

Entwurf und Implementierung einer DSP-Basierten Ansteuerung für Brus- hless DC-Motoren

Praktikumsbericht

vorgelegt von

Nguyen Trong Khoa

EDV.Nr.:951 990

dem Fachbereich VII – Elektrotechnik – Mechatronik – Optometrie

der Berliner Hochschule für Technik Berlin

vorgelegte Bachelorarbeit

zur Erlangung des akademischen Grades

Bachelor of Engineering (B.Eng.)

im Studiengang

Humanoide Robotik

Tag der Abgabe 22. Februar 2026



Studiere Zukunft

Gutachter

Prof. Dr. Peter Gober Berliner Hochschule für Technik

Christopher Böhm

DesCap GmbH

Kurzfassung

Die Kurzfassung gibt ein kurzes und prägnantes Bild der gesamten Arbeit. Sie soll den Leser neugierig machen und klarmachen, was zu erwarten ist. Erreichte Ergebnisse werden kurz umrissen.

Abstract

Bachelor and Master-Theses usually are often written in german. Nevertheless, their content may be interesting for people being not able to read german. In order to awaken their interest in this topic, an abstract in english is given. The experimental results and analysis are shown in short

Aufgabenblatt – durch Original austauschen !!!

Erklärung

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

Datum

Unterschrift

Inhaltsverzeichnis

1 Praktikumsbetrieb	3
1.1 Vorstellung des Betriebes	3
1.2 Weg zur Praktikumsstelle	3
2 Praktikumstätigkeiten	5
2.1 Hardware-Design	5
2.2 Hardware-Testen und Dokumentation	7
2.3 Filter	8
A Angehängtes: Die Dateien des Pakets	9

Abbildungsverzeichnis

Kapitel 1

Praktikumsbetrieb

1.1 Vorstellung des Betriebes

Die DesCap GmbH ist ein Start-up-Unternehmen mit Sitz in Berlin Alt-Mariendorf, welches sich im Bereich der Filtergeräte mit Schwerpunkt auf Atemschutz. Zu ihren Produkten, sowohl marktreif als auch unter Entwicklung, zählen vor allem filtrierende Gesichtsvisiere sowie Sensorsysteme zur Erfassung der Luftqualität und der sich darin befindenden Schadstoffe.

1.2 Weg zur Praktikumsstelle

Auf DesCap bin ich im April 2025, also etwa sechs Monate vor Praktikumsanfang, dank einer Email von Professor Hannes Höppner aufmerksam geworden. Da die Stellenausschreibung mich sehr angesprochen hat, habe ich mich entschieden, meine Bewerbung bei DesCap einzureichen. Der Bewerbungsprozess lief effizient und unkompliziert: es wurden insgesamt zwei Interviewgespräche geführt, das erste mit Ole (einem der Gründer) und Christopher (mein künftiger Kollege); beim zweiten Gespräch hat sich Sönke, der andere Gründer und zeitgleich Geschäftsführer, dazu gesellt. Der gesamte Bewerbungsablauf von der Bewerbung bis zur Vertragsunterzeichnung hat ungefähr 3 Wochen gedauert, und schon am 1 Mai 2025 begann mein erster Tag als Werkstudent. Zum Anfang des Wintersemesters 2025 verwandelte sich meine Werkstudentenstelle in ein sechsmonatiges Praktikum mit Schließung des Praktikumsvertrags. Dementsprechend erstreckte es sich vom 1 Oktober 2025 bis zum 1 April 2026. Es wurden durchschnittlich 20 Wochenstunden erbracht, sodass im Laufe der gesamten Praktikumsdauer 480 Stunden sich angehäuft haben.

Kapitel 2

Praktikumstätigkeiten

2.1 Hardware-Design

Meine Hauptaufgabe bei DesCap besteht in dem Designen und Validieren von PCBs für das Produkt "Work", welches zum Beispiel vor stäubigen Umgebungen bei handwerklichen Arbeiten schützt. Auf der Leiterplatte befinden sich unter anderem ein DC-Spannungswandler (zum gleichzeitigen Betreiben von zwei Lüftern) und ein MSP430 Mikrokontroller, mit dem sich alle Nutzerfunktionen realisieren lassen.

Schon bei der Einarbeitungsphase wurde darauf hingewiesen, dass ein komplett neues PCB-Design gemacht werden musste aufgrund des Mikrokontrollerwechsels (ATTiny zu MSP430). In den ersten zwei Wochen bekam ich die Aufgabe, mich mit der Schaltung des Vordesigns vertraut zu machen, wofür es enge Betreuung und Hilfestellung gab in Form von täglichen Treffen mit dem Engineering-Team. Dazu kamen auch kleinere Hardware-Recherchen, um etwa den richtigen Knopf oder die kostengünstigsten LEDs für das PCB zu finden.

Da ich im Vorfeld leider kein KiCad-Wissen verfügt habe, habe ich mich für 2 bis 3 Wochen im Anschluss mit den Grundlagen des PCB-Designs sowie der Bedienung von CAD-Softwares wie KiCad auseinandergesetzt. Das eigentliche Design hat 2 Wochen beansprucht, und während dieser Zeit gab es ständig Austausche, denn Christopher konnte sich über Github meinen Fortschritt ansehen und mir ggfs Rückmeldung geben. Da das neue Board über eine USB-Schnittstelle flasbar sein sollte und diese strenge Anforderungen bezüglich Routens und Impedanzmatchings stellte, haben wir uns zusätzlich noch externe Expertise für ein ausführliches Design-Review geholt.

Dabei habe ich gelernt, wie man mit welchen Tools bspw. die Impedanz eines USB-Diff-Pairs berechnen kann, da es der kritischste Aspekt war. Für dieses allererste Design kamen wir auf 86Ω , also eine akzeptable Abweichung vom Idealwert 90Ω . Bei dem Review wurde außerdem der für

uns sehr wichtige Gesichtspunkt EMV (elektromagnetische Verträglichkeit) besprochen zwecks Zertifizierung. Es wurden die "best practices" erklärt, wie man unter anderem DC-DC-Wandler routen oder welche Glättungskondensatoren man dafür nehmen soll.

Der nächste Schritt war der Bestellungsablauf. Hier habe ich mich an erster Stelle um die Auswahl von Komponenten gekümmert: geläufige Bauteile wie Widerstände oder Kondensatoren wurden nach ihren Werten optimiert (BOM-Konsolidierung); bei anderen Bauteilen wie Knöpfen, Schaltern oder LEDs wurde auf Kosten und langfristige Verfügbarkeit geachtet, damit zukünftig zeitkritischen großen Bestellungen nicht am Supply-chain scheitern.

Eine Woche etwa nach Bestellungsabschluss kamen die Leiterplatten von China bei uns in Berlin an. Das Flashen vom MSP430, welches zur Erinnerung am meisten Kopfschmerzen bereitet hat, hat entgegen Erwartungen auf Anhieb funktioniert. Wir hätten gedacht, dass wir ein paar Design-Runden bräuchten, um das halbwegs auf die Reihe zu bekommen. Christopher war demgemäß sehr glücklich und hat infolgedessen alle zum Mittagsessen in der Schindler-Mensa eingeladen.

Nach diesem enormen Meilenstein bestanden weitere Entwicklungsarbeiten, zumindest was das PCB-Design anging, nur noch in kleinen Modifikationen von einer Version zur anderen: manche Komponenten wurden gegen bessere und günstigere ausgetauscht, falsche Slkscreen-Texte wurden berichtet usw. Lediglich bei der fünften Iteration ist eine Herausforderung aufgetreten, nämlich dass der Bestand für unseren USB-Stecker bei JLCPCB zu gering war und die Nachlieferung unsere Zeiteinschränkungen überschritten hat. Nach tagelanger Recherche wurde für eine nachhaltigere Variante entschieden. Allerdings musste ich selber den Footprint davon erstellen, da er nicht in der Standard-KiCad-Library vorhanden war. Aufgrund dessen, dass man im Kicads Footprint-Editor bei komplexen Geometrien im Gegensatz zu Solidworks z.B. die Maßen und Abstände der Pads nicht genau in Verhältnis zueinander setzen kann, muss über Umwege gegangen werden. Letztendlich musste ich die mechanische Zeichnung vom Footprint in Solidworks umsetzen, diese im Anschluss mit Inkscape bearbeiten, um sie endlich als Vorlage in Kicad zu importieren.

Die Schwierigkeiten hören noch nicht auf. Die Pads des neuen USB-Steckers sind viel dichter bevölkert als beim alten, und das hat für einen veritablen Kampf gesorgt. Weil unser Stecker ein vertikaler und kein horizontaler ist wie die meisten USB-Stecker, müssen die D+ und D- Differentialleitungen in dem ganz schmalen Zwischenraum des Footprints überkreuz verbunden werden. Dies erfordert, dass ein Via dazwischen gequetscht werden muss, wodurch jedoch die DFM-Kapabilitäten (design for manufacturing) von unserem PCB-Produzenten JLCPCB regelrecht auf die Probe gesetzt werden.

Für das Produkt fungiert der oben erwähnte MSP430 als Funktionszentrale, und dieser muss natürlich auch programmiert werden. An dem Programmierprozess habe ich mich nicht direkt beteiligt, denn sonst wären mir die Aufgaben über den Kopf gewachsen. Nichtsdestoweniger musste ich mir die Datenblätter davon durchlesen, um am Ende den Code nachvollziehen zu können. Sowohl beim PCB-Design als auch beim Programmieren werfen mindestens 2 Leute einen Blick rein, um die Fehlerwahrscheinlichkeit zu minimieren. Meinen Beitrag zur MSP-Thematik habe ich nicht unbedingt auf C-Code-Ebene geleistet, sondern vielmehr auf Assembler-Ebene, denn ich musste etliche Schutzfunktionen implementieren und es ging nur, wenn man direkt den Linker-Skript modifizierte. Obwohl nicht notwendig, habe ich mich deswegen auch darin vertieft, wie Linker und Loader hinter der Kulisse funktionieren.

Abgesehen vom Hauptprodukt habe ich auch andere Designprojekte aufgenommen. Eines davon war ein Breakout-Board, das Partikelsensoren mit einem ESP32 verbindet. Dieses sollte Bestandteil von unserem Teststand zum Messen der Filtereffizienz, allerdings wurde es nicht in Betrieb genommen, weil wir uns anders überlegt haben.

Ein weiteres Projekt, an dem ich mich seit Kurzem beteilige, ist die Integration von Sensoren in das "WorkProdukt, die in der Lage sein sollen, sich untereinander mithilfe einer Basisstation über das LoRaWan-Protokoll Daten bezüglich der Innenraumluftqualität auszutauschen. Dies soll dem Zweck dienen, Simulationsergebnisse mit der wirklichen Betriebsfiltereffizienz zu vergleichen sowie Nutzer rechtzeitig warnen, sobald Grenzwerte überschritten werden. Dieses Projekt hat mir DesCap als Thema für die Bachelorarbeit vorgeschlagen und sieht vorerst folgende Unteraufgaben vor: Definieren von Anforderungen DesCaps (bzgl. Datenrate, Reichweite usw), Recherche und Auswahl eines geeigneten Mikrocontrollers bzw Entwicklungsboards, PCB-Prototypen entwerfen ...

2.2 Hardware-Testen und Dokumentation

Da es sich um Verbraucherprodukte handelt, ist das Testen genauso wichtig, wenn nicht sogar wichtiger, als das Designen von Leiterplatten. Hiermit verbringe ich also viel Zeit bei DesCap, um sicherzustellen, dass alles zuverlässig funktioniert und alle Missbrauchsszenarien seitens des Nutzers abgedeckt werden. Dafür haben wir uns schon nach Bestellungsabgabe der PCBs eine umfangreiche Teststrategie überlegt. Dazu gehören unter anderem folgende Punkte:

1. Welche Größen von welchen Komponenten müssen erfasst werden? Z.B. Temperatur und Entladespannung der Batterie, PWM-Duty-Cycle der Lüfter, Ladestrom ...
 2. Zulässige Wertebereiche für gemessene Signale
-

3. Durchschnittliche Standzeit in verschiedenen Betriebsmodi

Bei der ersten Batch wurde alles noch manuell erfasst, das heißt die PCBs enthielten zahlreiche Testpads, die beim Bedarf gelötet und an das Messgerät herangeführt werden mussten. Schnell haben wir festgestellt, dass es zwar funktioniert hat, aber recht mühsam war, wenn man mehrere Boards hintereinander testen wollte. Mit dieser Erkenntnis habe ich mir zur Aufgabe gemacht, einen kostengünstigen PCB-Test-jig zu bauen.

Der Aufbau entspricht einem typischen kommerziellen Testjig, mit kleinen Anpassungen: die Pogopins, die die SMT-Testpads auf dem PCB berühren sollen, werden auf einem Perfboard gelötet und auf der Unterseite des 3D-gedruckten PCB-Trägers angebracht. Die Oberseite hat einen Ausschnitt mit der gleichen Geometrie wie das PCB selbst. Die Pogopins werden auf der anderen Seite über Drähte mit dem Messgerät verbunden (in diesem Fall der Analog Discovery 3).

Auf diesem Wege können sehr aussagekräftige Kennlinien aufgezeichnet werden:

2.3 Filter

Anhang A

Angehängtes: Die Dateien des Pakets

Stylefile

Die Styledatei für diese Abschlussarbeit ist `bhtThesis.sty`, die in der Archivdatei vorliegt. Diese muss von L^AT_EX auffindbar sein, muss also in einem L^AT_EX bekannten Ordner liegen:

- Ubuntu-Linux: `$HOME/texmf/tex/latex/bhtThesis/bhtThesis.sty`
- MikTeX: `c:\localtexmf\tex\latex\bhtThesis\bhtThesis.sty`

Beispieldokument

Dieses Dokument befindet sich im Unterordner `tryout` des zip-files. Sie können diese Dateien in einen Ordner kopieren, in dem Sie schliesslich arbeiten werden. Die Dateien sind die folgenden

- `abstract_de.tex` Kurzfassung in deutscher Sprache
- `abstract_en.tex` Kurzfassung in englischer Sprache
- `anhang.tex` der Anhang
- `bhtThesis.bib` beinhaltet die zu zitierenden Literaturstellen und wird von bibTeX ausgewertet
- `main.pdf` ist die Ausgabedatei mit der Druckvorlage
- `main.tex` beinhaltet das Hauptdokument
- `makefile` realisiert das automatische mehrfache Übersetzen, hierfür muss `make` auf dem System installiert sein.
- `myapalike bst` beinhaltet die Formatierung für das Literaturverzeichnis

- `personalMacros.tex` kann einzelne, persönliche Macros beinhalten, die das Schreiben erleichtern
- `titelseiten.tex` realisiert alle Seiten bis zum Beginn des ersten Abschnittes
- Ordner `pictures`
 - `BHT-Logo-Basis.eps`
 - `BHT-Logo-Basis.pdf`
- Ordner `kapitel1`
 - `ch1.tex` Quelltext des Kapitel 1
 - Ordner `pictures`
 - * `schaltbild.pdf`
- Ordner `kapitel2`
 - `ch2.tex` Quelltext des Kapitel 2
 - Ordner `pictures`
 - * leer