

Entwicklung und testen eines Ultraschall-Entfernungsmessers als Vorbereitung eines Produktentwurfes

Projektarbeit

erstellt an der
Fachschule für Technik des Carl-Severing-Berufskollegs
für Metall- und Elektrotechnik der Stadt Bielefeld



Erstellt durch:

Eduard Meiser
Omar Hachimi
Stephan Dannat
FET6A

in Zusammenarbeit mit der Fa. Tinkerforge
betreut durch
Herr Simon

Bielefeld, 26. Februar 2018

Persönliche Erklärung

Hiermit bestätigen wir, dass die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt wurden. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internet-quellen) entnommen sind, wurden unter Angabe der Quellen kenntlich gemacht.

Ort, Datum

Eduard Meiser

Ort, Datum

Omar Hachimi

Ort, Datum

Stephan Dannat

Inhaltsverzeichnis

1	Lasten und Pflichtenheft	1
1.1	Lastenheft	1
1.1.1	Über Tinkerforge	1
1.1.2	Motivation	1
1.1.3	Aufgabenbeschreibung	1
1.2	Lösungsansatz	2
2	Management des Projektes	4
2.1	Trello	4
2.2	Github	4
3	Recherche der Funktionsweise	5
3.1	Sender	5
3.2	H-Brücke	5
3.3	Empfänger	5
3.4	Hochsetzsteller	6
3.5	Mikroprozessor	6
4	Wahl sowie alternative Bauteile	7
5	Open Source CAD-Programm Packet zur Erstellung von elektronischen Leiterplatten	8
5.1	Schaltplan	8
5.2	Platinenlayout	8
6	Entwicklung der Software zum Betrieb des Prototypen	10
6.1	Benötigte Register	10
6.1.1	Durchgeführte Berechnungen	10
6.1.2	Einstellung im Programm	10
6.2	Quellcodeentwurf	10
7	Messungen und Auswertung der Ergebnisse	13
7.1	PWM-Ausgabe	13
7.2	H-Brücke	14
7.3	Signalverlauf nach Verstärkung durch den OPV	14
7.4	aufgetretene Probleme	14
8	Ergebnisse der Forschung	15
9	Fazit aus den Ergebnissen für den Auftraggeber	16
10	Reflektion über den Projektablauf	17
11	Anhänge	18
11.1	Schaltpläne	18
11.2	Platinenlayout	18
11.3	Quellcode	18

1 Lasten und Pflichtenheft

1.1 Lastenheft

1.1.1 Über Tinkerforge

Die Tinkerforge GmbH wurde Ende 2011 mit dem Ziel gegründet, die Handhabung eingebetteter Systeme zu vereinfachen. Das Tinkerforge Baukastensystem besteht aus aktuell fast 80 verschiedenen Modulen, die vom Anwender flexibel für die jeweilige Aufgabe kombiniert werden können. Zu den Modulen zählen diverse Sensor- Aktor- und Schnittstellenmodule, die alle über Hochsprachen wie C#, Python und Java gesteuert werden können. Tinkerforge unterstützt aktuell 17 verschiedene Programmiersprachen. Sowohl Hardware als auch die Software aller Module sind OpenSource. Die Stärke des Baukastensystems ist aus Anwendersicht die enorme Flexibilität, die Einfachheit und die Schnelligkeit mit der Projekte realisiert werden können. Es eignet sich daher besonders im Bereich Rapid Prototyping. Daher findet das Tinkerforge Baukastensystem Anwendung in vielen Forschungsinstituten, in diversen Entwicklungsabteilungen bekannter Automobilhersteller und Ingenieurbüros.

1.1.2 Motivation

Diese Technikerarbeit soll die Grundlage zur Entwicklung eines Entfernungssensors für das Baukastensystem bilden, der auf einer Ultraschall-Entfernungsmessung basiert. Das Baukastensystem verfügt aktuell über so einen Sensor. Bei diesem handelt es sich aber im wesentlichen um ein zugekauftes Modul, welches nicht die gewünschten Leistungen liefert. Daher soll an einem zu entwerfenden Prototypen Forschung betrieben werden, um eine eigene Lösung entwerfen zu können.

1.1.3 Aufgabenbeschreibung

Innerhalb dieser Arbeit soll der Entwurf eines Prototypen des Entfernungssensors und die damit verbundene Forschung durchgeführt werden. Dabei ist durch Recherche zu erarbeiten, welche Möglichkeiten zur Realisierung zur Verfügung stehen. Durch Messungen am Prototypen soll festgestellt werden, welche dieser Möglichkeiten funktional und finanziell realisierbar sind, um ein eigenes Produkt zu erstellen. Sollte im Anschluss noch die Möglichkeit bestehen, sind die Ergebnisse in ein serienreifes Modul umzusetzen.

Diverse Teilaufgaben sind zu erledigen:

- **Recherche**

Zu Anfang muss recherchiert werden, welche Möglichkeiten es gibt mittels Ultraschall eine Entfernung zu ermitteln und wie diese technisch umgesetzt werden können. Zusätzlich müssen die Techniker sich mit dem Tinkerforge Baukastensystem und seiner internen Funktionsweise vertraut machen.

- **Bauteilauswahl**

Abhängig von der gewählten technischen Umsetzung müssen geeignete Komponenten ausgewählt werden. Die Auswahl sollte auch unter dem Gesichtspunkten Preis, der Bauteilverfügbarkeit und der technischen Anforderungen erfolgen.

- **Schaltplanentwurf und Layouterstellung**

Von Tinkerforge wird das Open Source CAD Programm KiCad verwendet. Mit diesem Programm ist ein Schaltplan für den Prototypen und anschließend ein Leiterplattenlayout zu erstellen.

- **Leiterplattenbestückung**

Die erstellte Leiterplatte wird von Tinkerforge in Auftrag gegeben. Diese muss mit den gewählten Komponenten bestückt werden. Die Tinkerforge GmbH stellt dazu die notwendigen Werkzeuge bereit.

- **Einrichten und Einarbeitung in die Tinkerforge Toolchain**

Viele Softwarekomponenten werden von der Tinkerforge Toolchain automatisch generiert. Um diese Nutzen zu können muss ein Linux System eingerichtet werden. Anschließend muss sich mit der Funktionsweise des Generators und der Softwareversionsverwaltung "Git" vertraut gemacht werden.

- **Testsoftware und Forschung**

Um Messungen an der Hardware durchführen zu können gilt es Programmblöcke zu entwerfen, mit denen die einzelnen Funktionen der Baugruppen getestet werden können. So soll ermittelt werden, wie zum einen das Ultraschallsignal effektiv ausgegeben werden kann und wie sich die Signalamplitude auf die Reichweite auswirkt und zum anderen wie das zurückkommende Signal verarbeitet werden kann. Auch soll erarbeitet werden, wie gut das Signal unter verschiedenen Bedingungen verarbeitet werden kann und ob eine zuverlässige Verarbeitungsqualität ohne großen Aufwand realisierbar ist.

1.2 Lösungsansatz

- Zur Recherche werden alle verfügbaren Quellen genutzt, um vergleichbare, existierende Systeme zu studieren und Datenblätter zu den verwendbaren Bauteilen zu erhalten. Auch werden verschiedene Alternativen verglichen, um eine, nach den vorhandenen Bedingungen, optimale Bauteilauswahl zu gewährleisten. Die Bauteile haben sowohl günstig, als auch qualitativ ausreichend zu sein, um bei den gegebenen Frequenzen schnell genug zu arbeiten und keine Störungen zu erzeugen.
- Zur Einarbeitung in das Open Source CAD Programm KiCad werden bereits vorhandene Pläne studiert und auf Basis dieser eigene Schaltpläne und Platinenlayouts entworfen. So wird durch die Übung sichergestellt, dass beim Entwurf der finalen Schaltpläne die Qualität der Pläne und des Platinenlayouts der gewünschten Norm entsprechen.
- Sobald die Leiterplatte und die benötigten Materialien geliefert werden, wird die Leiterplatte mit Hilfe der vorhandenen Werkzeuge bestückt. Dabei gilt es auf die Bauteilausrichtung und die maximalen Löttemperaturen der Bauteile zu achten. Auch sollten keines der Bauteile einer dauerhaften mechanischen Belastung ausgesetzt werden.
- Einen größeren Aufwand wird die Einarbeitung in die Toolchain, sowie die Programmierung bilden, da an den Schulen nicht die Zeit vorhanden ist, um tief in die komplexe Programmierung eingestiegen zu können, wie sie in derartigen Anwendungen anzutreffen ist. Entsprechend wird es in diesem Bereich viel zu lernen , zu üben, zu korrigieren und zu optimieren geben.
- Nach der Einarbeitung in die Programmierung wird eine Testsoftware zu entwickeln sein, um die verschiedenen Hardwarekomponenten zu testen und um Messungen an den Komponenten bei verschiedenen Zuständen durchführen zu können. Auch gilt es dann die eingehenden Signale mit Messgeräten zu kontrollieren und die Software zur Auswertung dieser anzupassen, um ein möglichst optimales Resultat zu erzielen.

- Als Resultat aus den Messungen gilt es festzulegen, ob und auf welche Weise sich das gewünschte Modul zur Entfernungsmessung realisieren lässt. Mit diesen Grundlagen kann die Tinkerforge GmbH einen Ultraschallsensor für die eigenene Produktpalette entwerfen. Die Entscheidungen werden durch die ausgewerteten Messergebnisse begründet und gefestigt.

2 Management des Projektes

2.1 Trello

Zur zeitlichen Planung und Übersicht des Ablaufes wurde auf das Onlinetool Trello zurückgegriffen. Dieses ist ein kostenfreies, web basiertes Projektmanagementtool. Es ermöglicht den Gruppenmitgliedern gleichzeitig von verschiedenen Orten auf die Oberfläche zuzugreifen und Änderungen vorzunehmen. So kann ein Teilnehmer auch neue Termine mit Kennzeichnung der Fälligkeit für andere Gruppenmitglieder einfügen, oder bereits erledigte Aufgaben für alle abhaken. Auch können hier relevante Dokumente, die alle Gruppenmitglieder lesen sollen hochgeladen, und bei Bedarf noch kommentiert werden. Für die Dokumentation lässt sich an diesem System auch wunderbar abgleichen, zu welchen Zeitpunkten die einzelnen Aufgaben abgeschlossen wurden.

2.2 Github

Bei Github handelt es sich um einen webbasierten Online-Dienst, der Server für Software- Entwicklungsprojekte mit einer Versionsverwaltung bereitstellt. So können alle Daten nach einer Änderung im Programm wieder hochgeladen und mit einem Kommentar versehen werden. Sollte nach mehreren Änderungen ein Problem auftreten, kann einfach auf eine ältere Version zurück gegriffen und so der Fehler eingegrenzt werden. auch kann ein Projekt auf mehrere Abschnitte aufgeteilt werden, damit mehrere Personen unabhängig voneinander daran arbeiten können. Nach der Bearbeitung können die Programmteile wieder zusammengefügt werden. Dabei ist erkennbar, welche Änderungen von wem vorgenommen wurden. So können alle Vorgänge jederzeit verfolgt werden, um eine größtmögliche Übersicht zu gewährleisten. Durch das kommentieren der Änderungen kann die Nachvollziehbarkeit dieser ebenfalls deutlich gesteigert werden. Des weiteren ist diese Plattform gerade für Unternehmen wie Tinkerforge, die ihren Quellcode als Open-Source anbieten besonders reizvoll, da die Nutzer hier schnell an alle Dateien ran kommen.

3 Recherche der Funktionsweise

3.1 Sender

Für den Sender (und empfänger) wurde ein auf Piezomodulen basierendes Kapselgehäuse verwendet. Dabei wurden für den Prototypen mehrere Sender verschiedener Hersteller bestellt, um Unterschiede der verschiedenpreisigen Bauteile zu ermitteln und festzustellen, welches Preissegment die nötige Qualität für die vorliegende Anwendung erfüllt.

3.2 H-Brücke

Der Lautsprecher wird mit einer Frequenz von 40kHz und einer Amplitude von 20V P-P betrieben. Um ein positives sowie negatives Rechtecksignal zu generieren wird eine H-Brücke verwendet. Sie Abb.x Funktion Block Diagramm vom IC A5950

In der Abb. Funktion Block Diagramm vom IC A5950 wird an OUT 1 sowie an OUT 2 eine Stromumkehrung erzielt, durch die Ansteuerung von den MOS-FET von dem Control Logic. Eine genaue Funktionsanalyse ist nicht erforderlich, weil an der H-Brücke nur ein Ultraschallsensor angeschlossen wird. Wichtig für die H-Brücke ist das sie mit 40kHz schalten kann die am Eingang. Ausgründen der internen Beschaltung und Toleranzen sind am Ausgang nicht die vollen 40kHz zu erwarten, sondern ein Verzug was später im Kapitel X.X behandelt wird.

3.3 Empfänger

Abb.1

In der obigen Schaltung(Bildverweis) ist zu sehen dass das ankommende Sinusförmige Signal, verstärkt und in ein digitales Signal umgewandelt wird. Die Schaltung wurde mit einem Hochpassfilter (CR Glied) bestückt bestehend aus C12 und R5 um unerwünschte Signalanteile von unter 40kHz zu unterdrücken. Der Widerstand wurde nach der e24 Reihe, anhand der folgende Berechnung, ausgewählt:

$$C = \frac{1}{2 * \pi * f_g * R} \Rightarrow \frac{1}{2 * \pi * 40kHz * C12 * 100\Omega} \approx 40pF$$

Die Kapazität des Kondensators C12 wurde an die Grenzfrequenz von 40 kHz und den Widerstand angepasst.

Für die Verstärkung der Amplitude so wie der Umwandlung des analog Signals in ein Rechtecksignal mit 40 KHz dienen die Operationsverstärker des Bausteines TLC272. Die Versorgungsspannung der OPV's von 3,3V wird durch den Kondensator C16 (EMV Störfilter) stabilisiert.

Für die Verstärkung der Amplitude ist der Operationsverstärker TLC272 U2B als nicht invertierenden Verstärker geschaltet. Wenn eine Gleichspannung anliegt, wirkt der Kondensator (C10) in der Operationsverstärkerschaltung als Impedanzwandler, also mit einer Verstärkung von eins, geht nun die Eingangsfrequenz hoch, nimmt der widerstand des Kondensators (C10) ab, somit beginnt der Operationsverstärker auch zu verstärken, und zwar mit zunehmender Verstärkung, bis irgendwann die Impedanz des Kondensators vernachlässigt werden kann und die Verstärkung nur noch durch das Verhältnis der Widerstände beeinflusst wird, R6 ist zudem notwendig um das schwingen der Amplitude zu verhindern, somit kann die Verstärkung mit folgender Formel berechnet werden:

$$V_u = R6 + R8 + \frac{R12}{R6}.$$

Die Z-Diode D2 ist für die Spannungsstabilisierung und als Sicherung da.

Für die Umwandlung des Analogen Signales in ein Digitales wurde der Operationsverstärker TLC272 U2C als Komparator geschaltet. Beim auftreten von Differenzen zwischen den eingangs Signalen, wechselt der Ausgang des Komparators zwischen Low (0 Volt) auf High (3,3 Volt).

3.4 Hochsetzsteller

Der Hochsetzsteller dient dazu, aus den 5V Versorgungsspannung eine (für die Versuche variable) höhere Spannung für den Sendebetrieb zu schaffen. So kann der Schalldruck der ausgegeben wird erhöht werden (größere Reichweite/größeres Rücksignal)

Die Funktionsweise des Hochsetzstellers (Spannungspumpe/Aufwärtswandler/Aufwärtsregler) ist relativ simpel und findet in vielen Bereichen Anwendung. Grundsätzlich wird eine Induktivität in Reihe mit einer Freilaufdiode vor einen Ladecondensator geschaltet. Dieser liegt parallel zum Ausgang. Zwischen der Spule und der Diode ist ein Schalter angeschlossen, der die Spule gegen Masse schaltet. So lädt sich die Spule bei Betätigung des Schalters auf (durch den Stromfluss entsteht ein Magnetfeld) und beim Öffnen steigt die Spannung am sekundären Ende der Spule, durch das zusammenbrechende Magnetfeld, an und lädt den Kondensator auf. Dieser Vorgang wird wiederholt, bis der Kondensator so weit aufgeladen ist, dass die Ausgangsspannung den gewünschten Wert hält. Dann wird die Schaltfrequenz auf das mindestmaß verringert, um den Wert zu halten. Natürlich ist die mögliche Ausgangsspannung nicht unbegrenzt über das Schaltspiel regelbar, sondern ist auch von den Baugrößen der Bauteile abhängig. Mit einer Induktivität von 10mH und einem Kapazität von 40uF lässt sich die Ausgangsspannung bei 5v Eingangsspannung zwischen 6v und 20v einstellen.

3.5 Mikroprozessor

Der Infineon XMC 1xxx48 gehört zu der Familie der ARM Cortex -M0 Prozessoren und ist ein 32-bit Industrial Microcontroller und wird mit 48MHz externer Clock betrieben. Die 48 steht für die Anzahl der Pins. Der interne Timer läuft mit 96Mhz. Neben bietet der XMC einen 12 bit A/D Wandler, welcher für die Analogmessung eine viel genauere Auflösung bieten kann als ein 8 bit A/D Wandler. Die Betriebsspannung des Prozessors beträgt 3,3V.

<https://www.infineon.com/cms/en/product/microcontroller/32-bit-industrial-microcontroller-based-on-arm-cortex-m/32-bit-xmc1000-industrial-microcontroller-arm-cortex-m0/#>

4 Wahl sowie alternative Bauteile

Für den Prototypen wurden mehrere verschiedene Ultraschallkapseln verwendet, um einen Vergleich der Qualität der Bauteile herstellen zu können. Für die Operationsverstärker wurde das IC LS232 verwendet, da dieses IC bereits zwei OPV's enthält.

5 Open Source CAD-Programm Packet zur Erstellung von elektronischen Leiterplatten

Das Open Source CAD-Programm KiCAD ist eine Anwendung zum Erstellen von Schaltplänen und elektronischen Leiterplatten. Hier lassen sich auf einfache Weise Schaltpläne erstellen und nach einer Prüfung auf fehlerfreie Verdrahtung zur vereinfachten Platinenerstellung nutzen.

5.1 Schaltplan

Zur Signalerzeugung wurde ein Hochsetzsteller eingesetzt, um eine höhere Spannung erzeugen zu können, als auf dem System zur Verfügung steht. So höhere Amplituden auf den Sender gegeben werden, um eine höhere Signalstärke zu erreichen. Danach wird das Signal auf eine H-Brücke geleitet, diese dient dazu aus dem Gleichsignal ein Wechselsignal mit einer Frequenz von 40kHz zu generieren. Dieses Wechselsignal wird auf den Sender gegeben um das Spannungssignal in Schallwellen umzuwandeln.

Wird ein zurückkommendes Signal empfangen, so wandelt der Empfänger die Schallwellen in ein Sinussignal um. Dieses Signal wird auf einen nicht invertierenden Verstärker geleitet, um die Signal-Amplitude zu verstärken und anschließend durch einen Komperator in ein Rechteckiges(digitales) Signal umgewandelt. Dieses Signal wird im Prozessor verarbeitet, um aus der Zeit zwischen senden und empfangen des Signals in die Entfernung zum betreffenden Objekt umzuwandeln.

Beim Schaltplanentwurf gilt es auch auf gewisse Physikalische Eigenschaften von Leitungen und Bauteilen zu achten. So sollten bei Anschluss längerer Leitungen, Condensatoren zum ausfiltern eingefangener Funksignale und anderer EMV-Belastungen, angebracht werden. Auch benötigen Bauteile wie Platinen Kondensatoren an der Spannungsversorgung, um auch kleinste Schwankungen dieser zu vermeiden.

5.2 Platinenlayout

Bei dem Entwurf eines Platinenlayouts gibt es viele Möglichkeiten ein Ergebniss zu erzielen. So können alle Bauteile so angeordnet werden, dass alle parallelen Bauteile sauber nebeneinander aufgereiht werden, und die in Reihe dazu liegenden Bauteile auch darunter angeordnet sind. So sähe die Platine zwar ähnlich eines Kontaktplanes aus, allerdings ist diese Variante aus Sicht der EMV nicht sonderlich empfehlenswert.

Auch können die Bauteile wie im Schaltplan in Gruppen zusammen gelegt werden und der Schaltplan auf der Platine 1:1 nachgebildet werden. Auch bei dieser Variante ergeben sich gelegentlich Probleme, was die Führung der Leitungen, und vor allem den Verlauf der Ströme angeht.

So sollte ein Augenmerk auf den Stromführenden Leitungen liegen. Je höher der Strom ist, desto breiter ist die Leitung auszulegen und sie sollte auch möglichst kurz gehalten werden um weniger EMV störungen zu erzeugen. Auch sollte die Rückführung (GND) günstigerweise als eigene Leiterschicht ausgeführt werden, um einen großen Leiterquerschnitt zu gewährleisten. So kann bei der Rückführung der Ströme auch das Risiko vermieden werden, durch die Bildung von größeren Schleifen Antennen zu erzeugen. Die GND Schicht sollte so wenig wie möglich unterbrochen werden, vorallem sind Unterbrechungen quer zur Stromflussrichtung zu vermeiden. Zusätzlich ist zu beachten, dass Kondensatoren meistens zur Verringerung von Störungen nahe digitalen Bauteilen angebracht werden sollten. Die optimale Platzierung ist direkt an den Pinns, so dass die Leiterbahn mit einem höheren Querschnitt auf den Kondensator geht, und dann mit leicht verringertem

Querschnitt direkt auf die Pinne des IC-Bauteils verläuft.

6 Entwicklung der Software zum Betrieb des Prototypen

6.1 Benötigte Register

Um das Programm zu entwerfen gab es die Möglichkeit, alle Zähler per Hand zu programmieren und so ein recht aufwendiges und langsames System zu entwerfen, oder die im Datenblatt hinterlegten Informationen zu bereits im Prozessor hinterlegten Zählregistern und interruptfähigen Eingängen zu erarbeiten. Durch diese aufwendige Recherche konnte ein grundlegendes Verständniss der Zählereinheit ccu40 erarbeitet werden, mit deren Hilfe sowohl ein Pulsweitenmoduliertes Rechtecksignal, als auch mehrere Zähler realisiert werden konnten.

6.1.1 Durchgeführte Berechnungen

Auch für die Programmierung waren diverse Berechnungen notwendig. So musste zur Erzeugung der Ultraschallimpulse ein Pulsweitenmoduliertes Rechteck signal geschaffen werden. Dafür wurde ein Timer der ccu40 auf einen Takt von 40kHz eingestellt. So musste bei einem Timertakt von 96MHz eine Periodendauer von 2400 Takten und ein Compare-Wert von 1200 Takten eingestellt werden. Im Zählvorgang des Timers wird der Ausgang nach Erreichen des Compare-Wertes auf 1, und nach Erreichen der Periodendauer wieder auf 0 gesetzt. Dadurch ergibt sich eine Periodendauer von 25us, was einer Frequenz von 40kHz entspricht.

Auch zur Erfassung der Zeit, die vergeht bis das Echo des Ultraschall-Impulses zurück kommt wird über einen Timer der ccu40 erfasst.

6.1.2 Einstellung im Programm

6.2 Quellcodeentwurf

Um Fehler in der Zeitnahme zu verringern sollten Zeitwerte, die ausgelesen werden sollen direkt am Anfang des Interrupts in Variablen gespeichert werden und nicht erst innerhalb anderer Anweisungen oder Schleifen, da das schon deutliche Abweichungen mit sich bringt.

Programmstruktur: Anstatt alles in der main.c an Programmcode zu verfassen was bei sehr komplexen Programmen schnell zu unübersichtlichkeit führt sowie die Bearbeitung an Bugs was dadurch noch extrem erschwert wird. Somit werden die Funktionen die in der main.c auftauchen ausgelagert und in einzelnen Funktion verschachtel.

Siehe Abb x.x

```
1 #include <stdio.h>
2 #include <stdbool.h>
3 #include "configs/config.h"
4 /****Eigene Include Dateien****/
5
6 #include "bricklib2/bootloader/bootloader.h"
7 #include "system_timer/system_timer.h"
8 #include "bricklib2/logging/logging.h"
9 #include "communication.h"
10 #include "a16pt.h"
11 int main(void)
```

```

12 {
13     logging_init();
14     logd("Start Distance US V2 Bricklet/n/r"); Fuer den DBUGModus TXPin P0_12
15     communication_init(); //Funktionsaufruf
16     a16pt_init(); //Funktionsaufruf
17     while(true)
18     {
19         a16pt_tick(); //Funktionsaufruf
20         bootloader_tick(); //Funktionsaufruf
21         communication_tick(); //
22         Funktionsaufruf
23     }
24 }

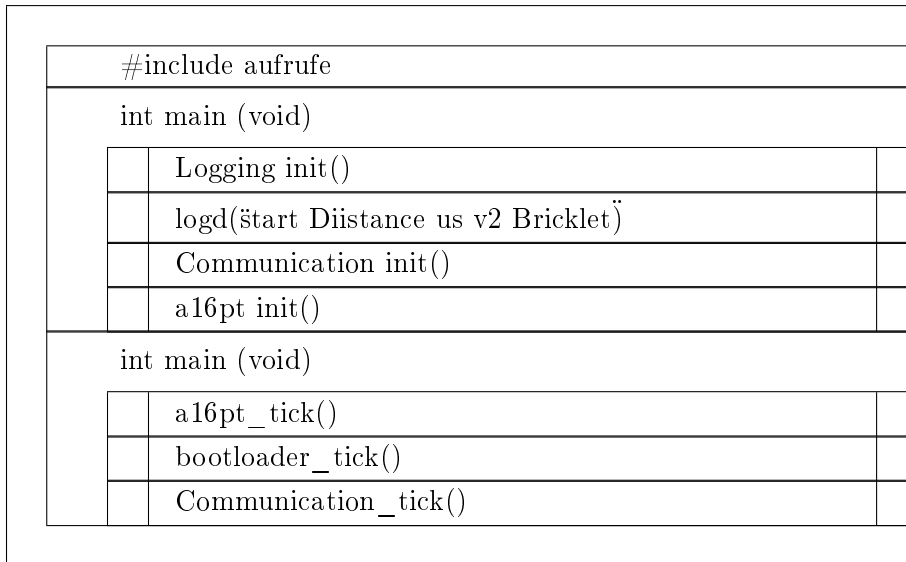
```

Den Inhalt von den „Eigene-Include_Dateien“ bis zur a16pt.h können wir vernachlässigen weil sie Standard mässig vor eingestellt werden. Die meiste Änderung findet in der a16pt.h statt wo der Fokus hauptsächlich liegt. In der a16pt.h werden die Funktionen definiert die dann in der main.c aufgerufen werden. In der a16pt.c stehen die Funktionsanweisungen dazu. So können auch einzelne Header-Dateien erstellt werden wo nur die Ports oder andere Werte definiert werden. Dieses geschieht mit dem Ziel nur einzelne Dateien bearbeiten zu müssen, um Änderung an Ports schnell und effizient durchführen zu können, ohne mehrere Dateien suchen zu müssen um eine Änderung zu bewirken.

```

1 #ifndef A16PT_H
2 #define A16PT_H
3 #include <stdint.h>
4 void a16pt_init(void); //Funktionsdefinition
5 void a16pt_tick(void); //Funktionsdefinition
6 uint16_t a16pt_get_distance(void); //Funktionsdefinition

```



Im Main stehen vor allem die Aufrufe der verschiedenen benötigten Funktionen. Diese Aufteilung auf verschiedene Blöcke (und in diesem Falle auch Dateien) dient der Vereinfachung der Abläufe und der Übersicht. So sieht man in der Main jetzt deutlich, welche Funktionen beim Starten initialisiert werden, und welche Unterprogramme regelmäßig aufgerufen werden. Auch vereinfacht diese Struktur gerade bei Prototypen das Testen der Funktion, so kann im Falle einer fehlerhaften Funktion einfach der Aufruf auskommentiert werden um zu testen, ob der Fehler wirklich von der Funktion herrührt. Dadurch müssen nicht etliche Zeilen Programmcode der Funktion auskommentiert werden, wodurch schnell Fehler entstehen könnten, durch übriggebliebene Zeichen, oder gar beim entfernen der Auskommentierung gelöschte Zeichen.

In der a16pt werden die für die Entfernungsmessung notwendigen Funktionen und die Interrupt anweisungen abgearbeitet, außerdem werden die Ein- und Ausgangsports hier geschaltet.

7 Messungen und Auswertung der Ergebnisse

Zu Beginn wurden Messungen direkt an dem Prozessor durchgeführt, um zu kontrollieren, dass die im Programm errechneten Signale auch die vorgesehenen Frequenzen erreichen. Anschließend wurden einzelne Baugruppen mit dem Prozessor kombiniert, um das Zusammenspiel dieser auswerten zu können. Danach wurden die Baugruppen um den Sensor erweitert und auch dieses Zusammenspiel wurde ausgewertet. Zum Abschluss wurde dann die komplette Platine im Betrieb getestet und optimiert.

7.1 PWM-Ausgabe

Um das Signal für die Entfernungsmessung zu generieren wurde der Mikrocontroller so programmiert, dass zehn Impulse mit einer Frequenz von 40kHz ausgegeben werden. Danach erfolgt eine Pause, um das zurückkehrende Signal abzuwarten und auszuwerten.

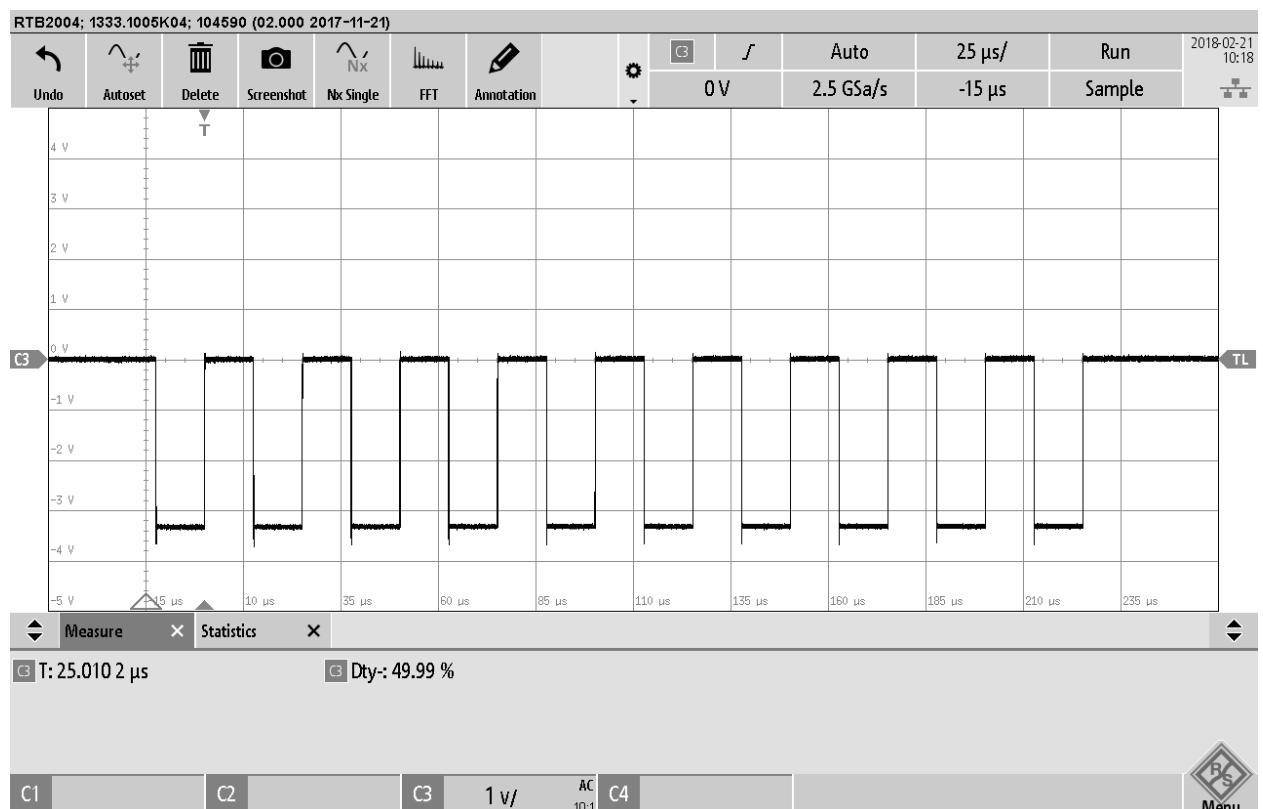


Abbildung 7.1: PWM-Burst auf 40kHz Basis

In der Abbildung 7.1 ist zu sehen, dass ein Burst aus zehn Impulsen mit einer Periodendauer von jeweils 25μs generiert wurde. Diese Messung wurde direkt an dem Mikrocontroller vorgenommen, um sicherzustellen dass die H-Brücke mit der passenden Frequenz angesteuert wird.

7.2 H-Brücke

Die H-Brücke wurde durch einen IXDN602, einen MOSFET ersetzt. Dieser wurde als Low-Side Driver-geschaltet. Dies geschah, weil erstens die Beschaltung des OPV nur für eine positive Halbwelle funktioniert und zweitens die Beschaltung der H-Brücke fehlerhaft war. So wurde einer der Ausgänge auf Masse gelegt, was im Betrieb einen Kurzschluss verursachen würde. Um diese Probleme zu beheben, mussten der Sender- und der Empfängerkreis aufgetrennt werden und eine zweite Ultrashallkapsel eingesetzt werden.

7.3 Signalverlauf nach Verstärkung durch den OPV

7.4 aufgetretene Probleme

Zur Inbetriebnahme der H-Brücke mussten Korrekturen vorgenommen werden. Denn diese gab zu Anfang noch keine Signale aus. Es wurde übersehen, dass der OCLSEL-Pin (Oszillatorauswahl) bei der Beschaltung berücksichtigt werden muss. So wurde der Prototyp um eine Drahtbrücke erweitert.

8 Ergebnisse der Forschung

Die Verwendung der H-Brücke war eine Fehlentscheidung, diese wurde durch einen MOSFET ersetzt. Das Problem mit der H-Brücke bestand darin, dass das gewählte Bauteil zum einen nicht zwangsläufig die gewünschte Schaltfrequenz halten kann.

9 Fazit aus den Ergebnissen für den Auftraggeber

10 Reflektion über den Projektablauf

11 Anhänge

11.1 Schaltpläne

11.2 Platinenlayout

11.3 Quellcode

Hier könnte Ihr Quellcode (inklusive Werbung) stehen
weitere Abbildungen

Abbildungsverzeichnis

7.1	PWM-Burst auf 40kHz Basis	13
-----	-------------------------------------	----

Tabellenverzeichnis