

# PRODUKTENTWICKLUNG 2

HOCHSCHULE LUZERN  
TECHNIK & ARCHITEKTUR

---

## Stepper Treiber

Konzeptbeschreibung

---

*Autoren:*  
Bettina WYSS  
Daniel WINZ

*Projektgruppe:*  
PREN-ET

Horw  
14. Juni 2015

## Inhaltsverzeichnis

<b>1</b>	<b>Fachgruppe Elektrotechnik</b>	<b>2</b>
<b>2</b>	<b>Schrittmotor</b>	<b>2</b>
2.1	Schritte . . . . .	3
2.2	Wicklungen . . . . .	3
<b>3</b>	<b>Stepper Motoransteuerung</b>	<b>5</b>
3.1	Grundsätzliches zur Ansteuerung . . . . .	5
3.2	Treiberstufe . . . . .	5
<b>4</b>	<b>Stepper Driver L6480</b>	<b>6</b>
4.1	Funktionsbeschreibung . . . . .	6
4.2	Schnittstelle . . . . .	6
4.3	Typical Application . . . . .	7
<b>5</b>	<b>Realisierung</b>	<b>8</b>
5.1	Hardware . . . . .	8
5.2	Schema . . . . .	8
5.3	Print Design . . . . .	11
5.4	Software . . . . .	13
5.4.1	Test auf Computer . . . . .	15

## 1 Fachgruppe Elektrotechnik

Elektrotechnik-Studierende aus mehreren Gruppen schliessen sich zusammen, um gemeinsame Probleme anzugehen. Dabei handelt es sich um die benötigte Hard- und Software, um Motoren anzusteuern und gegebenenfalls zu regeln. In diesem Zusammenschluss werden drei Gruppen gebildet, um Lösungen für DC-, Stepper- und Brushless-Motoren auszuarbeiten. Die Idee besteht darin, dass nicht jede Gruppe für dasselbe Problem womöglich denselben Lösungsansatz verfolgt, sondern die Ressourcen kombiniert, Synergien nutzt, um eine bessere Lösung zu erarbeiten. Auf diese Weise kann das teamübergreifende Arbeiten im Rahmen des PREN erlernt und geübt werden. Somit wird die Idee der Interdisziplinarität im erweiterten Sinn Rechnung getragen. Die Gruppen und deren Mitglieder sind in Tabelle 1 aufgeführt.

Team	Mitglied	Github-Name	DC	BLDC	Stepper
27	Daniel Winz	daniw		•	•
32	Yves Studer	ystuder		•	
33	Flavio Kreiliger	Flavinsky	•		•
38	Bettina Wyss	BettyET			•
39	Ervin Mazlagić	ninux	•		

Tabelle 1: Übersicht der PREN-ET Projektgruppen

Für den Austausch und die Ablage von Daten und Unterlagen wird die Plattform Github ausgewählt, da damit via Git<sup>1</sup> versioniert werden kann. Dazu wird auf Github die Organisation PREN-ET gegründet. Diese ist unter <https://github.com/PREN-ET> einsehbar. für die einzelnen Projekte werden Repositories angelegt. Diese sind in Tabelle 2 aufgeführt.

Repository	Link	Beschreibung
info	<a href="https://github.com/PREN-ET/info">https://github.com/PREN-ET/info</a>	Allgemeine Informationen zur Organisation von PREN-ET
doc	<a href="https://github.com/PREN-ET/doc">https://github.com/PREN-ET/doc</a>	Dokumentation
dc	<a href="https://github.com/PREN-ET/dc">https://github.com/PREN-ET/dc</a>	Treiber für Gleichstrommotoren
bldc	<a href="https://github.com/PREN-ET/bldc">https://github.com/PREN-ET/bldc</a>	Treiber für Brushless Motoren
stepper	<a href="https://github.com/PREN-ET/stepper">https://github.com/PREN-ET/stepper</a>	Treiber für Schrittmotoren
frdm	<a href="https://github.com/PREN-ET/frdm">https://github.com/PREN-ET/frdm</a>	Beispiele zur Ansteuerung mittels FRDM-KL25Z

Tabelle 2: Übersicht der PREN-ET Repositories

## 2 Schrittmotor

Schrittmotoren oder auch Stepper genannt, sind Synchronmotoren, bei welchen der Rotor um einen bestimmten Winkel gedreht werden kann. So ist die Rotorposition ohne zusätzliche Sensoren bekannt. Dabei ist zu beachten, dass der Motor keine Schritte verliert, was bei Überlast geschehen kann. Da die meisten Schrittmotorensysteme Open- Loop Systeme sind, entsteht eine dauernde Positionsabweichung bei einem Schrittverlust. Grundsätzlich wird zwischen zwei Schrittmotortypen unterschieden:

- Permanentmagnetmotor
- Reluktanzmotor

Der Permanentmagnetmotor besitzt als Rotor einen Permanentmagneten. Beim Reluktanzmotor besteht der Rotor aus einem gezahnten Weicheisenkern. Permanentmagnetmotoren erreichen eine kleinere Schrittfrequenz, besitzen jedoch ein grösseres Drehmoment als Reluktanzmotoren. Die Kombination aus Reluktanzmotor und Permanentmagnetmotor ist ein Hybridmotor. Ein Hybridmotor verbindet die Vorteile von Reluktanz- und Permanentmotor.

<sup>1</sup>Verteiltes Versionskontrollsystem

## 2.1 Schritte

Der Vollschrittbetrieb kann einphasig oder auch zweiphasig gesteuert werden. Beim einphasigen Vollschrittbetrieb sind immer zwei gegenüber liegende Pole aktiv. Beim zweiphasigen Vollschrittbetrieb werden jeweils zwei nebeneinander liegende Pole aktiv. Im Halbschrittbetrieb werden die beiden Vollschrittbetriebsarten kombiniert. So kann der Schrittwinkel halbiert werden. Zusätzlich kann der Schrittmotor mit Mikroschritten betrieben werden. Dabei folgt der Strom der sinusförmigen Referenzspannung. (Vgl. Seite 5)

Wie bei anderen Motoren hat auch bei Schrittmotoren die Anzahl Polpaare einen Einfluss auf dessen Drehzahl.

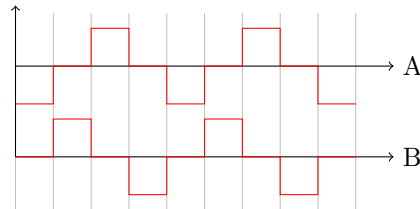


Abbildung 1: Vollschritt

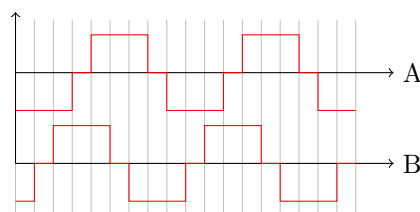


Abbildung 2: Halbschritt

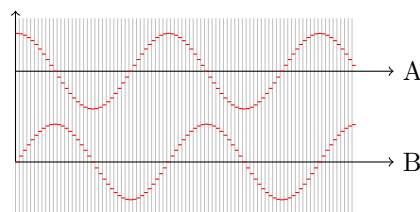


Abbildung 3: Mikroschritt

Die Polpaare bilden eine Übersetzung zwischen der elektrischen und der mechanischen Rotation. Die meisten gängigen Schrittmotoren besitzen 50 Polpaare. Da eine elektrische Umdrehung aus vier Schritten besteht ergeben sich bei 50 Polpaaren 200 Schritte für eine Umdrehung. Dies ergibt eine Schrittauflösung von  $1.8^\circ$  bei Vollschrittbetrieb.

## 2.2 Wicklungen

Ein Schrittmotor besitzt zwei getrennte Wicklungen. Diese sind rechtwinklig zueinander angeordnet. Bei Motoren mit mehr als einem Polpaar sind entsprechend mehr Wicklungen verbaut. Beim Aufbau der Wicklungen wird zwischen unipolaren und bipolaren Wicklungen unterschieden. Die unipolare Wicklung besitzt einen Mittelabgriff. Dieser wird meistens mit der Versorgungsspannung verbunden. Die beiden anderen Enden werden nun mit einer Lowside Treiberstufe angesteuert. Dadurch kann der magnetische Fluss mit einem geringen elektronischen Aufwand umgekehrt werden. Jedoch ist immer nur eine Hälfte der Wicklung aktiv. Zudem ist die Herstellung aufwendiger und somit auch teurer. Bei einer bipolaren Wicklung fehlt der mittlere Abgriff. Die Umkehrung des magnetischen Flusses muss über die Umkehrung der angelegten Spannung erfolgen. Deshalb muss für beide Wicklungen jeweils eine Brückenschaltung verwendet werden. Da bei bipolaren Schrittmotoren immer die gesamte Wicklung verwendet wird, kann damit ein grösseres Drehmoment erzeugt werden als mit einem gleich grossen unipolaren Schrittmotor. Die Ansteuerung der beiden Wicklungsarten ist in Abbildung 4 ersichtlich. (?, ?)

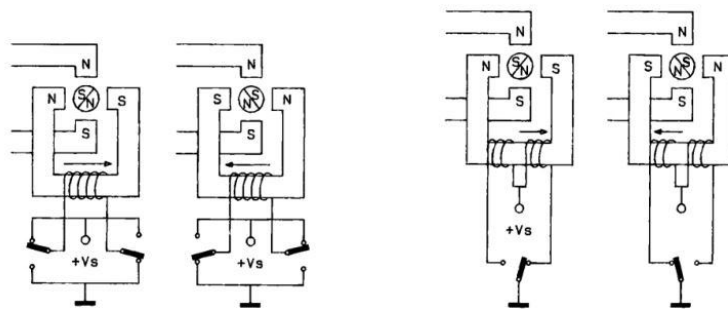


Abbildung 4: bipolarer und unipolarer Betrieb (?, ?)

### 3 Stepper Motoransteuerung

#### 3.1 Grundsätzliches zur Ansteuerung

Grundsätzlich besteht die Ansteuerung aus drei Teilen, wie in Abbildung 5 gezeigt. Um die Ansteuerung zu realisieren, gibt es eine Vielzahl von integrierten Schaltkreisen. Diese unterscheiden sich wie folgt:

- **Interfaces:** Einzelanschlüsse, einfache Busschnittstellen oder Mikrocontrollerschnittstellen wie SPI, I2C
- **Steuerfunktionen:** Einzelne Schritte oder Bewegungsabläufe (Motion Control Function)
- **Schaltungsintegration:** Steuerung und Treiberstufe als getrennte Schaltkreise, oder in einem Schaltkreis zusammengefasst.

(?, ?)

Der gewählte integrierte Schaltkreis ist der L6480 von STMicroelectronics. Dieser wird über die SPI Schnittstelle gesteuert und besitzt eine Motion Control Engine. Die Treiberstufe wird extern realisiert. (Vgl. Kapitel 4)



Abbildung 5: Komponenten der Ansteuerung eines Schrittmotors

#### 3.2 Treiberstufe

Wird ein Schrittmotor unipolar betrieben, so können die vier Wicklungen direkt mit Lowside Treibern angesteuert werden. Für den bipolaren Betrieb benötigt man für beide Wicklungen je eine H-Brücke. Die einfachste Methode ist es, den Strom nur durch den Wicklungswiderstand zu begrenzen. Der Nachteil ist, dass die Zeitkonstante durch den Wicklungswiderstand und die Induktivität bestimmt ist, und so bei höheren Schrittfrequenzen der gewünschte Strom und damit das Drehmoment nicht mehr erreicht wird. Deshalb wird ein zusätzlicher Vorwiderstand in Serie geschaltet, und so die Zeitkonstante verkleinert. Typische Verhältnisse sind vierfacher- oder fünffacher Widerstand, was eine vierfache bzw. fünffache Speisespannung voraussetzt. Diese Methode wiederum führt zu einer höheren Verlustleistung in den Widerständen. Im Ruhezustand ist es sinnvoll, den Strom soweit zu senken, dass das Haltemoment nicht unterschritten wird. Eine Spannungsumschaltung hat den weiteren Vorteil, dass so beim Anfahren eine steilere Stromkurve erreicht werden kann. (Vgl. Abbildung 6) Der Schrittmotor kann

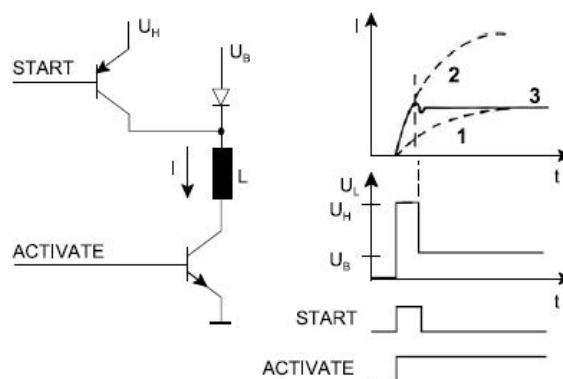


Abbildung 6: Spannungsumschaltung (?, ?)

alternativ auch stromgesteuert betrieben werden. Dabei folgt der Stromverlauf dem Verlauf einer Referenzspannung (Sollwert). Der Stromverlauf wird auf den Sollwert geregelt. Die Betriebsspannung muss so nicht stabilisiert werden.

## 4 Stepper Driver L6480

Wie bereits im Kapitel 3.1 erwähnt, wird der L6480 von STMicroelectronics verwendet.

### 4.1 Funktionsbeschreibung

Diese Schrittmotorensteuerung ist für den Betrieb von zweiphasigen (Vgl. Kapitel 2) Schrittmotoren mit Mikrosteps (Vgl. Kapitel 2) geeignet. Der L6480 erreicht eine maximale Auflösung von einem 1/128 Schritt. Die Steuerung generiert intern die PWM- Signale für die Motorenansteuerung. Alternativ kann auch mit Vollschritten oder Halbschritten gearbeitet werden. Die beiden H- Brücken werden extern mit N-Kanal MOSFETs realisiert. Es können Bewegungsprofile konfiguriert werden, so dass die Motoren definiert anfahren, abbremsen oder ein Punkt direkt angefahren werden kann. So kann der Aufwand bei der Mikrocontrollerprogrammierung verringert werden. Die Befehle werden über eine SPI- Schnittstelle übertragen. Die absolute Position ist in einem 22- Bit Register gespeichert. Der Bereich liegt dementsprechend zwischen  $-2^{21}$  und  $2^{21} - 1$ . (?, ?)

### 4.2 Schnittstelle

Der steuernde Mikrocontroller benötigt 8 Pins für die Kommunikation mit dem L6480 (?, ?) :

Pin	IO	Funktion
$\overline{FLAG}$	Output (Open Drain)	Wird bei einem Fehler intern auf GND gezogen.
$\overline{BUSY}$ / SYNC	Output (Open Drain)	Wird während dem Ausführen eines Befehls intern auf GND gezogen.
$\overline{STBY}/\overline{RESET}$	Input	Standby- und Resetmodus, falls extern GND anliegt.
STCK	Input	Im Step-Clock- Mode führt jede positive Flanke an diesem Pin zu einem Schritt.
<b>SPI</b>		
$\overline{CS}$	Input	Chip Select: Falls extern GND anliegt, startet die Kommunikation. Um die Kommunikation zu beenden, muss $\overline{CS}$ extern auf High gehalten werden.
CK	Input	Serial Clock: Synchronisierung der Kommunikation.
SDO	Output	Slave Data Out: Daten für den Mikrocontroller.
SDI	Input	Slave Data In: Befehle und Daten für den L6480.

Tabelle 4: Schnittstelle des Treibers L6480

### 4.3 Typical Application

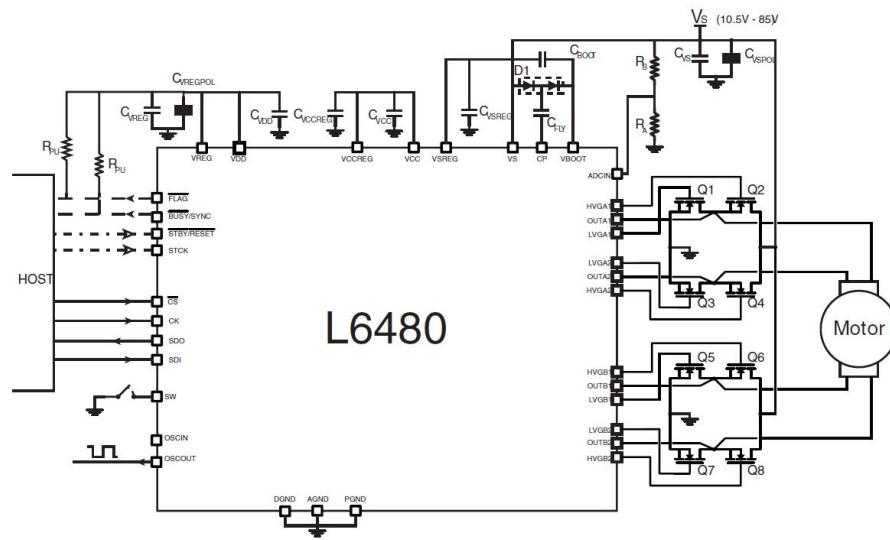


Abbildung 7: Typical Application (?, ?)



## 5 Realisierung

### 5.1 Hardware

Der L6480 besitzt, wie in Abschnitt 4 beschrieben, eine SPI-Schnittstelle. Über diese Schnittstelle soll der Steppertreiber die Befehle des Freedomboards erhalten. Der Steppertreiberprint wird mit Stiftleisten bestückt und kann so direkt auf das Freedomboard aufgesteckt werden. Es wird so keine Kabelverbindung benötigt und die Elektronik bleibt kompakt.

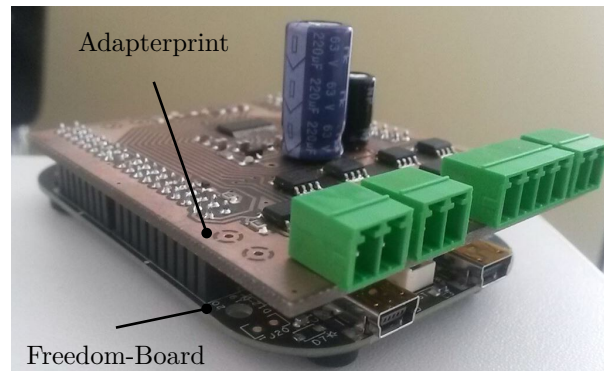


Abbildung 8: Freedomboard und Steppertreiberprint

Der Steppertreiberprint beinhaltet grundsätzlich den Treiber für den Schrittmotor, realisiert mit dem Baustein L6480 und einer externen H-Brücke aus N-FET. Zusätzlich sind Anschlüsse für andere Motoren (z.B. BLDC-Motor) auf dem Print integriert. Im Folgenden wird die Entwicklung dieses Adapterprints vorgestellt. Ab Seite 8 ist das Schema der Schaltung und ab Seite 11 die Beschreibung des Printdesigns zu finden.

### 5.2 Schema

Als Motorencontroller wird der integrierte Schaltkreis L6480 von STMicroelectronics verwendet. Dieser wird über eine SPI-Schnittstelle angesteuert. Der L6480 bietet die Möglichkeit, Bewegungsprofile zu konfigurieren. Die verschiedenen Betriebsarten werden im Kapitel 5.4 vorgestellt. Die H-Brücke wird extern mit N-FETs realisiert. Die Beschaltung des L6480 kann aus dem Datenblatt (? , ?) entnommen werden.

Das revidierte Schema ist auf Seite 10 abgebildet.

#### Speisung

Es gibt zwei Möglichkeiten, wie das IC gespeist werden kann:

- Motorenspannung, interne Spannungsregler generieren die nötige Gatespannung sowie die Logikspannung
- Externe Spannungsregler

Es wird die erste Möglichkeit, die Speisung mit nur der Motorenspannung gewählt. Als Motorenspeisung ist ein Bereich von 10.5 ... 85V zulässig. Bei einer Betriebsspannung von mehr als 25V sind MOSFET mit einer höheren Sperrspannung notwendig. Zusätzliche Spannungsregler sind nicht notwendig.

#### Motor supply voltage compensation

Der Motorcontroller bietet die Möglichkeit, Schwankungen der Motorenspannung zu erkennen und so die Amplitude des PWM-Sinussignales am Schrittmotor zu regeln. Dazu muss der Eingang ADCIN des Controllers korrekt beschaltet werden. An diesem Analogeingang soll bei korrekter Motorspannung 1/2 der Logikspannung anliegen. In diesem Fall bedeutet dies, dass bei normaler Betriebsspannung 1.65V über dem Widerstand R4 anliegen müssen.

#### LED Speisung

Damit sofort ersichtlich ist, ob der Adapterprint gespiesen ist, wird eine LED vorgesehen. Diese wird zwischen die Speisung über einen Vorwiderstand an GND angeschlossen. Somit ist alles in Ordnung, falls die LED leuchtet.

### LED Fehler

Der Motorencontroller kann so konfiguriert werden, dass verschiedene Fehler angezeigt werden können. Dies beinhaltet zum Beispiel Schrittverluste, Überstrom, Unterspannungserkennung oder Überhitzung des Controllers. Dazu dient ein Open-Drain Pin  $\overline{FLAG}$ , welcher im Fehlerfall auf GND zieht. Der Pin wird so beschaltet, dass eine LED den Fehlerfall auch optisch sichtbar macht.

### Pinbelegung

Die Kommunikation mit dem Motorencontroller und die Ansteuerung des Pneumatikventils wird mit dem Freedom-Board realisiert. Die folgende Tabelle stellt die Schnittstelle zwischen dem Freedomboard und den Anschlüssen auf dem Adapterprint dar.

Pin	Stepper-board	FRDM-KL25Z
<b>IO L6480</b>		
$\overline{FLAG}$	J2 Pin 2	PTA13
$\overline{BUSY}$ / SYNC	J1 Pin 2	PTA1
$\overline{STBY/RESET}$	J2 Pin 11	PTA17
STCK	J2 Pin 9	PTA16
<b>SPI L6480</b>		
$\overline{CS}$	J9 Pin 13	PTE4
CK	J9 Pin 9	PTE2
MOSI	J9 Pin 11	PTE3
MISO	J2 Pin 20	PTE1
<b>Pneumatik</b>		
Ventil	J2 Pin 18	PTE0

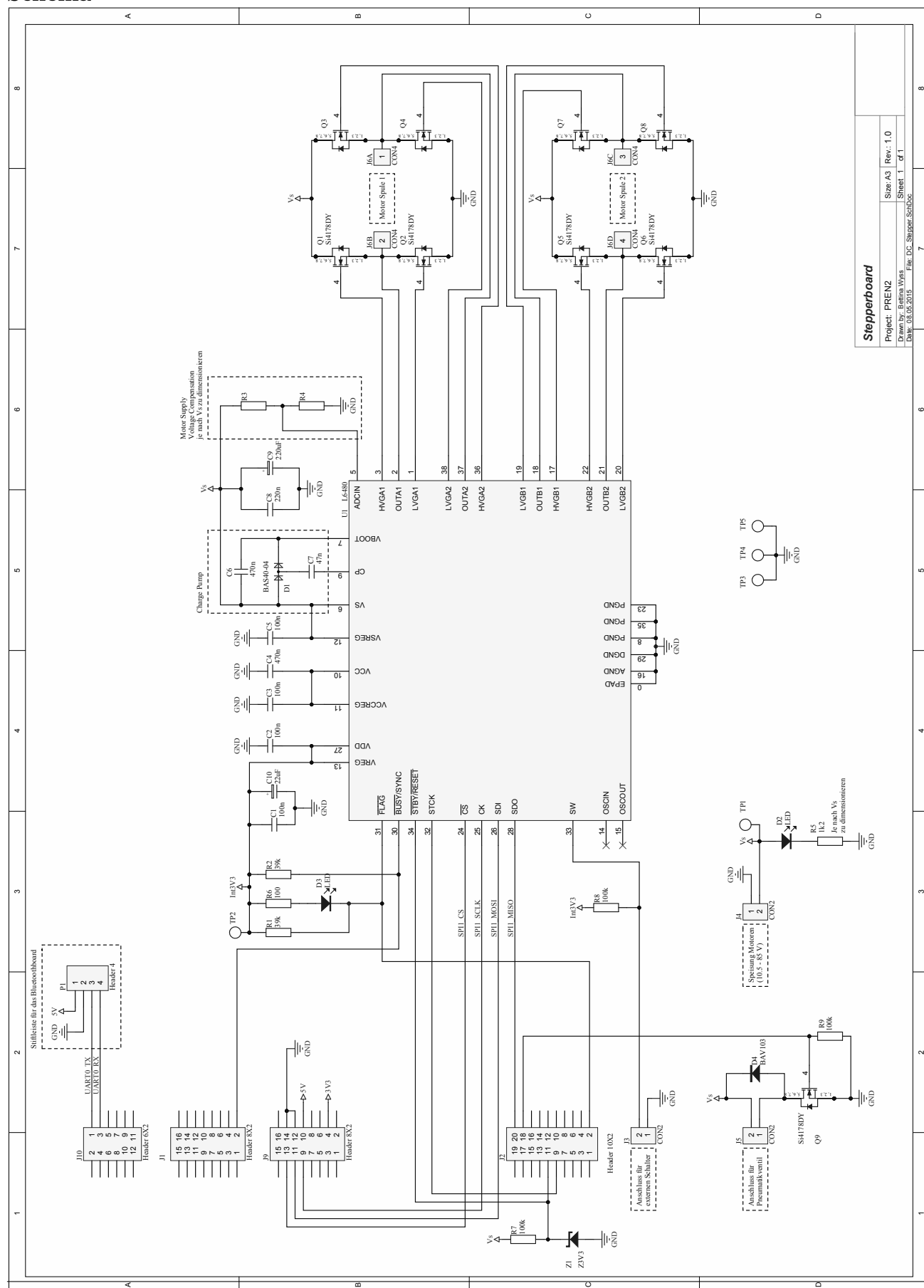
Tabelle 6: Pinbelegung

### Zusätzliche Bestückung

Zusätzlich kann ein Bluetoothmodul bestückt werden. Weitere Bestückungen...

Es wird mit dem Tool Altium Designer gearbeitet. Das überarbeitete Schema des Steppertreiberprints ist auf der Seite 10 zu finden.

## Schema



14. Juni 2015

10

## Bestückungsdokumente

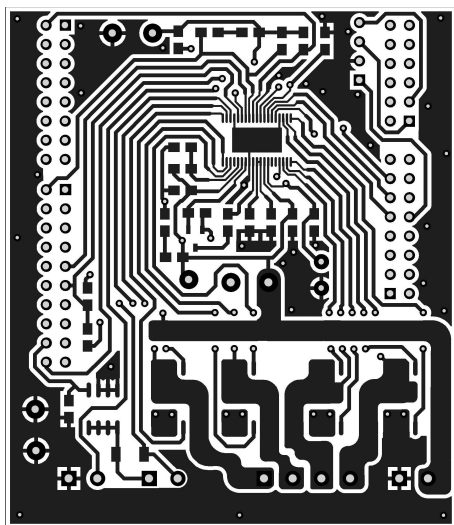


Abbildung 9: Top Layer

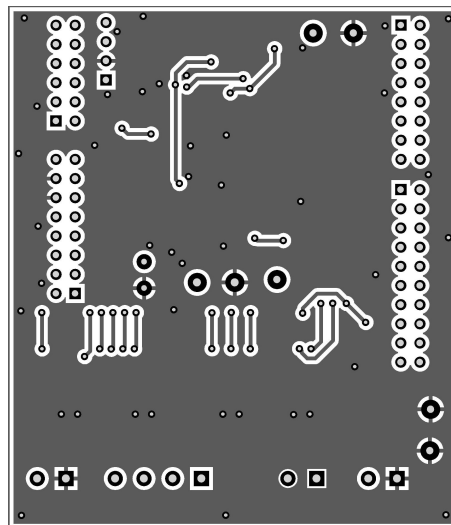


Abbildung 10: Bottom Layer

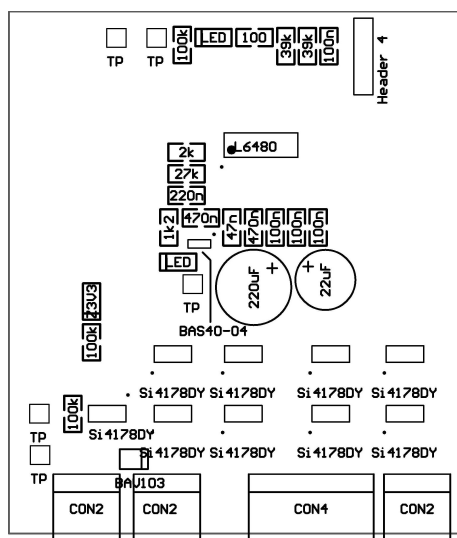


Abbildung 11: Werte

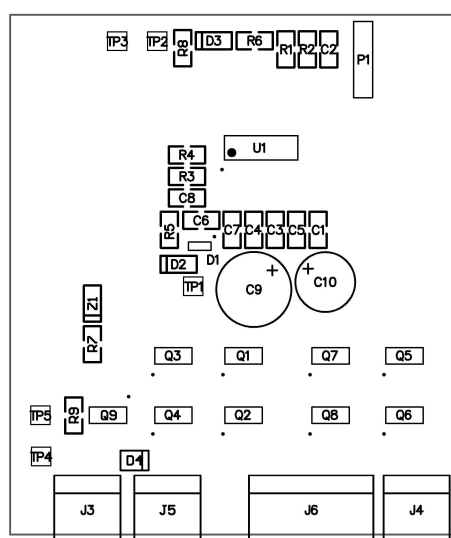


Abbildung 12: Bezeichnungen

### 5.3 Print Design

Der Adapterprint soll auf das Freedom-Board aufgesteckt werden und möglichst klein sein. Der Print hat eine Grösse von 60mm x 70mm. Darauf befinden sich die Anschlüsse für die Speisung, für den Motor, für weitere Motoren und für einen End- oder Notschalter. Dazu werden stabile Leiterplattenanschlüsse gewählt, welche auch die hohen Phasenströme des Motors aushalten und zudem eine genug hohe Spannungsfestigkeit besitzen, da die Motorenspannung bis 85V gewählt werden kann. Die eingesetzten Leiterplattenanschlüsse sind für eine Spannung bis 300V und einem Strom bis 8A geeignet. Dies ist ausreichend für diese Anwendung. Alle Bauteile sind SMD, ausser den Leiterplattenanschlüssen und den Kondensatoren.

Bauteil	Bezeichnung	Lieferant	Bestellnummer	Anz
Motor Contoller	L6480H	Mouser	511-L6480H	1
Shottkydiode	BAS40-04-G	Mouser	78-BAS40-04-E3-08	1
Zenerdiode	BZX585-B3V3	Mouser	771-BZX585-B3V3	1
Freilaufdiode	BAV103	Mouser	512-BAV103	1
FET	Si4178DY	PTA16	781-SI4178DY-TI-GE3	9
LED	LSQ976	Mouser	720-LSQ976-NR-1	2

Tabelle 8: Stückliste (Bauteile nicht an Lager)

Grundsätzlich werden folgende Regeln beim Design eingehalten:

- Leiterbahnbreite 20mil
- Abstände zwischen Leiterbahnen und Polygonen 20mil
- keine rechten Winkel
- möglichst wenig Auskreuzungen auf dem Bottomlayer
- genügend grosse Pads
- Verbindungen möglichst kurz halten

Da die Pins des L6480 näher als 20mil zueinander sind, können dort die Regel für die Leiterbahnbreite und für den Abstand nicht eingehalten werden. Deshalb werden für dieses Bauteil eigene Regeln definiert. Die Leiterbahnen sind um den L6480 nur 8mil breit, werden jedoch sofort auf 20mil verdickt. Die Motorenphasen werden nicht mit Leiterbahnen an die Anschlüsse geführt, sondern mit breiten Polygonen verbunden. Die Auskreuzungen auf dem Bottomlayer werden vermieden, da einerseits der erste Prototyp ohne Durchkontaktierung hergestellt wird, andererseits da die GND-Fläche auf dem Bottomlayer möglichst nicht "verschnitten" werden soll, um so EMV-Störungen abzuschirmen. Nach dem Layouten des ersten Prototypes folgt das Bestellen der Bauteile, welche nicht im Elektroniklabor vorhanden sind. Die Stückliste ist in der Tabelle 8 zu finden. Der erste Prototyp wird von Hand durchkontaktiert. Nach dem Bestücken mit der halbautomatischen Bestückungsmaschine und dem Löten im Ofen wird der Prototyp in Betrieb genommen. Die Inbetriebnahme zeigt, dass einige wenige Änderungen im Layout nötig sind. Diese werden in einer überarbeiteten Version des Adapterprints berücksichtigt. Der zweite Print wird maschinell durchkontaktiert. Die Dokumente zur ersten Version sind im Anhang zu finden. Die Bestückungsdokumente sind auf Seite 11 abgebildet.

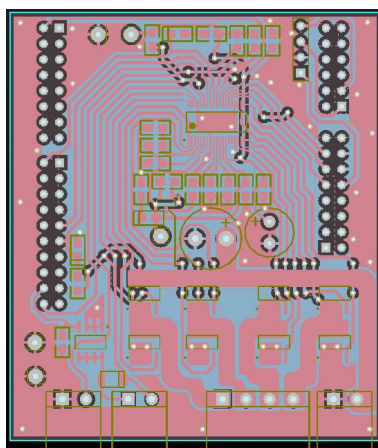


Abbildung 13: Printdesign mit Altium Designer

## 5.4 Software

Für die einfache Einbindung des Schrittmotortreibers in die Software der jeweiligen Teams wird eine Bibliothek erstellt. Diese besteht aus den Dateien `16480.c` und `16480.h`, welche in eigene Software eingebunden werden kann.

Listing 1: Einbinden der Bibliothek

```
1 #include 16480.h
```

Diese Bibliothek bietet Funktionen um alle Register des Schrittmotortreibers zu beschreiben und zu lesen. Ausserdem ist jeder Befehl des Treibers implementiert. Die Bibliothek ist mit Doxygen dokumentiert. In der entsprechenden Dokumentation ist die Schnittstelle der Bibliothek beschrieben. Aufgrund des Umfangs ist diese nicht im Anhang aufgeführt. Sie kann aber mit Doxygen generiert und als separates Dokument abgegeben werden. Damit die Bibliothek mit dem Schrittmotortreiber kommunizieren kann, muss eine SPI Schnittstelle zur bereitgestellt werden. Dazu sind die Funktionen `spi_write(uint8_t *data)` und `spi_read(uint8_t *data)` so zu ergänzen, dass sie jeweils ein Byte über SPI senden respektive empfangen. Über die Konstante `PL_FRDM` ist die SPI Schnittstelle auf einem FRDM-KL25Z von Freescale einschaltbar. Alle dafür notwendigen Komponenten sind in Tabelle 9 aufgeführt.

Listing 2: Definition der Plattform für das FRDM-KL25Z

```
1 #define PL_FRDM
```

Name	Komponente	Beschreibung
Stepperspi	SynchroMaster	SPI
WAIT1	Wait	Delays
STP_BSY	ExtInt	Interrupt für das Signal Busy

Tabelle 9: Komponenten bei Verwendung auf einem FRDM-KL25Z

Zudem kann mit der Konstante `PL_HAS_SHELL` die Unterstützung einer Shell-Komponente aktiviert werden. Die für die Shell zusätzlich notwendigen Komponenten sind in Tabelle 10 aufgeführt.

Listing 3: Definition für Shell-Unterstützung

```
1 #define PL_HAS_SHELL
```

Name	Komponente	Beschreibung
CLS1	Shell	Shell
UTIL1	Utility	Funktionen für die Verarbeitung von Strings
Shell	-	Funktionen für das Senden von Strings über die Shell (teamspezifisch)

Tabelle 10: Komponenten bei Verwendung der Shell

Folgende Befehle werden vom Parser der Bibliothek geparkt:

Befehl	Beschreibung
16480 help	Hilfe zur Shell
16480 status	Status des Treibers
16480 run (f r) <speed> (f r) f r <speed>	Drehung mit konstanter Geschwindigkeit Drehrichtung forward → Vorwärts reverse → Rückwärts Geschwindigkeit
16480 goto (f r) <position> (f r) f r <position>	absolute Positionierung Drehrichtung forward → Vorwärts reverse → Rückwärts Position in Anzahl Schritten
16480 move (f r) <steps> (f r) f r <steps>	relative Positionierung relative Positionierung forward → Vorwärts reverse → Rückwärts relative Positionierung
16480 reset	Anhalten mit Rampe
16480 hardstop	sofortiges Anhalten
16480 softstop	Reset des Treibers
16480 init (f r) <speed> (f r) f r <speed>	Initialisierung des Treibers Initialisierung des Treibers forward → Vorwärts reverse → Rückwärts Initialisierung des Treibers
16480 home go	Drehung zu Initialposition
16480 home set	Setze aktuelle Position als Initialposition
16480 softhiz	Anhalten mit Rampe mit anschliessendem deaktivieren der Treiberstufe
16480 kval (h r a d) <value> (h r a d) h r a d <value>	Setzen des Motorstroms der einzelnen Betriebsphasen Betriebsphase hold → Stillstand run → Fahrt mit konstanter Geschwindigkeit acceleration → Beschleunigung deceleration → Abbremsen Wert für Motorstrom

Tabelle 11: Befehle der Shell

### 5.4.1 Test auf Computer

Wird die Konstante PL\_FRDM nicht gesetzt, kann die Bibliothek auf einem Computer kompiliert und getestet werden. Das ermöglicht das Testen der Bibliothek ohne Hardware. Die Daten werden dabei nicht über SPI geschickt, sondern auf die Standardausgabe ausgegeben.

Listing 4: Beispielprogramm für den Test der Bibliothek auf dem Computer

```
1 #include "stdio.h"
2 #include "drv/l6480.h"
3
4 int main(void) {
5     l6480_init();
6     printf("hardhiz\n");
7     l6480_cmd_hardhiz();
8     printf("run_1_2\n");
9     l6480_cmd_run(1, 2);
10    return 0;
11 }
```

Listing 5: Ausgabe vom Testprogramm in Listing 4

```
1 hardhiz
2 write: 0xA8
3 run 1 2
4 write: 0x51
5 write: 0x0
6 write: 0x0
7 write: 0x2
```



## Abbildungsverzeichnis

1	Vollschritt . . . . .	3
2	Halbschritt . . . . .	3
3	Mikroschritt . . . . .	3
4	bipolarer und unipolarer Betrieb . . . . .	4
5	Komponenten der Ansteuerung eines Schrittmotors . . . . .	5
6	Spannungsumschaltung . . . . .	5
7	Typical Application . . . . .	7
8	Freedomboard und Steppertreiberprint . . . . .	8
9	Top Layer . . . . .	11
10	Bottom Layer . . . . .	11
11	Werte . . . . .	11
12	Bezeichnungen . . . . .	11
13	Printdesign mit Altium Designer . . . . .	12

## Tabellenverzeichnis

1	Übersicht der PREN-ET Projektgruppen . . . . .	2
2	Übersicht der PREN-ET Repositorys . . . . .	2
4	Schnittstelle des Treibers L6480 . . . . .	6
6	Pinbelegung . . . . .	9
8	Stückliste (Bauteile nicht an Lager) . . . . .	12
9	Komponenten bei Verwendung auf einem FRDM-KL25Z . . . . .	13
10	Komponenten bei Verwendung der Shell . . . . .	13
11	Befehle der Shell . . . . .	14