

# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>ii</b>
<b>1 Das Unternehmen</b>	<b>1</b>
<b>2 Projekte und Tätigkeiten während des Praktikums</b>	<b>3</b>
<b>3 Projektbeispiele</b>	<b>6</b>
3.1 Terminplanung und Entwicklungsablauf . . . . .	6
3.2 Umsetzung . . . . .	6
3.2.1 Power Management . . . . .	7
3.2.2 digitale Eingänge . . . . .	8
3.2.3 analoge Eingänge . . . . .	8
3.2.4 digitale Ausgänge . . . . .	9
3.2.5 USB Host / Slave / OTG . . . . .	9
3.2.6 Serielles debug Interface . . . . .	9
3.2.7 Skriptsprache LUA . . . . .	10
3.2.8 Planung der Bestückung . . . . .	10
<b>4 Erfahrungen und Zusammenfassung</b>	<b>12</b>

# Kurzfassung

Im Zuge des Praktikums wurde der erste Prototyp einer Industriellen Steuerung entwickelt. Dabei mussten diverse Schaltpläne erstellt werden, sowie das Layout der einzelnen Platinen. Dabei standen Schutzbeschaltungen an erster Stelle, da die Steuerung im ESD Umfeld betrieben werden muss. Zu Beginn wurde ein Konzept erstellt, welches dann nach und nach implementiert wurde. Zwischendurch wurden immer wieder Überprüfungen durchgeführt um Fehler zu minimieren. Die Platine des ersten Prototyps wurde bei Eurocircuits gefertigt. Die Bestückung der über 300 SMD Teile erfolgte händisch. In der Testphase stellten sich einige, wenige Fehler heraus, diese konnten größtenteils direkt am Prototyp korrigiert werden. Während der Fertigung wurden bereits die ersten Treiber in C/C++ erstellt. Um Benutzerprogramme ausführen zu können wurde eLua portiert und in Betrieb genommen. Zusätzlich wurde ein Konzept für die Ethernet Kommunikation entwickelt. Zu guter Letzt wurde ein kleiner Teststand entwickelt welcher die Funktionen des Prototyps demonstriert.

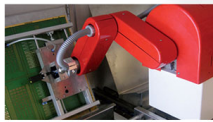
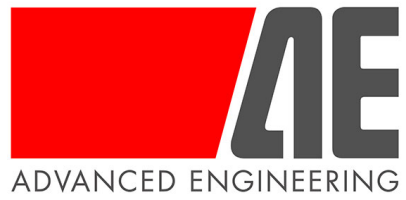
# Kapitel 1

## Das Unternehmen

Die Firma Advanced Engineering GmbH in Salzburg Hallein arbeitet seit 1991 als weltweiter Ausrüster für die PCB Industrie. Zu den Kernprodukten zählen einerseits Umladeroboter sowie Maylar Pealer. In der Industrie ist es wichtig die einzelnen Verarbeitungsschritte möglichst effizient zu verbinden. Dazu werden spezielle Buffer sowie Rüst-Roboter benötigt. Die Advanced Engineering GmbH verbindet genau diese mit inovativen Robotern, welche einen reibungslosen Betrieb ermöglichen. Der Kundenstamm zählt von Bosch in Deutschland über CMK Japan bis hin zu Samsung bzw. LG in Korea. Dabei wird speziell auf Kundenwünsche eingegangen. Alle Produkte weisen eine Ausfallsrate von unter einem Promill auf. Bei einem Betrieb von 24 / 7. Das wohl herausragendse Produkt ist der Maylar Peeler. Maylar<sup>1</sup>, eine spezielle Schutzfolie, welche sich auf den einzelnen Schichten der für die Herstellung eines PCB verwendeten Teilen befindet. Diese muss vor den nächsten Verarbeitungsschritten abgezogen werden, was händisch nahezu unmöglich ist. Die Advanced Engineering GmbH schafft dafür Abhilfe. Mit einer Geschwindigkeit von über 9 Meter pro Minute kann der Maylar Peeler diese Folien abziehen und braucht dabei gerade mal unter einem Drittel des Platzes der Konkurrenz. Das ist mitunter ein Grund das die Produkte der Advanced Engineering GmbH großes Interesse in Korea genießen. Alle Produkte sind zu 100 Prozent *Made in Austria* und werden von einem relativ kleinem Team produziert.

---

<sup>1</sup>Biaxial orientierte Polyester-Folie kurz boPET



**Abbildung 1.1:** Firmen-Logo sowie Produktabbildungen.

## Kapitel 2

# Projekte und Tätigkeiten während des Praktikums

Im Zuge des Praktikums stand die Entwicklung des Prototypen einer Industriellen Steuerung im Vordergrund. Die von der Advanced Engineering GmbH entwickelnden Roboter benötigen spezielle Steuerungen. Anfangs wurden SPS Systeme von verschiedenen Herstellern verwendet und getestet, jedoch mit nur mäßigem Erfolg. Speziell beim verarbeiten von Folien, da die dabei entstehende elektrostatische Entladung die Hardware stört, ja oft sogar beschädigt. Um diese Probleme zu lösen wurde eine externe Firma beauftragt, eine spezielle Steuerung zu entwickeln. Diese basiert auf einem Industrie PC mit via PCI-104 angeschlossenen Module, wie Schrittmotor-treiber oder Ein- Ausgangs- Karten. Diese Steuerung wird nun seit vielen Jahren mit so gut wie keinem Zwischenfall verwendet. Die Nachfrage der Roboter stieg in den letzten Jahren stark an und da die Steuerung für jeden dieser Roboter zugekauft werden muss, entstehen dabei enorme Kosten. Speziell für kleine Geräte ist die Verwendung dieser Steuerung wirtschaftlich sehr fragwürdig. Deshalb die Idee selbst eine Steuerung auf neuestem technologischem Stand zu entwickeln.

Die Anforderung ist klar. Es soll eine modulare Steuerung entwickelt werden, welche mindestens genau so robust wie die Aktuelle ist. Es sollten keine günstigen Bauteile verwendet werden, nur das Beste kommt in Frage. Natürlich sollten die Kosten auch möglichst gering gehalten werden. Da es sich dabei um eine Eigenentwicklung handelt sind die Kosten des fertigen Produktes generell um einiges geringer als wenn jede einzelne Steuerung anzukaufen wäre. Dadurch rückte dieses Kriterium in den Hintergrund. Gemeinsam mit einem Studienkollegen stellten wir uns der Herausforderung. Im ersten Teil des Praktikums erstellten wir gemeinsam ein Konzept, im Weiteren wurden die Themen aufgeteilt und abgearbeitet.

Die Steuerung soll grundsätzlich für kleine Maschinen verwendet werden. Durch die modulare Aufbauweise kann diese idealerweise auch größere

Maschinen steuern. Dabei galt es die Ausfallsicherheit zu beachten, da die aktuelle Steuerung eine Ausfallsrate von unter einem Promill aufweist. Das Umfeld der Steuerung ist jedoch alles andere als ideal für Elektronik, speziell im Hinblick auf ESD. Dazu müssen spezielle Schutzbeschaltungen wie TVS-Dioden implementiert werden. Grundsätzlich muss die Steuerung mit unterschiedlichen Potentialen arbeiten können. Deshalb müssen alle Bereiche voneinander galvanisch getrennt werden. Um die Wartungsfreundlichkeit zu erhöhen werden alle Schnittstellen überwacht und dadurch werden Fehler automatisch erkannt. Mit der Steuerung sollten möglichst viele benötigte Aktoren und Sensoren direkt verwendet werden, ohne Wandler oder Treiber zu benötigen.

Der von uns angedachte Prototyp umfasst 16 digitale Eingänge sowie 16 digitale Ausgänge und 8 analoge Eingänge. Zusätzlich können 3 Dreiphasen Schrittmotoren angeschlossen werden. Für die Kommunikation ist Ethernet vorgesehen sowie CAN. Um Ethernet möglichst einfach verwenden zu können ist ein 10/100 Mbit Switch verbaut, um die Steuersignale weiter schleifen zu können. Als Prozessor dient ein embedded Arm von ST mit 170 MHz. Siehe Abbildung 2.1 für Details.

Das geplante System soll ohne Betriebssystem auskommen. Dadurch kann die benötigte Rechenleistung dramatisch reduziert werden, da lediglich nur der zur Steuerung benötigte Code ausgeführt wird. Für jede verwendete Maschine wird dazu ein spezielles Programm erstellt und die CPU damit programmiert. Natürlich bringen Betriebssysteme wie Linux viele Vorteile mit sich, wie zum Beispiel Speicherverwaltung, jedoch benötigen diese auch viel leistungsstärkere Hardware.

Alle Treiber für die verbaute Hardware wurden in C / C++ erstellt und bieten einfache Schnittstellen. Der HAL<sup>1</sup> wurde grundsätzlich mit Macros und Pointer implementiert um einfache Migration zu ermöglichen. Zusätzlich wurde ein LUA Interpreter implementiert, mittels diesem es möglich ist, kleine Benutzerprogramme ausführen zu können. Diese kleinen Programme liegen auf einer verbauten micro SD Karte. Es besteht weiterhin die Möglichkeit diese Programme via Netzwerk zu verwalten. Dazu können Programme gestartet, beendet, verändert sowie schrittweise ausgeführt werden. Dadurch können Fehler leichter gefunden werden.

Grundsätzlich wurde das Projekt von uns selbständig entwickelt. Während der Design Phase wurden immer wieder gegenseitige Kontrollen durchgeführt um die Entstehung von Fehlern möglichst gering zu halten.

---

<sup>1</sup>Hardwareabstraktionsschicht

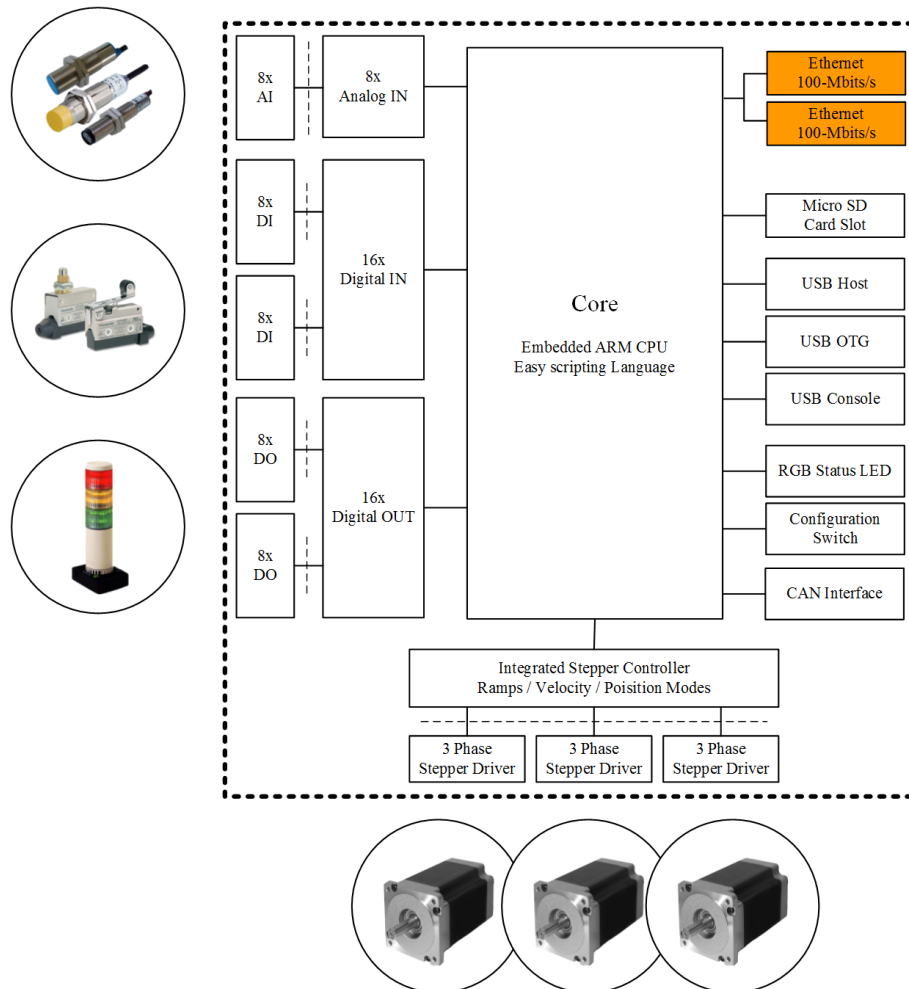


Abbildung 2.1: Blockdiagramm der Steuerung.

## Kapitel 3

# Projektbeispiele

### 3.1 Terminplanung und Entwicklungsablauf

Das Projekt wurde in die unten angeführten Punkte eingeteilt und auf eine Mindestdauer von drei Monaten geplant. Die Fertigstellung der Schaltpläne sowie die Inbetriebnahme wurden jeweils als Meilensteine angesehen und konnten auch eingehalten werden.

- Abklärung der Anforderungen
- Entwurf eines groben Konzepts
- Verfeinerung des Konzepts durch die Recherche der Teile, erste Festlegung von essentiellen Bauteilen
- Realisierung der Schaltungsteile (6. Woche)
- Review der Schaltungsteile des Kollegen
- Platine routen
- Finalisieren für die Produktion
- Implementierung der Hardware-Abstraktionsschicht und der Treiber während die Platine in der Produktion ist
- Schritt für Schritt Inbetriebnahme der Komponenten mit deren Treiber (9. Woche)
- Portierung und Integration des LUA Interpreters

### 3.2 Umsetzung

Es ist damit zu rechnen das die Steuerung starken ESD Störungen ausgesetzt ist. Deshalb müssen alle Komponenten möglichst robust ausgeführt werden, sowie müssen spezielle Schutzbeschaltungen implementiert werden. Zusätzlich soll die Steuerung in verschiedensten Maschinen eingesetzt werden, deshalb sollte diese modular vergrößert werden können. Da es dabei zwangsläufig zu unterschiedlichen Potentialen kommt, müssen möglichst viele Teile der Steuerung zueinander galvanisch getrennt werden.



Nach der gemeinsamen Kozeptionierung wurden die Themen aufgeteilt. Mein Fokus lag in den folgenden Punkten.

- Power Management
- digitale Eingänge (DI)
- digitale Ausgänge (DO)
- analoge Eingänge (AI)
- USB Host / Slave / OTG
- Serielles debug Interface
- Skriptsprache LUA
- Planung der Bestückung
- Ethernet Kommunikationskonzept

Die meisten der genannten Punkte umfassten Hardware sowie Software Entwicklung. Die Platine wurde mittels Altium Designer erstellt, in diesem Tool konnten alle Schaltpläne sowie das Layout erstellt werden. Dieses musste jedoch zu beging des Praktikums erlernt werden. Im Weiteren werden einige der Punkte beschrieben.

### 3.2.1 Power Management

Um die Platine und alle Bauteile mit ausreichend Spannung zu versorgen musste eine spezielle Spannungsversorgung implementiert werden. Auf der Platine werden 3 möglichst getrennte Spannungen benötigt. Die Erzeugung ist zweistufig. Die erste Stufe ist ein Schaltregler, welcher von einer Eingangsspannung zwischen 6 und 36 Volt konstante 5 V erzeugt, bei bis zu 2 Ampere. Es wurde ein spezieller integrierter Schaltregler von Texas Instruments gewählt, da dieser einen sehr großen Arbeitsbereich aufweist. Wird mehr Leistung benötigt gibt es von Texas Instruments alternative Bauteile, welche bis zu 10A liefern können. Diese sind kompatibel und könnten jederzeit getauscht werden, jedoch sollten 2 A mehr als ausreichend sein. Diese 5 V werden nach Spannungs- und Strom- Glättung als Versorgung der meisten Peripherie Bauteile verwendet. In der zweiten Stufe wird aus den 5 V mittels einem Linearregler 3,3 V erzeugt. Diese wird unmittelbar am Ausgang des Reglers zwei mal separat gefiltert, da diese Spannung einerseits als Versorgung für den Analogteil verwendet wird, sowie als Versorgung der CPU und dem Ethernetswitch dient. Diese Trennung ermöglicht es eine möglichst saubere Spannung dem Analogteil zur Verfügung zu stellen. Um die Spannungen möglichst verlustfrei über die Platine zu verteilen, besteht diese aus 4 Lagen - Versorgung / Signale / Signale / Erde. Eingangsseitig ist die Schaltung durch TVS Dioden gegen Überspannung geschützt. Dieses Design stellte sich später in der Testphase als sehr zuverlässig heraus. Lediglich die Wärmeentwicklung ist etwas höher als angenommen, jedoch immer noch im Rahmen. Der Wärmeentwicklung könnte mittels einer durchkontaktierter

Fläche Abhilfe geschaffen werden. Die Implementierung wurde grundsätzlich zu Beginn der Designphase durchgeführt, wurde jedoch im Laufe der Entwicklung mehrmals verändert, um den Leistungsanforderungen gerecht zu werden.

### 3.2.2 digitale Eingänge

Auf der Platine befinden sich 16 digitale Eingänge. Diese werden in 2 Gruppen zu je 8 Eingänge geteilt. Dabei sind alle Eingänge vom System galvanisch getrennt, sowie die Gruppen zueinander. Jede Eingangsgruppe hat ihre eigene Versorgungsspannung diese muss extern angeschlossen werden. Es ist auch möglich die Spannung mit der Systemspannung zu speisen, dann jedoch geht die galvanische Trennung verloren. Jeder Eingang ist separat mit einem Dioden Array ESD geschützt, sowie jede Gruppe mittels einer TVS Diode gegen Überspannung. Die erste Implementierung sah für die galvanische Trennung Optokoppler vor. Diese bieten jedoch wenig Flexibilität und sind relativ groß. Schlussendlich wurde ein spezieller Bauteil von Infineon verbaut. Mittels diesem Bauteil ist es möglich über den gesamten Spannungsbereich gültige Zustände zu erkennen. Sowie können Kurzschlüsse und Kabelbrüche erkannt werden. Dieser Bauteil ist speziell für industrielle Steuerungen gedacht und eignet sich dafür hervorragend. Zusätzlich wurde zu jedem Eingang eine LED hinzugefügt, welche den Status des Eingangs zeigt. Dieser Bauteil ist mittels einem parallelen Bus an die CPU angebunden. Um die galvanische Trennung über alle Lagen der Platine sicher zu stellen wurden auch die Versorgungslagen unterbrochen und eigene Flächen mit der getrennten Spannung angelegt.

### 3.2.3 analoge Eingänge

Es sollten 8 analoge 0 - 10 Volt Eingänge konstruiert werden. Um Spannungen von 0 - 10 Volt zu messen reicht grundsätzlich ein ADC und ein Spannungsteiler. Jedoch entsteht dann ein Problem, wenn die Spannung 10V überschreitet, das kann den ADC beschädigen. Dieses Problem wurde mittels einer Clamping Schaltung gelöst. Diese begrenzt die Spannung auf ca. 11 Volt, diese Spannung ist mittels dem ADC gerade noch messbar und dadurch kann eine Überspannung erkannt werden. Jeder Eingang wurde mittels ESD Schutzdioden zusätzlich geschützt. Die galvanische Trennung der analogen Signale stellte sich als sehr knifflig heraus. Es gibt zwar spezielle getrennte Operationsverstärker, jedoch sind diese unerschwinglich teuer. Das Trennen der analogen Signale ist dadurch unwirtschaftlich, weshalb die Trennung nach dem ADC durchgeführt wurde. Der ADC ist mittels SPI an den Prozessor angebunden. Zusätzlich benötigt der ADC eine Logikspannung. Es musste also nicht nur der SPI getrennt werden sondern auch die Spannung. Dazu wurde ein ADUM Bauteil von Analog Devices verwendet,

dieser ist in unterschiedlichsten Ausführungen erhältlich und stellt derzeit die modernste galvanische Trennung dar. In diesem Fall wurde ein ADUM mit SPI und DC-DC Wandler verbaut. Die getrennte Spannung nach dem ADUM ist jedoch alles andere als sauber und nicht direkt für den ADC geeignet, weshalb die Spannung mittels einem LDO erneut geglättet wird. Zusätzlich musste ein Level Shifter verbaut werden da die Spannungslevel des SPI von 5 Volt auf 3,3 Volt umgesetzt werden mussten. Um übersprechen der einzelnen analogen Signalkanäle zu minimieren wurden diese geschirmt geroutet. Der Analogteil konnte am Prototyp ohne Probleme in Betrieb genommen werden und erfüllte dabei alle Anforderungen.

### 3.2.4 digitale Ausgänge

Diese sind Grundsätzlich gleich wie die digitalen Eingänge aufgebaut. Es wurde erneut ein Bauteil von Infineon verwendet. Dieser beinhaltet nicht nur die galvanische Trennung sondern auch einen Leistungstreiber. So kann jeder Ausgang mit bis zu 40V betrieben, sowie mit bis zu 1,1A belastet werden. Der verwendete Bauteil ist Kurzschlussfest und überwacht zusätzlich die Belastung der einzelnen Ausgänge.

### 3.2.5 USB Host / Slave / OTG

Für Erweiterungen sowie für Interaktion mit dem Benutzer wurde eine USB Schnittstelle eingeplant. Dieser kann als Host, Slave oder OTG betrieben werden. Dafür wurde eine USB A Buchse sowie eine micro USB Buchse verbaut. Der verwendete Prozessor besitzt bereits den dafür benötigten USB-Core. Es musste also nur noch die erforderliche Beschaltung realisiert werden. Als Schutz wurden wieder ESD Schutzdioden sowie TSV Dioden verbaut. Zusätzlich wurde ein kleiner Leistungsschalter mit Stromüberwachung verbaut. Dieser stellt USB Geräte maximal 500mA zur Verfügung. Werden die 500mA überschritten, wird die Versorgung abgeschaltet und der Prozessor erhält eine Störmeldung. Da USB ein schneller serieller Bus ist, müssen im Layout einige Eigenschaften beachtet werden. So dürfen die Längen der differentiellen Paare nur gering von einander abweichen. Sowie muss auch die Impedanz eingehalten werden. Diese Bestimmungen können im Altium Designer eingegeben werden dieser beachtet dann automatisch die Einhaltung.

### 3.2.6 Serielles debug Interface

Diese Schnittstelle ist vor allen während der Entwicklungsphase der Treiber sehr brauchbar. Da es eine einfache serielle Schnittstelle darstellt und sehr einfach unter Windows verwendet werden kann. Der verwendete Prozessor bietet eine Vielzahl an Schnittstellen, darunter auch einige UARTS. Diese Signale könnten einfach mittels einem Level Shifter zu RS232 Signale gewandelt werden, da jedoch RS232 Schnittstellen kaum noch verwendet werden,

wurde der UART via USB übertragen. Dazu wurde ein Bauteil von FTDI verwendet. Dieser ist sehr gut integriert und kommt ohne externe Bauteile aus, es wurden lediglich Schutzbauteile für den ESD Schutz verbaut. Dieser wird von Windows als virtuelle COM-Schnittstelle erkannt und ermöglicht eine Kommunikation mit bis zu 3 Mbaud.

### 3.2.7 Skriptsprache LUA

Um dem Endbenutzer eine einfache Programmierung ohne C / C++ zu ermöglichen wurde ein skriptsprachen Interpreter implementiert. Dazu wurde eLua verwendet. Der Endbenutzer kann einfach Lua Programme, erstellen diese haben eine VB ähnlichen Syntax. Da Lua natürlich im Vergleich zu C / C++ sehr langsam ist wurden alle wichtigen Treiber als C++ Methoden implementiert. Diese werden lediglich von Lua aufgerufen. So gibt es als Beispiel eine Methode MoveUntillInput diese bewegt einen Motor mit einstellbarer Geschwindigkeit und Beschleunigung in eine Richtung, bis ein spezieller Eingang geschaltet wird. Da diese Methode komplett in C++ implementiert wurde ist die Verzögerungszeit minimal.

### 3.2.8 Planung der Bestückung

Der erste Prototyp wurde händisch bestückt. Grundsätzlich wurde während der Design Phase darauf geachtet, nicht all zu kleine SMD Bauteile zu verwenden. So wurden BGA und QFN Gehäuse vermieden. Da einige Bauteile jedoch nur als QFN verfügbar sind, mussten diese Gehäuse verbaut werden. Um die Bestückung zu erleichtern wurde eine kleine Führung entwickelt, diese ermöglicht es Bauteile ohne zu zittern zu platzieren. Zusätzlich können die Bauteile mittels einem Vakuumstift gehalten werden. Im Ersten Schritt wurde mittels einer Schablone eine Lötpaste auf alle zu lötenden Pads aufgetragen. Danach wurden die Bauteile mittels der Hilfsvorrichtung platziert. Danach wurde die Platine mittels einem Reflow Ofen gelötet. In [Abbildung 3.1](#) sind einige Fotos der Entwicklungsphase zu sehen.

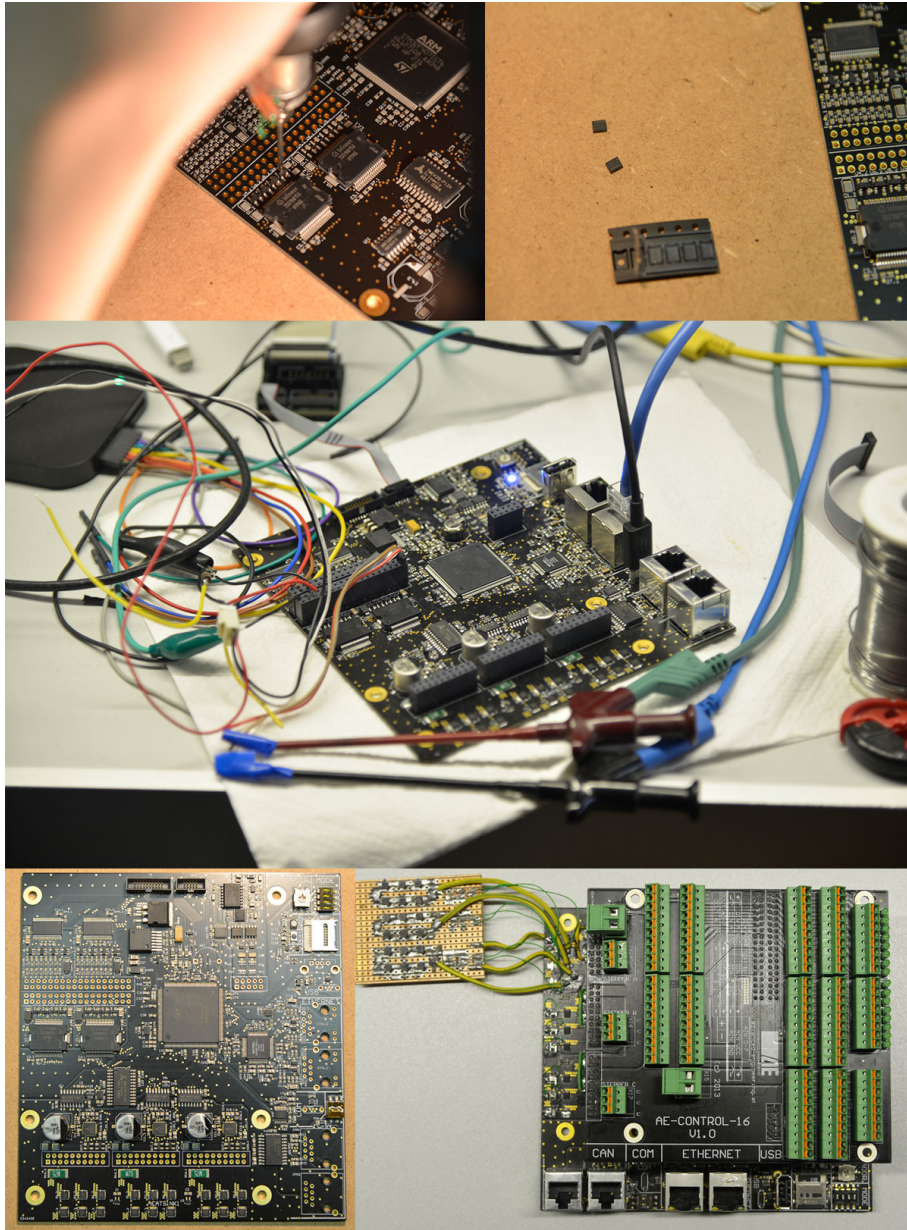


Abbildung 3.1: Einige Impressionen der Entwicklungsphase.

## Kapitel 4

# Erfahrungen und Zusammenfassung

Durch die Möglichkeit, sehr selbständig zu arbeiten konnte ich während des Praktikums sehr viele neue Erfahrungen sammeln. Die Firma Advanced Engineering GmbH stellte großes Vertrauen in meine Arbeit und dadurch konnte sehr frei gearbeitet werden. Es wurde nicht nur der Prototyp einer sehr viel versprechenden Steuerung entwickelt es wurde viel mehr ein ganzes Umfeld geschaffen. So wurde als Beispiel eine IDE für Arm Prozessoren auf Eclipse Basis zusammengestellt oder auch ein kleiner manueller Bestückungsroboter entwickelt, welcher es ermöglicht sehr einfach und genau Bauteile zu platzieren. Das Konzept der entwickelten Steuerung hat nicht nur mein Team sondern auch die Firma überzeugt und die Weiterentwicklung ist bereits besiegelt. Das gesamte Praktikum bestand zu 100 Prozent aus Teamarbeit und wäre anders auch nicht möglich gewesen. Im Team wurden immer wieder Phasen eingelegt in denen die Arbeit des anderen kontrolliert wurde. Dadurch konnten viele Fehler noch vor der ersten Testphase beseitigt werden. Erste Tests des Prototypen ergaben nur wenige kleine Fehler, diese konnten alle beseitigt werden, ohne neue Platinen zu fertigen. Im Team konnte ich mich mit meinem Kollegen jederzeit ergänzen und es wurden immer wieder neue Ideen geschmiedet. So existierten bereits während der Produktion des ersten Prototypen konkrete Ideen für weitere Entwicklungen. Derzeit arbeiten wir bereits an einer Weiterentwicklung unseres Projektes, diese umfasst einen viel Stärkere CPU, einen ARM Dual Core mit 1,8 GHz, sowie Anschlussmöglichkeiten für Bildschirme mit Touchscreen. Sowie wird die Netzwerkkommunikation überarbeitet, diese verwendet derzeit einfaches switched Ethernet. Die Weiterentwicklung erhält zusätzlich einen Echtzeit industriellen Ethernet Bus EtherCAT von Beckhoff. Die neue Version befindet sich derzeit noch in der Planungsphase. Ein erster Prototyp soll jedoch in den nächsten Monaten implementiert werden. Desweiteren gibt es Pläne einige Peripherien mittels der Verwendung von FPGA's zusammenzufassen.

Konkret bei der Ansteuerung von Motoren. So soll ein FPGA entwickelt werden welcher die unterschiedlichsten Motoren von Schrittmotoren bis zu BLDC ansteuern kann. Auch wird eLua in der neuen Version vermutlich durch Mono auf Realtime Linux Basis ersetzt. Grundsätzlich wurde eine sehr innovative Plattform geschaffen welche nun nach und nach Gestalt annimmt.