

BACHELORTHESIS
René Zimmermann

Entwicklung und Aufbau eines Steuergerätes zur Datenverarbeitung und Regelung einer Wärmetauscher- Schaltschrankkühlung in Bereichen mit erhöhter Strahlenbelastung

FAKULTÄT TECHNIK UND INFORMATIK
Department Informations- und Elektrotechnik

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

René Zimmermann

Entwicklung und Aufbau eines Steuergerätes zur
Datenverarbeitung und Regelung einer
Wärmetauscher-Schaltschrankkühlung in Bereichen
mit erhöhter Strahlenbelastung

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Elektro- und Informationstechnik*
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Annabella Rauscher-Scheibe
Zweitgutachter: Prof. Dr. Björn Lange

Eingereicht am: 20. August 2021

René Zimmermann

Thema der Arbeit

Entwicklung und Aufbau eines Steuergerätes zur Datenverarbeitung und Regelung einer Wärmetauscher-Schaltschrankkühlung in Bereichen mit erhöhter Strahlenbelastung

Stichworte

Steuergerät, Schaltschrankkühlung, Strahlenbelastung, Rittal, Arduino

Kurzzusammenfassung

Das zunehmende Alter und die zusätzliche Strahlenbelastung lassen immer mehr Steuergeräte der Schaltschrankkühlung im Tunnel des Linearbeschleunigers XFEL ausfallen. Da der Support zu diesen Geräten eingestellt wurde und der Umbau auf die neuen Systeme zu kostenintensiv ist, wird ein dualer Student vom Deutschen Elektronen-Synchrotron mit der Aufgabe betraut, das Problem durch Entwicklung eines Ersatzes zu lösen.

René Zimmermann

Title of Thesis

Development and construction of a control unit for data processing and control of a heat exchanger switch cabinet cooling in areas with increased radiation exposure

Keywords

Control unit, control cabinet cooling, radiation exposure, Rittal, Arduino

Abstract

Increasing age and additional radiation exposure are causing more and more control units of the control cabinet cooling system in the tunnel of the linear accelerator XFEL to fail. Since support for these devices has been discontinued and conversion to the new systems is too cost-intensive, a dual student from the Deutsches Elektronen-Synchrotron is entrusted with the task of solving the problem by developing a replacement.

Inhaltsverzeichnis

Abbildungsverzeichnis	VII
Tabellenverzeichnis	VIII
1 Einleitung	1
1.1 Allgemeines	1
1.2 Motivation	1
1.3 Zusammenfassung der Aufgabe	2
1.4 Aufbau der Ausarbeitung	3
2 Analyse der Rahmenbedingungen	4
2.1 Standort der Anlage	4
2.1.1 Allgemeines über den Standort	4
2.1.2 Deutsches Elektronen-Synchrotron DESY	6
2.1.3 European XFEL	8
2.2 Anlagenüberblick	10
2.2.1 Rittal	10
2.2.2 Schaltschrank-Kühlsystem	11
2.3 Aktuelle Problematik	14
2.3.1 Problemübersicht	14
2.3.2 Zukünftige Probleme	15
3 Anforderungsanalyse	16
3.1 Anforderungen an das System	16
3.1.1 Vorgaben des Auftraggebers	16
3.1.2 Mögliche Lösungsansätze	17
3.2 Reengineering durch Reverse Engineering	18
3.2.1 Begriffserklärung und Ziele	18
3.2.2 Reverse Engineering des Rittal Produktes	19
3.2.3 Reverse Engineering des Temperatursignals	21
3.2.4 Erworbene Kenntnisse	23
3.3 Wahl der Umsetzung	23
3.3.1 Wahl der Plattform	24

3.3.2	Einplatinencomputer	24
3.3.3	Mikrocontroller-Board	25
3.3.4	Eigenentwicklung	26
3.4	Hardwareauswahl	28
3.4.1	Allgemeines zur Bauteilwahl	28
3.4.2	Allgemeines zur Prozessorwahl	28
3.4.3	Bauteile des Netzteils	30
3.4.4	Speicherbaustein	31
3.4.5	Relais und Ansteuerung	32
3.5	Aufbau des Systems	32
3.5.1	Netzteil	32
3.5.2	Ein- und Ausgänge	33
3.5.3	Prozessoren	34
3.5.4	Peripherie	34
4	Theoretische Hintergründe	36
4.1	Bauteile eines Linearbeschleunigers	36
4.1.1	Allgemeines über Linearbeschleuniger	36
4.1.2	Hohlraumresonator	37
4.1.3	Undulator	38
4.1.4	Strahlstopper (Dump)	39
4.2	Ionisierende Strahlung	40
4.3	Speicherbausteine	40
4.3.1	EEPROM	41
4.3.2	Flash-Speicher	41
4.4	Mikrocontroller	42
4.4.1	Prozessor	42
4.4.2	Bootloader	43
4.4.3	Arbeitsspeicher	44
4.4.4	Watchdog	44
4.5	I ² C Bussystem	45
4.6	SNMP Protokoll	47
4.7	Leiterplattenfertigung	48
4.7.1	Programme zur Leiterplattenfertigung	48
4.7.2	Anfertigen eines Layouts	49
4.7.3	Optimieren des Layouts	50

4.7.4	Herstellung einer Leiterplatte	51
4.8	Arduino IDE	52
4.8.1	Bedeutung IDE	52
4.8.2	Programmiersprache	53
4.8.3	Programmbibliotheken	54
4.8.4	Vor- und Nachteile der Arduino IDE	54
4.9	Prüfung der Elektromagnetischen Verträglichkeit	55
4.9.1	Durchführung dieser Prüfung	56
4.9.2	Änderungen im Fehlerfall	57
5	Finale Hardwareumsetzung des Prototyps	58
5.1	Überblick	58
5.2	Wechselbarkeit der Bauteile	64
5.2.1	Austauschbare Prozessoren	64
5.2.2	Wechselbares EEPROM	65
5.2.3	Sicherungen und Überprüfung der Spannung	65
6	Umsetzung der Software	66
6.1	Überblick	67
6.2	Ablauf einzelner Funktionen	71
6.2.1	Schreiben des Logs auf den EEPROM	71
6.2.2	Ansteuerung des Lüftermoduls	73
6.3	Sicherheitsfeatures	76
6.4	Graphische Nutzeroberfläche über Webinterface	77
7	Überprüfung und Test	79
7.1	Ergebnis der Testphase	79
7.2	Ausblick	80
8	Fazit	82
	Literaturverzeichnis	84
	A Anhang	89
	Selbstständigkeitserklärung	90

Abbildungsverzeichnis

2.1	Position der Schaltschränke im Tunnel [18]	5
2.2	Gelände des Deutschen Elektronen-Synchrotrons mit Beschleunigern [40] .	7
2.3	Übersicht der XFEL Beschleunigeranlage [17]	9
2.4	Übersicht der Schaltschrank-Kühlanlage in der Draufsicht [45]	11
2.5	Produktbild der CMC-TC PU Rück- und Vorderseite [43]	12
2.6	Aufbau der Schaltschrank-Kühlung mit nummerierten Bauteilen [45] . . .	13
3.1	Abbildung eines Arduino Uno mit allen Anschlüssen [2]	27
4.1	Abbildung eines Beschleunigermoduls im XFEL [38]	36
4.2	Supraleitender Hohlraumresonator aus Niob [8]	38
4.3	Aufbau eines I ² C Datenpaketes [22]	46
4.4	Zeitdiagramm einer I ² C Kommunikation [22]	46
5.1	Scan der Platinenunterseite des Prototyps	59
5.2	Numerierter Scan der Platinenoberseite des Prototyps	60
5.3	Fertig bestückte Leiterplatte des Steuergerätes	62
5.4	Prozessoradapter für den Prototyp	63
6.1	Programmablaufplan des primären Prozessors	67
6.2	Programmablaufplan des sekundären Prozessors	69
6.3	Programmablaufplan der Funktion zum Schreiben eines Log-Eintrags . . .	71
6.4	Programmablaufplan der Funktion zum Kommunizieren mit einem Lüf- tersteuergerät: Empfangen von Daten	73
6.5	Programmablaufplan der Funktion zum Kommunizieren mit dem Lüfter- steuergerät: Senden von Daten	75
6.6	Darstellung der Daten auf der Webseite, oberer Teil	77
6.7	Darstellung der Daten auf der Webseite, unterer Teil, gekürzt	78

Tabellenverzeichnis

3.1	Darstellung der Übertragenen Daten mit dem Lüftermodul [44]	21
4.1	Maximal erlaubte Taktraten I ² C Bus [21]	45

1 Einleitung

1.1 Allgemeines

Diese Bachelorthesis ist eine Ausarbeitung einer Aufgabe vom Deutschen Elektronen-Synchrotron in Hamburg-Bahrenfeld. Diese Aufgabe wurde einem der firmeninternen dualen Studenten zugetragen, welcher in diesem Fall auch für diese Ausarbeitung verantwortlich ist. Das Ziel dieser Bachelorthesis ist es, das im Kapitel 1.3 genannte Problem zu beheben, indem ein Steuergerät entwickelt wird, welches letztendlich und nach den wichtigsten Prüfungen in Serie produziert werden kann, um das derzeit verwendete Gerät zu ersetzen.

1.2 Motivation

Bereits im September 2013 bekam die Firma Rittal, ein deutscher Schaltschrankhersteller, die Zusage zur Ausstattung des Linearbeschleunigers European XFEL mit Schaltschränken und dem dazugehörigen Kühlsystem [Vgl. 41].

Die Eröffnung des European XFEL fand Ende 2016 statt und im Mai 2017 wurde der erste Röntgenlaserstrahl erzeugt.

Im Jahre 2019 fingen die ersten Steuergeräte an auszufallen und somit keine geregelte Temperatur in den Schaltschränken zu garantieren.

Ein kurzer Reset der Geräte genügte zum Lösen des Problems. Dies war jedoch nicht von Dauer und die ersten Geräte mussten nach kurzer Zeit ausgetauscht werden. Da seitens Rittal die Produktion der Steuergeräte eingestellt wurde, weil eine neue Generation zum Verkauf steht, wurde der Support eingestellt. Folglich können keine neuen Geräte dieser Art gekauft werden.

Die neue Generation der Steuergeräte ist jedoch nicht mit den bereits verbauten kompatibel und deshalb wurden zunächst einige alte, unbenutzte oder gebrauchte Ersatzgeräte beschafft. Da diese nicht unbegrenzt zur Verfügung stehen und es zudem keine Garantie mehr auf diese Steuergeräte gibt, muss eine andere Lösung des Problems gefunden werden.

Ein Umbau auf die neuen Systeme würde den finanziellen Rahmen sprengen, da der Beschleuniger zum Umbau über den gesamten Zeitraum ausgeschaltet werden muss. Die Anschaffungskosten der neuen Schaltschränke wären hoch und der Einbau dieser stellt eine noch höhere Kostenstelle dar. Der Standort in einem kilometerlangen, mehrere Meter unter der Erde befindlichen Tunnel, verursacht in diesem Falle besagte Einbaukosten.

Es wurde entschlossen die Steuergeräte durch eine Eigenentwicklung zu ersetzen und einen dualen Studenten vom Deutschen Elektronen-Synchrotron mit dieser vielseitigen Aufgabe zu beauftragen. Da dieses Projekt, je nach Umsetzung, aus sehr vielen Teilbereichen der Elektro- und Informationstechnik besteht, ist dies eine sehr interessante, anspruchsvolle und lehrreiche Aufgabe. Außerdem spart sie der Forschungseinrichtung sehr viel Geld.

Je nach Umsetzung, werden die Bereiche Mikroprozessortechnik, Programmierung, Regelungstechnik und Elektronik angewandt. Natürlich lässt sich das Thema nicht nur auf diese Teile begrenzen, jedoch bilden sie den Hauptteil der Aufgabe.

1.3 Zusammenfassung der Aufgabe

Wie bereits in der Motivation angedeutet, handelt es sich um ein Problem, welches eine große und ausgewogene Aufgabe für diese Thesis darstellt. Es werden viele Bereiche der Elektro- und Informationstechnik abgedeckt und das Wissen, welches durch das Studium vermittelt wurde, kann genutzt und gefestigt werden.

Die Aufgabe ist es, ein Steuergerät zu entwickeln und aufzubauen, welches das aktuelle Gerät von Rittal ersetzt und die Regelung und Ansteuerung der vorhandenen Lüfter und des Regelkugelhahns übernimmt. Das Steuergerät befindet sich in einer Wasser-Luft-Wärmetauscher Einheit, welche die Kühlung von anliegenden Schaltschränken und den beinhalteten Geräten übernimmt.

Es gilt zu überprüfen auf welche Art und Weise dieses Steuergerät ersetzt werden kann und auf welche äußeren Umstände geachtet werden muss. Um das Projekt umsetzen zu können, wird analysiert welche Rahmenbedingungen bei der Arbeit mit dem Gerät berücksichtigt werden müssen. Dadurch entsteht ein Bild des Umfeldes der Anlage und worauf bei der Umsetzung geachtet werden muss.

Nach einer Analyse der Aufgabenanforderungen, können mehrere Lösungsansätze erstellt und ein Prototyp entwickelt werden.

1.4 Aufbau der Ausarbeitung

Zunächst werden die Rahmenbedingungen analysiert und somit ein Gesamtbild des Standortes und der zugehörigen Firmen geschaffen. Danach folgt die so genannte Anforderungsanalyse, mit der die Aufgabe analysiert und somit mehrere Lösungsansätze geschaffen werden. In diesem Abschnitt wird erläutert welche Wahl am Besten ist und warum diese Umsetzung gewählt wurde.

Im folgenden werden in der Analyse die Wahl der Bauteile und die Anordnung dieser erläutert.

Darauf folgen die theoretischen Grundlagen, die dem Leser zum Nachschlagen angeboten werden. Im Text wird gezeigt in welchem Kapitel sich die zugehörigen Grundlagen befinden. Dieses Kapitel ist sehr umfangreich, da viele unterschiedliche Themengebiete behandelt werden.

Nach den theoretischen Grundlagen wird das fertige Ergebnis beschrieben. Zunächst die Hardware- und danach die Softwareumsetzung.

Nach verkünden der Testergebnisse werden die bestehenden Probleme, deren Lösungsansätze und zukünftige Verbesserungen des Steuergerätes dargestellt.

Nach einem Fazit kommt diese Ausarbeitung zum Ende und es können die Anhänge begutachtet werden.

2 Analyse der Rahmenbedingungen

Die Analyse der Rahmenbedingungen stellt einen wichtigen Teil dieser Ausarbeitung dar. Mit Hilfe dieser können die Aufgabe und die äußeren Umstände des Problems nachvollzogen werden und es wird ein besseres Verständnis des Aufbaus der Anlage ermöglicht. Zunächst wird der Standort der Anlage beschrieben.

2.1 Standort der Anlage

Im Folgenden werden der Standort der Kühlanlage und die äußeren Umstände beschrieben. Zunächst ein allgemeiner Überblick mit darauf folgender genauer Beschreibung des Standortes.

2.1.1 Allgemeines über den Standort

Der Standort der Schaltschrank-Kühlanlagen ist im Tunnel des sogenannten European XFEL, mehr dazu im Abschnitt 2.1.3 (S. 8). Dieser befindet sich zwischen dem Campus des Deutschen Elektronen-Synchrotron in Bahrenfeld, Hamburg und dem Gelände des European XFEL in Schenefeld, Schleswig-Holstein. In diesem Tunnel befinden sich, neben der Röhre des Linearbeschleunigers, viele technische Gerätschaften zur Versorgung der Magnete des Beschleunigers, der Aufrechterhaltung des Vakuums und der Überwachung der Anlage. Der Aufbau und die Funktion dieser Anlage wird im oben genannten Abschnitt zum European XFEL erläutert.

Ein großer Teil der Geräte zur Steuerung und Überwachung befindet sich in Schaltschränken unter der Beschleuniger-Röhre. Zur Verdeutlichung der örtlichen Gegebenheiten ist in der folgenden Abbildung ein kurzes Stück der Anlage im Tunnel zu sehen. Die Position der abgebildeten Schaltschränke ist im Tunnel direkt hinter dem Injektor und stellt eine der ersten Beschleunigerstrecken dar.



Abbildung 2.1: Position der Schaltschränke im Tunnel [18]

Wie in der oben gezeigten Darstellung zu sehen ist, befindet sich eine Abschirmung gegenüber ionisierender Strahlung um die Schaltschränke. Mehr über diese Abschirmung kann im Kapitel 4.2 auf Seite 40 gelesen werden. Die Schaltschränke sind die grau-weißen Einheiten aus Metall. Die Abschirmung hält sich teilweise auf den stabilsten Punkten dieser Schränke. Über den Schaltschränken ist die gelbe Röhre des Linearbeschleunigers zu sehen.

Die Schaltschränke, oder auch Racks genannt, werden vom hessischen Hersteller Rittal gefertigt und beinhalten, neben den Rahmen für 19 Zoll-Einschübe, eine Einheit zur Kühlung der Schaltschränke. Diese Einheit kann neben einem Schaltschrank oder zwischen zwei Schaltschränken des selben Herstellers verbaut werden. Bei dieser Kühlungseinheit handelt es sich um einen Wasser-Luft-Wärmetauscher. Über einen Regelkugelhahn kann dessen Durchflussmenge und mittels drei Lüftern der Luftfluss gesteuert werden. Mit der verbauten Temperaturüberwachung kann somit die Temperatur im Schaltschrank auf einen eingestellten Wert geregelt werden.

Neben der Hardware zur Durchführung der Temperaturregelung befinden sich die zugehörigen Steuergeräte in unmittelbarer Nähe dazu.

Eines dieser Steuergeräte, mit welchem sich diese Ausarbeitung beschäftigt, ist das Hauptsteuergerät. Es kommuniziert mit den insgesamt vier anderen Steuergeräten.

Drei dieser Steuergeräte sind mit jeweils einem Lüfter und den dazugehörigen Temperatursensoren verbunden. Das vierte Steuergerät dient zur Ansteuerung des Wasserventils und ist mit den zugehörigen Temperatursensoren, sowie Feuchtigkeitssensor, Leckagesensor und Durchflusszähler verbunden.

Der genaue Aufbau der Kühlanlage wird in Kapitel 2.2.2 (S. 11) beschrieben.

2.1.2 Deutsches Elektronen-Synchrotron DESY

Das Deutsche Elektronen-Synchrotron ist ein Forschungszentrum für Grundlagenforschung im naturwissenschaftlichen Bereich. Es besitzt zwei Standorte in Deutschland, einen in Zeuthen, Brandenburg und einen in Hamburg-Bahrenfeld.

Auf dem Gelände des Deutschen Elektronen-Synchrotrons, im weiteren mit der Abkürzung DESY genannt, sind viele Anlagen zur Forschung an und mit Teilchen aufgebaut. Es wird in den Bereichen der Photonen, Teilchen- und Astroteilchenphysik geforscht und besitzt als weiteren Schwerpunkt die Arbeit mit Teilchenbeschleunigern. Diese werden dort entwickelt, geplant und gebaut.

DESY wurde am 18. Dezember 1959 in Hamburg gegründet und ist Mitglied der Helmholtz-Gemeinschaft. Seit der Gründung wurde der Campus um viele Beschleunigeranlagen ergänzt, welche zum Teil in internationaler Zusammenarbeit finanziert, gebaut und betreut werden.

Der Standort in Zeuthen wurde am 1. Januar 1992 aus dem damaligen Institut für Hochenergiephysik gegründet [Vgl. 12].

Eines der größten und das erste internationale Projekt war der Bau des Beschleunigers HERA. So beteiligten sich mehr als 45 Institute aus zwölf Ländern am Anlagenbau des Beschleunigers, dessen Ring mit einem Umfang von 6336 m der größte bei DESY in Hamburg-Bahrenfeld errichtete Beschleuniger ist [Vgl. 13].

Die folgende Abbildung zeigt eine Auswahl der sich am DESY befindlichen Beschleuniger und verdeutlicht die Größe des HERA Beschleunigers.

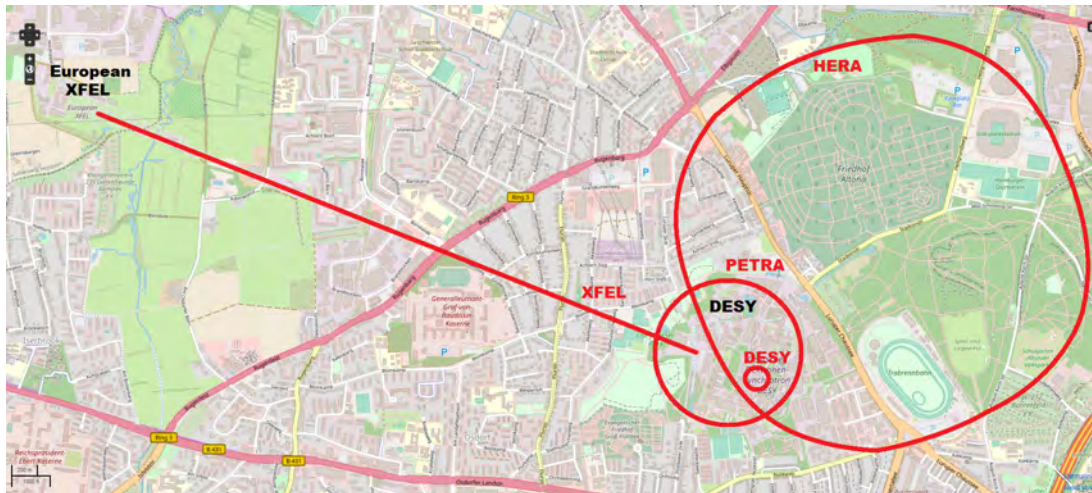


Abbildung 2.2: Gelände des Deutschen Elektronen-Synchrotrons mit Beschleunigern [40]

In der Abbildung 2.2 sind die Ausmaße der riesigen HERA-Tunnelanlage im Untergrund von Bahrenfeld zu sehen. Insgesamt gibt es vier Zugänge mit einem dazugehörigen Gelände zum Beschleuniger. Des Weiteren sind die Beschleuniger PETRA, DESY und XFEL gekennzeichnet.

Der erste Teilchenbeschleuniger war der eben genannte DESY, dessen Bau im Jahr 1960 begann. Am 1. Januar 1964 wurden die ersten Elektronen im Synchrotron beschleunigt. Dem ersten Beschleuniger folgten über die kommenden Jahre viele weitere, wie zum Beispiel DORIS, PETRA, FLASH und andere. Unter Beteiligung von DESY und anderen Instituten wurde der Beschleuniger XFEL gebaut.

Unter den vielen Forschungserfolgen, welche bei DESY gemacht wurden, steht zum Beispiel die Entdeckung des Gluons. 1978 wurde dieses Teilchen an dem PETRA Beschleuniger entdeckt [Vgl. 11]. Des Weiteren wurde 2009 der Nobelpreis an Prof. Ada E. Yonath verliehen, nachdem sie bei DESY die Struktur und Funktion der Ribosomen erforschte. Dies ermöglicht neue Ansatzpunkte zum Erforschen neuer Medikamente wie zum Beispiel Antibiotika [Vgl. 10].

2.1.3 European XFEL

Bei dem European XFEL handelt es sich um eine internationale Forschungseinrichtung, welche mit Hilfe von Röntgenlasern arbeitet. Insgesamt sind zwölf Länder an der Einrichtung beteiligt und es besteht eine enge Zusammenarbeit mit dem Deutschen Elektronen-Synchrotron. Zwischen den Standorten Hamburg-Bahrenfeld und Schenefeld befindet sich eine Tunnelanlage, welche mehrere Zugänge besitzt.

XFEL steht für X-Ray Free-Electron Laser, übersetzt Freie-Elektronen-Laser mit Röntgenstrahlung. Dieser Röntgenlaser kann genutzt werden um Viren, Zellen, Moleküle und chemische Reaktionen in einer dreidimensionalen Detailaufnahme festzuhalten. Dies geschieht nicht über einen kontinuierlichen Laserstrahl, sondern durch Röntgenblitze mit einer Dauer von nur Femtosekunden, welche auf die Probe gelenkt werden.

Der Planungsbeginn der 3,4 km langen Anlage war Anfang 2003 und das Ende der Bauphase, der weitestgehend unterirdischen Anlage, im Jahre 2016.

Nach der Eröffnung im Oktober 2016 wurde im Mai 2017 der erste Röntgenlaserstrahl erzeugt und die Einweihung und erste Forschungsarbeiten fanden im September 2017 statt. Seitdem wird in den Experimentierhallen auf dem Campus in Schenefeld geforscht [Vgl. 29].

Der Strahl wird auf dem DESY Gelände erzeugt und wird in dem Tunnelstück zwischen den Standorten beschleunigt. Es handelt sich in dem Fall nicht um einzelne Elektronen, die beschleunigt werden, sondern um Elektronenpakete. Diese Pakete werden nach der Beschleunigung in verschiedenen Anordnungen von Magneten in Form gebracht und erneut beschleunigt.

Die Beschleunigung der Elektronen findet in so genannten Hohlraumresonatoren, auch Cavities genannt, statt. Mehr dazu im Kapitel 4.1 (S. 36).

Nach der Beschleunigung werden die Elektronen gezielt in sogenannten Undulatoren, siehe Kapitel 4.1, auf Kurvenfahrt geschickt um somit Synchrotronstrahlung in Form von Röntgenstrahlung zu erzeugen. Die Elektronen werden danach von dem Röntgenlicht getrennt und die Laserimpulse werden dann über spezielle Spiegel in die einzelnen Experimentierhütten umgelenkt.

Im folgenden ist eine Übersicht der Anlage zu sehen. Der eigentliche Beschleunigertunnel bis zur Aufspaltung der Elektronen in der Abzweigung XS1, ist stark eingekürzt, jedoch kann der weitere Verlauf gut nachvollzogen werden.

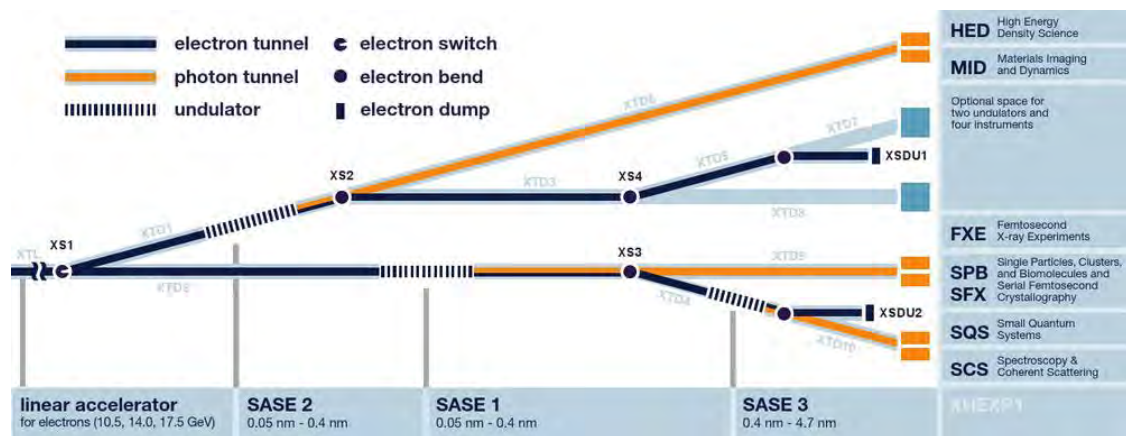


Abbildung 2.3: Übersicht der XFEL Beschleunigeranlage [17]

Wie in der Grafik 2.3 zu erkennen ist, startet die Beschleunigung nach der Erzeugung und Injizierung der Elektronenpakete im XTL. Nach der Beschleunigung wird der Elektronenstrahl im XS1 auf zwei Strahlstrecken aufgeteilt. Auf beiden Strecken werden die Elektronen durch Undulatoren zum Aussenden der Röntgenstrahlung gebracht.

Die Elektronen und Photonen verlaufen in zwei getrennten Rohren. Während der Röntgenläser direkt zu den Experimentierhütten geleitet wird, werden die Elektronenpakete umgelenkt. Diese werden zum Teil erneut durch Undulatoren geschickt und letztendlich in den Elektronenabsorbern, oder auch Dumps genannt, siehe Kapitel 4.1.4 (S. 39), gestoppt.

Forschungsgebiete

Seitdem der Nutzerbetrieb im European XFEL startete, wurde mit Hilfe dieses Beschleunigers an vielen Experimenten aus den Fachgebieten der Physik, Chemie, Biologie, Materialwissenschaft und Nanotechnologie geforscht.

Es wurden biologische Strukturen wie Zellen, Proteine und Membranen analysiert, welche zur Entwicklung neuer Arzneimittel und Therapien beiträgt.

Untersuchungen an Pflanzen können den Vorgang der Photosynthese genauer beobachten und so die Forschung im Energiesektor voran treiben.

Außerdem wurden Experimente am Coronavirus durchgeführt um die Bekämpfung dieses Virus voranzutreiben.

Der Anwendung sind keine Grenzen gesetzt und es werden zukünftig viele Erfolge in der Wissenschaft zu beobachten sein.

2.2 Anlagenüberblick

Nach der Analyse des Standortes an sich befasst sich dieses Kapitel mit dem Hersteller des Schaltschrank-Kühlsystems und mit dem Kühlsystem an sich.

2.2.1 Rittal

Rittal ist ein deutscher Hersteller und Anbieter für verschiedene Schaltschranksysteme, Klimatisierung, Stromverteilung, IT-Infrastruktur, Software und Service. Der Sitz ist in Herborn in Hessen und dort wurde die Firma 1961 gegründet. Es ist Mitglied der Friedhelm Loh Group und beschäftigt 9700 Mitarbeiter in acht Produktionsstätten und 58 Tochtergesellschaften.

Die Sparte der Schaltschranksysteme bezieht sich auf viele Arten von Schaltschränken. Diese unterscheiden sich in ihren Bauformen, den Materialien oder der Art der Einbausysteme. Spezielle Schaltschranksysteme sind unter anderem elektromagnetisch geschirmte oder erdbebensichere Systeme.

Klimatisierung wird bei Rittal durch Eigenständige Kühlgeräte oder verschiedene Kühl- und Heizsysteme für Schaltschränke angeboten. Diese Systeme können lüfterlos, mit Lüftern oder auch als Wasser-Luft-Wärmetauscher erstanden werden. Bei dieser Ausarbeitung handelt es sich um die Steuerung für ein Wasser-Luft-Wärmetauscher System.

Die anderen Sparten beziehen sich unter anderem auf Sammelschienensysteme im vorbereiteten Schaltschrank oder auf Überwachungssysteme für die IT-Sicherheit.

Das Wasser-Luft-Wärmetauscher System wird Liquid Cooling Package, kurz LCP, genannt. Es besteht aus einem oder zwei Standardschaltschränken, in deren Mitte ein 30 cm breites Element zur Kühlung angebracht ist. Im folgenden Kapitel 2.2.2 wird der Aufbau und die Funktion des Systems erläutert [Vgl. 46].

2.2.2 Schaltschrank-Kühlsystem

In der folgenden Übersicht ist dargestellt, wie das System funktioniert und aufgebaut ist. Es zeigt den Aufbau im Schaltschrank in der Draufsicht und stellt die Luftführung dar.

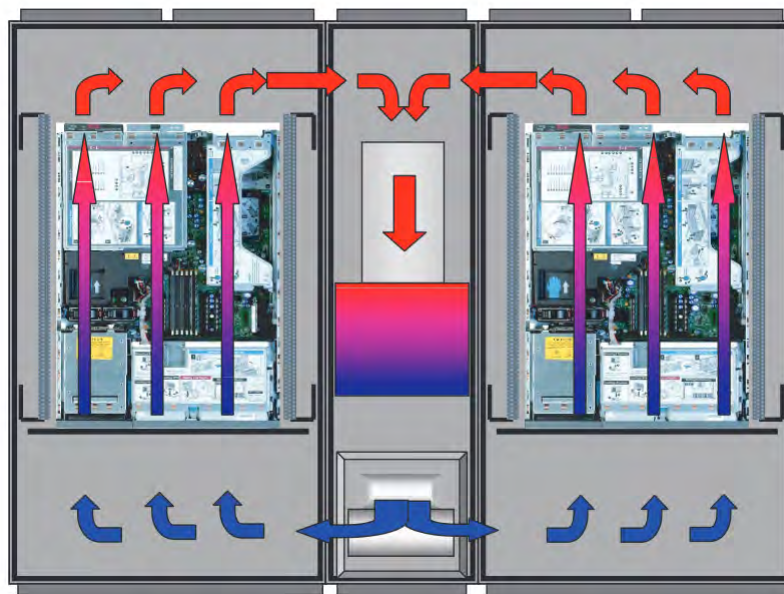


Abbildung 2.4: Übersicht der Schaltschrank-Kühlanlage in der Draufsicht [45]

In der oben gezeigten Abbildung wird der Luftfluss in der genannten Schaltschrank-Einheit dargestellt. Unten mittig im Bild befinden sich die Lüfter, welche die Luft im Schaltschrank durch den Wasser-Luft-Wärmetauscher zieht. Diese gekühlte Luft strömt links und rechts durch die verbauten 19-Zoll-Einschübe, nimmt die Abwärme dieser auf und wird wieder durch den Wärmetauscher gezogen. Der Wärmetauscher wird durch das rot-blaue Bauteil in der Mitte der Abbildung dargestellt.

Das Hauptsteuergerät, die Computer Multi Control Top Concept Processing Unit, kurz CMC-TC PU, kommuniziert zusätzlich über Ethernet und einer Website mit den Anwendern. Es können außerdem Daten per SNMP abgefragt und empfangen werden. Dieses Protokoll wird im Kapitel 4.6 auf Seite 47 behandelt.

Die Hardware des CMC-TC PU ist in einem 13,5 cm x 12,8 cm x 4,5 cm großem Gehäuse verbaut und ist auf dem folgenden Bild zu sehen.



Abbildung 2.5: Produktbild der CMC-TC PU Rück- und Vorderseite [43]

Das untere Steuergerät zeigt die Rückseite des Gerätes. Hier sind links Anschlüsse für vier Zusatzsensoren, welche jedoch nicht genutzt werden. Daneben befindet sich der so genannte P-I²C Anschluss und zwei Buchsen für ein Grafikdisplay und ein Alarmrelais. Bei den verbleibenden Anschlüssen handelt es sich um den Netzwerk- und dem runden Stromversorgungsanschluss.

Das obere Steuergerät zeigt die Frontansicht mit dem Taster zum Zurücksetzen der Fehlermeldungen, den Status-LEDs und einem seriellen Anschluss.

Die folgende Abbildung zeigt die Position der Steuergeräte im Schaltschrank. Die wichtigsten Bauteile sind nummeriert und werden in der Bildbeschreibung aufgezählt. Wie bereits erwähnt, sind die Ansteuerung der Lüfter und die Lüfter selbst redundant verbaut. Dies beschreibt die Wiederholung der Nummern.



Abbildung 2.6: Aufbau der Schaltschrank-Kühlung mit nummerierten Bauteilen [45]

Auf der Abbildung ist der Schaltschrank mit geöffneter Tür zu sehen. In diesem Mittelstück befindet sich die Technik zur Kühlung. Es werden die mit Zahlen markierten Bauteile aufgezählt:

- 1: Netzteil zur Spannungsversorgung der Steuergeräte und der Lüfter
- 2: CMC-TC PU Hauptsteuergerät
- 3: Kühlwasser-Steuergerät
- 4: Lüfter mit Saugseite nach hinten
- 5: Luftzufuhr Steuergerät

Hinter den gezeigten Bauteilen befinden sich der Wasser-Luft-Wärmetauscher, die Zuleitungen und die Temperatursensoren.

2.3 Aktuelle Problematik

Dieser Abschnitt behandelt die aktuelle Situation in den Schaltschränken im European XFEL. Es wird ein kurzer Überblick der Problematik geschaffen, welche Bereiche am meisten Betroffen sind und welche zukünftigen Probleme dadurch und durch andere Einflüsse entstehen. Eine kurze Beschreibung des Aufwands eines Neuaufbaus soll einen Einblick in die finanzielle Situation geben und warum das Verwenden der bestehenden Anlage besser ist als ein Neuaufbau.

2.3.1 Problemübersicht

Durch den Standort im Tunnel unter einem Linearbeschleuniger tritt, trotz Abschirmung, ionisierende Strahlung auf, die sowohl Bits in Speichermedien manipulieren, als auch die integrierten Schaltkreise der komplexen Bauteile beschädigen können. Mehr dazu kann im Kapitel 4.2 (S. 40) gelesen werden. Dies sorgt zu immer häufiger auftretenden Ausfällen der Elektronik der Anlage.

Hinzu kommen die Auswirkungen der langen Laufzeit. Elektrolytkondensatoren trocknen, deren Kapazität sinkt und Spannungen werden weniger gut geglättet und gehalten, wodurch das System instabil werden kann. Diese Auswirkungen lassen sich nur durch eine aufwändige Reparatur beheben, liegen jedoch im Rahmen des Möglichen.

Die lange Laufzeit hat auch Auswirkungen auf Speichermedien. Bauteile oder Bereiche in integrierten Schaltkreisen, welche zur Speicherung von Logs und Konfigurationen dienen, verschleißten durch häufiges Überschreiben. Der Grund dafür kann in Kapitel 4.3.1 auf Seite 41 nachgelesen werden. Dieses Problem kann jedoch nicht ohne weiteres repariert werden.

Zusätzlich, zu den bereits genannten Problemen, kommt es in der Nähe von so genannten Strahlstoppfern zu einer besonders hohen Strahlungsbelastung. Mehr zu den Strahlstoppfern kann in Kapitel 4.1.4 (S. 39) nachgelesen werden. Hier war die Ausfallrate besonders hoch.

Zusammen mit dem eingestellten Support der Steuergeräte ergibt dies ein Problem, welches normalerweise nur durch den Tausch der gesamten Kühlanlage gelöst werden kann.

2.3.2 Zukünftige Probleme

Noch sind ein paar Ersatzgeräte bei DESY verfügbar und auch auf dem Gebrauchtwarenmarkt werden noch einige angeboten. Jedoch ist bei diesen die Laufzeit unbekannt und es können die selben Probleme mit den Speicherbausteinen auftreten wie bei den bestehenden Steuergeräten.

Manche in den gekühlten Schaltschränken befindlichen Geräte sind auf eine geregelte Temperatur angewiesen. Ein Ausfall eines Steuergerätes bedeutet in diesem Fall, dass die Lüfter mit voller Leistung angesteuert werden und sich die Temperatur im Schaltschrank ändert. Dies wirkt sich auf die verbauten Gerätschaften aus und es können dadurch Messergebnisse verfälscht werden.

Wird keine Alternative zum Austausch der gesamten Anlage gefunden, so kann davon ausgegangen werden, dass dies sehr kostenintensiv wird.

Es handelt sich um über 150 Racks, deren Kühlung erneuert werden müsste. Der European XFEL darf während des Wechsels nicht laufen, die Abschirmungen müssten hochgenommen und die Kühlelemente getauscht werden. Der Standort in einem Tunnel erschwert den Transport der Schaltschränke. Verbunden mit den Arbeitskosten ergibt dies eine hohe Gesamtsumme.

Durch diese Problematiken kann ein Neuaufbau der Kühlanlage finanziell nicht in Frage kommen.

3 Anforderungsanalyse

3.1 Anforderungen an das System

Dieses Kapitel befasst sich mit der Analyse der geforderten Aufgaben. Es wird analysiert, welche Vorgaben der Auftraggeber fordert und welche Maßnahmen zur sicheren Funktion des Gerätes beitragen.

Im Anschluss werden mehrere Lösungsvorschläge geprüft und eines davon ausgewählt.

3.1.1 Vorgaben des Auftraggebers

Nach einer Schilderung der aktuellen Situation, wurde die Aufgabe formuliert, eine Steuerung oder Regelung der Schaltschrankkühlung zu entwickeln, damit das derzeit verbaute System von Rittal auf lange Sicht ersetzt werden kann.

Die Vorgaben des Auftraggebers wurden nach einer kurzen Diskussion und Ideenfindung detaillierter und durch einige Ergänzungen nach kurzer Zeit final. Es wurden folgende Punkte ausgehandelt, die eine Entwicklung eines Steuergerätes behandeln.

- Ersetzen des aktuellen Steuergerätes
- Die vorhandenen sekundären Steuergeräte ansteuern, wenn möglich
- Bietet Zusätzlich drei Relais-Anschlüsse, die über Ethernet schaltbar sind
- Verfügt über ein Webinterface zur Datenübersicht
- Passt in das vorhandene Gehäuse
- Sendet über SNMP Werte an das vorhandene Überwachungssystem
- Erstellt Log-Daten zur Fehlererkennung
- Ausfallsicherheit, auch in Bezug auf ionisierende Strahlung

Diese Punkte decken die aktuelle Funktion komplett ab und als zusätzliche Funktion können die drei Relais geschaltet werden.

3.1.2 Mögliche Lösungsansätze

Dieser Abschnitt behandelt die Abwägung, auf welche Art und Weise die Anforderungen erfüllt werden können. Es werden die drei folgenden Möglichkeiten in Betracht gezogen.

Temperatursteuerung über einen Zweipunktregler

Die erste Möglichkeit soll verdeutlichen, wie schnell sich die Anforderungen geändert haben und wie von einer einfachen, jedoch gut funktionierenden Idee, ein Ansatz zum Aufbau eines Steuergerätes entstand.

Eine Überlegung zum Ersetzen der defekten Steuergeräte war die Verwendung eines einfachen Zweipunktreglers. Die Funktion eines Zweipunktreglers ist ähnlich der Funktion des Bimetalls im Bügeleisen. Dieses schaltet unter einem gewissen Sollwert die Spannung ein und die Oberfläche wird durch die Heizdrähte erhitzt. Beim Überschreiten dieses Sollwertes öffnet das Bimetall den Kontakt und schaltet die Spannungsversorgung ab.

Die Umsetzung der Idee ist folgende. Alle verbauten Steuergeräte aus dem Schaltschrank werden entfernt und die Anschlüsse der Lüfter werden direkt über ein Relais mit dem Zweipunktregler verbunden.

Dieser Lösungsansatz ist ein einfacher, kostengünstiger Ansatz, dessen Funktion voll und ganz gegeben ist. Jedoch fehlt in diesem Fall die Überwachung der Temperaturen, welche für die in den Schaltschränken verbauten, teils temperatursensiblen, Geräte sehr wichtig ist. Die Temperaturschwankungen wären zu hoch und somit fällt diese Option aus. Die gewünschte Funktion der schaltbaren Relais über Ethernet wäre somit auch ausgefallen.

Neuaufbau der Kühlanlagenelektronik

Ein weiterer Ansatz zur Lösung der Elektronikprobleme ist der Neuaufbau der gesamten Schaltschrank Elektronik. Dieser Ansatz bringt die Vorteile, alle genannten Anforderungen einzuhalten und das System nach Belieben umzubauen.

Die Vorteile überwiegen den Nachteilen, jedoch sind zwei negative Punkte in Form von Platzmangel und der Ansteuerung der Lüfter vorhanden.

Der Platzmangel entsteht, da es sich, wie bereits in der Abbildung 2.6 zu sehen ist,

um insgesamt vier kleine Steuergeräte handelt. Diese sind in dem Mittelteil des Schaltschranks verteilt. Somit muss ein Halter entworfen werden, welcher das Gehäuse des neuen Steuergerätes an der gewünschten Stelle positioniert. Es muss ein Gehäuse gewählt werden, welches genug Platz für die genutzten Ein- und Ausgänge besitzt.

Der zweite Nachteil ist die Ansteuerung der Lüfter, denn diese werden fest mit der Netzspannung verbunden und über einen extra Anschluss gesteuert. Die Signale müssen abgegriffen und analysiert werden.

Entwicklung eines neuen Steuergerätes

Der vorherige Ansatz war bereits sehr gut, jedoch gibt es eine dritte Option, welche in Betracht zieht, nur das Hauptsteuergerät des Systems zu ersetzen. Die Sekundärsteuergeräte sind sehr einfach aufgebaut und da diese bis zu diesem Zeitpunkt noch nicht ausgefallen sind, können diese weiterverwendet werden.

Es kommt bei der richtigen Auswahl der neuen Komponenten im Steuergerät zu keinem Platzmangel und es kann sogar das Gehäuse des alten Steuergeräts weiterverwendet werden. Es muss lediglich eine neue Frontplatte gefräst werden, da neue Anschlüsse für die Relais ergänzt werden.

Ein großer Nachteil ist auch hier das Analysieren der ausgegebenen Daten, welche die Datenübertragung zwischen den Geräten betrifft. Kann dieses Signal erfasst und entschlüsselt werden, so steht diesem Ansatz nichts im Wege. Die Erfassung der übertragenen Daten wird im folgenden Kapitel 3.2 behandelt.

3.2 Reengineering durch Reverse Engineering

In diesem Kapitel wird die Bedeutung und die Methodik des Reverse Engineerings erläutert, da dieses beispielsweise beim Ersetzen von Elektroniken eine große Rolle spielt. Auch in diesem Projekt wurde die Methode zum Verständnis der übertragenen Daten und dem Auswerten der Temperatursensoren angewandt.

3.2.1 Begriffserklärung und Ziele

Das Reverse Engineering ist Teil des Reengineering und bezeichnet die Umkehrung des Entwicklungsprozesses von einem fertigen System bis hin zu Konstruktionszeichnungen

und Quellcodes. Es wird in der Produktentwicklung, der Qualitätsprüfung und auch bei der Fehlersuche genutzt.

Der Begriff stammt aus dem Maschinenbau und wurde genutzt, wenn reale Objekte mit Hilfe von verschiedenen Mess- und Scanverfahren zu einem CAD-Modell umgewandelt wurden.

In der Software-Entwicklung bedeutet dies die Analyse von vorhandenen Maschinencodes oder der Dekompilierung dieser.

Wird Reverse Engineering bei der Entwicklung von Hardware genutzt, so bereitet dies kaum Probleme, da das Patentrecht in diesem Falle nur bei identischen Kopien greift. Beim der Software Analyse ist das Dekompilieren jedoch verboten und wird durch die Gesetzgebung der EU, siehe [19], verboten. Das Verbot greift nicht im Falle der Anpassung von Schnittstellen.

Werden nach der Analyse durch Reverse Engineering neue Anforderungen in das Konzept integriert und ein neues Produkt erstellt, so wird dies durch den Begriff Forward Engineering beschrieben. [Vgl. 1]

3.2.2 Reverse Engineering des Rittal Produktes

Die Hardware des zur Zeit verwendeten Steuergerätes ist für die Entwicklung einer neuen Elektronik trivial. Es müssen nur die vorhandenen Anschlüsse für die Stromversorgung und die Ein- und Ausgänge übernommen werden.

Das Augenmerk liegt auf dem Anschluss des Bussystems, welches von Rittal genutzt wird. Nur so können die gesendeten Daten analysiert und im besten Fall im neuen Programm übernommen werden.

Das studieren der Bedienungsanleitung war in diesem Fall sehr wichtig. Durch Anmerkungen wie in dem folgenden Zitat, fiel es leichter die Datenübertragung zu analysieren.

“Der 8-polige Steckverbinder X5 (AMP MTA-Stiftleiste 640457-8) enthält den sog. Power-I²C-Bus. Belegung des I²C-Bus: 1 - n.c., 2 - Gnd, 3 - n.c., 4 - Gnd, 5 - Gnd, 6 - + 24 V, 7 - P-SDA, 8 - P-SCL.” [44, S. 77]

Es handelt sich um einen sogenannten "P-I²C"Bus, das "P" steht für "Power". Mit diesen Kenntnissen wurde ein Oszilloskop angeschlossen und es ergab sich, wie durch die Anschlüsse SCL und SDA erwartet, ein typisches I²C Muster, mehr dazu im Abschnitt 3.2.4 (S. 23). Da die Spannung nicht höher als Standard war, also 5 V, resultiert daraus,

dass "Power" nicht die Daten- und Taktleitung betrifft.

Eine kurze Analyse des Anschlusses, in dem Zitat wird ein älteres CMC Modell beschrieben, zeigte zwei zusätzliche Leitungen zu den beiden benötigten für den Bus. Diese beiden Leitungen versorgen die sekundären Steuergeräte mit 24 V und der dazugehörigen Masseleitung. Der Begriff "Power" beschreibt nur die Anschlussart mit der zusätzlichen Stromversorgung.

Nach der Analyse der Übertragungsart werden die Übertragenen Daten analysiert. Über einen Mikrocontroller werden die Adressen der Steuergeräte ermittelt und die Daten über den "Request"-Befehl angefragt. Der verwendete Mikrocontroller ist ein Arduino Uno und es wird die Wire Bibliothek genutzt, welche einen Zugriff auf das Bussystem mit Hilfe von simplen Befehlen erlaubt. Mehr dazu im Kapitel 4.8.3 (S. 54).

Es sind insgesamt vier Adressen vorhanden. Der Unterschied zwischen dem Kühlwassersteuergerät und der Lüftersteuergeräte kann durch die Adressen direkt erkannt und das Bitmuster analysiert werden.

Das empfangene Bitmuster kann durch ein kleines selbstgeschriebenes Programm in regelmäßigen Abständen auf der Arduino Konsole auf dem PC Monitor ausgegeben und somit der Einfluss von Sensoren und dem Anschluss des Lüfters auf dieses Muster erkannt werden. Da diese Bitmuster nicht wie ein normaler Text gelesen werden können, befindet sich im folgenden Abschnitt 3.2.3 ein Beispiel für das Dekodieren des Temperatursignals. Die empfangenen Signale können so alle entschlüsselt und das Senden von Signalen an das Steuergerät behandelt werden. Durch zeitintensives ausprobieren von verschiedenen langen Byte-Anordnungen, konnte zunächst keine Verbindung zum Steuergerät aufgebaut werden. Nach erneutem recherchieren in der Bedienungsanleitung einer anderen Version des Geräts wurde ein Hinweis über den Aufbau der gesendeten Nachrichten gefunden. Mit diesem Hinweis kann, ohne Hinzunahme eines Oszilloskops, ein passendes Bytemuster durch gezieltes Ausprobieren erstellt und vermerkt werden.

Letztendlich wurden so alle benötigten Signale der Steuergeräte dekodiert und konnten durch das neue System ausgelesen und angesteuert werden.

Der letzte unbekanntes Softwarebestandteil sind die Adressen des SNMP. Diese können mit Hilfe eines SNMP-Browsers und der Eingabe der IP-Adresse schnell herausgefunden werden. Die vorhandenen SNMP Adressen, auch OID genannt, mehr dazu in Kapitel 4.6 (S. 47), sollen der Einfachheit halber übernommen werden. Werden diese Adressen übertragen, so kann das neue System ohne große Änderungen im Server eingebunden und ausgelesen werden.

3.2.3 Reverse Engineering des Temperatursignals

Das Entschlüsseln des übertragenen Temperatursignals ist ein sehr gutes Beispiel für Reverse Engineering. Es gab weder einen Anhaltspunkt, welche Daten zu dem Signal gehören, noch Angaben über den Typ des Temperatursensors.

Zunächst waren nur sehr wenige der übertragenen Daten, welche über eine Abfrage in der Software eingeholt wurden, klar. Die Übertragung erfolgt Byteweise, so werden vom Steuergerät des Kühlwassers und der Luftstromregelung 8 Byte übertragen.

Die folgende Tabelle stellt die Telegrammlänge einer älteren CMC Version dar. Es wurde davon ausgegangen, dass dies beibehalten wurde.

Tabelle 3.1: Darstellung der Übertragenen Daten mit dem Lüftermodul [44]

Byte	1	2	3	4	5	6	7	8
Name	“A”	Tmp1	Tmp2	Tmp3	Dz1	Dz2	Status	Chk

Das erste Byte ist eine Kennung, diese ist nicht weiter von Bedeutung. Die folgenden drei Bytes sollen für drei mögliche Temperaturfühler stehen, jedoch ist in diesem Fall nur Tmp1 verbaut. Die Drehzahlen Dz1 und Dz2 stellen gemessene Drehzahlimpulse der Lüfter dar. Status enthält in der Version des Handbuches keine Daten und Chk stellt die Checksumme der Bytes 1 bis 6 dar.

Es war nun klar, dass sich die drei Bytes nach der Kennung genauer angesehen werden sollten.

Zum Entschlüsseln der jeweiligen Bytes wurde zunächst der Temperatursensor für die eingehende Temperatur entfernt. Es ergab sich eine Änderung im Bytemuster an zwei Bytes, statt nur einem. Im Zuge dessen wurde dieser Sensor wieder angeschlossen und der für die ausgehende Temperatur entfernt.

Daraus wurde ersichtlich, dass sich der Wert eines anderen Bytes komplett geändert hat und bei einem zweiten Byte nur ein wenig. Diese Erkenntnis führte dazu das Bitmuster des einen Bytes genauer zu untersuchen.

Durch gezieltes anschließen und entfernen der Sensoren, sowie dem Erwärmen dieser konnte die Funktion dieses Bytes erkannt werden. Es werden die oberen 4 Bits für den

Sensor der ausgehenden und die unteren 4 Bits für den Sensor der eingehenden Temperatur genutzt.

Es galt nun herauszufinden, auf welche Art und Weise die Hälfte des einen mit dem jeweils ganzen anderen Byte zusammen verwendet werden. Dazu wurde der Temperatursensor zusammen mit einem digitalen Thermometer auf verschiedene Temperaturen gebracht und die zugehörigen Werte notiert. Es wurde dadurch klar, dass sich das jeweils andere Byte bei geringeren Temperaturänderungen verändert und der dazugehörige geteilte Wert bei höheren Temperaturen.

Das Verknüpfen des halben und des gesamten Bytes stellte sich als erfolgreich heraus und wurde wie folgt umgesetzt. Das geteilte Byte wurde als "in1" definiert:

```
byte in1 = airValuesDevice[3];
in1 = in1 & (15); // 15 = 00001111
d1 = (in1 << 8) + airValuesDevice[2];
```

Zunächst wird der empfangene Wert als Variable gespeichert. Die Analyse der Daten ergab, dass die rechten 4 Bits des Bytes für die eingehende Temperatur verantwortlich ist. Dementsprechend müssen die anderen Werte null gesetzt werden. Dies erfolgt durch das Verunden mit der Zahl 15.

Im nächsten Schritt werden die vier oberen Bits um acht Stellen nach links verschoben und die nun fehlenden acht Bits aus dem dazugehörenden anderen Byte addiert.

Dieser Versuch erbrachte eine Zahl, die bei Temperaturänderung ein lineares Verhalten aufwies.

Durch das Notieren der Zahlenwerte zu der dazugehörigen Temperatur konnte, über mehrere herausgesuchte Punkte, eine Geradengleichung aufgestellt werden. Durch Umstellen kann dies zu einer Temperatur in Grad Celsius führen.

Dies ist ein Beispiel für die Herangehensweise unbekannte Werte so umzuwandeln, dass die zugehörigen Sensoren und Steuergeräte auch mit dem neuen Steuergerät kommunizieren können.

3.2.4 Erworbene Kenntnisse

Das Ziel des Reverse Engineerings des Rittalproduktes war nicht eine exakte Kopie zu erzeugen, es galt nur der Fehleranalyse und der Analyse der Datenübertragung mit den anderen Steuergeräten.

Die Software, deren Ablauf und die Fehlerbehandlung sind eigenständig erarbeitete Leistungen. Eine hundertprozentige Portierung wäre in keinem Fall möglich gewesen, da es sich um eine komplett verschiedene Prozessorarchitektur handelt.

Genauso eigenständig wurde auch die Hardware ausgewählt und das Layout der Leiterplatte erzeugt, nur die Positionen der hinteren Anschlüsse im Gehäuse wurden übernommen, damit der Neuaufbau in dem vorhandenen Gehäuse durchgeführt werden kann.

Das Reverse Engineering ergab viele Informationen zum Nutzen der bereits verlegten Sensoren und der anderen Steuergeräte. Durch das Auslesen der Daten anderer CMC-Geräte wurden außerdem die Parameter der PID-Regelung bekannt.

Es ergaben sich unter anderem folgende Erkenntnisse:

- Taktrate und Funktion des Bussystems zur Kommunikation mit den anderen Steuergeräten
- Datenstruktur und Adressen der Steuergeräte
- Ansteuerung der Lüfter und Auswertung der internen Fehlercodes
- Werte der PID-Regelung
- Umwandlung der Werte der Temperatursensoren
- Allgemeines Verhalten in bestimmten Fehlerfällen

Dank dieser Informationen konnte die Entwicklung eines neuen Steuergerätes durchgeführt werden.

3.3 Wahl der Umsetzung

Dieser Abschnitt behandelt den Prozess zur Auswahl der benötigten Bauteile, je nach Art und Weise des Aufbaus des Systems. Zunächst wird analysiert auf Welche Art und

Weise das Projekt umgesetzt wird. Danach wird auf die Wahl der einzelnen Bauteile eingegangen.

3.3.1 Wahl der Plattform

Unter Berücksichtigung der Rahmenbedingungen und der erworbenen Kenntnisse durch das Reverse Engineering, werden im Folgenden mehrere Möglichkeiten der Projektumsetzung durchleuchtet und deren Vor- und Nachteile analysiert.

3.3.2 Einplatinencomputer

Es gibt eine große Vielfalt an Einplatinencomputern. Berühmte Hersteller sind unter anderem die Raspberry Pi Foundation, Lemaker.org mit dem Banana Pi oder auch das PandaBoard einer freien Entwickler Community. Im großen und ganzen gehört der Hersteller Arduino mit Systemen wie dem Arduino Uno oder Nano auch hierzu, jedoch gibt es den Unterschied, dass diese Entwicklerboards kein eigenes Betriebssystem benötigen, sondern die benötigten Hardwareinformationen und Pinbelegungen in einem Bootloader untergebracht sind. Deshalb werden sie unter dem nachfolgenden Punkt 3.3.3 separat behandelt.

Die oben genannten Einplatinencomputer besitzen eine große Auswahl an freiprogrammierbaren Ein- und Ausgängen, sowie integrierten Analog-Digital-Wandlern. Außerdem sind Netzwerkschnittstellen, USB und Videoausgänge auf der Platine vorhanden, welche auch in kleinen Außenabmessungen verfügbar sind.

Es werden viele Softwarepakete für diese Systeme angeboten, welche die Forderungen einfach umsetzbar machen. So kann zum Beispiel über einen Befehl eine Software für die Umsetzung des SNMP installiert und konfiguriert, oder über die Systemeinstellungen die Netzwerkadresse angepasst werden.

Die Vorteile scheinen den Nachteilen zu überwiegen, denn es ist alles benötigte auf kleinem Raum vorhanden. Jedoch bieten diese Vorteile viele Nachteile, denn durch die benötigte Netzwerkanbindung müssen die Systeme regelmäßig aktualisiert werden. Dies birgt die Gefahr eines Systemausfalls.

Die Komplexität dieser Systeme bietet in diesem Anwendungsfall einen anderen Nachteil. Sollte ionisierende Strahlung den internen Speicher in Form der SD- oder Micro SD-Karte beschädigen, so reicht es, wenn ein Bit des Betriebssystems verändert wird und somit

das System zum Ausfall gebracht wird. Dieser Fall kann bei allen möglichen Systemen auftreten, jedoch sind in diesem Fall mehrere hundert Megabyte bis zu einem Gigabyte an Daten gespeichert, welche nur das Betriebssystem darstellen.

Schlussendlich lässt sich sagen, dass diese Form von System zu komplex für den Anwendungsfall ist und nur ein Bruchteil von dem genutzt wird, was tatsächlich zur Verfügung steht.

3.3.3 Mikrocontroller-Board

Ein Mikrocontroller-Board besitzt den großen Unterschied, dass das System im größten Teil als integrierte Schaltung in einem Bauteil vorhanden ist. So ist zum Beispiel der Arbeitsspeicher integriert, statt als extra Bauteil auf der Platine zu finden.

Natürlich kommt ein Mikrocontroller nicht ohne zusätzliche Schaltungen aus, so finden sich auf einem solchen Board eine Einheit zur Spannungsumwandlung, bzw. Spannungstabilisierung oder Bauteile, welche die Kommunikation im Netzwerk oder zu einem PC ermöglichen. Die Ein- und Ausgänge werden mit Buchsen- oder Stiftleisten realisiert, damit jeder Pin des Mikrocontrollers beschaltet werden kann. Dies lässt die Abmessungen der Boards schnell größer werden, welches durch zusätzliche Bauteile, wie einem Netzwerkmodul oder einem Anschluss für I²C nicht den Anforderungen eines kleinen Gehäuses entsprechen.

Vorteile gegenüber eines Einplatinencomputers sind in diesem Fall die fehlende Komplexität, da kein eigenes Betriebssystem installiert werden muss. Unter kleinerem Speicheraufwand übernimmt dies ein installierter Bootloader, dessen Funktion in Kapitel 4.4.2 (S. 43) erklärt wird. Es wird außerdem ein festes Programm mit moderatem Speicheraufwand in einer Schleife ausgeführt.

Die Prozessorarchitektur, siehe Kapitel 4.4.1 (S. 42), ist in diesem Fall eine 8-Bit Architektur und ist somit deutlich weniger komplex als die der Einplatinencomputer. Dies ist unter Berücksichtigung der geringen benötigten Rechenleistung und der Umstände des Standortes ein großer Vorteil.

Da das System redundant aufgebaut wird, kann nur unter großem Platzaufwand ein zweites Board im Gehäuse platziert werden. Dies ist der Grund warum eine Eigenentwicklung die bessere Alternative sein kann.

3.3.4 Eigenentwicklung

Eine weitere Möglichkeit zur Umsetzung stellt eine komplette Eigenentwicklung dar. Diese Eigenentwicklung kann auf zwei Arten umgesetzt werden. Beide Entwicklungen haben gemeinsam, dass ein Leiterplatten-Layout erstellt werden muss, damit dieses Layout auf einen Platinen-Rohling geätzt oder gedruckt werden kann. Mehr dazu im Kapitel 4.7 (S. 48).

Die erste Möglichkeit kann durch ein Leiterplattendesign umgesetzt werden, auf dem die beiden benötigten Mikrocontroller-Boards über die Platine mit den Ein- und Ausgängen und der restlichen externen Hardware verbunden werden. Da die Mikrocontroller-Boards jedoch schwer in dem vorhandenen Gehäuse unterzubringen sind, wird die zweite Variante genutzt. Grund dafür ist, dass neben den beiden Boards zusätzlich ein Netzteil zur Spannungsumwandlung von 24 V auf 5 V, Buchsen zum Anschluss der anderen Steuergeräte, 24 V Spannungsversorgung, drei Relais mit Anschlüssen und ein Netzwerkmodul Platz finden müssen. Durch diesen Platzmangel wird die zweite Variante durchgeführt.

Die Platzeinsparung auf der Leiterplatte wird durch das Minimieren der Schaltung um den eigentlichen Mikroprozessor erreicht. So können nicht genutzte Ein- und Ausgänge von diesem stillgelegt werden und die genutzten direkt mit den Bauteilen verbunden werden. Diese Platzeinsparung ist möglich, da der Mikroprozessor, neben der Grundschaltung, den geringsten Platz des Boards einnimmt. Der meiste Platz wird durch das Bereitstellen aller verfügbaren Ein- und Ausgänge eingenommen. So ist beispielsweise auf einem Arduino Uno Board auf der einen Seite eine Buchsenleiste mit 14 Anschlüssen und auf der anderen eine Leiste mit 18 Anschlüssen.

Die auf der nächsten Seite folgende Abbildung zeigt einen Arduino Uno in der Draufsicht.

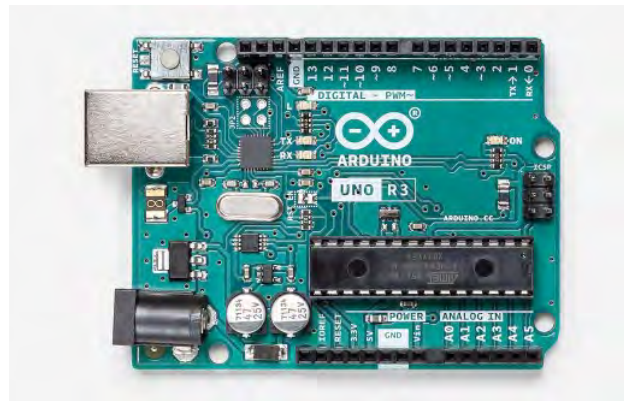


Abbildung 3.1: Abbildung eines Arduino Uno mit allen Anschlüssen [2]

Auf der Abbildung 3.1 ist zu sehen, dass der eigentliche Mikroprozessor, das schwarze Bauteil unter der Beschreibung Uno R3, sehr wenig Platz einnimmt. Zur Verdeutlichung der Größenverhältnisse: Der Stromanschluss nimmt etwa den halben Platz des Mikroprozessors ein. Der meiste Platz ist freie Fläche oder mit vereinzelter Zusatzbeschaltung versehen.

Zusätzlich sind auf jedem Board Bauteile zur Spannungsversorgung und ein USB Anschluss, welcher zur Seriellen Ausgabe und zur Kommunikation mit dem PC dient. Die integrierte Schnittstelle zur Kommunikation mit dem PC ist zwar recht hilfreich, jedoch wird diese im fertigen Produkt nicht benötigt. Das Prototypenboard sollte jedoch eine solche Schnittstelle besitzen.

Die Wahl einer Eigenentwicklung führt dazu, dass neben der auf dem fertigen Mikrocontroller-Board genutzten Prozessoren, auch andere Prozessoren genutzt werden können. Diese können je nach Einsatzzweck variieren. Welche Kriterien für die Wahl des Prozessors gelten, wird im Abschnitt 3.4.2 (S. 28) behandelt.

Neben der freien Prozessorwahl hat diese Vorgehensweise den Vorteil der freien Positionierung der Bauteile auf der Platine. Es können außerdem Testpunkte zur Überprüfung von Signalen und Spannungen platziert werden. Der einzige Nachteil ist das Erstellen eines Layouts für die Platine, statt einer Verdrahtung nach Schaltplan. Dieses wird jedoch gerne in Kauf genommen.

Alles in allem ist eine Umsetzung auf Leiterplatte deutlich robuster und professioneller als eine fliegende Verdrahtung mit zwei Mikrocontroller-Boards in einem Gehäuse und wird deshalb für dieses Projekt durchgeführt.

3.4 Hardwareauswahl

3.4.1 Allgemeines zur Bauteilwahl

Die Wahl der Bauteile fiel durch die äußeren Gegebenheiten und der dadurch resultierenden Einfachheit der Hardware auf wechselbare Bauteile. Alle wichtigen Bauteile, die durch Strahlung gestört werden könnten, sind auswechselbar und in Sockeln verbaut. Eine Abschirmung gegenüber ionisierender Strahlung wäre machbar, jedoch durch ein Verbauen von einem extra hergestellten Abschirmungsblech für jedes Gehäuse zu aufwändig. Ein schneller Wechsel von Bauteilen ist im Fall eines Ausfalls effizienter. Die einfache Reparatur von Geräten sollte immer gegeben sein und sorgt für ein Nachhaltiges Produkt.

3.4.2 Allgemeines zur Prozessorwahl

Dieser Abschnitt befasst sich mit der Wahl des passenden Prozessors. Es müssen die äußeren Umstände berücksichtigt werden und es muss auf die Geschwindigkeit und die Speicherkapazität geachtet werden.

Da bereits viele Projekte mit einem Arduino umgesetzt werden, kann der Prozessor dieses Mikrocontroller-Boards in Frage kommen, jedoch muss vorher ein Vergleich mit anderen Prozessoren durchgeführt werden.

Vergleich mit strahlenfesten Prozessoren

Mit einem Arduino Uno können viele Projekte realisiert werden. Den Anwendungsmöglichkeiten sind keine Grenzen gesetzt und es kann vieles gesteuert, geregelt, automatisiert oder vernetzt werden. Beim Arduino Uno übernimmt dies der Prozessor ATmega328p von Microchip AVR. Dieser Hersteller bietet eine Vielfalt von Mikroprozessoren in verschiedenen Architekturen an. Die 8-Bit Prozessoren dieses Herstellers sind alle, wenn auch bei manchen über programmiertechnische Umwege, mit den Arduino-Bibliotheken kompatibel.

Da der Prozessor mitunter ionisierender Strahlung ausgesetzt ist, weil sich das Strahlrohr oder eventuell ein Dump in der Nähe befinden, fiel das erste Augenmerk auf strahlungsfeste 8-Bit Prozessoren dieses Herstellers.

Der so genannte ATmegaS128 basiert auf dem originalen ATmega128 und wurde mit einer Energiedosis von 300 Gray getestet. Dies entspricht dem MIL-STD-883, welcher ein Teststandard für Mikroelektroniken im militärischen und Luftfahrt Bereich darstellt [Vgl. 20]. Die Einheit Gray besitzt die gleiche Definition wie die Maßeinheit Sievert.

Eine Energiedosis von 300 Gray entspricht etwa der halben aufgenommenen Dosis der Technik der Ulysses Sonde, die von 1990 bis 2009 um die Sonne kreiste. Die Elektronik der Juno Sonde, welche durch Umrundung des Jupiters Aufnahmen von diesem machte, nahm unter einer 1 cm Titanabschirmung etwa 250 Gray auf. Zum Vergleich: die Sonde Juno nahm an der Oberfläche eine Energiedosis von 1 Megagray auf [Vgl. 31].

Diese Daten veranschaulichen die Effizienz einer Abschirmung.

Da es sich bei dem Standort nicht um den Weltraum handelt und die Warnsysteme im Tunnel bereits bei 4 mGy innerhalb von zwei Stunden Alarm schlagen, ist dieser Prozessor stark überdimensioniert [Vgl. 24].

Laut Messwerten der DESY-internen Strahlenschutzgruppe, herrscht in der Nähe der Strahlstopper, durch Aktivierung dieser, eine Dosisleistung von 0,2 mSV/h, also einer Leistung von 0,0002 Gray pro Stunde. Dies gilt jedoch nur für die Zeit außerhalb des Strahlbetriebes.

Aktuelle Technologie besitzt eine Strahlungsresistenz bis zu einer Dosis von 50 bis 100 Gray. Somit kann sich ein Satellit auch mit normaler Hardware mehrere Jahre im Erdorbit befinden. Die Strahlungsfeste Hardware wird somit nur für ausfallsichere Module benötigt. Durch erhöhte Sonnenaktivität kann die Dosisleistung kurzzeitig ansteigen und könnte für den Ausfall von normaler Hardware im Orbit sorgen. Aus diesem Grund wird für wichtige Anwendungen strahlungsfeste Hardware genutzt [Vgl. 31].

Durch die Erkenntnis, dass auch normale technische Bauteile eine ausreichende Strahlenfestigkeit besitzen, fällt die Wahl auf einen handelsüblichen Prozessor aus dem Hause Microchip AVR.

Im Vergleich zum Vorgänger-Steuergerät werden jedoch zwei Prozessoren genutzt, um eine Ausfallsicherheit zu garantieren. Außerdem ist der Prozessor durch die Prozessorarchitektur mit 8 Bit deutlich einfacher aufgebaut und dementsprechend strahlungsfester als der Vorgänger.

Prozessorwahl

Die Prozessorwahl fällt aus persönlicher Erfahrung auf den ATmega328p, da bereits viel mit diesem gearbeitet wurde. Leider kann der Prozessor nicht für beide Anwendungen genutzt werden. Für die Kommunikation mit den Steuergeräten und mit dem anderen Prozessor reicht dieser aus, jedoch reicht der Speicher nicht für das Ansteuern der Steuergeräte zusätzlich zur Kommunikation über Ethernet. Hinzu kommt ein großer Speicherplatzbedarf für das SNMP.

Aus diesem Grund wurde die Produktpalette des Herstellers durchsucht und der passende Primärprozessor entdeckt. Es handelt sich um den ATmega1284p, welcher genug Speicher für alle Anwendungen besitzt. Dieser Prozessor ist, wie auch der 328p im DIP-Gehäuseformat erhältlich und lässt sich somit problemlos austauschen.

Der ATmega328p besitzt 32 kB Programmspeicher und 2 kB RAM, der ATmega1284p besitzt 128 kB Programmspeicher und 16 kB RAM. Beide verursachen einen geringen Stromverbrauch und werden bei 5 V betrieben. Die Anzahl der Ein- und Ausgänge, sowie die Schnittstellen zum Kommunizieren mit anderer Hardware sind ausreichend vorhanden.

Bei beiden handelt es sich um 8-Bit Mikrocontroller.

3.4.3 Bauteile des Netzteils

Das Netzteil muss ausfallsicher sein. Je weniger Bauteile verwendet werden, desto weniger fehleranfällig ist dieses. Die Anforderung ergab, dass aus 24 V eine 5 V und eine 3,3 V Spannung erzeugt werden muss.

Die Mikroprozessoren, Relais und Speicherbausteine werden mit 5 V und der Netzwerkdapter mit 3,3 V versorgt.

Nach einer Recherche über die Robustheit einiger Schaltspannungsregler fiel die Wahl auf einen LM2594, der die 24 V in 5 V umwandelt. Ein LD1117 wandelt die 5 V in 3,3 V um.

Die Minimalbeschaltung dieser Bauteile ist gering und sie laufen seit Jahren in anderen Projekten und Geräten. Die Wahl fiel auf diese Bauteile, da sie in einem Fernsehgerät von 2008 verbaut wurden und dort, trotz langer Laufzeit, noch kein einziges mal ausgefallen sind.

Nachdem die Wahl auf die genannten Bauteile gefallen ist, wurden bei erneuter, genauerer Untersuchung der Steuergeräte von Rittal die gleichen Bauteile für die Spannungsumwandlung entdeckt. Dies bestätigt die Zuverlässigkeit der Spannungswandler.

Zu den Spannungswandlern kommen diverse Dioden, Varistoren und Sicherungen zum Schutz gegen Überspannung zum Einsatz.

Es werden Dioden zum Schutz vor Verpolung eingesetzt, damit in einem Fehlerfall, beim Überprüfen der Module in der Werkstatt, keine Schäden entstehen können.

Ein Varistor in der 24 V-Versorgung schützt die Schaltung vor hoher Überspannung und eine Zenerdiode sorgt für eine stabile Spannung. Die 24 V-Versorgung ist mit einer trägen 2 A Sicherung abgesichert. Sie schützt die Leiterbahnen vor dem Durchbrennen und wurde mit Reserven für diese ausgelegt.

Die 5 V-Versorgung ist ebenfalls über eine vorab ausgelegte träge 1 A Sicherung abgesichert und mittels Zenerdiode wird die Spannung stabil gehalten. Die 3,3 V-Spannungsversorgung wird nur über eine träge 200 mA Sicherung geschützt.

Trotz Schutzbeschaltung nimmt der Netzteiltrakt des neuen Steuergerätes wenig Platz ein und die anderen Bauteile können nach belieben positioniert werden.

3.4.4 Speicherbaustein

Da der gewählte Mikroprozessor nicht genug Speicherplatz für eine große Anzahl an Logs bietet, muss eine Möglichkeit gefunden werden, die aktuellen Log Einträge zu speichern. Für diesen Fall gibt es so genannte EEPROMs, deren Funktion in 4.3.1 (S. 41) erklärt wird.

Ein über I²C angesteuerter Speicher ist prädestiniert für dieses Projekt, da viel Information auf einfache Art und Weise gespeichert werden kann. Die Wahl der Speichergröße fiel auf ein Bauteil mit 1024 kB großem Speicher. Dies bietet den Vorteil, dass über mehrere Jahre Daten gespeichert werden können, bevor der Baustein erneut beschrieben wird. Durch dieses seltene Überschreiben besitzt dieser Baustein eine sehr hohe Lebensdauer. Durch ein Gehäuse im DIP-8 Format kann der EEPROM bei einem möglichen Totalausfall über ein externes Programm ausgelesen und ausgewertet werden.

3.4.5 Relais und Ansteuerung

Dem Relais galt nur wenig Aufmerksamkeit. Da dieses, laut Auftraggeber, nur selten betätigt werden soll, Somit fiel die Wahl auf ein klassisches Relais, welches per Magnetspule und Anker die Kontakte schaltet.

Ein Halbleiterrelais, welches durch kontaktloses Schalten nahezu verschleißfrei ist, wird somit nicht benötigt. Die Wahl fiel auf ein Omron G5V Print-Relais, welches günstig in der Anschaffung ist, kleine Abmessungen besitzt und die geforderte Spannung und den Strom liefern kann.

Durch die Stromaufnahme von 30 mA des Relais und einem maximalen Ausgangsstrom von 20 mA des Mikroprozessors, muss ein geeigneter Treiber verwendet werden. Dieser Treiber wird durch eine Darlington-Schaltung mit Transistoren realisiert. Diese ermöglicht es die Relais mit dem Mikroprozessor anzuschließen.

Darlington-Schaltungen gibt es als Arrays auf einem integrierten Schaltkreis, in diesem Fall als ULN2003A mit 16 Pins und 7 Kanälen.

Durch die Kombination aus Relais und Darlington Array kann so die Anforderung von extern ansteuerbaren Bauteilen erfüllt werden.

3.5 Aufbau des Systems

Dieses Kapitel befasst sich mit einer kurzen Beschreibung des Aufbaus des Systems. Dieses kann in wenige Abschnitte unterteilt und somit besser beschrieben werden. Im folgenden werden die Abschnitte Spannungsumwandlung im Netzteil, Ein- und Ausgänge, Datenverarbeitung über die Prozessoren und Peripherie erläutert.

3.5.1 Netzteil

Das Netzteil dient zur Umwandlung der bereits verfügbaren 24 V, welche über das Netzteil von Rittal bereitgestellt werden. Die Umwandlung dieser Spannung wird bereits in dem alten Steuergerät durchgeführt und soll auch in dem neuen stattfinden. So wird die Herstellung oder der Kauf eines neuen externen Netzteils vermieden, welches Kosten spart und nachhaltig ist.

Der Aufbau des Netzteils ist so klein wie möglich. Die Anordnung der Bauteile ergibt

sich aus dem Schaltplan und die Bauteile folgen der festgelegten Umwandlung der Spannungen. So kann sich der LM2594 nur hinter der Schutzbeschaltung der 24 V-Versorgung befinden, da er aus dieser Spannung die geforderten 5 V umwandelt.

Die Beschaltung der Spannungswandler befindet sich so nah wie möglich an den Wandlern. Das Netzteil befindet sich direkt hinter dem Anschluss zur Spannungsversorgung.

3.5.2 Ein- und Ausgänge

Die Position der Ein- und Ausgänge ist Gehäusebedingt vorgegeben. So ist auf der geschlossenen Seite ein serieller Anschluss in Form einer RJ-11 Buchse, ein Sichtfenster für die Status-LED, ein weiteres für die Aktivität der seriellen Schnittstelle und eine flexible Stelle für das Drücken des Reset-Tasters. Die Abmessungen des alten Steuergerätes werden aufgenommen und bei der Fertigung des neuen Steuergerätes berücksichtigt.

Anders ist es jedoch an der Frontseite. Dort befindet sich eine schraubbare Frontplatte, die modifiziert werden kann. Die Platzverhältnisse sind zwar vorgegeben, es ist jedoch genug Platz zum beliebigen Unterbringen der geschalteten Relais-Ausgänge, der I²C Schnittstelle, dem Ethernet Anschluss, sowie dem Netzanschluss zur Spannungsversorgung.

Wie bereits beschrieben, werden jeweils eine RJ-45 Buchse für die Ethernet- und die I²C-Schnittstelle benötigt. Es muss genau gekennzeichnet werden, um welche Schnittstelle es sich handelt, da es bei Verwechslung zur Zerstörung des Netzwerkgerätes kommen kann. Der Grund dafür ist die Spannung von 24 V und die zugehörige Masse, die an zwei Pins des I²C Anschlusses angeschlossen sind.

Die RJ-11 Buchse bleibt bestehen, sie wird zur Anzeige der seriellen Meldungen, sowie zur Konfiguration des primären Prozessors genutzt. Durch diese wird ein Ausbau des Gerätes und ein Anschließen eines Programmieradapters vermieden.

Wichtig ist der Netzanschluss, welcher über einen dreipoligen Anschluss der Firma Kycon, namens KPJX-3S-S, geschieht. Dieser wurde vom Steuergerät von Rittal genutzt und wird somit auch für das neue Gerät verwendet.

Die Wahl des Anschlusses der drei Relais-Ausgänge fiel auf einen häufig verwendeten Anschluss von Molex. Der Anschluss ist in Buchsenform und für die Montage auf einer PCB geeignet. Diese Buchsen sind platzsparend und für den Einsatzzweck geeignet.

Damit keine Verwechslungen stattfinden wird der Anschluss des nachträglich einzubauenden Durchflusssensor, welcher bei vorhandenem Fluss des Kühlwassers das Signal ändert, über einen dreipoligen Anschluss von Molex realisiert. So kann kein Stecker vertauscht werden und die Funktion ist gewährleistet.

3.5.3 Prozessoren

Die Prozessoren werden so Platziert, dass die Länge der Leiterbahnen so kurz wie möglich ist. Eine parallele Platzierung ist in diesem Fall nicht umsetzbar, da dies durch die Platzverhältnisse nicht möglich ist.

Eine optimale Positionierung ist eine L-Form. Auf diese Weise können die Leitungen kurz gehalten und die Datenleitungen zum Netzwerkadapter getrennt von den anderen Datenleitungen verlegt werden.

Die Prozessoren werden sich in der Mitte der Leiterplatte befinden.

3.5.4 Peripherie

Mit Peripherie sind in diesem Fall Module gemeint, welche abseits des Mikroprozessors genutzt werden. Diese sind kein Teil des Systems und werden über I²C oder einen anderen Bus angesteuert. Die Auswahl ist bei solchen Modulen fast unbegrenzt, jedoch werden in diesem System zwei dieser Module genutzt.

Eines dieser Module ist das Netzwerkmodul. Dieses ist ein eigenständiger integrierter Schaltkreis, welcher mit dem Mikroprozessor über die Pins MOSI und MISO, also den Anschlüssen für das Serial Peripheral Interface, auch SPI, kommunizieren. Dies ist ein eigenständiges, synchron serielles Bussystem, welches seit 1987 verwendet wird [Vgl. 48]. Das Netzwerkmodul basiert in diesem Fall auf dem W5500 der Firma WIZnet, ist günstig in der Anschaffung und kann sehr leicht in das System eingebunden werden.

Das zweite externe Modul ist ein EEPROM, welcher über I²C angesteuert wird. Auch dieser ist ein eigenständiger integrierter Schaltkreis und kann einfach in das System eingebunden werden.

Bei den Relais handelt es sich auch um eine Art Peripherie, diese sind jedoch nicht

über einen Bus ansteuerbar, sondern dienen in diesem Fall nur als Schalter, die bei Ansteuerung durch das Darlington-Transistor-Array 24 V durchschalten.

Die Position der Relais wird sich so nah wie möglich bei den Anschlüssen der Frontplatte befinden. Das Transistor-Array wird an einen passenden Platz zwischen den Prozessoren und den Relais gesetzt.

Das Netzwerkmodul wird so nah es geht an dem primären Prozessor und nah am Rand zur Frontplatte positioniert. So sind die Wege der Verdrahtung kurz und das Netzkabel kann verbunden werden.

Die Position des EEPROMs wird so nah wie möglich an beiden Prozessoren sein. So werden die Leitungswege kurz gehalten und die Datenübertragung ist weniger stör anfällig.

4 Theoretische Hintergründe

4.1 Bauteile eines Linearbeschleunigers

4.1.1 Allgemeines über Linearbeschleuniger

Die Beschleunigung der Elektronen wird in einem Linearbeschleuniger umgesetzt. Es wird ein Linearbeschleuniger genutzt, da bei diesem die Bremsstrahlung eines Ringbeschleunigers entfällt, mehr dazu im Kapitel 4.2 (S. 40). Diese Bremsstrahlung, auch Synchrotronstrahlung wird jedoch später gezielt in Undulatoren, Kapitel 4.1.3 (S. 38), erzeugt. Auf dem folgenden Bild ist eine Beschleuniger-Einheit abgebildet. In dieser gelben Röhre werden die Elektronenpakete streckenweise beschleunigt.



Abbildung 4.1: Abbildung eines Beschleunigermoduls im XFEL [38]

In diesen Beschleunigermodulen befinden sich Hohlraumresonatoren, welche bei Temperaturen nahe dem absoluten Nullpunkt supraleitend werden, mehr dazu im nächsten Kapitel. Um diese Temperaturen zu erreichen wird ein Teil dieser Röhren mit flüssigem

Helium gekühlt. In einem anderen Teil des Beschleunigermoduls, dem Strahlrohr, muss ein Ultrahochvakuum vorhanden sein. In dem Strahlrohr befinden sich die Elektronenpakete, die beschleunigt werden.

In den Modulen befinden sich außerdem Magnete zum Fokussieren und korrigieren der Flugbahn.

Die Module wurden von vielen Instituten aus aller Welt gebaut, getestet und verbaut.

Die Magnetstrukturen zu Korrektur befinden sich jedoch nicht nur in den Beschleunigermodulen, sondern sind auch als externe Module verbaut. Vor allem vor den Beschleunigermodulen werden sie zum Erstellen eines gezielt geformten Elektronenpakets und zur Richtungskorrektur dieser genutzt.

4.1.2 Hohlraumresonator

Der Hohlraumresonator, auch Kavität oder Cavity, ist das Herzstück eines Beschleunigers. Dieser besitzt einen Hohlraum, welcher eine bestimmte berechnete Form besitzt, in dem hochfrequente elektromagnetische Wellen erzeugt werden. Mit der Form des Hohlraums und der Frequenz der Welle wird durch gezieltes Abstimmen eine hin und her reflektierende Welle erzeugt, damit sich eine stehende Welle, meist in verschiedenen Moden, ausbildet. Es kommt zur Resonanz. Trifft nun ein Teilchen, in diesem Falle ein Elektron, mit der richtigen Geschwindigkeit auf diese Welle, so wird es beschleunigt. Da im falschen Zeitpunkt die Teilchen abgebremst werden, kann kein kontinuierlicher Strahl aus Elektronen beschleunigt werden, sondern nur Elektronenpakete.

In alten Beschleunigeranlagen musste diese Frequenz in den Resonatoren an die Geschwindigkeit der Teilchen angepasst werden. Auf diesem Synchronisieren basiert der Name "Synchrotron". [Vgl. 49]

Ein Hohlraumresonator besteht aus dem Material Niob, welches, wie auch Eisen, Silber und Gold, zu den Übergangsmetallen gehört. Die Eigenschaften bei Temperaturen nahe dem absoluten Nullpunkt machen dieses Metall so besonders, denn es wird zum Supraleiter, sein Widerstand strebt also gegen Null. Durch diese Eigenschaft geht bei der Übertragung von Energie keine Energie in Form von Wärme verloren. Ein Hohlraumresonator aus Niob ist somit sehr effizient. [Vgl. 51]

Die folgende Abbildung zeigt einen supraleitenden Hohlraumresonator, wie er im Teilchenbeschleuniger DESY verwendet wird.



Abbildung 4.2: Supraleitender Hohlraumresonator aus Niob [8]

An den beiden Enden befinden sich Anschlüsse zum Verbinden mit dem Strahlrohr. Die Kammern des Resonators sind auf dieser Abbildung zu erkennen. Dieser Resonator wurde aus hochreinem Niob gefertigt.

4.1.3 Undulator

Ein Undulator ist ein Gerät mit dem Synchrotronstrahlung erzeugt werden kann. In diesem Bauteil wird ein elektrisch geladenes Teilchen durch ein wechselndes Magnetfeld abgelenkt. Ist die Geschwindigkeit dieses Teilchens annähernd Lichtgeschwindigkeit, so

gibt es bei der Beschleunigung quer zur Flugrichtung relevante Mengen elektromagnetischer Strahlung in Form von Synchrotronstrahlung ab. Diese gehört zur ionisierenden Strahlung, Kapitel 4.2 (S. 40). Diese Strahlung wird auch Bremsstrahlung genannt.

Ein Undulator besteht aus vielen Dipolmagneten, die mit abwechselnder Ausrichtung zueinander in Reihe angeordnet sind. Es entsteht eine Sinusförmige Flugbahn, sobald sich ein Teilchen durch dieses Feld bewegt. Diese Undulatoren werden in so genannte Undulatorstrecken, bestehend aus vielen 5 m langen Segmenten, untergebracht.

Die Magnete können in Form von Elektromagneten, Permanentmagneten oder auch durch Hybride Magnete umgesetzt werden. Über dem Abstand zum Elektronenstrahl kann die Wellenlänge der emittierten Synchrotronstrahlung eingestellt werden. [Vgl. 17]

4.1.4 Strahlstopper (Dump)

Im gesamten XFEL Tunnel kommt es bei der Beschleunigung der Elektronen zu ionisierender Strahlung. Dies lässt sich nicht vermeiden, da diese gewollt erzeugt wird, oder z.B. bei der Korrektur der Flugbahn der Elektronenpakete entsteht.

In manchen Fällen, wenn zum Beispiel ein Notaus betätigt wird, müssen die im Beschleuniger befindlichen Elektronen schnellstmöglich aus dem Strahlrohr befördert werden. Die beschleunigten Elektronen werden in diesem Fall magnetisch umgelenkt und in einen massiven Block aus Metall und Beton, oder anderen Materialien geschossen. Bei der Abbremsung entsteht radioaktive Strahlung, welche den Dump aktiviert.

Je nach beschleunigter Materie und Energie der Teilchen ist ein solcher Strahlstopper beispielsweise durch einen 7 m langen, 70 cm x 70 cm breiten und hohen, mit Stahl umhüllten, Graphitblock umgesetzt. Dieser ist wassergekühlt. Solch ein 750 Tonnen schwerer Strahlstopper wird am LHC in Cern verwendet um die Strahlenergie abzuleiten und diese in Wärmeenergie umzuwandeln. [Vgl. 25]

Ein weiterer Einsatzzweck eines Strahlstoppers befindet sich nach der Abzweigung des Röntgenstrahls zu den Experimentierhütten. Ab dieser Stelle wird nur noch die Röntgenstrahlung gebraucht und die Elektronen werden in einen Dump abgeleitet. Dieser ist größer als die Dumps für den Notaus der Anlage, da er dauerhaft mit den nicht mehr verwendeten Elektronen beschossen wird.

Die durch das Abbremsen entstehende Strahlung ist problematisch für viele Elektroniken in der Nähe. So auch in dem Schaltschrank, in dem dieses Projekt getestet wird.

Dieser befindet sich direkt neben einem Strahlstopper. Andere Abteilungen des DESY sind durch die erhöhte Strahlung bereits aus diesem Schaltschrank ausgezogen, da die verbaute Elektronik Fehler aufwies.

Der Testlauf wird zeigen, ob die neue Elektronik dieser Strahlenbelastung standhält.

4.2 Ionisierende Strahlung

Ionisierende Strahlung ist, genau wie das sichtbare Licht, eine Energieübertragung in Form einer elektromagnetischen Welle. In diesem Fall erfolgt jedoch ein größerer Energietransport pro Photon als beim sichtbaren Licht oder Infrarotstrahlung.

Tritt diese Strahlungsart in Materie ein, so kann diese dadurch verändert werden, da die Energie der ionisierenden Strahlung chemische Verbindungen aufbrechen, oder Atome und Moleküle ionisieren kann. Unter Ionisierung versteht sich das Herausschlagen von Elektronen aus der Atomhülle, welches das zugehörige Atom oder Molekül elektrisch positiv geladen sein lässt. Besitzen Teilchen eine elektrische Ladung, so werden diese Ionen genannt.

Bei Ionisierung oder Veränderung der Moleküle von lebenden Zellen oder Organismen werden diese beschädigt.

In elektronischen Schaltungen kann durch den Einfluss der ionisierenden Strahlung eine Veränderung der übertragenen und gespeicherten Daten stattfinden. Die Werte einer Null, bzw., Low-Pegels, können somit durch die energiereiche Strahlung auf Eins, bzw. High-Pegel, gebracht werden. Dies sorgt beim Aufrufen von Programmteilen oder Werten für Fehler, die die Systeme zum Absturz bringen können.

Je feiner und näher beieinander die Elektronik ist, desto häufiger treten diese Fehler auf. Ionisierende Strahlung lässt sich durch Materialien wie Beton, Blei und andere dichte Metalle abschirmen.

Ionisierende Strahlung kann beim Zerfall von Atomkernen entstehen und auch technisch Erzeugt werden. Röntgenstrahlung gehört zu der Kategorie der ionisierenden Strahlung und ist Teil der Synchrotronstrahlung. [Vgl. 5]

4.3 Speicherbausteine

Es gibt viele Arten von Speichertechnologien. Diese reichen von Lochkarten, über magnetische Speicher bis zum 3D-NAND-Flash Speicher. In diesem Kapitel werden die

verwendeten Arten EEPROM und Flash-Speicher, welche vom Mikrocontroller und zur Speicherung von Log-Daten genutzt werden, genauer beschrieben. Der flüchtige RAM Speicher wird im Kapitel 4.4.3 (S. 44) beschrieben, da dieser bei diesem Projekt direkt zum Mikroprozessor gehört.

4.3.1 EEPROM

Ein EEPROM ist ein nicht-flüchtiger Speicher. Wenn die Stromversorgung unterbrochen wird, bleiben die beschriebenen Daten bestehen. Ein EEPROM besitzt eine begrenzte Anzahl an Programmierzyklen und zum Löschen ist, nicht wie beim EPROM, kein UV-Licht notwendig.

Durch die begrenzte Speicherzyklen-Anzahl, ist dieser Speicher geeignet um dauerhafte Konfigurationen, Log-Daten oder den Programmcode eines Mikroprozessors zu speichern. Zur kurzen Zwischenspeicherung sollte zum Beispiel RAM Speicher genutzt werden.

EEPROMS können, im Vergleich zum Flash-Speicher, byteweise beschrieben und gelöscht werden.

EEPROM kommt vom englischen Wort “Electrically Erasable Programmable Read-Only Memory”, welcher durch einen Halbleiterspeicher umgesetzt wird, der sich mit Spannungsimpulsen beschreiben oder löschen lässt. Die Speicherung ist im Vergleich zum Flash-Speicher langsamer. Interessant ist jedoch der Aufbau, welcher durch eine Matrix aus Feldeffekttransistoren und deren verbundene Steueranschlüsse, Floating-Gates, umgesetzt wird. Jeder Transistor kann somit eine Datenmenge von einem Bit speichern. Das isolierte Floating-Gate hält die Ladung im Transistor.

Das Löschen der gespeicherten Daten erfolgt durch das Anlegen eines größeren Spannungsimpulses am Transistor, das Auslesen hingegen wird durch eine niedrige angelegte Spannung an die Drain-Source-Anschlüsse durchgeführt.

Die Begrenzung der Schreibzyklen ist auf Leckeffekte am Floating-Gate zurückzuführen und sorgt für eine Begrenzung von mehreren tausend bis zu einer Million Zyklen. Diese Zyklenzahl ist in der Datenverarbeitung jedoch nicht immer ausreichend. [Vgl. 34]

4.3.2 Flash-Speicher

Eine andere Speicherart ist der sogenannte Flash-Speicher, oder Flash-EEPROM. Er ist ein nicht-flüchtiger Speicher und arbeitet schneller als der klassische EEPROM. Wird

der Flash-Speicher neu beschrieben, so muss zunächst ein ganzer Sektor an Daten gelöscht werden. Dies ist beim EEPROM nicht der Fall und liegt an der verwendeten Ansteuerungs-Elektronik. Das Beschreiben funktioniert jedoch gleich.

Die Speicherung im Flash-EEPROM ist, bis auf die Ansteuerung, identisch. Es wird mit Feldeffekttransistoren und der Verschaltung der Floating-Gates gearbeitet.

Die Ansteuerung wird in NAND- und NOR-Flash unterteilt. Beim NAND-Flash sind die Speicherzellen seriell miteinander verschaltet, beim NOR parallel.

Durch das Löschen eines ganzen Sektors ist die Anzahl der Schreibzyklen zum Teil noch weiter eingeschränkt und endet mit einer Schreib- und Löschyklen Anzahl von zehn- bis einhunderttausend Zyklen für NOR und bis zu einer Million Zyklen für NAND. Diese begrenzte Zyklenzahl führte beim amerikanischen Autohersteller Tesla zu Problemen mit dem Bildschirm und dem Multimediasystem, welche auf einen verschlissenen Speicher zurückzuführen sind [Vgl. 23].

Diese Speicherart wird zum Beispiel in USB-Sticks und Smartphones genutzt. [Vgl. 6]

4.4 Mikrocontroller

4.4.1 Prozessor

Prozessorarchitektur

Ein wichtiges Merkmal eines Prozessors ist die Prozessorarchitektur. Diese beschreibt den Aufbau der Prozessorkerne und wird in Architekturen wie 8-, 16-, 32- und 64-Bit unterteilt.

Die 8-Bit-Architektur zum Beispiel, wie sie auch im verwendeten Prozessor genutzt wird, beschreibt eine Verarbeitungsbreite von 8 Bit. Die Verarbeitungsbreite steht für das verarbeitbare Datenwort im Prozessor und wird auch Busbreite genannt. Dies gibt die Datenmenge an, welche in einem Schritt von der arithmetisch-logischen Einheit des Prozessors verarbeitet werden kann.

Die Tatsache, dass nur ein Byte pro Takt verarbeitet wird und die Hardware dahinter dementsprechend einfacher aufgebaut ist, sorgt für einen robusten und weniger Fehleranfälligen Prozessor, der für dieses Projekt prädestiniert ist.

Ein Raspberry Pi basiert auf der Arm-Architektur, welcher in älteren Systemen eine Wortbreite von 32 und heutzutage von 64 Bit besitzt. Diese besondere Architektur zeichnet sich durch einen effizienten Befehlssatz aus und ermöglicht eine besser Geschwindigkeit in der Ausführung von Befehlen und eine geringere Stromaufnahme.

32 oder auch 64 Bit Prozessoren sind mittlerweile in jedem leistungsstarken Gerät verbaut. Dies ist der Fall, da durch die großen Wortbreiten auch größere Adressbereiche zur Verfügung stehen. Es kann somit sehr viel Speicher adressiert werden, jedoch sind größere Architekturen für die Speicheradressierung zumindest in den nächsten Jahren nicht notwendig. [Vgl. 30]

4.4.2 Bootloader

Ein Bootloader, auch Startprogramm genannt, ist eine Software, die neben der Firmware zu den wichtigsten Programmen auf einem System zählt. Die Firmware gibt dem Prozessor, bzw. dem System, die benötigten Informationen, um was es sich bei dem System handelt und wo welche Register liegen. Das System ist dadurch in der Lage Dinge in den Arbeitsspeicher zu laden, Positionen der Ein- und Ausgangsregister zu kennen und durch diese Informationen überhaupt starten zu können.

Nachdem das System mit der Firmware gestartet wurde, kommt der so genannte Bootloader zum Einsatz. Dieser Bootloader befindet sich bei einem PC auf der Festplatte im Master Boot Record, kurz MBR. Sollte ein PC einmal nicht starten, so kann dies an einem Fehler im MBR zurückzuführen sein. Dieser Fehler kann durch Reparaturoptionen wiederhergestellt werden und der PC startet wieder.

Der Bootloader eines Mikroprozessors liegt meistens am Anfang des Flash-Speichers und wird von dort aus geladen. Die Aufgabe von diesem liegt darin, die Vermittlung zwischen Hard- und Software durchzuführen. Wird in der Software ein Ausgang angesteuert, so wird die Anfrage zu dem Register geleitet, welches vom Bootloader angegeben wurde. Der Bootloader gibt außerdem den Systemtakt, das Pinout und die Unterstützung einiger Bussysteme und Protokolle vor. [Vgl. 26]

4.4.3 Arbeitsspeicher

Im Vergleich zu den EEPROM Speichern handelt es sich bei dem Arbeitsspeicher um einen flüchtigen Speicher. Sobald er von der Spannung getrennt wird, verfallen die gespeicherten Daten.

Der Vorteil des Random Access Memory, kurz RAM, ist, dass er unbegrenzt beschreibbar und deutlich schneller ist. In diesem Speicher werden Variablen oder andere oft veränderte Daten oder Programmteile abgelegt.

Arbeitsspeicher ist meist direkt im Prozessor integriert und wird direkt über diesen adressiert, damit die Zugriffszeiten sehr gering bleiben. Dieser RAM wird in mehrere Arten unterteilt. Zwei davon sind DRAM, also dynamisches RAM und SRAM, statisches RAM.

Dynamisches RAM wird in Computern verwendet. Dieses wird durch winzige Kondensatoren realisiert, deren Ladung sich jedoch durch die auftretenden Leckströme entlädt und die gespeicherte Information gelöscht wird. Dies tritt trotz angelegter Versorgungsspannung auf. Aus diesem Grund müssen die Daten stets aufgefrischt werden.

Statisches RAM wird durch bistabile Kippstufen realisiert und ist im Vergleich zum DRAM durch mehr Bauelemente für größeres Speichervolumen zu teuer. Deshalb wird er mit kleinerem Speichervolumen als Arbeitsspeicher für Mikroprozessoren verwendet. [Vgl. 27]

4.4.4 Watchdog

Ein wichtiges, in einigen Mikroprozessoren integriertes, Feature ist der Watchdog. Übersetzt bedeutet dies Wachhund und dies ist auch seine Aufgabe. Der Watchdog enthält einen separaten Zähler, welcher in einem bestimmten Takt getaktet wird. Über einen Vorteiler kann eine erwünschte Zyklenzahl eingestellt werden und er beginnt von null an hoch zu zählen. Wird die eingestellte Zyklenzahl erreicht, so wird ein Reset für den Prozessor ausgelöst.

In der Hauptschleife des Programms wird der Watchdog regelmäßig zurückgesetzt. Hängt sich das Programm nun durch einen Fehler im Code, durch äußere Einflüsse oder durch das Warten auf ein anderes Bauteil durch abgebrochene Kommunikation auf, so startet der Prozessor nach Erreichen der gewünschten Zyklenzahl neu. Programmteile und Variablen werden erneut in den Arbeitsspeicher geladen und das Programm sollte wieder laufen. [Vgl. 35]

4.5 I²C Bussystem

Ein wichtiges Bussystem, welches Bestandteil des Projektes ist, ist das I²C. Es dient zur Kommunikation mit anderen Bauteilen und, in diesem Fall, mit den Prozessoren der anderen Steuergeräte.

I² steht für Inter-Integrated Circuit. Das Bussystem wurde 1982 von Philips Semiconductors entwickelt und es handelt sich um einen seriellen Datenbus mit zwei Datenleitungen. Aus lizenzrechtlichen Gründen wird häufig der Name Two-Wire-Interface, kurz TWI, genutzt.

Das Bussystem wurde als Master-Slave-Bus entwickelt, so wird eine Datenübertragung stets durch einen Master angefordert. Der angeschlossene Slave reagiert auf diese Anfrage. Es können jedoch auch mehrere Master und Master, welche zeitgleich als Slave programmiert sind, in den Bus eingebunden werden.

Eine der beiden Signalleitungen ist die Taktleitung, Serial Clock, kurz SCL, und bei der anderen handelt es sich um die Datenleitung, Serial Data, kurz SDA. Die Takte werden in verschiedene Modi eingeteilt. [Vgl. 21]

Die folgende Tabelle zeigt eine Übersicht der Betriebsmodi.

Tabelle 4.1: Maximal erlaubte Taktraten I²C Bus [21]

Modus	Maximale Übertragungsrate	Richtung
Standard Mode	100 kHz	bidirektional
Fast Mode	400 kHz	bidirektional
Fast Mode Plus	1 MHz	bidirektional
High Speed Mode	3,4 MHz	bidirektional
Ultra Fast Mode	5 MHz	unidirektional

Der Aufbau der Datenpakete ist festgelegt und wird auf der folgenden Abbildung veranschaulicht.

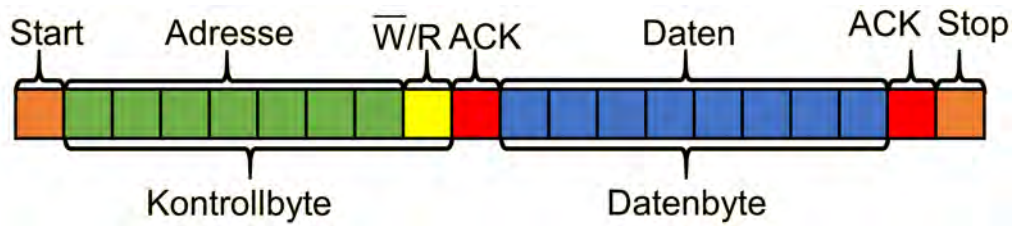


Abbildung 4.3: Aufbau eines I²C Datenpaketes [22]

Nach dem Startbit folgen die sieben Bits für die Adresse, gefolgt von einem R/\bar{W} -Bit. Daraufhin folgt ein Ack Bit, das Acknowledge, welches bestätigt, dass der Slave die Daten bis zu diesem Punkt empfangen hat. Empfängt der Master dieses Ack, so fährt er mit der Datenübertragung fort. Nach einem Byte für die Daten erfolgt eine erneute Acknowledge-Abfrage, gefolgt von dem Stop Bit, um das Ende der Übertragung zu verkünden.

Die Adressierung erfolgt durch das erste, vom Master gesendete, Byte, von dem die ersten 7 Bits für die Adresse und das letzte für das R/\bar{W} -Bit steht. Das letzte Bit gibt somit an, ob der Slave Daten zu senden oder zu empfangen hat. Mit den sieben Adress-Bits können bis zu 128 Geräte adressiert werden.

In der folgenden Abbildung wird die Datenübertragung an sich veranschaulicht. Es werden die Signale beider Datenleitungen zur gleichen Zeit verglichen.

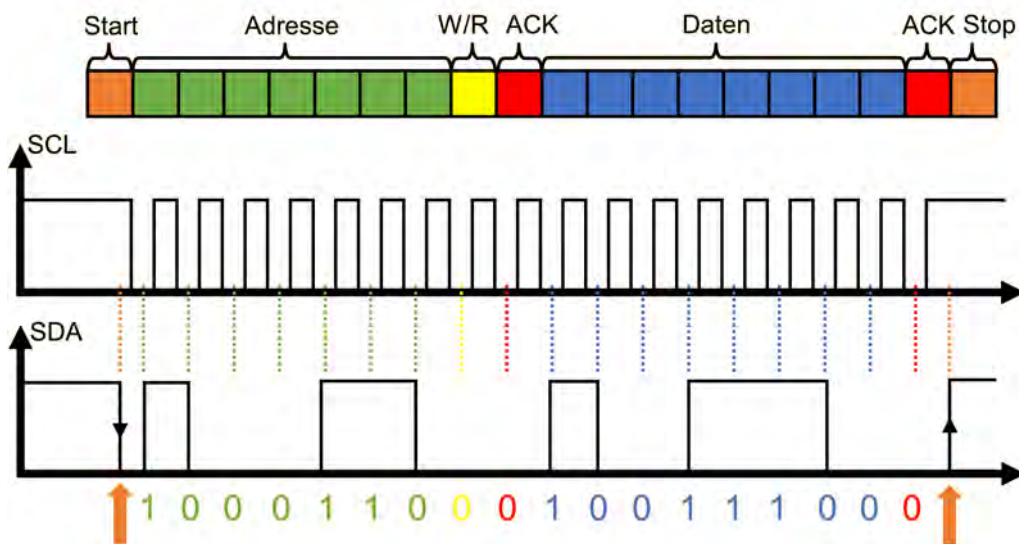


Abbildung 4.4: Zeitdiagramm einer I²C Kommunikation [22]

Über das Setzen des Dateneingangs auf Low, während sich die Taktleitung auf High befindet, wird der Start der Kommunikation angekündigt. Es folgen, passend zum Takt der SCL Leitung, die gesendeten Bits auf der Datenleitung. Der Pegel der Datenleitung wird immer nur dann geändert, sobald der Takt auf Low ist. Dies ist die Regel für die Datenübertragung. Bei der Start- oder Stop-Bedingung darf das Bit jedoch geändert werden, während die Taktleitung auf High steht. Dies passiert, wie es auf der Abbildung zu erkennen ist, am Ende der Übertragung. Der Pegel der Datenleitung wird, während der Takt auf High ist, auch auf einen High-Pegel gesetzt und die Datenübertragung ist abgeschlossen.

4.6 SNMP Protokoll

Ein weiteres wichtiges Netzwerkprotokoll dieses Projektes ist das Simple Network Management Protocol, kurz SNMP. Die erste Version dieses Protokolls wurde bereits 1988 definiert und fand zunächst unverschlüsselt statt. Die Sicherheit wurde jedoch erst in der dritten Version verbessert.

Bei der in diesem Projekt genutzten Version handelt es sich um die zweite Version.

Die Funktion des SNMP ist das Überwachen, Steuern und Fehlererkennen von Netzwerkelementen von einer zentralen Station aus. Zu diesen Elementen gehören Computer, Server, Drucker, Router, Steuergeräte und viele weitere. Die Einfachheit, Vielseitigkeit und Modularität sind ausschlaggebend für die Entwicklung dieses Standards, welcher von den meisten Endgeräten und Managementprogrammen unterstützt wird.

Der ATmega kann diesen Standard unterstützen, jedoch leiten viele Beispiele und andere Projekte, die mit diesem Protokoll arbeiten, auf veraltete Software und nicht mehr existente Bibliotheken weiter. Die Anpassung einer anderen Bibliothek ist jedoch in Arbeit. Bei der verwendeten Version ist die Größe der Adresswerte der so genannten OID durch falsche Datentypen begrenzt, welches jedoch geändert wird.

Object Identifier, kurz OID, sind die eindeutigen Adressen, welche die Adressenstruktur zu einem Managed Object beschreibt. Managed Objects sind auslesbare oder steuerbare Werte, welche durch die Manager Software abgefragt werden können.

Die Object Identifier sind in einer Management Information Base, kurz MIB, niedergeschrieben, welche die Adressstruktur eines Baumes besitzt. Die in dem MIB-Baum

verwendeten Adressen sind eindeutig und können bei der Internet Assigned Numbers Authority, kurz IANA, kostenlos beantragt werden. Die IANA ist für die Zuordnung von Nummern und Namen im Internet verantwortlich und verwaltet unter anderem IP-Adressen. Die Struktur der MIB besitzt eine feste Syntax und kann unter anderem als Textdatei in bestehende MIB-Verwaltungen eingepflegt werden.

Manager fragen die Daten von Agenten ab, ähnlich dem Master-Slave-System. Diese Abfragen werden durch Befehle wie GET- oder GETNEXT-REQUEST abgefragt und durch SET-REQUEST verändert. Darüber hinaus gibt es noch weitere Befehle wie eine TRAP, bei der ein Agent unaufgefordert Daten an einen Manager weiterleitet. Andere Befehle sind ähnliche oder zusammengefasste Abwandlungen der oben genannten Befehle. [Vgl. 39]

4.7 Leiterplattenfertigung

Ein sehr wichtiger Teil dieses Projektes ist das Erstellen eines Layouts der Leiterplatte und das Herstellen dieser.

Zunächst wird erläutert mit welchen Hilfsmitteln die Leiterplatte erstellt werden kann, danach wie dieses Layout erstellt wird und worauf zu achten ist. Darauf folgt eine Vorstellung der Leiterplattenherstellung.

4.7.1 Programme zur Leiterplattenfertigung

Es gibt eine Menge verschiedener Programme zur Platinenherstellung. Die Auswahl reicht von Programmen mit denen nur das Layout erstellt werden kann, über Programme, die zusätzlich eine Projekt- und Bauteilbibliothek verwalten können, bis zu Programmen, die ein so genanntes Auto-Routing anbieten.

Beim Auto-Routing handelt es sich um das automatische Anlegen von Leiterbahnen nach Schaltplan. Es werden die bestmöglichen Wege berechnet und umgesetzt. Dies ist bei mehrlagigen Leiterplatten unabdingbar, da es ohne solche Funktionen zu sehr vielen Überschneidungen kommt.

Ein sehr beliebtes und in diesem Projekt benutztes, Programm ist EAGLE. Dies ist die Abkürzung von Einfach Anzuwendender Grafischer Layout-Editor. Es können Schaltpläne entworfen werden, mit denen das Layout erstellt wird. Sind zusätzlich zu den

Abmessungen der Bauteile auch CAD-Modelle vorhanden, so kann die fertig bestückte Leiterplatte als 3D-Modell begutachtet werden.

Die kostenfreie Version bietet sehr eingeschränkten Projektumfang, reicht jedoch für kleine Projekte aus. Die Studentenversion bietet genug Möglichkeiten für viele Anwendungen.

Ein weiteres Programm ist Altium, welches einen ähnlichen Funktionsumfang besitzt wie EAGLE. Die Studentenversion bietet auch hier viele Vorteile.

Die kostenlose Alternative ist KiCad, welches dank Open Source ständig weiterentwickelt wird.

Welches Programm zur Erstellung von Schaltplänen und danach zum Anfertigen eines Layouts genutzt wird, bleibt jedem selbst überlassen. Bezahlpflichtige Programme besitzen jedoch den Ruf bessere Auto-Router zu besitzen.

Der Erste Schritt zum fertigen Layout ist das Erstellen eines Schaltplans. Dadurch weiß das Programm beim Anfertigen des Layouts, welches Bauteil an welchen Pin eines anderen Bauteils angeschlossen werden muss. [Vgl. 42]

4.7.2 Anfertigen eines Layouts

Nach dem Erstellen eines Schaltplans wird das Layout erstellt. Hierzu müssen zunächst die Abmessungen der Leiterplatte eingegeben werden. Danach werden die platztechnisch vorgegebenen Bauteile an ihre Position gesetzt und die frei beweglichen Bauteile taktisch, mit kurzen Wegen für Datenleitungen, positioniert.

Danach muss das Programm wissen welche Breite die Leiterbahnen besitzen. Diese Breite ist abhängig von den zu erwartenden Strömen. Für diesen Fall gibt es Tabellen, Faustformeln oder genaue Berechnungen. Es ist dem Nutzer nun selbst überlassen die Leiterbahnen entweder durch Auto-Routing oder selbst zu verlegen.

An manchen Stellen lässt es sich nicht vermeiden die Leiterbahnen auf die andere Seite der Platine zu setzen, damit Überschneidungen vermieden werden.

4.7.3 Optimieren des Layouts

Die Auto-Routing Funktion ist eine gute Hilfe, wenn es um das Planen der Leiterbahnen geht. Doch das Programm weiß nicht alles über die Funktion der Bauteile. So ist es möglich, dass eine mittels Pulsweitenmodulation getaktete Leitung direkt neben einer Datenleitung verläuft. In diesem Fall kann es durch Übersprechen zu einer Störung in der Datenübertragung kommen. Wichtige Leiterbahnen wie diese sollten aus diesem Grund vor dem Auto-Routing manuell verlegt, oder nach dem Routing genau überprüft werden.

Es muss außerdem darauf geachtet werden, dass die Leiterbahnen möglichst kurz sind. Lange Leiterbahnen können wie Antennen wirken und Störungen empfangen oder aussenden. Durch das Definieren von ungenutzten Flächen als Masseflächen kann diesem entgegengewirkt werden. Die Masse sollte zudem sternförmig angeordnet sein, damit keine Ströme im Kreis fließen.

Weitere Störungen können vermieden werden, indem sich die Entstörkondensatoren direkt an den Versorgungspins der dazugehörigen ICs befinden. Dies kann von den meisten Auto-Routern nicht überprüft werden, da diese die Bauteile und deren Verwendung nicht kennen.

Es kann zudem hilfreich sein, die Anschlussbeine der bedrahteten Bauteile als Durchführungen zur anderen Platinenseite zu nutzen. So müssen keine aufwändigen Durchkontaktierungen geschaffen werden und die Herstellung wird einfacher. Die Herstellung wird außerdem vereinfacht, wenn die Winkel der Leiterbahnen zueinander minimal 45° betragen.

Wie bereits erwähnt sollten die Leiterbahnen sich nicht gegenseitig stören. Neben PWM-Signalen können auch analoge durch digitale Signale gestört werden. Dies wird durch taktisches Positionieren, sowie durch genug Abstand umgesetzt. Zwischen Klein- und Netzspannung sollte ebenfalls eine klare Trennung, sowie genug Abstand eingehalten werden. [Vgl. 36]

Wird auf diese Dinge geachtet, so kann einem guten Layout nichts im Wege stehen.

4.7.4 Herstellung einer Leiterplatte

Wurde das Layout erstellt, so müssen für die Herstellung Gerber-Daten erstellt werden. Diese geben, ähnlich einer Fräsdatei für CNC-Fräsmaschinen, den Ursprung und den Endpunkt der Leiterbahnen, sowie die Position der Bohrungen als Koordinaten an. Des Weiteren enthalten diese Daten im Gerber-Format Informationen wie zum Beispiel die gewünschten Farben der Lackierung, Material und die Art der Durchkontaktierungen. Es kann auch die von dem Entwicklungsprogramm erstellte Platinen-Datei verschickt werden, jedoch kann somit das Layout entwendet und dessen Funktion nachvollzogen werden.

Die Daten werden an einen Hersteller der Wahl gesendet und dieser leitet die Fertigung ein. Gefertigt werden Leiterplatten auf zwei verschiedene Arten. Eines der Verfahren ist subtraktiv, es wird einer mit Kupfer überzogenen Leiterplatte Material entnommen, um so Leiterbahnen zu formen. Dieses Entfernen wird durch ätzen oder stanzen und bei Prototypen über fräsen erreicht.

Das additive Verfahren wird durch das Auftragen des negativen Leiterbahnbildes und durch ein anschließendes Metallbad erreicht. [Vgl. 4]

In diesem Projekt wurde für die erste Version eine Platinenfräse genutzt. Die zweite Version wurde beim Hersteller geätzt. Aus diesem Grund wird das Ätzen genauer erläutert.

Wird eine Leiterplatte geätzt, so werden zunächst die Daten ausgewertet und auf die zugehörigen Systeme übertragen. Das Bohren und Erstellen der Durchkontaktierungen wird meist zuerst durchgeführt. Darauf folgt das Plotten einer Maske aus den Leiterplattendaten. Auf die Platine wird ein lichtempfindlicher Film, auch Resist genannt, aufgetragen und mit der Maske des Leiterplattenbildes belichtet. Daraufhin wird diese entwickelt. Die belichteten Flächen sind, wie es der Name Resist schon sagt, gegen das Ätzmittel resistent. Je nach Art des Fotolackes kann aber auch die nicht belichtete Fläche gegen das Ätzmittel resistent sein.

Die Ätzlösung kann aus verschiedenen Chemikalien bestehen. Es werden zum Beispiel Eisen-III-Chlorid, Ammoniumpersulfat, Natriumpersulfat oder eine Mischung aus Salzsäure und Wasserstoffperoxid genutzt [Vgl. 37].

Nach dem Ätzen wird die Leiterplatte gespült, getrocknet und überprüft. Sollten keine Fehler vorhanden sein, so wird die Leiterplatte nachbearbeitet und versandt. [Vgl. 16]

4.8 Arduino IDE

Die verwendete Software zum Programmieren ist die Arduino IDE. In diesem Kapitel werden, nach einer kurzen Beschreibung dieser Software, die Vorteile und Nachteile aufgezählt. Daraufhin wird die Wahl dieser Programmierumgebung erläutert.

4.8.1 Bedeutung IDE

Der Name IDE stammt vom englischen Integrated Development Environment und steht für eine integrierte Entwicklungsumgebung. Zu den bekanntesten integrierten Entwicklungsumgebungen gehören unter anderem Visual Studio von Microsoft, Arduino, Eclipse oder auch Android Studio von Google.

Diese Programme können zum Teil für nur eine Programmiersprache, manchmal auch für mehrere Sprachen verwendet werden und sind somit vielseitig einsetzbar. Sie bestehen aus einer Sammlung von Computerprogrammen, einer Art Werkzeugsammlung zur Softwareentwicklung. Das Augenmerk einer IDE ist der wegfallende Medienbruch, da keine Daten zu einer anderen Software verschoben werden müssen, um die Entwicklung zu vervollständigen.

Die vorhandenen Tools einer IDE sind unter anderem ein Editor, welcher Quelltext formatiert und Syntax hervorhebt, Compiler, Linker, Debugger, Interpreter und andere Tools zum Erleichtern der Arbeit. [Vgl. 33]

Ein sehr wichtiger Bestandteil ist der Compiler. Dieser Compiler wandelt den in einer Programmiersprache geschriebenen Code in die binäre Sprache um, welche vom Prozessor verstanden wird. Dieser kann keinen Klartext lesen, sondern nur Anweisungen in Form von 0 und 1 ausführen. [Vgl. 50]

Im Gegensatz zum Compiler liest ein Interpreter den Quellcode Zeile für Zeile aus und führt diesen direkt aus. Somit kann das Verhalten des Programms während der Ausführung analysiert werden. Es muss nicht der gesamte Quellcode per Compiler umgewandelt und ausgeführt werden. Ein Interpreter ist somit sehr gut zum Analysieren von Fehlern geeignet, da dieser den Code an einer Fehlerstelle nicht auslesen kann und stoppt. Dadurch ist ein Interpreter zum Teil Bestandteil des Debuggers, kann jedoch auch unabhängig davon genutzt werden. [Vgl. 52]

Der Debugger ist ein weiterer wichtiger Baustein einer IDE. Dieser hilft dem Programmierer bei der Fehlersuche im Quellcode und lässt diesen Schritt für Schritt nachvollziehen, was gerade in dem Programm abläuft und wo Fehler auftreten. Ist in einem Code mit mehreren tausend Zeilen auch nur ein kleiner Fehler, so kann dieser mit Hilfe des Debuggers gefunden werden. Ein Feature ist zum Beispiel das Setzen sogenannter Haltepunkte, auch Breakpoints genannt, an denen der Debugger pausiert und die aktuell gespeicherten Variablen ausgibt. Es können außerdem Variablen ab diesem Punkt manuell verändert werden um das Verhalten des Programms zu testen. [Vgl. 3]

Ein Linker ist ein einfacher, jedoch sehr wichtiger Teil einer IDE, denn er verknüpft den verwendeten Code aus Bibliotheken mit dem geschriebenen Code und erzeugt ein ausführbares Programm. Diese Bibliotheken werden in 4.8.3 (S. 54) genauer beschrieben.

4.8.2 Programmiersprache

Das Programmieren ist nur mit Hilfe von Programmiersprachen möglich, denn es sind nur sehr wenige Menschen im Stande eine Programmierung mittels binärem Code durchzuführen. In diesem Fall hilft eine der vielfältigen Programmiersprachen.

Eine Programmiersprache formuliert Datenstrukturen und Algorithmen, welche sich aus schrittweisen Anweisungen, der sogenannten Syntax, zusammensetzen. [Vgl. 28]

Eine Programmiersprache bietet meistens folgende wichtige Bausteine, wie Ein- und Ausgabebefehle, oder eine Deklaration von Variablen zur Zwischenspeicherung von Daten. Hinzu kommen Bausteine, welche Mathematische Grundfunktionen einbinden und eine Verarbeitung von Zeichenketten oder Steueranweisungen für eine bedingte Ausführung ermöglichen. Ein Einbinden von Bibliotheken und Unterfunktionen wird zusätzlich angeboten.

Aus diesen Funktionen können höhere Funktionen erstellt werden, welche als Bibliothek gekapselt und somit als Bausteine für viele andere Programme genutzt werden können. Programmiersprachen werden in mehrere Kategorien unterteilt. So gibt es hardwarenahe Sprachen, höhere Programmiersprachen, Skriptsprachen, grafische Programmiersprachen und viele mehr. Diese Kategorien unterteilen die Vielzahl an Sprachen in die einzelnen Anwendungsgebiete.

Sehr häufig verwendete Programmiersprachen sind Java, C, C++ oder Python. Dies war jedoch nur ein sehr geringer Auszug aus den verfügbaren Sprachen. Dieses Projekt wurde mit der hardwarenahen Sprache C geschrieben. [Vgl. 32]

4.8.3 Programmbibliotheken

Bibliotheken bezeichnen eine Sammlung von Routinen und Unterprogramme, die Lösungswege für Problemstellungen anbieten, jedoch keine eigenständig laufenden Programme darstellen.

Mit Hilfe von Bibliotheken können auch komplexere Anwendungen, wie zum Beispiel der Anschluss eines Ethernetmoduls an ein Mikrocontroller Board, leicht umgesetzt werden. Statt selbst die benötigten Variablen zu definieren, sind die wichtigsten von ihnen bereits gespeichert. Auch unbekannte, vom Nutzer selbst zu programmierende Variablen, können per Aufruf einer zugehörigen Funktion mit den angehängten Daten hinzugefügt werden. So kann über eine kurze Funktion eine IP im Programmablauf definiert und das Netzwerkmodul über einen Befehl in der Programmschleife genutzt werden.

Diese Bibliotheken können selbst erstellt, über die IDE hinzugefügt oder frei im Internet heruntergeladen werden. Sie erlauben dem Anwender somit ein schnelles Arbeiten mit unbekanntem Modulen oder Programmabschnitten, ohne sich vorab mit der Materie auseinanderzusetzen.

In diesem Projekt wurden Bibliotheken wie ein Software PID-Regler, Ethernet, I²C oder SNMP genutzt, da das händische Programmieren dieser zu viel Zeit in Anspruch genommen hätte. [Vgl. 7]

4.8.4 Vor- und Nachteile der Arduino IDE

Die Vorteile der Arduino IDE sind, dass auch Nutzer mit weniger Programmiererfahrung diese nutzen können. Es sind viele Programmierbeispiele verfügbar und für jedes Problem gibt es eine Bibliothek, deren Funktionen ausführlich beschrieben sind. Das Schalten der Ein- und Ausgänge lässt sich per Befehl ausführen und es müssen keine Bits in den dazugehörigen Registern geändert werden.

Ein weiterer Vorteil ist die Community, mit deren Hilfe viele bekannte Fehler und Unstimmigkeiten ausgemerzt wurden. So gibt es in den deutsch- und englischsprachigen Foren sehr viel Unterstützung falls etwas nicht funktioniert. Dies ist jedoch kein Alleinstellungsmerkmal, da solche Foren auch für andere Programmiersprachen und IDE

vorhanden sind. Die Arduino IDE nimmt dem Nutzer Dinge ab, da viele Funktionen automatisch eingestellt werden und dies durch falsche manuelle Eingaben nicht zu Problemen führt.

Noch ein Vorteil ist die kostenlose Verfügbarkeit. Es kann problemlos von der Arduino Webseite heruntergeladen werden.

Ein Nachteil wurde bereits in den Vorteilen genannt: Die IDE greift dem Nutzer sehr unter die Arme, sodass sich kein Verständnis über die Materie aufbaut. Dies ist jedoch von den meisten Anwendern so gewollt, da sie entweder gut programmieren und sich nicht mit Mikrocontrollern auskennen, oder andersherum. Die IDE hat jedoch gewisse Grenzen, die jedoch erst auftreten, wenn sehr komplexe Anforderungen gestellt werden. Ein weiterer Nachteil ist jedoch das fehlende Debugging. Es werden zwar Fehler angezeigt und auch wo sich diese befinden, jedoch handelt es sich nur um Fehler in den Datentypen oder in der Syntax. Es kann kein Haltepunkt erstellt werden und es können keine Variablen mitten im Programm geändert werden.

Da die Vorteile den Nachteilen überwiegen wurde die Arduino IDE gewählt und der dazugehörige Bootloader auf den Mikroprozessor geladen.

4.9 Prüfung der Elektromagnetischen Verträglichkeit

Eine wichtige Prüfung für das Steuergerät stellt die Prüfung der elektromagnetischen Verträglichkeit, kurz EMV Prüfung, dar. Es sollte speziell bei diesem Steuergerät darauf geachtet werden, dass es keine anderen Geräte stört und selbst einen gewissen Schutz gegenüber anderen Störeinflüssen besitzt.

Aus diesem Grund wird im folgenden Kapiteln das Thema Prüfung der Elektromagnetischen Verträglichkeit behandelt, die Grundlagen dazu erklärt und wie die Probleme vermieden werden können.

Bevor es in die finale Phase des Projektes geht, muss dies offiziell überprüft werden, da zur Zeit noch Änderungen im Schaltplan und am Layout durchgeführt werden können.

4.9.1 Durchführung dieser Prüfung

Die Durchführung einer Prüfung der elektromagnetischen Verträglichkeit ist international, sowie europäisch genormt.

Die Einhaltung der Normen sorgt für eine störungsfreie Funktion verschiedener Geräte und muss durch unterschiedliche Tests überprüft werden. Diese unterscheiden sich nach Anwendungszweck des zu überprüfenden Geräts.

Des Weiteren sind die Prüfungen in mehrere Kopplungsarten unterteilt. Diese Arten sind galvanische, induktive, kapazitive Kopplung und eine Kopplung über das elektromagnetische Feld.

Die ersten drei Arten werden durch leitungsgebundene Störprüfungen getestet. Bei der vierten Prüfung wird die Störung durch ein elektromagnetisches Feld überprüft, welche die Kopplungsarten E-Feld, H-Feld und elektrostatische Felder umfassen. Diese Prüfungen werden für eine CE-Kennzeichnung benötigt. [Vgl. 15]

Die galvanisch gekoppelte Prüfung wird benötigt, um den Störeinfluss über die Anschlussleitungen, wie Spannungseingang, Signal-Ein- und -Ausgänge zu überprüfen.

Bei der galvanisch gekoppelten Prüfung werden Einbrüche der Netzspannung simuliert, sowie Einwirkungen von Schaltstörungen und von elektromagnetischen Entladungen überprüft. Der Einfluss von eingekoppelten Störungen über benachbarte Systeme wird in diesem Zuge mit geprüft.

Die induktiv gekoppelte Prüfung wird durchgeführt, indem eine Stromkoppelzange um den, wenn vorhanden, Kabelbaum gelegt wird. Wenn kein Kabelbaum vorhanden ist, oder als zusätzliche Prüfung, können einzelne Leitungen des Prüflings geprüft werden.

Es wird auf zwei Arten gemessen. Die Vorgehensweise Störfestigkeitsprüfung, kurz SF, wird durchgeführt, indem über die Stromkoppelzange ein hochfrequenter Strom eingepreßt wird. Die Funktion muss trotz dieses Stroms gegeben sein. Der hochfrequente Strom kann jedoch auch durch einen, sich wiederholenden, Einzelimpuls umgesetzt werden.

Die Methode Störaussendungsmessung, kurz SA, prüft die Magnetfelder des Kabelbaums oder der einzelnen Leitungen und wertet die Stärke dieser aus.

Die kapazitiv gekoppelte Prüfung findet auch in den beiden bereits genannten Bereichen statt. In diesem Fall wird über eine definierte Koppelkapazität ein Störsignal eingestreut, SF, oder der ausgekoppelte Störpegel über diese Koppelkapazität gemessen, SA.

Die strahlungsgekoppelte Prüfung kann über mehrere Verfahren erfolgen. So gibt es eine Antennenmessung, eine Messung über Streifenleitung und die Messungen über eine TEM-, bzw. GTEM-Zelle. Letztere sind die Abkürzungen für Transversal Elektromagnetische Welle und Gigahertz TEM-Zelle.

In diesem Abschnitt wird die bekannteste Form, die Antennenmessung, beschrieben. Es wird bei der Antennenmessung eine Antenne im definierten Abstand von 1, 3 oder 10 m zum Prüfling aufgestellt. Diese Messung muss in einem abgeschirmten Raum ohne fremde Störeinflüsse stattfinden. Der Frequenzbereich der Prüfung erstreckt sich zwischen 30 MHz bis zu 18 GHz.

Der Störfestigkeitstest wird in diesem Fall durch die Erzeugung einer definierten Feldstärke, welche auf den Prüfling gerichtet ist, durchgeführt. Dieser muss trotz Hochfrequenzfeld fehlerfrei funktionieren. Bei der Störaussendungsmessung wird die abgestrahlte Störfeldstärke gemessen. [Vgl. 14] [Vgl. 47]

4.9.2 Änderungen im Fehlerfall

Sollte die Messung fehlschlagen, gibt es mehrere Arten der Fehlerbehebung. Durch kürzere Leitungslängen kann die Leitungsinduktivität gesenkt werden. Zudem können die Leitungen dicht beieinander verlegt und bei Verdrahtungen verdrillt werden, um diese Induktivität zu verringern.

Bauteiltechnisch kann für jedes IC ein Stützkondensator verwendet werden. Außerdem ist es ratsam, für jeden Stromkreis eine separate Leitungsführung anzulegen.

Wie bereits bei den Layout-Tipps genannt, macht es Sinn Schleifen in der Masseführung zu vermeiden und diese Sternförmig aufzubauen.

Durch lokalisieren des störenden Bauteils und analysieren der Fehlerursache in diesem Bauteil kann zum Beispiel über eine Schirmung, oder durch Austausch mit einem ähnlichen Bauteil das Problem beseitigt werden. [Vgl. 9]

5 Finale Hardwareumsetzung des Prototyps

Dieses Kapitel befasst sich mit der Umsetzung der Hardware. Es wird das Layout beschrieben und danach der fertige Prototyp vorgestellt. Darauf folgen Hinweise zu den wichtigen Funktionen des Steuergerätes.

5.1 Überblick

Nach Erstellung des Layouts aus dem Schaltplan, welcher sich im Anhang auf der beigelegten CD befindet, wurde die erzeugte Gerberdatei an einen Leiterplattenhersteller geschickt, um die Platine ätzen zu lassen. Es wurden zunächst zwei Leiterplatten erstellt, damit ein fertiges Steuergerät im Schaltschrank laufen kann und ein zweites zum Test der Softwareoptimierung verfügbar ist.

Die fertige Leiterplatte wird auf den folgenden zwei Abbildungen vorgestellt. Auf der Oberseite der Leiterplatte befindet sich eine nachträglich eingefügte Nummerierung, welche zur Veranschaulichung der einzelnen Bereiche auf dieser dient. Alle Bauteile werden auf der Platinenoberseite montiert, deshalb befindet sich nur dort der Aufdruck der Bauteilbeschriftung.

Nach dem Ätzen wurde die Leiterplatte mit einem grünen Lötstopplack versehen und daraufhin mit den Labels und Umrissen der Bauteile bedruckt. Die Beschriftung der internen Bearbeitungsnummer von DESY, sowie die Kennzeichnung der Ober- und Unterseite wurden in Kupfer auf die Platine geätzt.

In der folgenden Abbildung ist die Unterseite der gefertigten Platine abgebildet.

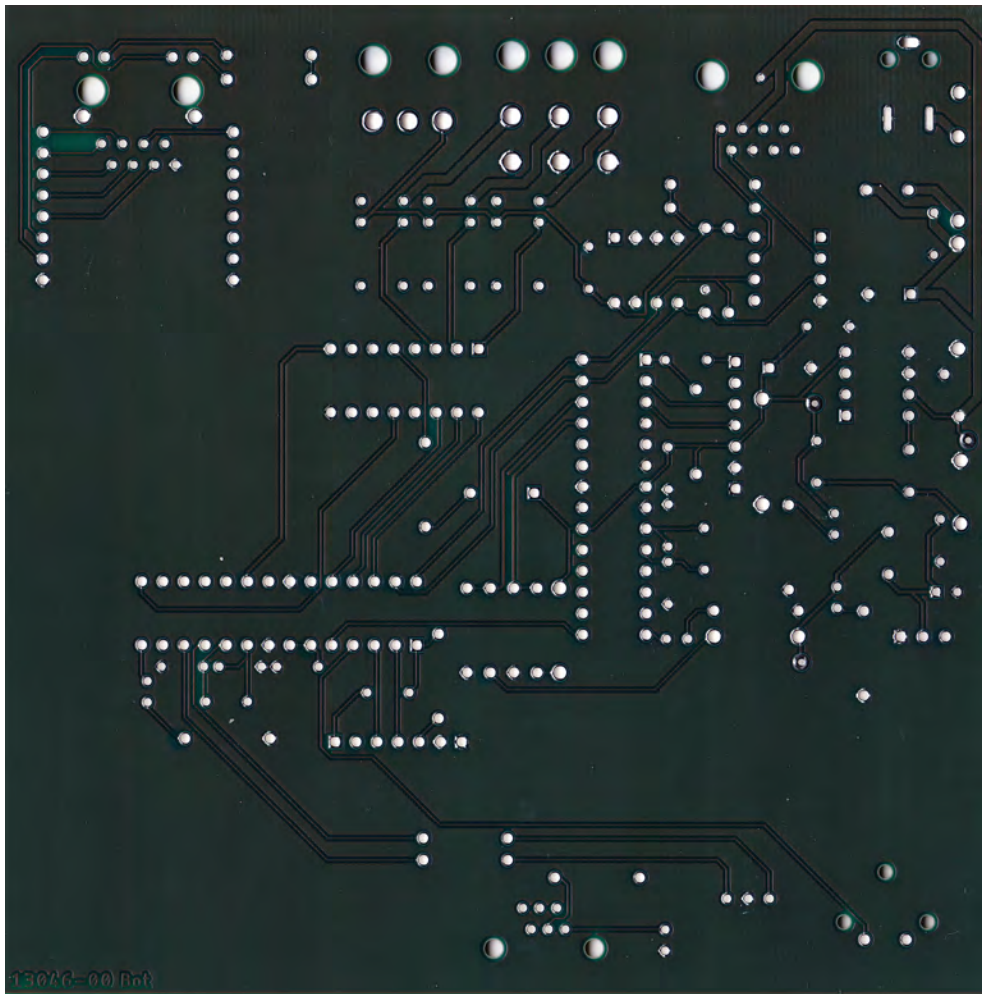


Abbildung 5.1: Scan der Platinenunterseite des Prototyps

Es ist zu erkennen, dass sich auf dieser Seite layoutbedingt sehr wenige Leiterbahnen befinden. Diese besitzen einen minimalen Winkel von 45° zueinander und verlaufen in den meisten Fällen parallel zueinander.

Bei genauerer Betrachtung können einzelne Durchführungen auf der Massefläche erkannt werden. Diese dienen zur Durchkontaktierung eingeschlossener Kupferflächen auf der Oberseite und stellen eine Verbindung zur Masse her.

Ohne Beschriftung und Bauteilaufdruck wirkt das Layout sehr unübersichtlich. Aus diesem Grund wird in der folgenden Abbildung die Oberseite der Leiterplatte gezeigt, welche sich auf der nächsten Seite befindet.

In der Abbildung der Oberseite wurden die jeweiligen Funktionsbereiche in rot gekennzeichnet.

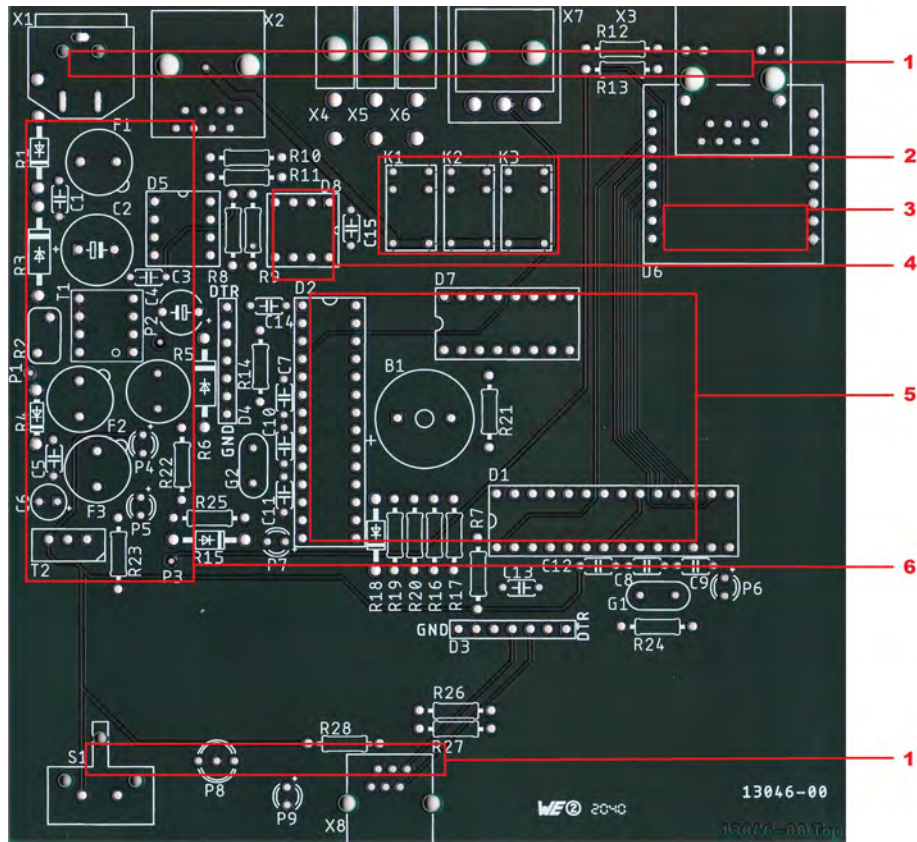


Abbildung 5.2: Nummerierter Scan der Platinenoberseite des Prototyps

In der Folgenden Aufzählung werden die einzelnen Bereiche der Leiterplatte beschrieben.

- 1: Ein- und Ausgänge der Leiterplatte
- 2: Relais zum Versorgen externer Bauteile mit 24 V
- 3: Netzwerkmodul
- 4: EEPROM zur Speicherung der Logs
- 5: Beide Prozessoren, links der sekundäre, rechts der primäre Prozessor
- 6: Bereich des Netzteils auf der Platine

Wie auf der Abbildung zu erkennen ist, sind die Leiterbahnen des Netzteilbereiches sehr kurz. Dies wirkt sehr gedrungen, jedoch ist dies die beste Möglichkeit um ein- und ausgehende Störungen zu vermeiden.

Die beiden Prozessoren sind L-förmig angeordnet. Beim primären Prozessor handelt es sich auf dieser Leiterplatte noch um einen ATmega328p, da dessen Funktion im Entwicklungsprozess noch gegeben war. Durch Weiterentwicklung der Software war der Speicher dieses Prozessors nicht mehr ausreichend. Für die Testphase wird der Sockel auf einen 40-poligen ATmega1284p adaptiert, dies ist auf der Abbildung 5.4 zu erkennen.

Die Außenmaße der Leiterplatte entsprechen den Maßen der Platine von Rittal und die Schaltung besteht nur aus sehr wenigen Bauteilen. Für die finale Version können noch ein paar Optimierungen stattfinden, dies wird jedoch in Kapitel 7.2 auf Seite 80 behandelt.

Das Bestücken der Leiterplatte erfolgte durch den Entwickler, wird jedoch, bei einer größeren Stückzahl, durch die firmeninterne Fertigung umgesetzt. Über eine so genannte Lötwellen werden die vorher maschinell und manuell bestückten Bauteile gezielt mit flüssigem Lötzinn gespült und so mit der Leiterplatte verlötet. Die Prozessoren und Speicherbausteine werden jedoch von der betreuenden Person verbaut, da deren Software- und Datenstände vor der Montage überspielt werden müssen. Mehr dazu im Kapitel 5.2 auf Seite 64.

Das Netzwerkmodul ragt über die Leiterplatte hinaus. Dieses wird jedoch durch eine Eigenanfertigung ersetzt. Dazu wird das Layout der Minimalbeschaltung, wie bereits auf dem Adapter, auf die Platine des Steuergerätes übertragen. Die Funktion wird jedoch durch einen kleineren Prototyp auf einer gefrästen Platine getestet. Das Netzwerkmodul ist bei dem im Schaltschrank befindlichen Steuergerät fest verlötet. Die Test-Leiterplatte besitzt jedoch Buchsenleisten zum Anschluss des Moduls.

Die folgende Abbildung zeigt die fertig bestückte Leiterplatte ohne den Prozessoradapter. Dieser wird separat gezeigt und erklärt.

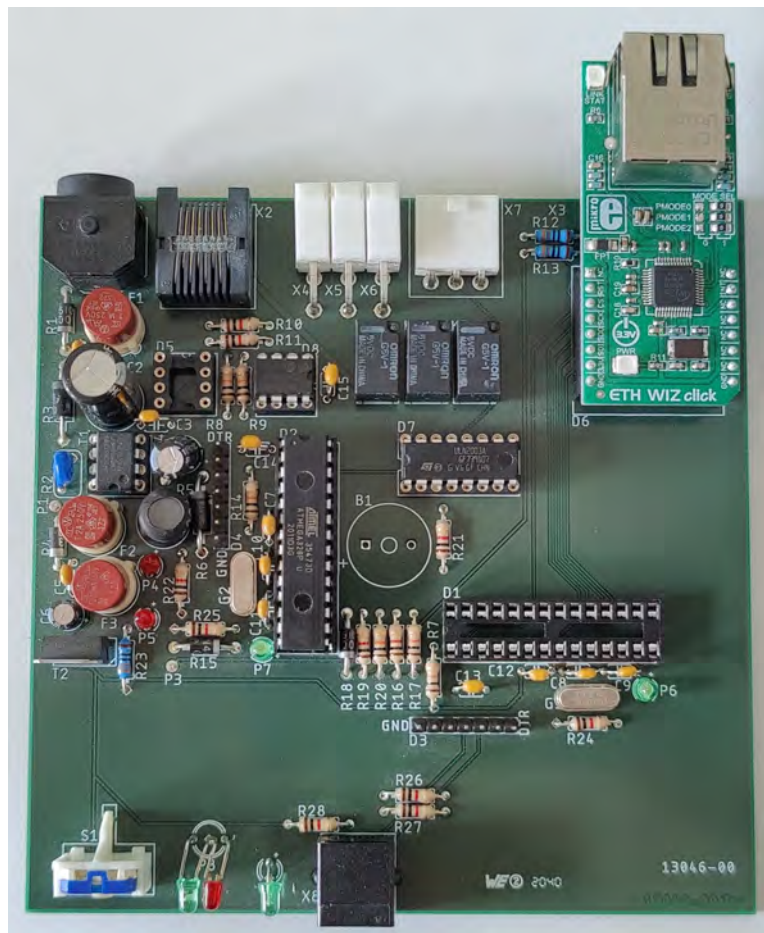


Abbildung 5.3: Fertig bestückte Leiterplatte des Steuergerätes

Auf der linken Seite befindet sich sichtbar der Bereich des Netzteils. Dort finden sich unter anderem die drei rot-braunen Kleinstsicherungen.

Bei dem fehlenden Bauteil des nicht belegten IC-Sockel handelt es sich um einen überflüssigen I²C-Puffer, dessen Funktion nach Erstellung des Layouts verworfen wurde. Der Buzzer, das fehlende Bauteil in der Mitte, wird ebenfalls nicht benötigt und wird aus dem finalen Layout entfernt. Die beiden nebeneinander befindlichen LEDs am unteren Rand werden durch eine Dual-LED ersetzt. Bei der gelieferten LED handelte es sich um ein falsches Pinout, aus diesem Grund wurde die RGB-LED durch zwei LEDs ersetzt.

Wie bereits erwähnt und auch auf der Abbildung zu erkennen, ragt der Netzwerkadapter über die Leiterplatte hinaus. Platztechnisch wäre es möglich gewesen diesen so weit auf die Leiterplatte zu verschieben bis der Anschluss bündig zur Kante steht.

Dieses Abstehen ist jedoch beabsichtigt, denn so kommt es nicht zu einer versehentlichen Verwechslung der P-I²C Buchse mit der Netzbuchse. Bei dem finalen Steuergerät wird dies jedoch ausreichend auf der Frontplatte gekennzeichnet.

Die folgende Abbildung zeigt den Adapter zur Montage des größeren ATmega1284p auf dem Sockel des ATmega328p.

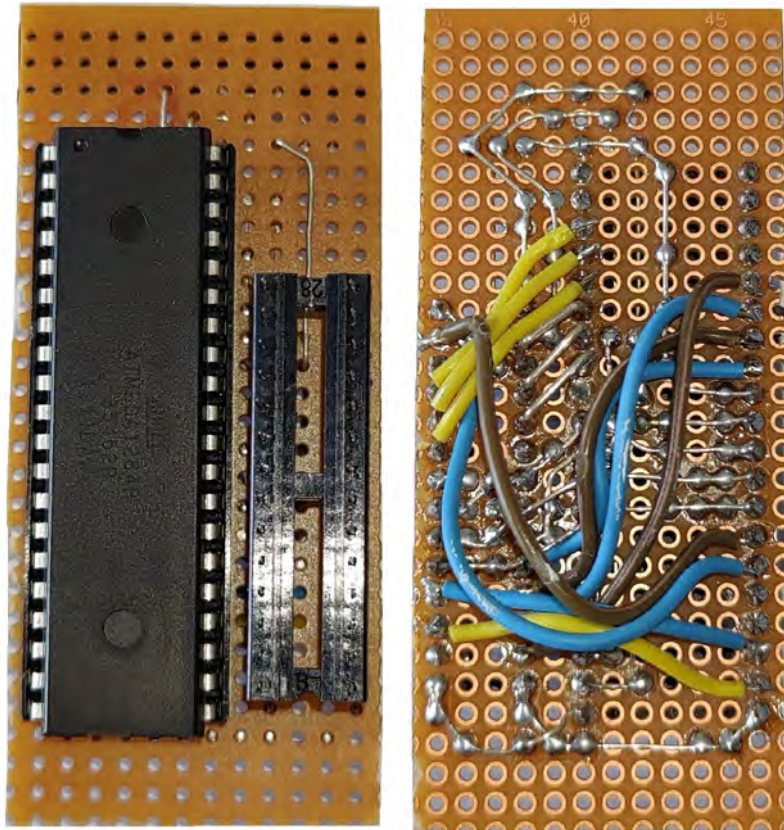


Abbildung 5.4: Prozessoradapter für den Prototyp

Die Funktion der Kombination aus festen Leiterbahnen aus Silberdraht und der fliegenden Verdrahtung ist gegeben und das Steuergerät kann auf diese Weise getestet werden. Sollte das System Störungen aufweisen, so wird zunächst der Adapter in Betracht gezogen. Eine Verdrahtung mittels Kupferlackdraht ist jedoch schneller und somit effizienter.

Der Prozessor wird beim finalen Layout fest eingeplant und der Adapter wird nicht mehr benötigt. Das Bestehen der EMV-Prüfung in dieser Form steht jedoch nicht fest, da Störeinflüsse über die langen Leitungen eingestreut werden können.

5.2 Wechselbarkeit der Bauteile

Da das Steuergerät eine gewisse Ausfallsicherheit gewährleisten soll, sind die wichtigsten Bauteile austauschbar. Kommt es zu einem Fehler, der einen der beiden Prozessoren betrifft, läuft das System problemlos weiter. Die Sicherheitsfunktionen werden im nächsten Kapitel beschrieben. Wenn ein Prozessor durch den Watchdog nicht neu startet, kann vor Ort ein Wechsel von diesem erfolgen.

Der Betreuer kann im Vorfeld die Prozessoren mit der richtigen Software bespielen und nur mit einem Schraubendreher und einem IC-Ausbauwerkzeug vor Ort am Steuergerät einen Wechsel durchführen. So entfällt der Ausbau und das Durchtauschen mit einem Ersatzgerät und die potenzielle Ausfallzeit beschränkt sich nur auf die Zeit während des Wechsels.

5.2.1 Austauschbare Prozessoren

Der Grund für die Verwendung von austauschbaren Prozessoren wurde bereits in der Einleitung dieses Kapitels beschrieben. Dieser Abschnitt behandelt das Aufspielen der Datenstände auf den Prozessor.

Vor dem Bestücken der Leiterplatte mit den Prozessoren und dem EEPROM müssen diese mit einem Datenstand bespielt werden. Die Prozessoren werden zunächst mit dem zugehörigen Bootloader bespielt, danach mit der passenden Software. Die Software muss insofern angepasst werden, dass die richtige IP und MAC Adresse eingetragen wird. Dies gilt für den primären Prozessor. Der Datenstand des sekundären Prozessors kann ohne Veränderungen überspielt werden, da dieser nicht mit dem Netzwerk verbunden ist.

Für das Überspielen wurde ein Aufsatz für einen Arduino Uno entwickelt, welches das Brennen der Bootloader ermöglicht. Dieser ist zur Zeit nur für einen ATmega1284p vorbereitet. Im Kapitel 7.2 auf Seite 80, wird jedoch die Planung einer neuen Vorrichtung beschrieben.

5.2.2 Wechselbares EEPROM

Das EEPROM ist ebenfalls austauschbar. Sollte die Strahlenbelastung so hoch sein, dass dieser beschädigt wird, kann er, nach einem kurzen Aufspielen des Datenstands auf einen neuen EEPROM, problemlos ausgewechselt werden. Die vorhandenen Logs werden so zwar nicht mehr auf der Webseite angezeigt, jedoch kann über die, im Kapitel zuvor genannte, Vorrichtung der Speicher problemlos ausgelesen werden.

Sollte der genutzte EEPROM nicht mehr produziert werden, so kann die Verwendung einer anderen Baureihe in der Software leicht angepasst werden.

5.2.3 Sicherungen und Überprüfung der Spannung

Zur Vereinfachung der Fehlersuche wurden drei Testpunkte auf die Leiterplatte gesetzt und beschriftet. Diese dienen zur Überprüfung der Spannungen und zeigen einen Fehler in der Spannungsversorgung, wenn dieser vorhanden ist.

Wird keine Spannung auf dem 24 V-Testpunkt gemessen, so ist es ratsam die zugehörige Sicherung zu testen und gegebenenfalls nach einer Kurzschlussprüfung zu wechseln. Ohne die Prüfung auf einen Kurzschluss kann eine neue Sicherung sofort auslösen und der Wechsel war erfolglos.

Dies gilt auch für die beiden anderen Spannungen.

6 Umsetzung der Software

Die Software stellt in diesem Projekt die wichtigste Komponente dar. Der Entwurf des Layouts und die Umsetzung auf der Leiterplatte ist der Funktion durch die Software unterlegen. Der Quellcode der Testversion befindet sich im Anhang auf der beigelegten CD.

Im Folgenden wird zunächst ein vereinfachter, allgemeiner Ablaufplan der Software der beiden Prozessoren gezeigt und die Funktion erläutert. Die Vereinfachung erfolgt durch das Umschreiben und Zusammenfassen der verwendeten Variablen. Zum Beispiel werden die Timer-Zuweisungen,

```
currentMillis = millis();  
noResponseTimer = millis();  
ledTimer = millis();  
EEPROMtimer = millis();  
flowTimer = millis();
```

mit "Zuweisung der Timer" abgekürzt. Der Ablaufplan wurde vereinfacht, damit der Platzverbrauch in dieser Ausarbeitung verringert wird.

Im folgenden Kapitel 6.2 (S. 71) werden einige Unterfunktionen speziell erklärt. Die Sicherheitsfunktionen sind zur Vereinfachung ausgeblendet und werden im Kapitel 6.3 auf Seite 76 erläutert.

6.1 Überblick

Die folgende Abbildung zeigt den vereinfachten Programmablaufplan des primären Prozessors.

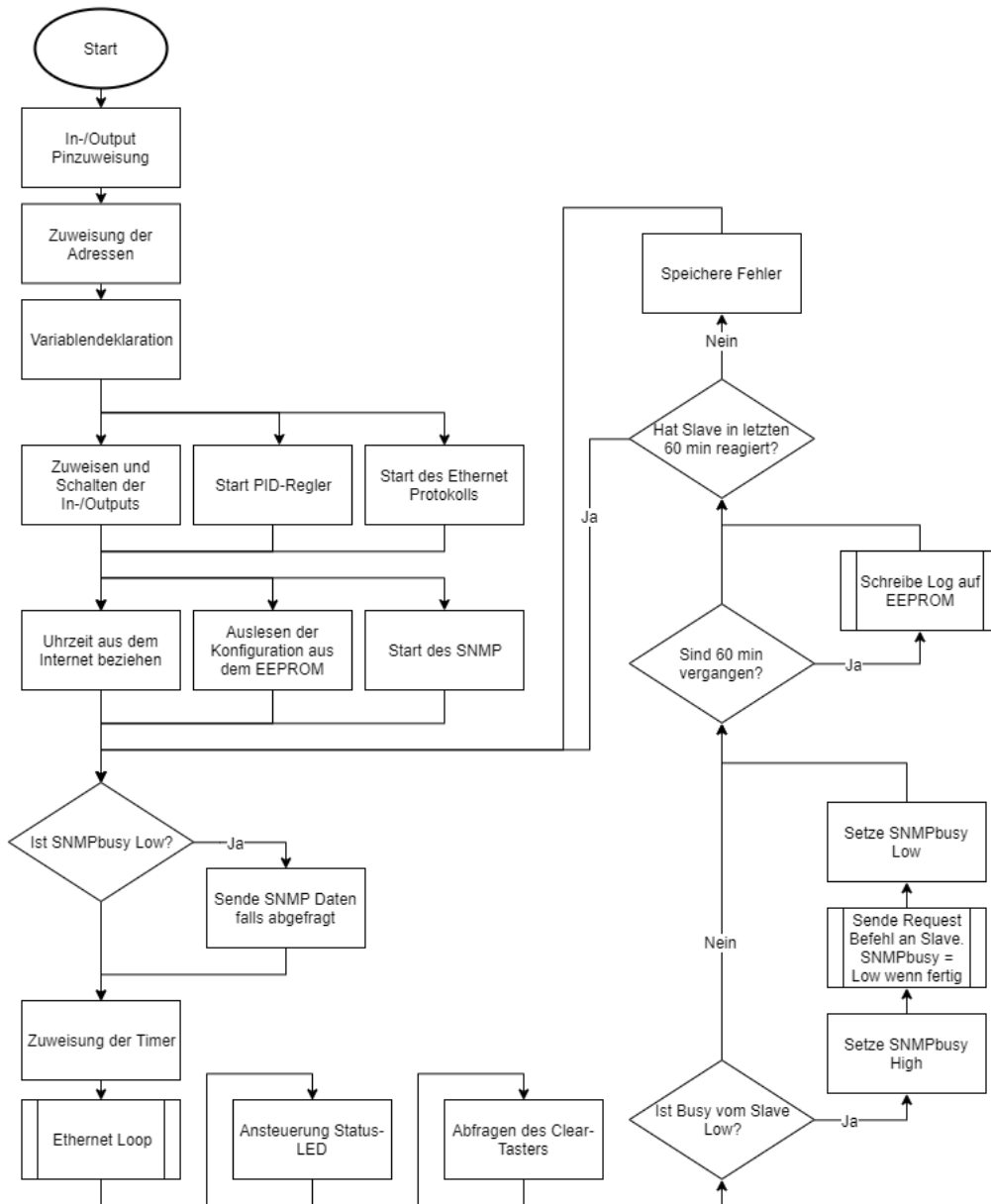


Abbildung 6.1: Programmablaufplan des primären Prozessors

Zunächst wird das Pinout zugewiesen und danach erfolgt die Zuweisung der Adressen für die Steuergeräte, dem sekundären Prozessor, Ethernet und SNMP. Daraufhin erfolgt die Variablendeklaration.

Die Setup-Funktion, in diesem Fall als zwei parallele Zuweisungsstränge definiert, führt eine Zuweisung und ein eventuelles Schalten der Ein- und Ausgänge durch. Zusätzlich werden der PID-Regler, das Ethernetprotokoll und das SNMP gestartet. Nachdem die Konfigurationsdaten aus dem EEPROM geladen wurden, wird die aktuelle Uhrzeit aus dem Internet abgerufen.

Es startet die Loop-Funktion, welche mit einer Abfrage der SNMPbusy Variable beginnt. Die Variable wird gesetzt, sobald ein Request an den sekundären Prozessor geschickt wird, damit es zu keiner Unterbrechung beim Empfang der Daten vom Slave kommt, wenn Daten über SNMP abgefragt werden.

Es folgt eine Zuweisung der Timer und ein Aufruf der Ethernet-Loop-Funktion, welche so häufig wie möglich aufgerufen werden muss.

Nach dem Empfang der Daten wird SNMPbusy wieder auf Low gesetzt. Und es erfolgt die nächste Zeitabfrage. Sind seit dem letzten Log-Eintrag 60 min vergangen, so wird ein neuer Eintrag geschrieben.

Am Ende wird die letzte Änderung des Busy-Eingangs vom sekundären Prozessor gecheckt. Hat sich der Pegel in den letzten 60 min nicht verändert, so geht der primäre Prozessor davon aus, dass etwas mit seinem Gegenüber nicht stimmt. In dem Fall greifen die Sicherheitsfunktionen.

Nach dieser Abfrage startet die Schleife von vorne.

Im Folgenden wird der vereinfachte Programmablaufplan des sekundären Prozessors dargestellt. Dieser ist einfacher aufgebaut, da es keine Netzwerkfunktion gibt. Auch hier sind die Sicherheitsfunktionen ausgeblendet, da diese gesondert beschrieben werden.

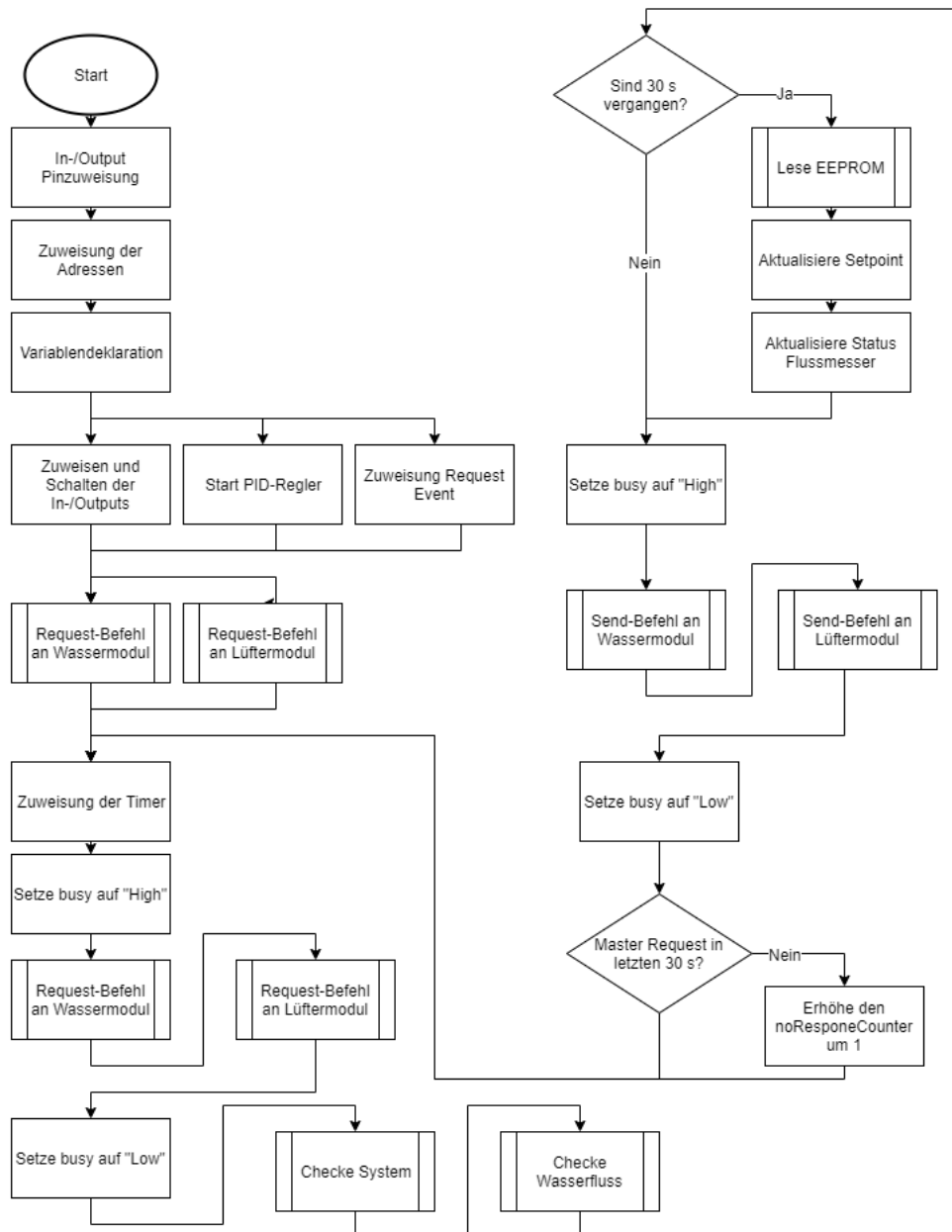


Abbildung 6.2: Programmablaufplan des sekundären Prozessors

Auch dieses Programm beginnt mit der Zuweisung des Pinouts und der benötigten Adressen zur Kommunikation mit dem Master und den anderen Steuergeräten. Daraufhin werden die Variablen deklariert.

In der Setup-Funktion, welche der Loop-Funktion vorangestellt ist, werden die Ein- und Ausgänge zugewiesen und nach Bedarf geschaltet. Der PID-Regler wird nach vorheriger Konfiguration gestartet und dem Prozessor die Reaktion auf ein Request-Event vom Master zugewiesen.

Der Loop startet ab jetzt, indem die Timer für den Ablauf aktualisiert werden und der Busy-Anschluss, welcher mit dem Master verbunden ist, auf High gesetzt wird. Dies signalisiert dem Master, dass dieser in diesem Moment keine Anfrage senden kann, da der Slave sonst die Datenübertragung zu den Steuergeräten abbrechen könnte.

Ist der Ausgang auf High gesetzt, so wird zunächst ein Request-Befehl an die beiden Steuergeräte geschickt, um die aktuellen Werte abzufragen.

Busy wird daraufhin wieder auf Low gesetzt und es beginnen Systemtests, welche unterbrochen werden können.

Die folgende Abfrage überprüft, ob seit dem letzten Auslesen 30 s vergangen sind. Wenn ja, dann werden der Setpoint und der Status des Durchflussmessers aus dem EEPROM gelesen und in die lokalen Variablen gespeichert. Danach, oder falls keine 30 s vergangen sind, wird Busy erneut auf High gesetzt und die berechneten Daten für die Regelung der Lüfter, sowie die Daten aus dem Systemcheck auf die Steuergeräte übertragen. Daraufhin wird der Busy-Anschluss wieder auf Low gesetzt.

Erfolgte innerhalb der letzten 30 s kein Request vom Master, so wird ein Zähler hoch gesetzt. Dieser dient zur zusätzlichen Identifikation von Fehlern des Master-Prozessors.

Die Schleife startet erneut und läuft so lange weiter bis die Fehlererkennung, im Falle eines nicht mehr reagierenden Masters, einen anderen Programmablauf startet.

Einzelne Zahlenangaben können sich jedoch noch verändern, falls diese zu gering oder deutlich zu hoch eingestellt sind.

6.2 Ablauf einzelner Funktionen

6.2.1 Schreiben des Logs auf den EEPROM

Bei der folgenden Abbildung handelt es sich um das Ablaufdiagramm des Schreibens von Log-Einträgen auf den EEPROM, welche in Abbildung 6.1 als Unterfunktion aufgerufen wird.

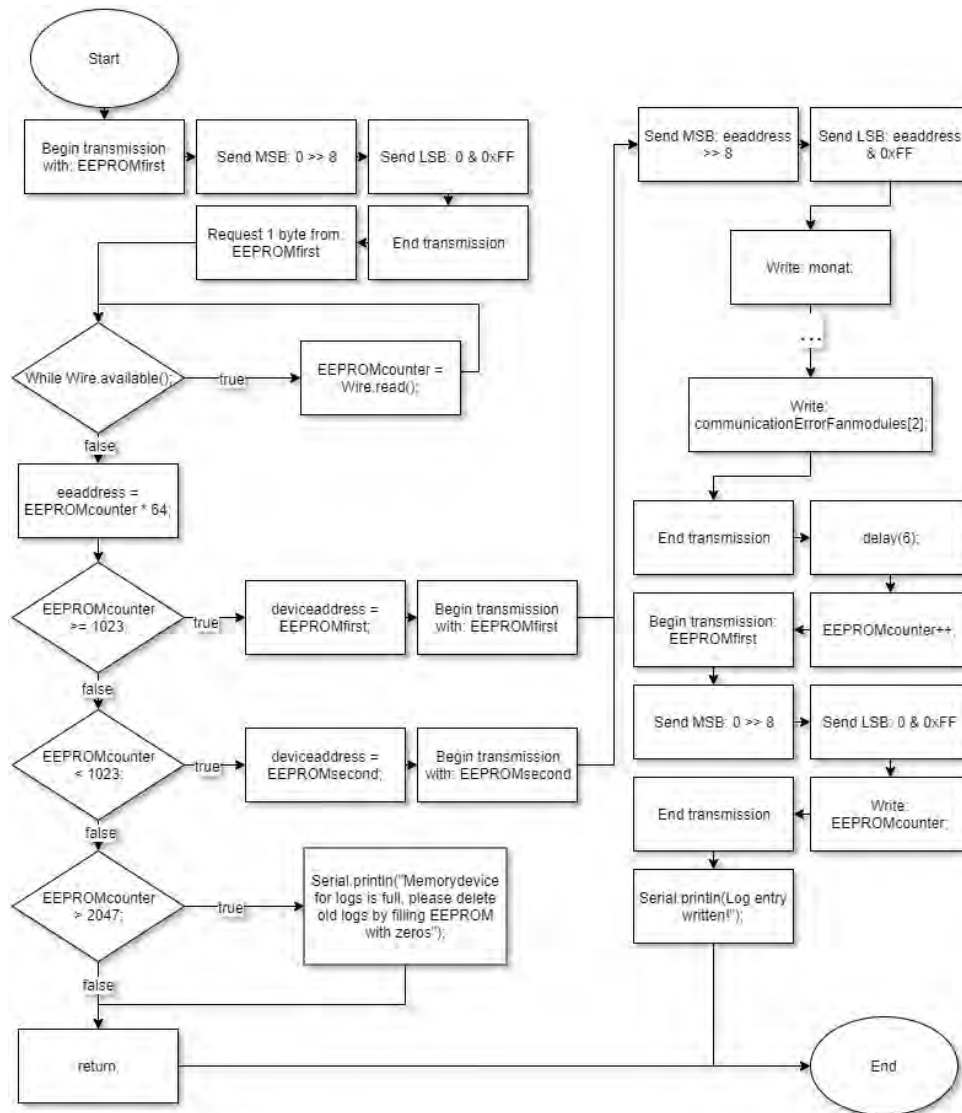


Abbildung 6.3: Programmablaufplan der Funktion zum Schreiben eines Log-Eintrags

Nach dem Start der Funktion wird eine Kommunikation mit dem EEPROM angekündigt. Die Adresse ist EEPROMfirst, welche eine der beiden Adressen für diesen Baustein ist. Je Adresse werden 512 kB adressiert, deshalb wird der EEPROM in EEPROMfirst und EEPROMsecond unterteilt.

Nach Ankündigung eines Datentransfers wird dem EEPROM zuerst das Most Significant Byte, kurz MSB und darauf das Least Significant Byte, kurz LSB übertragen. Diese Bytes sind nach Wertigkeit geordnet und werden dem EEPROM zur Lokalisierung des abzufragenden oder zu sendenden Bytes gesendet. Die Adresse 0 steht für das Erste Byte im Speicher und wurde als Speicherplatz für den EEPROMcounter festgelegt.

Durch Beenden der Datenübertragung, wird dem EEPROM signalisiert, dass Daten von dieser Adresse abgefragt werden.

Aus diesem Grund wird im nächsten Schritt ein Request-Befehl über ein Byte an EEPROMfirst gesendet und darauf die geforderten Daten empfangen.

Nach der Schleife wird die interne Adresse vom Speicherplatz ermittelt. Dafür wird der zuvor ausgelesene EEPROMcounter, die Anzahl der bereits geschriebenen Logs, mit 64 multipliziert. Die 64 ergibt sich durch das Schreiben von ganzen Seiten bei jedem Log-Eintrag, welche 64 Byte groß sind. Befindet sich die Adresse auf der Ersten Hälfte des EEPROMs, wird EEPROMfirst als Empfänger festgelegt, falls sich die interne Adresse auf der anderen Hälfte befindet, EEPROMsecond.

Sollte die Adresse größer als 2047 sein, so ist der Speicher voll. Im Test wird eine Meldung zum Wechsel ausgegeben, um die geschriebenen Logs zu kontrollieren. Danach wird die Funktion über Return beendet. In der finalen Version beginnt das Speichern wieder von vorne und der EEPROMcounter wird zurückgesetzt.

Nachdem die zugehörige Hälfte des Speichers ermittelt wurde, werden das MSB und LSB mit der zu beschreibenden Adresse übermittelt und die Daten in das EEPROM geladen. Die Datenübertragung wurde in der Abbildung mittels drei Punkte gekürzt. Es werden Daten wie das Datum, die Uhrzeit, sowie Temperaturen und weitere Daten übertragen. Diese können auf dem Webinterface im Kapitel 6.4 (S. 77) abgelesen werden.

Nach der Datenübertragung wird diese Beendet, eine kurze Pause zum Verarbeiten eingelegt und der EEPROMcounter erhöht. Dieses wird im ersten Byte des EEPROMs hinterlegt. Nachdem das Aktualisieren des Zählers abgeschlossen ist, wird eine Bestätigung zum Erstellen des Logs ausgegeben und die Unterfunktion beendet.

6.2.2 Ansteuerung des Lüftermoduls

Dieser Abschnitt befasst sich mit der Kommunikation mit den anderen Steuergeräten. So wird als Beispiel die Kommunikation mit dem Lüftermodul als Ablaufdiagramm veranschaulicht und beschrieben. Die nächste Abbildung zeigt das Ablaufdiagramm für das Empfangen von Daten.

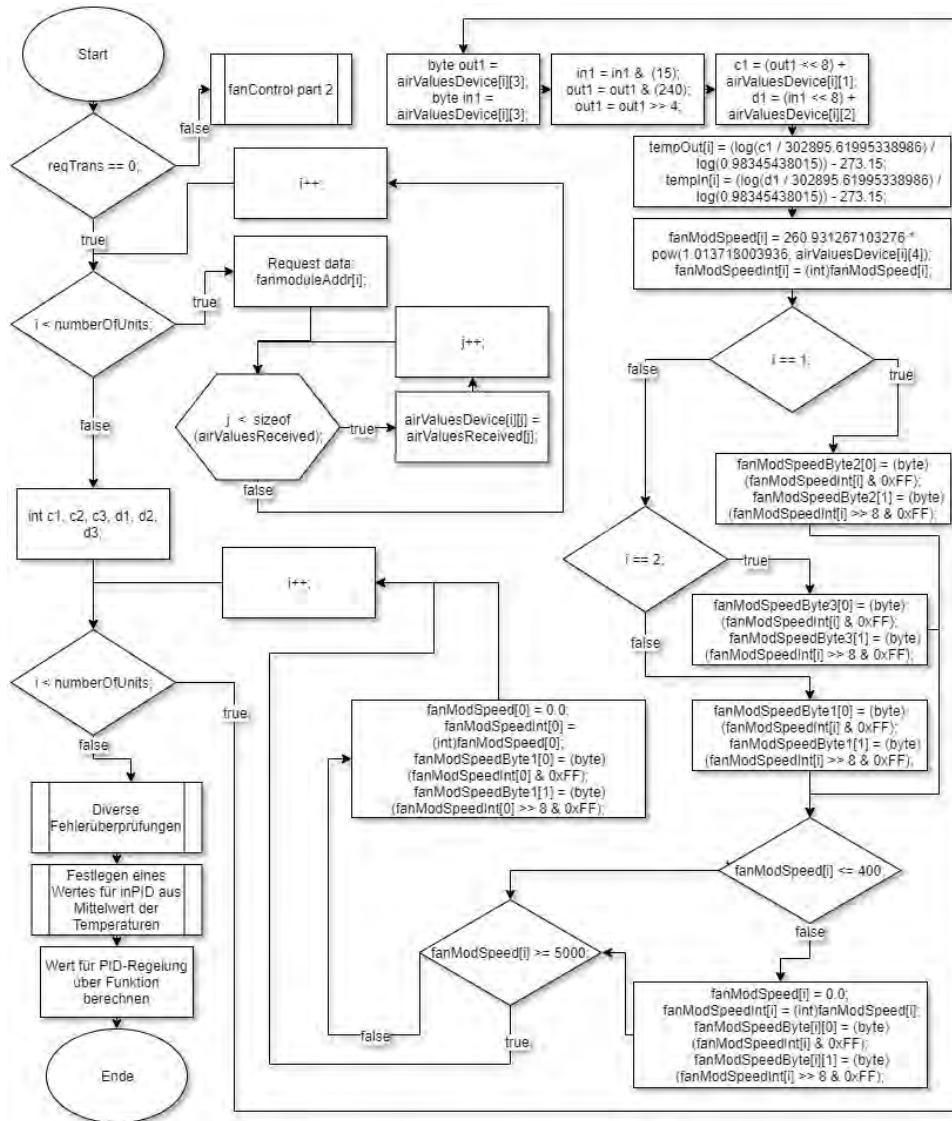


Abbildung 6.4: Programmablaufplan der Funktion zum Kommunizieren mit einem Lüftersteuergerät: Empfangen von Daten

Wie auf der Abbildung 6.4 zu sehen ist, nimmt die Lesbarkeit eines Ablaufplanes ab, sobald verschachtelte Schleifen vorhanden sind. Die Schleifen haben die Funktion den Programmieraufwand zu mindern, indem die Daten der Steuergeräte in drei Durchläufen dieser Schleife abgefragt werden.

Zunächst erfolgt eine Abfrage von reqTrans. Diese Variable gibt die Art der Kommunikation an. Entweder werden Daten ausgelesen oder auf die Geräte geschrieben. Besitzt reqTrans den Wert null, so werden Daten ausgelesen.

Ist der Wert ausgelesen, so wird eine Schleife gestartet, die drei Mal durchläuft. Für jedes Steuergerät wird ein Durchlauf gestartet, der zum Auslesen und Zuordnen der Daten dient. Zu Beginn jedes Schleifendurchlaufs wird das, zum Index passende, Steuergerät ausgelesen und in einer zugehörigen Schleife den airValuesDevice mit dem richtigen Index zugewiesen.

Nach dem Durchlaufen der verschachtelten Schleifen wird die bereits beschriebene Berechnung der Temperatur und die Berechnung der Lüfterdrehzahl durchgeführt. Diese ist recht unübersichtlich, da viele Rechenoperationen stattfinden.

Nach der Initialisierung der Temperaturvariablen startet eine Schleife, die für drei Durchläufe die Werte des dazugehörigen Steuergerätes berechnet. Es wird die Temperatur berechnet und daraufhin, über Abfragen zum Index, die jeweilige Lüfterdrehzahl bestimmt. Der Wert wird anschließend geprüft, der Index erhöht und die Schleife startet erneut.

Nach dem Durchlaufen der Schleife werden Fehler überprüft und ein Wert für die Berechnung in der PID-Funktion ermittelt, welcher im Anschluss berechnet wird.

Danach endet die Funktion.

Die folgende Abbildung zeigt das Ablaufdiagramm zum Senden von Daten. Durch die Berechnung des PID-Wertes beim Empfangen, ist diese deutlich einfacher aufgebaut.

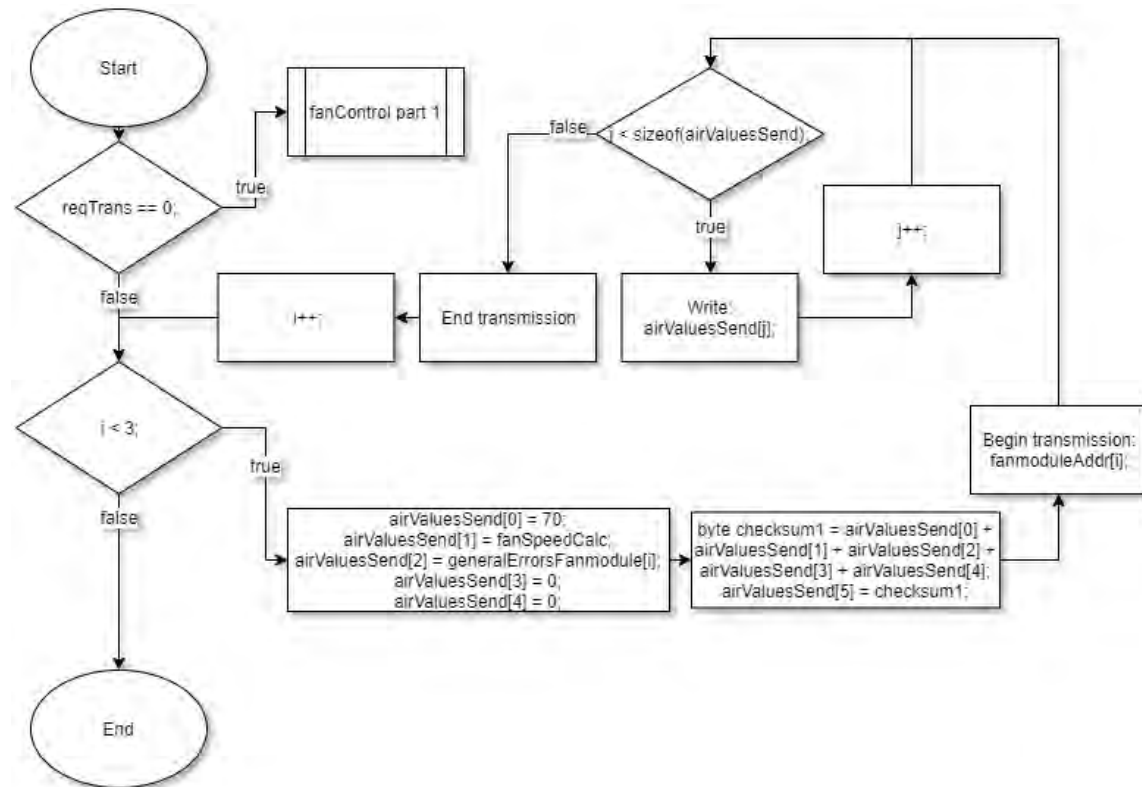


Abbildung 6.5: Programmablaufplan der Funktion zum Kommunizieren mit dem Lüftersteuergerät: Senden von Daten

Beträgt der Wert von reqTrans 1, so werden Daten an das Lüftermodul gesendet. In einer Schleife für die drei Steuergeräte, werden jeweils die passenden, indizierten Daten zwischengespeichert, die Checksumme ermittelt und eine Datenübertragung an die jeweilige Adresse gestartet. Die Übertragung der Daten erfolgt mittels Schleife.

Wurden alle Daten gesendet, so wird die Bedingung der Schleife nicht mehr erfüllt und die Übertragung wird beendet. Daraufhin erhöht sich der Index zur Adressierung der Geräte und die Schleife beginnt erneut. Wurden alle Daten übertragen endet die Funktion.

6.3 Sicherheitsfeatures

Dieses Kapitel behandelt die vorhandenen Sicherheitssysteme. Diese wurden im letzten Kapitel nur angeschnitten, um die Übersichtlichkeit zu verbessern.

Ein wichtiger Teil des Sicherheitssystems ist die gegenseitige Überwachung der Prozessoren. Diese lassen einen Zähler laufen, der sich immer dann zurücksetzt, sobald eine Kommunikation mit dem Gegenüber stattfand. Dieser Vorgang erkennt Probleme relativ genau, da die jeweiligen Prozessoren mittels Watchdog überwacht werden. Bringt ein Reset durch den Watchdog keinen Erfolg, so erkennt der andere Prozessor diesen Fehler, da weiterhin keine Kommunikation hergestellt wird.

Wie oben erwähnt wird ein Watchdog genutzt der den Prozessor neu startet sobald dieser sich aufhängt. Dies gewährleistet die Funktion im Falle eines Bitfehlers durch ionisierende Strahlung. Wurde der Speicherstand beschädigt kann der Watchdog jedoch nicht zur Lösung des Problems beitragen.

Fällt ein Prozessor aus, so erkennt der andere Prozessor diesen Fehler und schaltet in den "Singlecore-Modus" um. Dieser Modus ändert beim sekundären Prozessor die Einträge im Log und setzt den Busy-Ausgang auf einen Low-Pegel. Das Ausschalten des Ausgangs bezweckt eine mögliche Kommunikation mit dem primären Prozessor, falls dieser wieder funktionieren sollte. Dies ist sinnvoll, falls sich der primäre Prozessor durch einen sporadischen Netzwerkfehler aufhängt. Das System wird weiterhin durch den sekundären Prozessor geregelt und die Temperatur wird gehalten. Eine Lösung des Problems muss somit nicht zeitnah gefunden werden.

Von außen kann dieser Fehler erkannt werden, wenn das System nicht mehr über Netzwerk oder SNMP antwortet.

Falls der sekundäre Prozessor ausfällt, schaltet der primäre Prozessor ebenfalls in diesen Modus. Es ändert sich die Zugriffszeit über SNMP und Netzwerk, da der Prozessor die Regelung der anderen Steuergeräte übernimmt. In diesem Fall blockiert der Prozessor die Kommunikation nach außen, während die Daten der Steuergeräte ausgelesen oder Daten an diese gesendet werden. So kommt es zu keinen Unterbrechungen bei der Regelung. Das Webinterface oder das Abfragen über SNMP hängt sich jedoch eventuell, bis zur nächsten Aktualisierung der Daten, auf.

Es wird eine Anfrage in definierten Zeitabständen an den sekundären Prozessor gesendet, für den Fall, dass dieser wieder läuft.

Der primäre Prozessor erkennt einen Fehler seines Gegenübers, indem dieser die Zeit zählt, in der sich der Slave im beschäftigten Zustand befindet. Dies geschieht über das

Auslesen des Busy-Eingangs. Zusätzlich wird geprüft, ob dieser antwortet, wenn ein Request gesendet wird.

Der Fehler wird über das Webinterface und SNMP ausgegeben.

Fällt das gesamte Steuergerät aus, greift ein bereits vorhandener Sicherheitsmechanismus, welcher die Lüfter automatisch auf Vollast laufen lässt und den Regelkugelhahn voll öffnet.

6.4 Graphische Nutzeroberfläche über Webinterface

Bei dem Webinterface handelt es sich um eine einfache, in HTML programmierte Oberfläche. Da das Gerät in einem sicheren Netzwerk steht, konnte auf einen Zugriff per Passwort verzichtet werden. In der folgenden Abbildung wird die simple Oberfläche gezeigt.

```
CMC-TC self-development v0.7  
Date: 11/7/2021 Time: 10:02:17 IP: 192.168.169.62 MAC: 0.D0.93.28.E0.41  


---

System overview  
CPU Status:  
CPU 1: OK  
CPU 2: OK  
Power supply:  
5V value: 5.00 V  
3.3V value: 3.30 V  


---

Errors  
Water module errors:  
No watermodule errors detected.  
Fan module errors:  
No fanmodule errors detected.  


---

Control unit data  
Setpoint: 23 °C  
Flowmeter: No flowmeter installed  
  
Data of water module:  
Temp In: 18 °C  
Temp Out: 20 °C  
Valve opened: 100.00 %  
Data of fan module 1: Data of fan module 2: Data of fan module 3:  
Temp 1 In: 21 °C Temp 2 In: 21 °C Temp 3 In: 21 °C  
Temp 1 Out: 24 °C Temp 2 Out: 23 °C Temp 3 Out: 23 °C  
Fan 1 rpm: 522 Fan 2 rpm: 515 Fan 3 rpm: 522  


---


```

Abbildung 6.6: Darstellung der Daten auf der Webseite, oberer Teil

Wie bereits zu sehen ist, sind alle wichtigen Daten vorhanden. Es wird der Systemstatus angezeigt und darunter die aktuellen Fehler.

In dem nächsten Abteil werden die aktuellen Temperatur- und andere wichtige Systemwerte der anderen Steuergeräte angezeigt.

In der nächsten Abbildung wird die untere Hälfte des Webinterfaces gezeigt. Zur Übersichtlichkeit wurde die zweite Hälfte der Logs gekürzt.

Switch additional relay (one switching operation at a time)

Relay 1: **OFF** Relay 2: **OFF** Relay 3: **OFF**

ON ON ON

OFF OFF OFF

Last 24 logs

09/7/2021 22:59:58 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,522,522 Errors: 0,0,0,0,0,0,0,0
09/7/2021 23:59:39 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,515,522 Errors: 0,0,0,0,0,0,0,0
10/7/2021 00:59:52 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,515,522 Errors: 0,0,0,0,0,0,0,0
10/7/2021 01:59:33 Temps: 18,21,21,21,20,24,23,23 Speeds: 529,515,522 Errors: 0,0,0,0,0,0,0,0
10/7/2021 02:59:45 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,522,522 Errors: 0,0,0,0,0,0,0,0
10/7/2021 03:59:57 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,515,515 Errors: 0,0,0,0,0,0,0,0
10/7/2021 04:59:39 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,515,522 Errors: 0,0,0,0,0,0,0,0
10/7/2021 05:59:51 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,515,522 Errors: 0,0,0,0,0,0,0,0
10/7/2021 06:59:33 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,515,522 Errors: 0,0,0,0,0,0,0,0
10/7/2021 07:59:44 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,522,522 Errors: 0,0,0,0,0,0,0,0
10/7/2021 08:59:57 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,515,515 Errors: 0,0,0,0,0,0,0,0
10/7/2021 09:59:39 Temps: 18,21,21,21,20,24,23,23 Speeds: 522,515,522 Errors: 0,0,0,0,0,0,0,0
...

Refresh data

Configuration of flowmeter and setpoint

Flowmeter: **not installed** Change setpoint here:

installed setpoint value (18-30):

not installed

Abbildung 6.7: Darstellung der Daten auf der Webseite, unterer Teil, gekürzt

Unter den Systemwerten befinden sich hier die drei Schaltflächen zum Schalten der zusätzlichen Relais. Darunter befinden sich die Logs der letzten 24 Stunden, welche in dieser Version manuell über Refresh data auf den aktuellen Stand gebracht werden müssen. Die weiteren 12 Logs wurden zur Übersichtlichkeit durch drei Punkte ersetzt.

Der Aufbau der Logs besteht aus Datum, Zeit und Temperaturen in folgender Reihenfolge: einmal Wasser eingehend, dreimal Luft eingehend, einmal Wasser ausgehend, dreimal Luft ausgehend. Darauf folgen die Lüfterdrehzahlen in richtiger Reihenfolge und darauf eine festgelegte Reihenfolge von Daten aus dem Fehlerspeicher.

Unter den Logs befindet sich eine Konfigurationsoberfläche. Dort kann per Mausklick der Durchflussmesser eingestellt, oder der Setpoint verändert werden.

7 Überprüfung und Test

7.1 Ergebnis der Testphase

Zur Zeit befindet sich das Steuergerät in einem sonst leeren Schaltschrank in der Nähe eines Strahlstoppers. Dieser Test läuft seit einigen Wochen und brachte bereits einige Erkenntnisse über Lücken in der Software und Probleme bei der Kommunikation mit den anderen Steuergeräten.

Die Testphase ist sehr wichtig, da so Fehler erkannt werden, an die beim Debuggen nicht gedacht wurde und welche nur durch eine lange Laufzeit und die Umstände in der Umgebung auftreten können. Zudem können die Betriebsbedingungen im Büro nicht simuliert werden, um die Funktion des Steuergerätes zu überprüfen.

Zunächst fiel auf, dass die Log-Daten nicht vernünftig gespeichert wurden. Die Regelung funktionierte jedoch tadellos und auch die Kommunikation über Ethernet lief sehr gut. Durch einen falsch programmierten Auslesevorgang des Log-EEPROMs kam es zu Fehlern beim Speichern und es wurden Adressen übersprungen. Die übersprungenen Adressen führten zu einer Ausgabe der Logs mit den gespeicherten Daten und sich drei dazwischen befindlichen Zeilen mit nicht definierten Zahlen. Dieser Fehler wurde bereits behoben.

Ein weiterer Fehler war die zu scharf eingestellte gegenseitige Kontrolle der Prozessoren. Da der Master-Prozessor bei Anfragen über Ethernet einen zeitlich verzögerten Request an den Slave-Prozessor sendete, sprang der sekundäre Prozessor kurz in ein Notprogramm. Dieses wurde durch die nachfolgende Überprüfung auf Anwesenheit zwar wieder verlassen, jedoch wurden, durch fehlendes Zurücksetzen einer Variable, Logs auch von diesem Prozessor gespeichert.

Dieser Fehler führte zu der Änderung, dass beide Prozessoren stündlich die Logs speichern und beide auf der Webseite ausgegeben werden. Dieses wird noch angepasst.

Die Ausgabe der Logs auf der Webseite führt in seltenen Fällen zu Anzeigefehlern. Diese erscheinen, da die Daten beim Laden der Logs direkt vom Speicher abgelesen werden. Dies führt zur Änderung, dass die Logs automatisch beim Speichern auch in einen Zwischenspeicher zur Ausgabe auf der Webseite geladen werden.

Zeitgleich wird mit dieser Änderung eine Abfrage älterer Logs eingebaut, bei der ein Datum eingegeben werden kann und die Logs dieses Tages ausgegeben werden.

Nach einigen Wochen Betriebszeit stürzte der sekundäre Prozessor ab und kommunizierte nicht mehr mit den anderen Steuergeräten. Die Ursache wird noch im Programmcode nachgesehen, jedoch fiel durch diesen Fehler auf, dass der Watchdog für diesen Prozessor nicht ordnungsgemäß funktionierte. Dies ist in Bearbeitung.

Das Verhalten des Steuergerätes wird noch ein paar Wochen beobachtet. Dies setzt voraus, dass die Softwareupdates mit den Änderungen in der Zwischenzeit aufgespielt werden, um diese Änderungen auf Fehler zu überprüfen.

Wichtig ist auch ein langfristiger Test des SNMP, da dieses durch noch fehlende Kompatibilität nicht ordnungsgemäß funktioniert.

Besteht dieser Prototyp den Test, kann am finalen Produkt gearbeitet werden. Treten während der Produktion der Steuergeräte neue Probleme auf, so kann die Software auch nachträglich angepasst werden. Sind die Geräte jedoch im Tunnel verbaut, so muss für ein Update jedes Steuergerät einzeln auf den neuen Stand gebracht werden. Dies erfolgt über den Ausbau des Steuergerätes aus dem Schaltschrank, öffnen des Gehäuses und dem Überspielen der Software per Programmieradapter.

Ein Update über Ethernet ist nicht möglich.

7.2 Ausblick

Die Entwicklung ist fast beendet und es werden nur noch Optimierungen an der Software durchgeführt. Die Testphase ist sehr erfolgreich und zeigte bereits wichtige Sicherheitslücken und Änderungswünsche auf. Die Testphase wird noch ein paar Wochen weiterlaufen und danach kommt es zur EMV Prüfung und zur Besprechung der CE Prüfung. Wenn diese erfolgreich sind, kommt es zur finalen Planung der Hardware und zur Fertigung der ersten Leiterplatten.

Hinzu kommt der Aufbau einer Vorrichtung zum Überspielen der Bootloader und der Datenstände auf die Prozessoren. Hierfür wird ein Arduino Uno mit einer Leiterplatte verbunden und die jeweiligen Sockel für die Mikroprozessoren und den EEPROM angeschlossen. Über den PC können so die Daten übertragen und die alten Datenstände

eines ausgebauten EEPROMs ausgelesen werden. Es werden Sockel gewählt, in die die Bauteile leicht eingesetzt und herausgenommen werden können.

Bevor die Leiterplatten in die finale Phase gehen können, muss der Netzwerkadapter umgebaut und getestet werden. Nach dem erfolgreichen Test kann die Netzwerkanbindung fest in das Layout integriert und die finale Version erstellt werden.

Während sich die Leiterplatten in der Produktion befinden, werden die Anmerkungen zur Nutzung des Steuergerätes in ein gesamtes Dokument gefasst und alle Funktionen ausführlich beschrieben. Durch dieses Dokument sollen auch nicht unterwiesene Nutzer das Steuergerät bedienen und betreuen können.

8 Fazit

Zusammenfassend lässt sich sagen, dass die Planung und Entwicklung eines elektronischen Bauteils in diesem Umfang eine komplexe Aufgabe ist. Das erlernte Wissen des Studiums wird durch die praktische Anwendung vertieft und zeigt den Sinn hinter all diesen erworbenen Grundlagen. Durch die Entwicklung des neuen Steuergerätes ergibt sich eine Win-Win-Situation für DESY und dem Studenten, der dieses Projekt durchgeführt hat.

Zum Aufbau dieser Thesis lässt sich sagen, dass die Gliederung einen roten Leitfaden bietet, der zum Verständnis der Vorgänge beiträgt. Der Theorieteil mag den Lesefluss zwar unterbrechen, jedoch sind die einzelnen Themen kurz und bündig zusammengefasst. Dieser Teil ist aber wichtig, um einigen Überlegungen in der Vorbereitung und auch in der Umsetzung der Lösung gut folgen zu können. Der Seitenumfang ist so groß, da dieses Wissen aus sehr vielen verschiedenen Themengebieten stammt. Ein Verweis bei dem ersten Kontakt mit der eventuell unbekanntem Theorie führt den Leser zu den benötigten Informationen.

Die Theorie befindet sich zwischen der Anforderungsanalyse und der Ergebnisvorstellung, damit es zu einem besseren Verständnis in den Kapiteln der Umsetzung kommt.

Zum Projekt selber lässt sich sagen, dass dieses zum jetzigen Zeitpunkt zur vollsten Zufriedenheit umgesetzt wurde. Da sich der Prototyp in der Testphase befindet und mit Hilfe einiger Anpassungen in der Software so läuft, wie es in der Aufgabenstellung gefordert war. Der Großteil der Entwicklungsphase ist abgeschlossen. Das Problem mit der Kompatibilität des SNMP ist das einzige Problem, das noch behoben werden muss.

Die Prüfung der elektromagnetischen Verträglichkeit wird keine Probleme verursachen. Dank der vorausschauenden Positionierung von Entstörkondensatoren ist das System vor Spannungsschwingungen geschützt. Außerdem werden die ausgewählten Bauteile in vielen anderen, geprüften Elektrogeräten bereits verwendet.

Ist diese Prüfung abgeschlossen, wird der CE-Prüfung ebenso nichts im Wege stehen, damit das Gerät sorgenfrei eingesetzt werden kann.

Durch das Eigenständige herausuchen der zu verwendenden Bauteile und dem Eigen-

ständigen Bestellen, lässt sich der Kostenaufwand gut abschätzen. Es wird somit, trotz der langen Entwicklungszeit, ein guter und günstiger Ersatz zu der bestehenden Elektronik entwickelt, so wie es vom Auftraggeber gefordert wurde.

Zur Durchführung des Projektes lässt sich sagen, dass es bereits sehr interessant ist die Entwicklung eines Produktes von außen zu verfolgen, wie es in interessanten Dokumentationen geschieht. Jedoch ist es umso spannender dieses Produkt selbst zu planen, zu entwickeln und letztendlich herzustellen. Der Arbeitsaufwand hätte jedoch durch Aufteilung der einzelnen Bereiche auf mehrere Personen gemindert und somit die Entwicklung beschleunigt werden können, jedoch sollte dieses Projekt von einer Person durchgeführt werden um den Lerneffekt zu steigern.

Die eigenen Ansprüche, die Steuerungselektronik im alten Gehäuse zu installieren, halfen bei der Ideenfindung und gaben einen Ansporn das Projekt schnell zum Punkt der Konfiguration der Hardwarekomponenten voranzutreiben. So konnten nicht einfach irgendwelche Bauteile verwendet werden, sondern es musste auf die Platzverhältnisse geachtet werden.

Die Umsetzung von eigenen Sicherheitssystemen war außerdem sehr interessant, da zunächst ein Konzept erstellt und dies in den bereits geplanten Programmablauf eingepflegt werden musste. Das Überprüfen der Funktion per Debugger gab viele Einblicke in die Funktion der Software. Dies hilft beim besseren Verständnis der internen Datenübertragung.

Es wurde klar, dass sich der Wirkungsgrad der Arbeit durch eine gute Vorbereitung steigern lässt. Programmablaufpläne können das Programmieren deutlich vereinfachen und eine Grundstruktur bieten. Das Programm kann dann, durch Verfeinern des Ablaufplanes, einfach umgesetzt werden und es entstehen weniger Fehler durch Anpassung mitten im Programm. Außerdem können Änderungen so besser eingepflegt werden.

Es zeigte sich außerdem, dass die gewählte Art der Umsetzung richtig ist und vollständig umgesetzt werden kann.

Der Ablauf dieses Projektes führte zu einer erfolgreichen Entwicklung, wurde jedoch, durch die äußeren Umstände der aktuell herrschenden Pandemie, deutlich verlangsamt.

Literaturverzeichnis

- [1] ANGERMEIER, Dr. Georg: *Reverse Engineering*. 2017. – URL <https://www.projektmagazin.de/glossarterm/reverse-engineering>. – Zugriffsdatum: 2021-07-15
- [2] ARDUINO.CC: *ARDUINO UNO REV3*. 2021. – URL <https://store.arduino.cc/arduino-uno-rev3>. – Zugriffsdatum: 2021-07-20
- [3] AUGSTEN, Stephan: *Was ist ein Debugger?* 2018. – URL <https://www.dev-insider.de/was-ist-ein-debugger-a-730931/>. – Zugriffsdatum: 2021-07-29
- [4] BEHREND, Luise: *Leiterplattenherstellung*. 2017. – URL <https://service.projektlabor.tu-berlin.de/wordpress/wp-content/uploads/sites/19/2017/05/7973-Leiterplattenherstellung-pdf.pdf>. – Zugriffsdatum: 2021-07-23
- [5] BUNDESAMT FÜR STRAHLENSCHUTZ: *Was ist ionisierende Strahlung?* 2019. – URL <https://www.bfs.de/DE/themen/ion/einfuehrung/einfuehrung.html>. – Zugriffsdatum: 2021-07-28
- [6] BÖTTCHER, Sarah: *Was ist ein Flash-Speicher?* 2018. – URL <https://www.it-business.de/was-ist-ein-flash-speicher-a-818012/>. – Zugriffsdatum: 2021-07-28
- [7] DE-ACADEMIC.COM: *Bibliothek (Programmierung)*. 2021. – URL <https://de-academic.com/dic.nsf/dewiki/168872>. – Zugriffsdatum: 2021-07-30
- [8] DE-ACADEMIC.COM: *Hohlraumresonator*. 2021. – URL <https://de-academic.com/dic.nsf/dewiki/622486>. – Zugriffsdatum: 2021-07-28
- [9] DERVIS GÜNAYDIN, Dipl.-Ing. (FH) Hendrik H.: *Mit diesen Entstörungsmaßnahmen lassen sich typische EMV-Probleme vermeiden*. 2008. – URL <https://www.elektronikpraxis.vogel.de/mit-diesen-entstoerungsmassnahmen-lassen-sich-typische-emv-probleme-vermeiden-a-116691/>. – Zugriffsdatum: 2021-08-01

- [10] DEUTSCHES ELEKTRONEN-SYNCHROTRON: *Pressemeldung PM 7.10.09*. 2009. – URL https://www.desy.de/e428/e548/e4802/e11177/e84364/index_ger.html. – Zugriffsdatum: 2021-07-10
- [11] DEUTSCHES ELEKTRONEN-SYNCHROTRON: *60 Jahre DESY: Vom Hamburger Teilchenbeschleuniger zum globalen Forschungszentrum*. 2019. – URL https://www.desy.de/aktuelles/news_suche/index_ger.html?openDirectAnchor=1754&two_columns=0. – Zugriffsdatum: 2021-07-02
- [12] DEUTSCHES ELEKTRONEN-SYNCHROTRON: *Die Entschlüsselung der Materie*. 2020. – URL https://www.desy.de/ueber_desy/desy/index_ger.html. – Zugriffsdatum: 2021-07-02
- [13] DEUTSCHES ELEKTRONEN-SYNCHROTRON: *HERA - Teilchenphysikexperimente H1, ZEUS und HERMES*. 2020. – URL https://www.desy.de/forschung/anlagen__projekte/hera/index_ger.html. – Zugriffsdatum: 2021-07-02
- [14] DIRK FIEDLER, Dipl.-Ing. (FH) Hendrik H.: *EMV-Test: Unverträglichkeiten frühzeitig aufdecken*. 2020
- [15] ELECTRONICSPLANET.CH: *EMV Kopplungsarten*. 2021. – URL <https://www.electronicspanet.ch/emv/emv-kopplungsarten.php>. – Zugriffsdatum: 2021-08-01
- [16] EURO CIRCUITS: *Herstellung einer Leiterplatte – Schritt für Schritt erklärt*. 2021. – URL <https://www.eurocircuits.de/lehrfilme/>. – Zugriffsdatum: 2021-07-24
- [17] EUROPEAN XFEL: *Lichtquellen*. 2021. – URL https://www.xfel.eu/anlage/lichtquellen/index_ger.html. – Zugriffsdatum: 2021-07-09
- [18] EUROPEAN XFEL: *Virtuelle Tour durch den European XFEL*. 2021. – URL <https://www.xfel.eu/virtualtour/>. – Zugriffsdatum: 2021-07-10
- [19] EUROPÄISCHE UNION: *DIRECTIVE 2009/24/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL*. 2009. – URL <https://eur-lex.europa.eu/eli/dir/2009/24/oj>. – Zugriffsdatum: 2021-07-15
- [20] EVERYSPEC.COM: *MIL-STD-883L*. 2019. – URL http://everyspec.com/MIL-STD/MIL-STD-0800-0899/MIL-STD-883L_56323/. – Zugriffsdatum: 2021-07-22

- [21] EVISION-WEBSHOP.DE: *I²C*. 2021. – URL <https://evision-webshop.de/Total-Phase-Knowledge-Base/Promira/I2C-vs-SPI-Protokoll-Analysatoren-Unterschiede-und-Aehnlichkeiten>. – Zugriffsdatum: 2021-07-21
- [22] FERNANDEZ, Enrique: *I2C-Bus*. 2021. – URL <http://fmh-studios.de/theorie/informationstechnik/i2c-bus/>. – Zugriffsdatum: 2021-07-21
- [23] GREGOR HEBERMEHL, Thomas H.: *US-Behörde ordnet nach Tesla-Zögern Rückruf an*. 2021. – URL <https://www.auto-motor-und-sport.de/elektroauto/tesla-probleme-flashspeicher-rueckruf-nhtsa-reparatur-kostenlos/>. – Zugriffsdatum: 2021-07-17
- [24] HOFFMANN, Markus: *Ein einfaches Strahlungsalarm-System für den XFEL Main Linac*. 2018. – URL <https://bib-pubdb1.desy.de/record/407786/files/DESY-Technical-Note-18-01-1.00%20%282%29.pdf>. – Zugriffsdatum: 2021-07-22
- [25] HTTP://WWW.LHC-FACTS.CH: *Beam Dump*. 2021. – URL <http://www.lhc-facts.ch/index.php?page=beamdump>. – Zugriffsdatum: 2021-07-28
- [26] IONOS.DE: *Bootloader: Alle Informationen zum Manager des Systemstarts*. 2020. – URL <https://www.ionos.de/digitalguide/server/konfiguration/was-ist-ein-bootloader/>. – Zugriffsdatum: 2021-07-20
- [27] IONOS.DE: *RAM (Arbeitsspeicher): Was steckt dahinter?* 2020. – URL <https://www.ionos.de/digitalguide/server/knowhow/was-ist-ein-arbeitsspeicher/>. – Zugriffsdatum: 2021-07-20
- [28] LACKES, Prof. Dr. Richard: *Programmiersprache*. 2021. – URL <https://wirtschaftslexikon.gabler.de/definition/programmiersprache-42635>. – Zugriffsdatum: 2021-07-29
- [29] LANDSCHAFT DER FORSCHUNGSINFRASTRUKTUREN: *European XFEL – Ein Röntgenlaser der Superlative*. 2020. – URL <https://fis-landschaft.de/materie/european-xfel/>. – Zugriffsdatum: 2021-07-08
- [30] LEITENBERGER, Bernd: *Was ist der Vorteil eines 64 Bitters?* 2013. – URL <https://www.bernd-leitenberger.de/bitbreiten-bei-prozessoren.shtml>. – Zugriffsdatum: 2021-07-22

- [31] LEITENBERGER, Bernd: *Strahlenbelastung von Satelliten und Raumsonden*. 2016. – URL <https://www.bernd-leitenberger.de/strahlung.shtml>. – Zugriffsdatum: 2021-07-22
- [32] LERNEPROGRAMMIEREN.DE: *Übersicht über die Programmiersprachen (2021)*. 2021. – URL <https://lerneprogrammieren.de/uebersicht-ueber-die-programmiersprachen/>. – Zugriffsdatum: 2021-07-29
- [33] LUBER, Dipl.-Ing. (FH) Stefan: *Was ist eine IDE?* 2017. – URL <https://www.dev-insider.de/was-ist-eine-ide-a-600703/>. – Zugriffsdatum: 2021-07-28
- [34] LUBER, Dipl.-Ing. (FH) Stefan: *Was ist ein EEPROM?* 2019. – URL <https://www.storage-insider.de/was-ist-ein-eprom-a-831590/>. – Zugriffsdatum: 2021-07-28
- [35] MIKROCONTROLLER.NET: *AVR-GCC-Tutorial/Der Watchdog*. 2021. – URL https://www.mikrocontroller.net/articles/AVR-GCC-Tutorial/Der_Watchdog. – Zugriffsdatum: 2021-07-20
- [36] MIKROCONTROLLER.NET: *Richtiges Designen von Platinenlayouts*. 2021. – URL https://www.mikrocontroller.net/articles/Richtiges_Designen_von_Platinlayouts. – Zugriffsdatum: 2021-07-23
- [37] MILATZ, Harry: *Platine ätzen*. 2019. – URL <https://www.all4hardware4u.de/platinenherstellung/platine-aetzen/>. – Zugriffsdatum: 2021-07-24
- [38] MÜLLER-ELSNER, Heiner: *Der European XFEL-Beschleuniger während seiner Konstruktion im Tunnel*. 2021. – URL https://www.xfel.eu/anlage/beschleuniger/index_ger.html. – Zugriffsdatum: 2021-07-27
- [39] NIEDERMEIER, Thomas: *SNMP Grundlagen*. 2021. – URL https://www.thomas-krenn.com/de/wiki/SNMP_Grundlagen. – Zugriffsdatum: 2021-07-22
- [40] OPENSTREETMAP.DE: *Weltkarte*. 2021. – URL <https://www.openstreetmap.de/karte.html#>. – Zugriffsdatum: 2021-07-10
- [41] PRESSEBOX: *Rittal sorgt für Beschleunigung im Supertunnel*. 2013. – URL <https://www.pressebox.de/pressemitteilung/rittal-gmbh-co-kg/Rittal-sorgt-fuer-Beschleunigung-im-Supertunnel/boxid/622266>. – Zugriffsdatum: 2021-07-08

- [42] PROTONNEWS: *Unsere Top 10 der besten Softwareprogramme für die Entwicklung von Leiterplatten.* 2021. – URL <https://www.proto-electronics.com/de/blog/top-10-der-besten-softwareprogramme-entwicklung-von-leiterplatten>. – Zugriffsdatum: 2021-07-23
- [43] RITTAL: *Montage- und Bedienungsanleitung, CMC-TC Processing Unit II.* DK 7320.100. Herborn, Deutschland: , 08 2006. – Document: A29290 21 IT 74
- [44] RITTAL: *Rittal - Liquid Cooling Package, Betriebs- und Wartungsanleitung.* Herborn, Deutschland: , 10 2007
- [45] RITTAL: *LCP DESY (XFEL-Tunnel), Montage-, Installations- und Bedienungsanleitung.* Herborn, Deutschland: , Februar 2014
- [46] RITTAL: *Wir über uns, Komplexes einfach machen.* 2021. – URL <https://www.rittal.com/de-de/Unternehmen/Portraet/Wir-ueber-uns>. – Zugriffsdatum: 2021-07-09
- [47] ROHDE UND SCHWARZ: *EMV-Konformitätstests.* 2021. – URL https://www.rohde-schwarz.com/de/loesungen/test-and-measurement/emc-testing/emc-compliance/emv-konformitaet_253444.html. – Zugriffsdatum: 2021-08-01
- [48] SCHWERDTFEGER, Martin: *SPI - Serial Peripheral Interface.* 2000. – URL <https://web.archive.org/web/20190116220910/http://www.mct.de/faq/spi.html>. – Zugriffsdatum: 2021-07-23
- [49] SCHÖRNER-SADENIUS, Thomas: *Wie funktioniert ein Hohlraumresonator?* 2010. – URL https://www.weltmaschine.de/neuigkeiten/ask_an_expert/hohlraumresonator/. – Zugriffsdatum: 2021-07-27
- [50] T2INFORMATIK.DE: *Compiler - Was ist ein Compiler und welche Aufgaben hat er?* 2021. – URL <https://t2informatik.de/wissen-kompakt/compiler/>. – Zugriffsdatum: 2021-07-29
- [51] WWW.CHEMIE.DE: *Niob.* 2021. – URL <https://www.chemie.de/lexikon/Niob.html>. – Zugriffsdatum: 2021-07-28
- [52] XOVI.DE: *Was ist ein Interpreter?* 2021. – URL <https://www.xovi.de/was-ist-ein-interpreter/>. – Zugriffsdatum: 2021-07-29

A Anhang

Der Anhang zur Thesis befindet sich in digitaler Form auf der beiliegenden CD, welche bei der Prof. Dr. Annabella Rauscher-Scheibe oder beim Fakultätsservicebüro der Fakultät Technik und Informatik eingesehen werden kann.

Folgende Daten sind enthalten:

- Dateien zum Programmcode:
 - CMC-TC_v0.6_M.txt
 - CMC-TC_v0.6_S.txt
 - arduinoSetup.h.txt
- Datei zu den Schaltplänen:
 - Schaltpläne_Zimmermann_2359328.pdf
- Thesis in digitaler Form:
 - Bachelorthesis_Zimmermann_2359328.pdf

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original