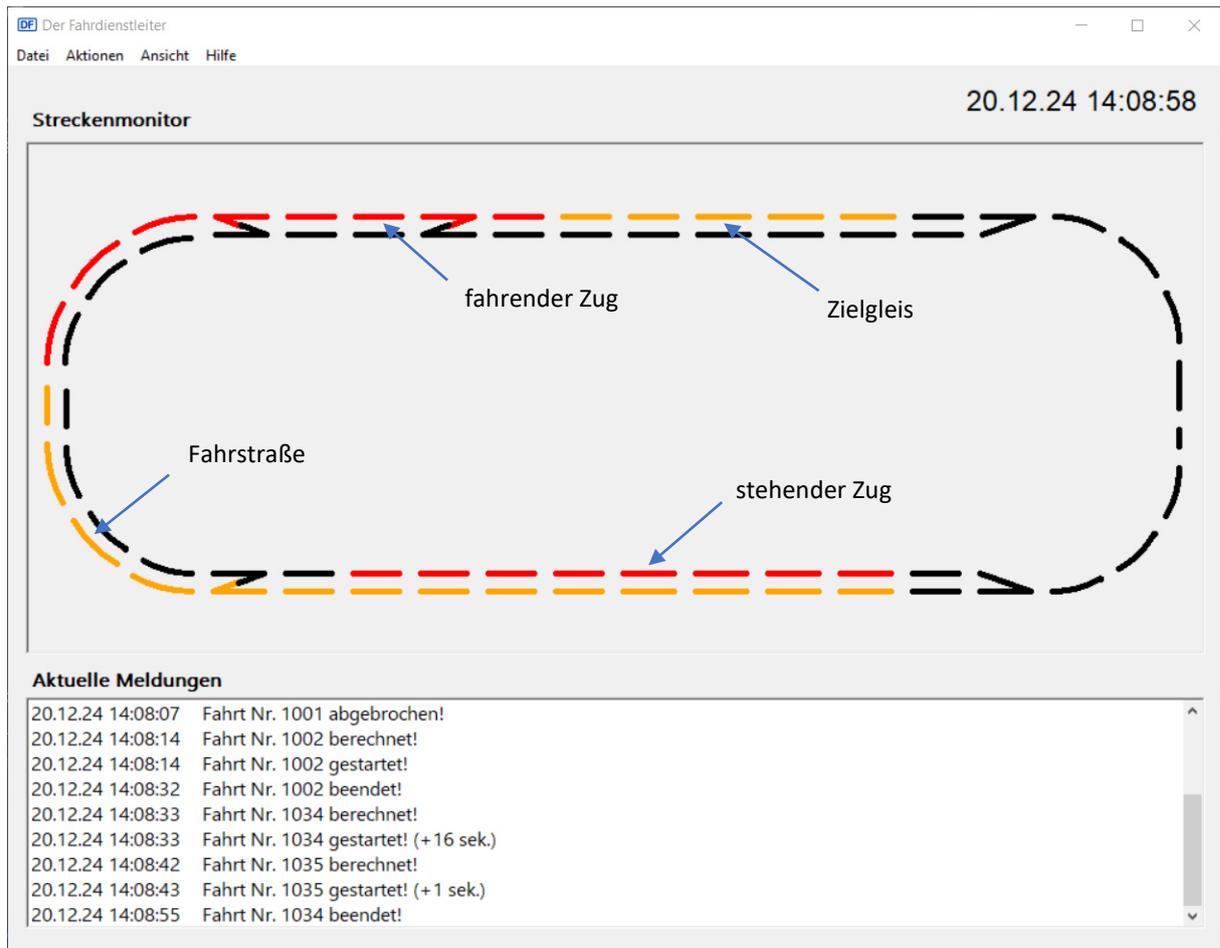


Abbildung 24 – Steuersoftware (Automatikbetrieb)

Handbetrieb

In Störungsfällen kann unter Umständen der autonome Betrieb nicht aufrechterhalten werden. Hierfür bietet die GUI den Modus „Handbetrieb“. Dieser lässt sich über das Untermenü „Aktionen“ in der Menüleiste aktivieren, worauf sich ein Dialogfenster öffnet. In der oberen Hälfte dieses Fensters können alle Stellwerke manuell gesteuert und ausgelesen werden. Weichen, Signale und Gleisstromrelais lassen sich in ihren Zuständen verstellen und die Magnet-Zählerstände können ausgelesen und zurückgesetzt werden. Die angezeigten Zustände (Kontrollpunkte der Radio Buttons) entsprechen stets den vom Stellwerk gesendeten Statusdaten. Damit lassen sich beispielsweise Weichenstörungen leicht erkennen, da dann keiner der beiden zugehörigen Radio Buttons aufleuchtet.

Im unteren Bereich des Fensters befinden sich zwei Zugfernsteuerungen, womit die Statusdaten zweier Züge zeitgleich angezeigt werden können. Mit den Fernsteuerungen lässt sich über die Pfeiltasten (links und rechts) der einzusehende oder zu steuernde Zug auswählen. Mit den Tasten W und S lässt sich die Fahrstufe variieren und die Tasten 0 (L), 1, 2, 3 und 4 schalten die Funktionen F0...F4. Zwischen den Fernsteuerungen umschalten lässt sich mit der Leertaste. Wird die Eingabetaste betätigt, wird ein Nothalt für den ausgewählten Zug ausgelöst. Die Rücktaste schaltet die Fahrtrichtung um.

Am unteren Fensterrand gibt es letztlich noch drei Buttons, mit denen sich der Bahnstrom ein und ausschalten („Freigabe“, „Evakuierung“) und ein globaler Nothaltauftrag („Nothalt global“) einleiten lässt.

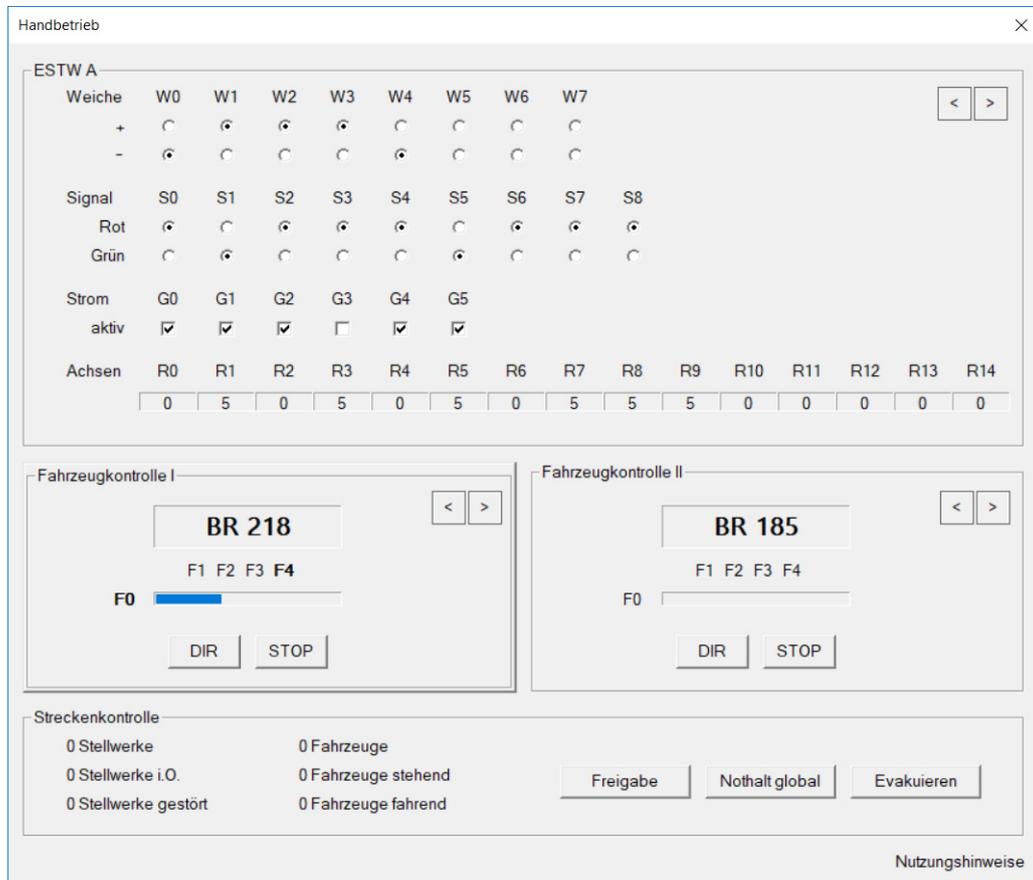


Abbildung 25 – Steuersoftware (Handbetrieb)

3.7.3 Kooperation von Frontend und Backend

Als Schnittstelle zwischen der GUI und CRailway dient die Klasse CRailMonitor. Beide Klassen werden von der GUI instanziiert. CRailMonitor verfügt über einen Vektor aus Elementen des struct DrawableSect. Dieser besteht jeweils aus einem Zeiger auf ein Element des sections-Vektors aus CRailway, womit sich eine vom Belegzustand abhängige farbige Markierung realisieren lässt, und Zusatzinformationen, die für die grafische Ausgabe relevant sind. Über einen Aufruf der Methode drawAll() kann die GUI den Streckenmonitor aktualisieren lassen, wobei abhängig von der Fenstergröße ein Skalierungsfaktor bestimmt wird, der eine optimale Ausnutzung des verfügbaren Zeichenbereichs ermöglicht. Die in jeder CSection hinterlegten Schienenstücke, aus denen der Abschnitt besteht, werden in einer Schleife ausgewertet und mittels der Windows GDI Funktionen (Graphics Device Interface) in den Zeichenbereich gezeichnet. Die Ausgabe arbeitet unabhängig von der Strecke. Durch das Editieren des sections-Vektors ist es möglich, die Software an jede beliebige Strecke anzupassen.

Im Handbetrieb-Modus greift das Frontend direkt auf die Member der CRailway-Klasse zu um etwa Weichen zu stellen oder Magnetzählerstände auszulesen.

3.7.4 Fahrstraßenbildung und Zugleitung

Die in 3.7.1 beschriebene Klassenstruktur soll letztendlich als Basis für die Implementierung eines autonomen Fahrbetriebs genutzt werden. Die Klasse `CRailway` wird dazu um einige Methoden erweitert.

Berechnung des Fahrwegs `createRoute()`

Die Berechnung des Fahrwegs erfolgt, wenn ein `CTrain-Element` abfahrbereit ist (`waitTime <= 0`). Der Algorithmus nutzt die doppelte Verkettung des Vektors `sections`, um den Weg vom aktuellen Standort des Zuges bis zum nächsten Bahnhof zu ermitteln. Abhängig von der Fahrtrichtung (`circulation`) nutzt er die `next` oder `prev`-Zeiger, um den folgenden Abschnitt zu ermitteln. Dieser muss frei sein, um der neuen Instanz von `CRoute` zugefügt werden zu dürfen. Abschnitte, die Weichen enthalten, besitzen mehrere Folgeabschnitt-Zeiger. Den nächsten freien Abschnitt liefern die Methoden `getNextUnblockedSection()` bzw. `getPrevUnblockedSection()` einer `CSection`-Instanz. Diese werden in einer Schleife aufgerufen, bis der letzte bezogene Abschnitt dem nächsten Bahnhof entspricht, an dem der Zug halten soll.

Ist kein Folgeabschnitt verfügbar der weder belegt noch reserviert ist, wird das `CRoute-Element` verworfen und mit dem nächsten Zug fortgefahren. Aufgrund der elliptischen Bauform der Strecke führen alle Gleise zum nächsten Bahnhof und der richtige Fahrweg wird immer gefunden. Für den Fall, dass die Strecke geändert wird, sodass der Erfolg nicht mehr garantiert ist, oder die verkettete Liste inkonsistent ist, gibt es einen Iterations-Zähler, der die Schleife nach Durchlaufen der maximal zulässigen Iteration abbricht.

Konnte zuvor ein Fahrweg erfolgreich berechnet werden, wird die Methode `prepareRouteJobs()` aufgerufen, die für den berechneten Fahrweg die sogenannten Weichen- und Signal-Stellaufträge (`turnoutJobs`, `signalJobs`) generiert. Diese enthalten Informationen über die Weichen, die zum Verschließen der Fahrstraße gestellt sein müssen, und die Signale, die nach dem Verschließen Fahrt zeigen sollen.

Das angewandte Verfahren für die Fahrwegsuche ist mit dem sogenannten Such-Echostrom-Prinzip vergleichbar, welches im Stellwerktyp `SpDrS60` zur Anwendung kommt [18], [19].

Generierung der Fahrstraße `lockRoute()`

Nachdem ein Fahrweg generiert wurde, müssen die enthaltenen Stellaufträge bearbeitet werden. Zunächst werden dazu in einer Schleife alle Weichen-Aufträge bearbeitet. Ist eine Weiche verstellt, so wird die Stellung in die richtige Lage veranlasst und mit dem nächsten Auftrag fortgefahren, um die Zeit bis zum Eintreffen der aktuellen Statusmeldung vom Stellwerk zu überbrücken. Vom zuständigen Stellwerk als fehlerhaft gemeldete Weichen resultieren in einem Verwerfen des Fahrwegs und einem Freimelden aller reservierten Abschnitte. Eine Warnmeldung wird ausgegeben, die den Anwender auffordert, die Weiche ggf. zu reparieren und über den Handbetrieb-Modus zu entstören.

Nach Ende der Schleife gibt eine Zählvariable Aufschluss darüber, wie viele Weichen-Stellaufträge unbearbeitet sind. Beträgt der Wert 0, erhält das `CRoute-Element` den Status `LOCKED` (verschlossen) und es erfolgt ein Aufruf von `setAllSignals()`. Diese Methode

bearbeitet alle Signal-Stellaufträge des angegebenen CRoute-Elements. Sie wird auch verwendet, um alle Signale auf Halt zu stellen, wenn nach Verschließen der Fahrstraße ein Stellwerk oder eine Weiche eine Störung aufweist.

Überwachung laufender Fahrten `locateTrains()`

Die Methode zum Überwachen der laufenden Fahrten dient der Erkennung von feindlichen Fahrten. In einer Schleife werden alle von den Stellwerken gemeldeten Magnetzählerstände mit zugehörigen zwischengespeicherten Werten verglichen. Bei einer Abweichung steht fest, dass mindestens ein Magnet noch nicht von der Software erfasst wurde. In diesem Fall wird zunächst der Sensor den Abschnitten zugeordnet, zwischen denen er installiert ist.

Zur Erkennung feindlicher Fahrten wird dann abgefragt, ob der Abschnitt zu einer Fahrstraße gehört und ob die Position des berechtigten Zuges an diesen Abschnitt grenzt oder diesen bereits beinhaltet. Ist die Fahrt genehmigt, wird der Zugpositions-Vektor aktualisiert. Danach muss anhand der Fahrtrichtung erkannt werden, in welchen Abschnitt aus- und in welchen eingefahren wird. Die Magnetzählerstände der betroffenen Abschnitte werden daraufhin aktualisiert, sodass sie die korrekte Anzahl der eingezählten Magnete enthalten.

Bei Detektion feindlicher Fahrten werden sofort die Gleisstromrelais abgeschaltet, die die Abschnitte am und um den Gefahrenpunkt versorgen. Anschließend wird eine Warnmeldung ausgegeben, die den Anwender auffordert, die Situation manuell zu behandeln.

Ein Abschnitt, in den mindestens ein Magnet eingezählt wurde, wird ausschließlich von Halt zeigenden Signalen umgeben.

Ausführung der Fahrten und Auflösung der Fahrstraßen `endRoutes()`

Die Methode `endRoutes()` überprüft die Zustände der gespeicherten CRoute-Elemente und regelt die Zuggeschwindigkeit. Befindet sich ein stehender fahrtberechtigter Zug auf einer gesicherten Fahrstraße, wird er auf die reguläre Fahrtgeschwindigkeit beschleunigt. Während des Einfahrens in den letzten Abschnitt der Fahrstraße wird die Geschwindigkeit reduziert, bevor der Zug nach vollständiger Einfahrt in das Zielgleis zum Stillstand gebracht wird. Vollständig durchgeführte Fahrten werden aus dem Vektor `routes` gelöscht.

4 Ergebnisse

Ziel der Durchführung war die Errichtung eines einfach aufgebauten, robusten Zugleit- und Sicherungssystems unter Einbezug des Standes der Technik und der Praktiken im realen Bahnbetrieb. Der Prozess der Durchführung soll in diesem Abschnitt unter Beleuchtung der erlangten Erkenntnisse und erreichten Ergebnisse aufgearbeitet werden.

4.1 Streckennetz

Als Grundlage für das Erproben der konzipierten Zugleit- und Sicherungsverfahren diente der physikalische Aufbau (siehe 3.1). Aus der Arbeit hervorgegangen ist ein Streckennetz, an welchem verschiedene Methoden zur Zugbeeinflussung und Gleisfreimeldung erprobt werden konnten. Weiterhin bot es Möglichkeiten der Simulation verschiedener Szenarien, wie sie im realen Schienenverkehr auftreten.

Im weiteren Verlauf erfolgte ein Ausbau der Streckenausrüstung. Eine hierfür konzipierte Elektronik aus diskreten Halbleiterbauteilen ermöglicht die mikrocontrollerbasierte Steuerung und Überwachung von Weichen. Die Elektronik arbeitet fehlerfrei und liefert stets den korrekten Weichenzustand.

Die Strecke wurde mit Signalen bestückt, die eine intuitive Wahrnehmung der Funktionsfähigkeit der Gleisfreimeldung erlaubt, die auf kleinen im Gleisbett befindlichen Hall-Sensoren in Verbindung mit an den Fahrzeugen installierten Magneten basiert. Zusätzlich erlauben sie die Nachverfolgung der Bildung von Fahrstraßen im autonomen Betriebsmodus.

Zum Ausschluss feindlicher Fahrten lassen sich ausgewählte Streckenabschnitte in Gefahrensituationen spannungsfreischalten. Der Bereich der abgeschalteten Spannung muss relativ groß sein, um sicherzustellen, dass die Lok unabhängig davon keine weitere Versorgung erhält, ob sie die Wagons zieht oder schiebt, bzw. sie ihren Schleifer am vorderen oder hinteren Drehgestell hat. Es entwickelte sich zur Praxis, immer mehrere Abschnitte abzuschalten, die die Gefahrenstelle umgeben, um dies sicherzustellen.

Die errichteten Stellwerke vereinen die Steuerung (Mikrocontroller) mit den nötigen Grundschaltungen für die einzelnen Elemente der Streckenausrüstung. Die Software läuft stabil, jedoch führten zwischenzeitlich Übertragungsfehler auf der Taktleitung des Stellwerk-Busses zu Abstürzen. Wegen der zustandsbasierten Arbeitsweise der I²C Hardwareschnittstelle können fälschlich erkannte Taktflanken dazu führen, dass der Teilnehmer sich dauerhaft im Slave-Receiver oder Slave-Transmitter-Modus befindet. Da die FSM nur bei einer durch 8 teilbaren Anzahl an verarbeiteten Takten in den Ruhezustand (Idle) zurückkehren kann, hängt sich das Programm in diesem Fall auf. In einer älteren Application note von Philips wird ein Verfahren zum beenden dieses Zustandes beschrieben, das in Fachkreisen als „Freitakten“ bezeichnet wird [20]. Da in diesem Fall jedoch die Möglichkeit der Änderung des Quellcodes der Slave-Software besteht, wurde stattdessen ein Verfahren entwickelt, das einen Timer verwendet, der die Zeit seit dem letzten Empfangen oder Senden eines vollständigen Bytes misst und nach Ablauf einer vordefinierten Zeit (50 ms) die I²C Schnittstelle zurücksetzt. Eine Teilnahme am Bus ist danach wieder möglich. Die Art der Verwendung des Timers gleicht dem Watchdog-Prinzip.

4.2 Bahnbetrieb

Das errichtete Streckennetz ermöglicht zunächst die manuelle Durchführung von Zugfahrten über den Handbetrieb-Modus der Steuersoftware „Der Fahrdienstleiter“. Durch die robuste Kommunikation zwischen der Software und dem Hauptmodul lässt sich die Generierung der Bitfolgen des Energie-Daten-Signals (Märklin-Motorola-Format) gezielt beeinflussen, sodass die Fahrzeug-Decoder die gewünschten Befehle erhalten. Das Verstellen der Fahrwegelemente (Weichen, Signale), das Steuern der Gleisstromrelais und das Auslesen der Magnetzähler erfolgt ebenfalls über den Handbetrieb.

Das der Inbetriebnahme vorangestellte ausführliche theoretische Testen des Algorithmus zur Generierung des Motorola-Formats macht sich dadurch bemerkbar, dass es während der gesamten Durchführungsphase zu keinem unvorhergesehenen Verhalten der Decoder kam. Das Zugsteuerungssystem kann somit als sehr zuverlässig eingestuft werden. Besondere Vorsicht ist jedoch bei Fahrtrichtungswechseln geboten. Anders als bei handelsüblichen Fahrpulten ist es in der Hauptmodul-Software möglich, sprunghaft von einer positiven Fahrstufe (Vorwärtsfahrt) auf eine negative Fahrstufe zu schalten. Die Fahrzeugdecoder unterstützen Richtungsänderungen ausschließlich im Stillstand. Daher muss eine geänderte Richtung immer mit der Fahrstufe „Stop“ eingeführt werden. Das Fahrzeug fährt sonst in dieselbe Richtung weiter.

4.3 Transfer auf die Realität

Das entwickelte Modell weist einige Gemeinsamkeiten und Unterschiede zu dem realen Schienenverkehr auf. Die Überwachung und Stellung von Weichen aus der Ferne ist in der Realität von hoher Wichtigkeit und lässt sich daher auch in dem entworfenen Zugleit- und Sicherungssystem finden. Die erbauten Signale gleichen zwar nicht den realen Vorbildern, sind aber bei ferngesteuerten Zügen auch nicht wirklich notwendig. Wie durch das ETCS-System verdeutlicht wird, muss eine Signalisierung nicht zwingend bestehen, wenn die Übertragung von Befehlen an die Fahrzeugelektronik über Funk erfolgen kann [6].

Das Zählen von Achsen durch induktive Gleiskontakte oder das Freimelden von Gleisabschnitten durch Gleisstromkreise erweist sich gemäß 3.3.4 als nicht umsetzbar. Daher wird mit dem Zählen von Magneten eine alternative Gleisfreimeldungsstrategie verfolgt.

Das Abschalten der Versorgungsspannung (Gleisstrom) deckt sich nicht mit den praktizierten Formen der Zugbeeinflussung. Um der Realität zu entsprechen, müssten alle Lokomotiven mit einer entsprechenden Fahrzeugausrüstung (etwa PZB) ausgerüstet werden. Aufgrund der kleinen Bauform erweist sich ein solches Vorhaben jedoch als schwer umsetzbar.

Die zentrale (Fern-)Steuerung mehrerer elektronischer Stellwerke (ESTW) ist eine gängige Praxis und wird durch die geschriebene Steuersoftware wahrheitsgetreu nachgebildet. Der autonome Betriebsmodus der Software zeichnet sich durch seine realitätsgetreue, jedoch vereinfachte Verfahrensweise bei der Berechnung von Fahrwegen und der Bildung von Fahrstraßen aus.

Insgesamt kann das Gesamtsystem als vereinfachte Nachbildung eines realen Vorbilds mit durch die Modellbauweise bedingten Anpassungen angesehen werden, die durch die Einrichtung eines autonomen Betriebs erweitert wurde. Die Streckenausrüstung befindet sich auf einem einheitlichen Stand der Technik und es gibt keine Variationen gleicher Ausrüstungstypen.

4.4 Stabilität des Systems

Das System arbeitet insbesondere in Hinblick auf das autonome Fahren zuverlässig. Das in 3.7.4 beschriebene Bilden von Fahrwegen und Verschließen dieser zu Fahrstraßen funktioniert ohne sicherheitsrelevante Probleme. Es ist jedoch zu berücksichtigen, dass der Algorithmus für die Berechnung von Fahrwegen auf dem Durchlaufen von hintereinanderliegenden Abschnitten basiert. Eine eventuelle Änderung der Strecke (etwa die Ergänzung von Abstellgleisen) kann dazu führen, dass Fahrwegberechnungen scheitern, weil sich der Algorithmus „verläuft“.

Eine weitere Instabilität ist durch die Weichen-Stellantriebe bedingt. Beim Stellen von Weichen fließen durch den Antrieb kurzzeitig hohe Ströme, die aufgrund einer Unterdimensionierung der vom Hersteller verbauten Endschalter für Kohleablagerungen an den kleinen Kontaktflächen sorgen, wodurch ein Übergangswiderstand entsteht. Dieser sorgt vereinzelt für Weichenstörungen, da das Stellwerk die Weiche als defekt erkennt, wenn dieser Widerstand zu groß wird. Das Einsetzen eines größeren Widerstandes zur Begrenzung des Stroms durch die Zenerdioden der Elektronik für die Zustandserkennung (siehe 3.3.1) kann Abhilfe schaffen, da dann durch den geringeren Ruhestrom die über dem Schalter abfallende Spannung geringer wäre. Die Änderung kann aber ein langsames Stabilisieren der Zustandsbits bewirken.

Der Stellwerk-Bus wird unter bisher nicht eindeutig eingegrenzten Umständen gestört. Die Störungen treten durch elektromagnetische Verkopplung der Leiterbahnen für die Gleisstrom-Endstufe und der Bus-Leitungen auf. Durch parasitäre Induktivitäten der Leiterbahnen entstandene Schwingungen konnten durch den Einsatz von RC-Dämpfungsgliedern stark reduziert werden, wodurch die Stärke der Störungen verringert wurde. In seltenen Fällen treten jedoch noch immer Übertragungsfehler auf, die wahrscheinlich durch Rückwirkungen der auf den Gleisen befindlichen Fahrzeug-Fahrmotoren zurückzuführen sind. Abhilfe können weitere Dämpfungsglieder schaffen, die direkt an den Einspeisepunkten der Gleisströme angebracht werden. Unabhängig hiervon ist die Verwendung von geschirmten Datenleitungen empfehlenswert.

4.5 Sicherheitslücken

Die Erkennung der Fahrzeugmagnete ist zwar sehr zuverlässig, jedoch ist es denkbar, dass Fehler bei der Zählung auftreten. Eine Abweichung der Anzahl der Magneten in Theorie und Praxis kann dazu führen, dass ein Zug seine Halteposition überfährt. Nur die Gleisstromabschaltung kann dann noch Unfälle verhindern.

Defekte Gleissensoren können außerdem nur anhand von unplausiblen Zählerständen erkannt werden. Ein an einer Gefahrenstelle befindlicher ausgefallener Sensor kann dafür sorgen, dass die Gleisstromabschaltung nicht wirksam wird. Der Anwender sollte daher die

in Fällen von falschen Zählungen ausgegebene Warnmeldung „Magnetzählerstand unplausibel“ stets zum Anlass einer Funktionsprüfung des betroffenen Sensors nehmen.

Ein auch in der Realität bekanntes Problem sind Deadlocks. Diese treten auf, wenn Züge so aufeinandertreffen, dass keine weiteren Fahrten mehr möglich sind. Dieses Szenario kann auf der errichteten Anlage auftreten, wenn alle Gleise der beiden Bahnhöfe belegt sind und alle Züge eine Fahrt auf der eingleisigen Strecke anstreben. Die autonome Logik wird dann keinen Fahrweg bilden können, was den Bahnverkehr bis zu einem manuellen Eingreifen komplett zum Erliegen bringen würde.

Im Falle eines Verbindungsabbruchs von der Steuersoftware zum Hauptmodul fahren derzeit alle Züge mit unveränderter Geschwindigkeit weiter. Um diese Lücke zu schließen, sollte das Hauptmodul die Zeit zwischen zwei empfangenen Requests messen und bei Überschreiten einer definierten Zeit den Verbindungsabbruch erkennen einen globalen Nothalt auslösen.

Ein Wechsel des Protokolls auf das UART-basierte *HAW Hamburg Safe Layer Protocol* (HHSLP) ist sinnvoll, da dieses sogenannte Heartbeat-Datenpakete zum periodischen Testen der Verbindung verwendet. Zudem bietet es die Möglichkeit, Verluste von Datenpakete zu detektieren, indem es gesendete Pakete mit einer laufenden Nummer versieht [21]. Das Protokoll basiert auf einem in der realen Bahninfrastruktur eingesetzten Kommunikationsverfahren.

4.6 Aussicht auf Erweiterungen

Die Ausgangsposition aller Züge muss in der aktuellen Softwareversion im Quellcode vorgegeben sein. Eine geeignete Erweiterung würde das regelmäßige Speichern der aktuellen Zugpositionen in einer Datei einbringen, womit beim Programmstart immer die vorigen Zugpositionen eingelesen werden könnten.

Für den Fall der Verbreitung der Software ohne den Quellcode wäre es außerdem nötig, die Definition der Strecke ebenfalls über eine Datei zuzulassen.

Als möglichen Ersatz für die Gleissensoren, der auch näher an dem realen Vorbild (induktive Gleissensoren) liegen würde, eignen sich Gabellichtschranken. Diese zählen die Achsen der Züge anstelle von Magneten, mit denen die Fahrzeuge vorher ausgestattet werden müssen. Lediglich die Installation der Lichtschranken ist mit höherem Aufwand verbunden.

5 Fazit

In Hinblick auf die gesetzten Ziele (siehe 2.3) ist diese Arbeit als erfolgreich einzustufen. Wie im Abschnitt 4 dargestellt, geht aus dieser Arbeit ein Modellbahnsystem hervor, welches über ein durchdachtes Zugsicherungssystem verfügt. Durch die Verwendung einheitlicher Streckenausrüstung auf ESTW-Basis wird die Kooperation der Teilsysteme gegenüber dem realen Streckennetz in Deutschland stark vereinfacht, da keine Kompatibilitätsprobleme auftreten.

Der gemeinsame Betrieb vieler verschiedener Technologien aus verschiedenen Epochen des technischen Fortschritts ist aufwändig und begünstigt das Auftreten von Störungen. Verspätungen und Zugausfälle sind häufig die Folge. Auf dieser Basis einen autonomen Bahnbetrieb zu realisieren, erscheint nicht sinnvoll. Das deutsche Streckennetz benötigt dringend umfangreiche Modernisierungsmaßnahmen, wie die Deutsche Bahn sie bereits angekündigt hat [22].

Aus dieser Arbeit geht unter Beachtung des Standes der Technik hervor, welche Teilsysteme erforderlich sind, um den Anforderungen der Zugleit- und Sicherungstechnik gerecht zu werden. Eine Signalisierungsanlage ist essenziell für den effizienten Bahnbetrieb, da diese die Kommunikation zwischen dem Fahrdienstleiter und dem Fahrpersonal präzisiert und erleichtert. Anlagen zur Überwachung von Fahrwegelementen ermöglichen es, aus der Ferne zweifelsfrei festzustellen, ob ein Streckenabschnitt befahrbar ist. Weichen müssen gestellt und Bahnübergänge geschlossen sein, damit einem Zug eine verschlossene Fahrstraße zugesprochen werden kann. Zur Erkennung genehmigter und feindlicher Fahrten ist jeder Abschnitt mit Freimeldeanlagen auszustatten. Das Einrichten von Gleisstromkreisen oder wahrheitsgetreu nachgebildeten Achszählern ist bedingt durch die Modellbauweise nicht möglich, jedoch kann die Funktion durch ein Ersatzsystem (Magnetsensoren im Gleisbett) nachgebildet werden.

Durch den flächendeckenden Ausbau des ETCS sind herkömmliche Gleisfreimeldeanlagen möglicherweise schon bald entbehrlich, da die entsprechend ausgerüsteten Fahrzeuge ihre Position über die streckenseitig installierten Eurobalisen erkennen und selbstständig ihre Vollständigkeit sicherstellen können (Integritätsprüfung). Das Fahrzeug allein kann somit über das Mobilfunknetz die nötigen Informationen an die Betriebszentrale senden, worauf diese einen Abschnitt als belegt oder frei melden kann.

Mit einem zusätzlichen Umbau auf einheitliche Stellwerkstechnik und einer Erneuerung veralteter oder anfälliger Streckenausrüstung wäre ETCS möglicherweise schon bald für den vorgesehenen autonomen Betrieb einsetzbar. Der in diesem Projekt eingerichtete autonome Betrieb stellt ein Anwendungsbeispiel dar. In der Realität bestehen weitere Problemstellungen, die bei einem Modellaufbau nicht zu beachten sind. Externe Einflüsse in den Bahnverkehr durch Menschen oder die Natur sind Faktoren, auf die besonders zu achten ist. Auch bedarf es einer Überarbeitung der Eisenbahn-Bau- und Betriebsordnung (EBO), die es bislang verbietet, Fahrzeuge ohne Fahrpersonal zu bewegen.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorstehende Bachelorthesis mit dem Titel

Bau und Konzeption eines Zugleit- und Sicherungssystems für autonomen Fahrbetrieb

selbstständig ohne fremde Hilfe gefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quelle kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Elmshorn, den 23.12.2024

6 Verzeichnisse

6.1 Quellenverzeichnis

- [1] U. Maschek, Sicherung des Schienenverkehrs, Springer, 2012.
- [2] „stellwerke.de,“ Holger Kötting, [Online]. Available: <http://www.stellwerke.de/formen/>. [Zugriff am 9 12 2024].
- [3] D. InfraGO, Fahrdienstvorschrift; Richtlinie 408.11 - 16, 2021.
- [4] J. Pacht, Systemtechnik des Schienenverkehrs, Vieweg+Teubner, 2008.
- [5] „stellwerke.blogspot,“ 2014. [Online]. Available: <https://stellwerke.blogspot.com/2014/12/fernsteuerung-in-rohr-1987.html>. [Zugriff am 20 12 2024].
- [6] L. Schnieder, European Train Control System, 2021.
- [7] D. M. König, „Das neue Märklin-Motorola-Format,“ 2000.
- [8] Infineon, „Datenblatt IRLZ44N (PD-94831),“ [Online]. Available: <https://www.infineon.com/cms/de/product/power/mosfet/n-channel/irlz44n/>. [Zugriff am 27 09 2024].
- [9] Sharp, „Datenblatt PC817,“ [Online]. Available: <https://www.farnell.com/datasheets/73758.pdf>. [Zugriff am 28 09 2024].
- [10] T. Instruments, „Datenblatt TM4C123GH6PM,“ [Online]. Available: <https://www.ti.com/lit/ds/symlink/tm4c123gh6pm.pdf>.
- [11] T. Instruments, „Datenblatt LM2678,“ [Online]. Available: <https://www.ti.com/lit/gpn/LM2678>.
- [12] T. Connectivity, „Datenblatt D2N Relais,“ [Online]. Available: https://cdn-reichelt.de/documents/datenblatt/C300/D2N05-167_DB_EN.pdf.
- [13] CDIL, „Datenblatt BC547B,“ [Online]. Available: https://cdn-reichelt.de/documents/datenblatt/A100/BC546_48-CDIL.pdf.
- [14] Braband, Steinhausen und Mück, „Dokumentation Bachelorprojekt – Achszähler für Modelleisenbahnen,“ Prof. Dr. Paweł Buczek, 2021.
- [15] Infineon, „Datenblatt TLE4905L,“ [Online]. Available: <https://www.infineon.com/cms/de/product/sensor/magnetic-sensors/magnetic-position-sensors/magnetic-switches/tle4905l/>. [Zugriff am 30 10 2024].

- [16] Microchip/AVR, „Datenblatt ATmega 2560,“ [Online]. Available: http://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf. [Zugriff am 10 12 2022].
- [17] Philips, „Application note UM10204,“ [Online]. Available: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>. [Zugriff am 19 12 2024].
- [18] D. Kunz, „Fahrwegsuche nach dem Spurplanprinzip,“ 2014.
- [19] J. Ernst, Das Sp Dr S60-Stellwerk, Bahn-Fachverlag, 1993.
- [20] NXP, „Application note AN10216,“ [Online]. Available: http://www.nxp.com/documents/application_note/AN10216.pdf. [Zugriff am 19 12 2024].
- [21] P. D. P. Buczek, „HAW Hamburg Safe Layer Protocol Spezifikation (auf Anfrage ausgeteilt)“.
- [22] D. Bahn, „Umfassendes Infrastrukturprogramm für mehr Kapazität und Qualität,“ [Online]. Available: https://www.deutschebahn.com/de/presse/suche_Medienpakete/Umfassendes-Infrastrukturprogramm-fuer-mehr-Kapazitaet-und-Qualitaet-7250964. [Zugriff am 22 12 2024].
- [23] „stellwerke.info / Stellwerksliste,“ Arbeitskreis Stellwerk e.V., [Online]. Available: <https://stellwerke.info/stw>. [Zugriff am 9 12 2024].
- [24] Märklin, „Gleispläne,“ [Online]. Available: https://www.maerklin.de/fileadmin/media/service/faq/Brosch%C3%BCre_C-Gleis_Gleispl%C3%A4ne_Erg%C3%A4nzungspackungen.pdf. [Zugriff am 30 05 2024].

6.2 Literaturverzeichnis

- U. Maschek – Sicherung des Schienenverkehrs
- J. Pacht - Systemtechnik des Schienenverkehrs
- L. Schnieder - European Train Control System
- DB Infra Go – Fahrdienstvorschrift; Richtlinien 408.01 - 06
- Dr. M. König, „Das neue Märklin-Motorola-Format“

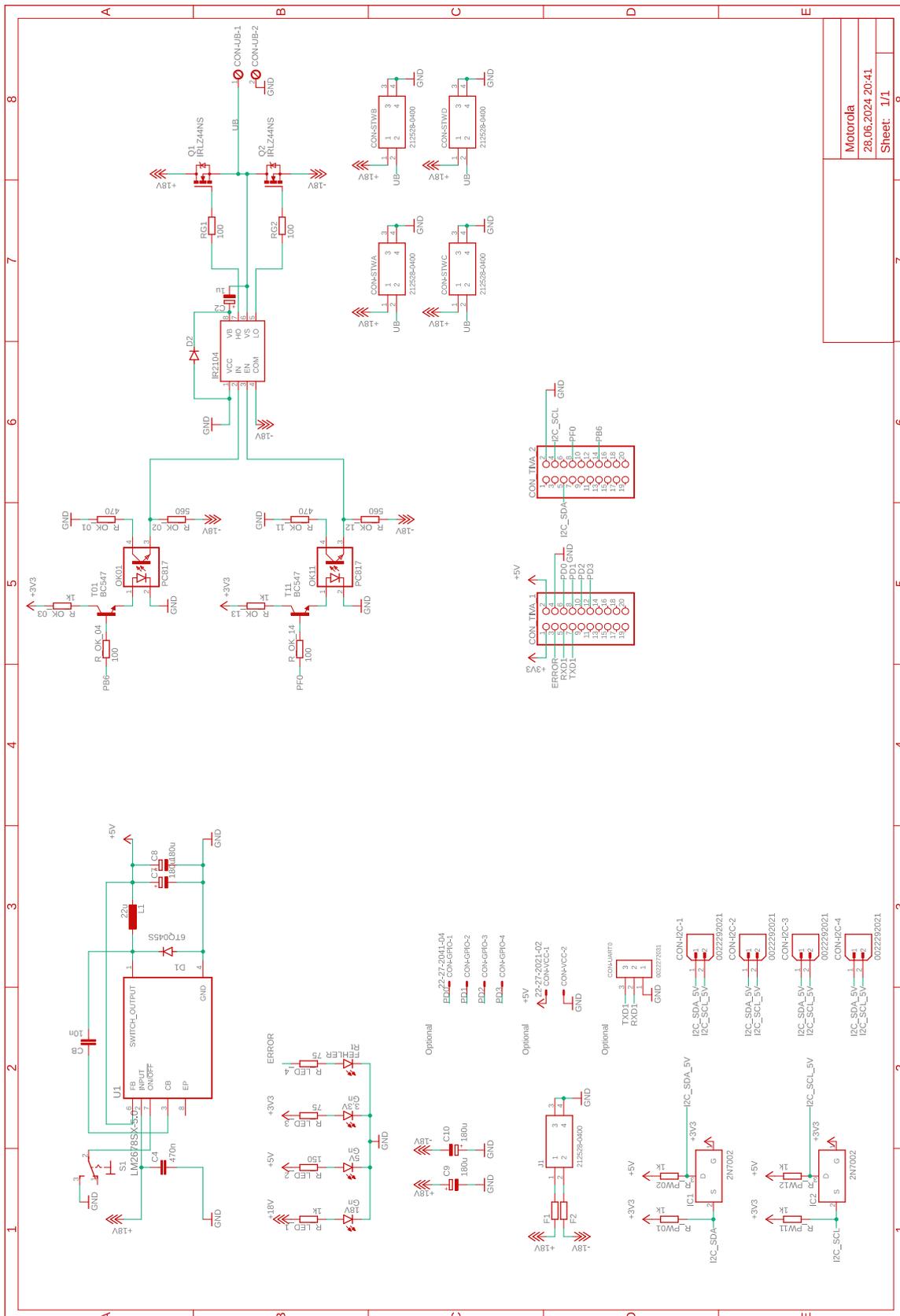
6.3 Abbildungs- und Tabellenverzeichnis

Abbildung 1 – Stelltisch eines Gleisbildstellwerks [6]	7
Abbildung 2 – C-Gleis mit Mittelleiter	10
Abbildung 3 – Gleisabschnittsplan (Gesamtkonzept)	11
Abbildung 4 – Bitfolge "01" im Motorola-Format.....	13
Abbildung 5 – Aufbau eines Motorola-Datenpakets [9]	13
Abbildung 6 – Sendereihenfolge der Datenpakete beim Motorola-Format	16
Abbildung 7 – Motorola-Endstufe: Halbbrücke mit Treiber (alle Spannungen beziehen sich auf GND als Bezugspotential)	18
Abbildung 8 – Motorola-Endstufe: Galvanische Trennung zwischen Controller und Treiber-IC mittels Optokoppler	20
Abbildung 9 – Optokoppler Eingangssignal (CH 2) und Ausgangssignal (CH1)	21
Abbildung 10 – Hauptmodul: Pegelwandler 3,3 V ↔ 5 V für I ² C Kommunikation	22
Abbildung 11 – Hauptmodul: 5 V Spannungsversorgung mittels Schaltregler	23
Abbildung 12 – Platinenlayout Hauptmodul.....	24
Abbildung 13 – Märklin Weichenantrieb ohne Gehäuse (Artikelnr. 74491)	24
Abbildung 14 – Stellwerk-Grundschialtung: Weichensteuerung und Rückmeldung.....	25
Abbildung 15 – Halt zeigendes Signal	26
Abbildung 16 – Stellwerk-Grundschialtung: Signalsteuerung	26
Abbildung 17 – Stellwerk-Grundschialtung: Gleisstromabschaltung	27
Abbildung 18 – Gleisstrom-Achszähler: Messung zweier Drehgestelle (je 2 Achsen); Messung des 2. DG fehlerhaft durch verschmutztes Rad	28
Abbildung 19 – Hall-Gleissensor.....	29
Abbildung 20 – Stellwerk-Grundschialtung: Hall-Sensor Beibeschialtung	29
Abbildung 21 – Platinenlayout Stellwerkmodul.....	30
Abbildung 22 – Struktur der Hauptmodul-Software.....	34
Abbildung 23 – Struktur der Stellwerk-Software	37
Abbildung 24 – Steuersoftware (Automatikbetrieb)	41
Abbildung 25 – Steuersoftware (Handbetrieb).....	42
Tabelle 1 – Liste verfügbarer ETCS-Betriebsmodi	9
Tabelle 2 – Verwendete Produkte der Firma Märklin.....	10
Tabelle 3 – Rechenbeispiel zu den Adressbits	13
Tabelle 4 – Kodierung der Fahrstufen beim Motorola-Format [9]	14
Tabelle 5 – Rechenbeispiel zu den S-Bits	14
Tabelle 6 – Fahrstufen-Kategorien bei Motorola II [9]	15
Tabelle 7 – Kategorien zur Funktionssteuerung [9]	15
Tabelle 8 – Sonderfälle des Motorola-Formats zwecks Abwärtskompatibilität [9]	16
Tabelle 9 – Rechenbeispiel zu den Sonderfällen.....	16
Tabelle 10 – Sendereihenfolge der Motorola-Datenpakete (1 Decoder angemeldet) [9]	17
Tabelle 11 – Sendereihenfolge der Motorola-Datenpakete (2 Decoder angemeldet) [9]	17
Tabelle 12 – Schaltzustände der Halbbrückenschialtung	18
Tabelle 13 – Features des TM4C123G Mikrocontrollers [12]	21

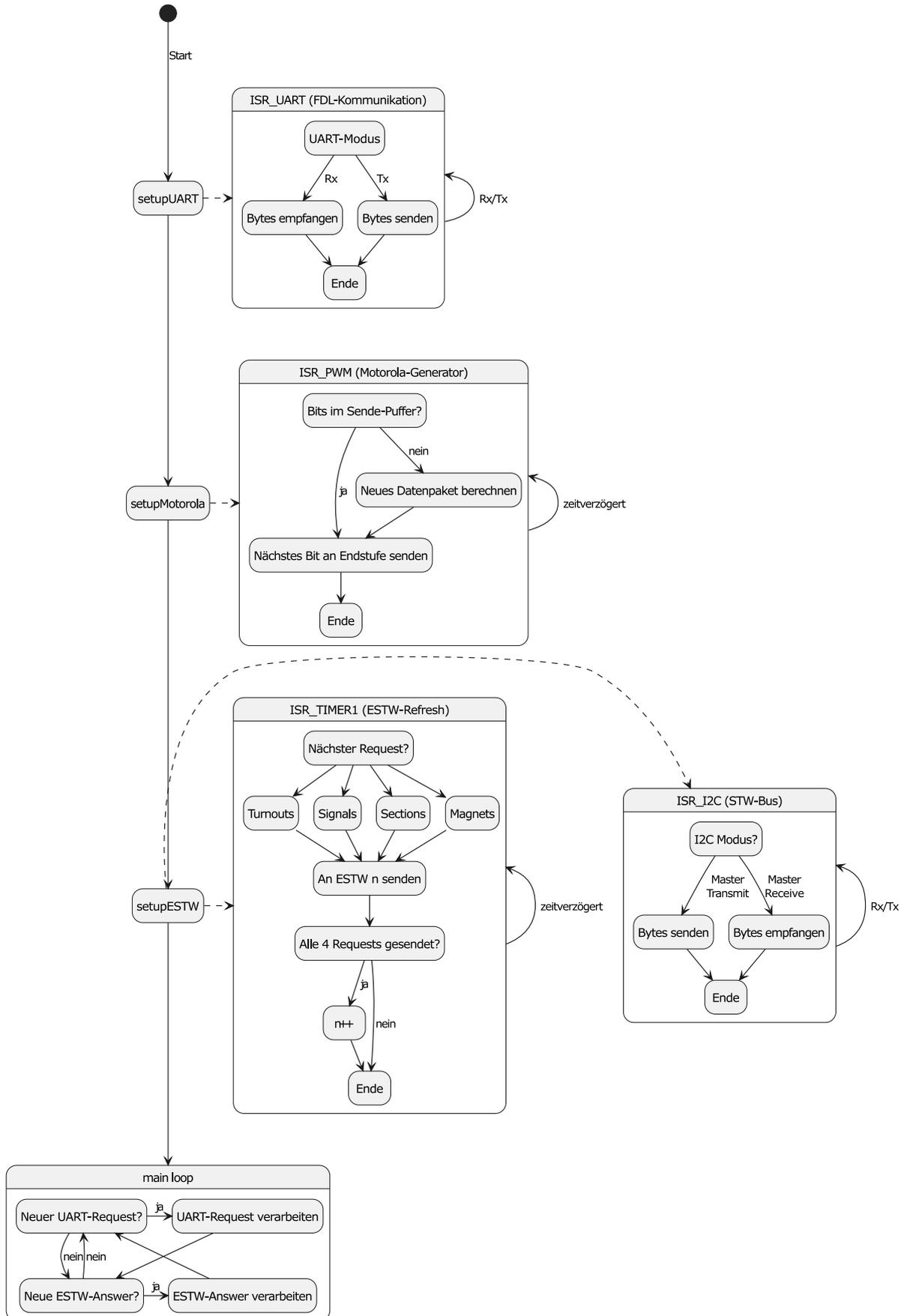
Tabelle 14 – Zustände des bidirektionalen Pegelwandlers.....	22
Tabelle 15 – Belegung der Weichen-Zustandsbits.....	26
Tabelle 16 – Quantität der Grundsaltungen des Stellwerkmoduls.....	30
Tabelle 17 – Liste aller Stellwerk-Bus-Requests.....	31
Tabelle 18 – Liste aller UART-Requests.....	33

7 Anhang

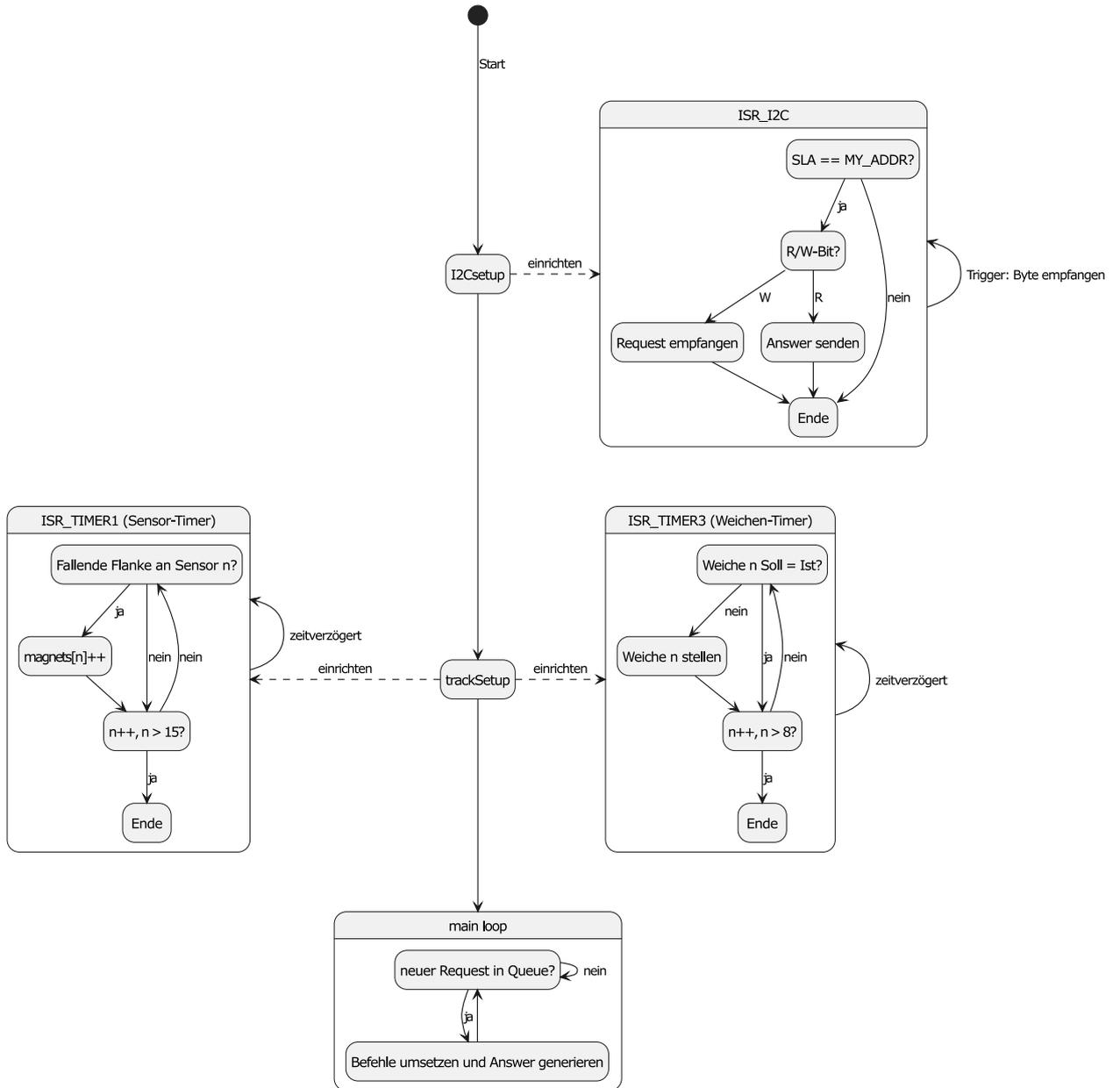
7.1 Schaltplan Hauptmodul



7.3 Programmfluss-Diagramm Hauptmodul



7.4 Programmfluss-Diagramm Stellwerkmodul



7.5 Programmfluss-Diagramm Fahrstraßenbildung

