

PRODUKTENTWICKLUNG 1

HOCHSCHULE LUZERN

TECHNIK & ARCHITEKTUR

Stepper Treiber

Konzeptbeschreibung

Autoren:

Bettina WYSS

Daniel WINZ

Projektgruppe:

PREN-ET

Horw
10. Juni 2015

Inhaltsverzeichnis

| | |
|---|-----------|
| 1 Fachgruppe Elektrotechnik | 2 |
| 2 Schrittmotor | 2 |
| 2.1 Schritte | 3 |
| 2.2 Wicklungen | 3 |
| 3 Stepper Motoransteuerung | 5 |
| 3.1 Grundsätzliches zur Ansteuerung | 5 |
| 3.2 Treiberstufe | 5 |
| 4 Stepper Driver L6480 | 6 |
| 4.1 Funktionsbeschreibung | 6 |
| 4.2 Schnittstelle | 6 |
| 4.3 Typical Application | 7 |
| 5 Realisierung | 8 |
| 5.1 Hardware | 8 |
| 5.2 Schema | 8 |
| 5.3 Print Design | 11 |
| 5.4 Software | 13 |
| 6 Ausblick | 14 |
| Literatur- und Quellenverzeichnis | 15 |

1 Fachgruppe Elektrotechnik

Elektrotechnik-Studierenden aus mehreren Gruppen haben sich zusammengeschlossen 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 wo möglich 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 Idee der Interdisziplinarität im erweiterten Sinn Rechnung getragen. Die Gruppen und deren Mitglieder sind in Tabelle 1 aufgeführt. Für den Austausch und die Ablage von Daten und Unterlagen

| Team | Mitglied | Github | 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

wird die Plattform Github ausgewählt, da damit via Git¹ 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 | https://github.com/PREN-ET/info | Allgemeine Informationen zur Organisation von PREN-ET |
| doc | https://github.com/PREN-ET/doc | Dokumentation |
| dc | https://github.com/PREN-ET/dc | Treiber für Gleichstrommotoren |
| bldc | https://github.com/PREN-ET/bldc | Treiber für Brushless Motoren |
| stepper | https://github.com/PREN-ET/stepper | Treiber für Schrittmotoren |
| frdm | https://github.com/PREN-ET/frdm | 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.

¹Verteiltes 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.

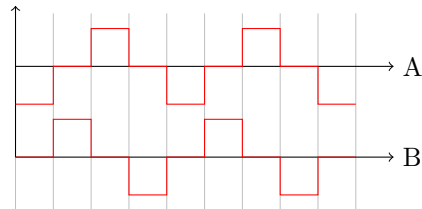


Abbildung 1: Vollschritt

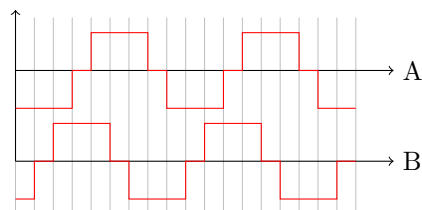


Abbildung 2: Halbschritt

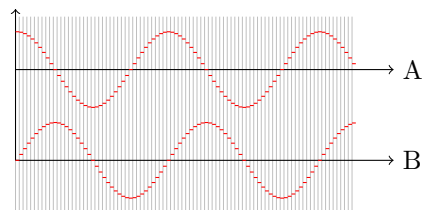


Abbildung 3: Mikroschritt

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° 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. (Prof. Dr. Wolfgang Matthes, 2007)

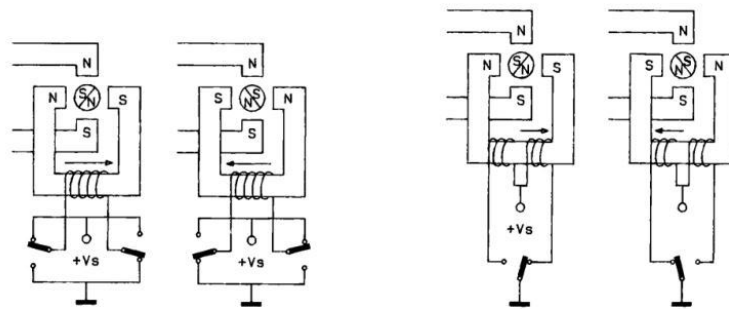


Abbildung 4: bipolarer und unipolarer Betrieb (Prof. Dr. Wolfgang Matthes, 2007)

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.

(Prof. Dr. Wolfgang Matthes, 2007)

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

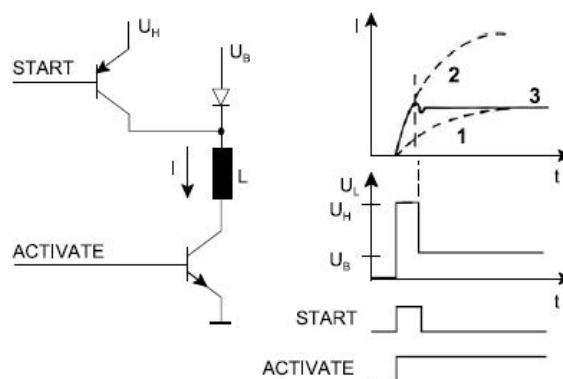


Abbildung 6: Spannungsumschaltung (Thomas Hopkins, 2012)

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 $2^{21} - 1$. (STMicroelectronics, 2012)

4.2 Schnittstelle

Der steuernde Mikrocontroller benötigt 8 Pins für die Kommunikation mit dem L6480 (STMicroelectronics, 2012) :

| 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. |
| SPI | | |
| \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

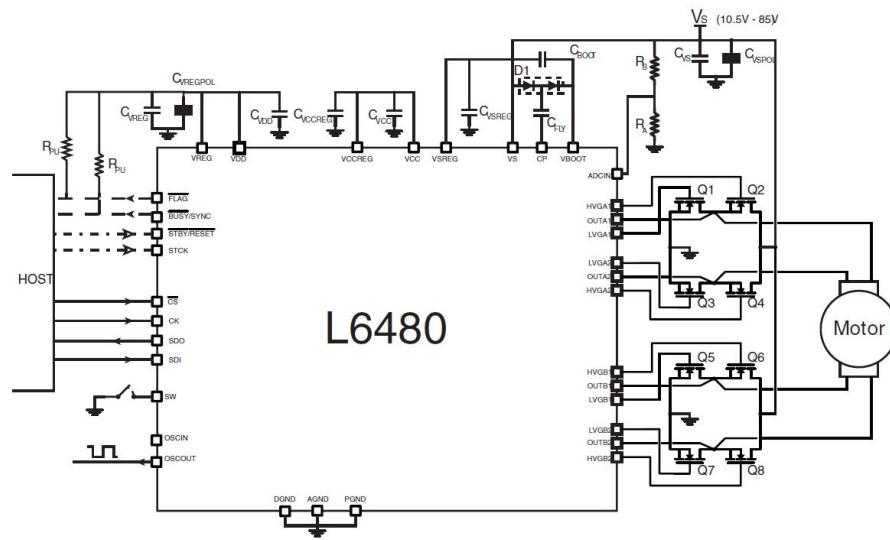


Abbildung 7: Typical Application (STMicroelectronics, 2012)

5 Realisierung

5.1 Hardware

Der L6480 besitzt, wie im 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.

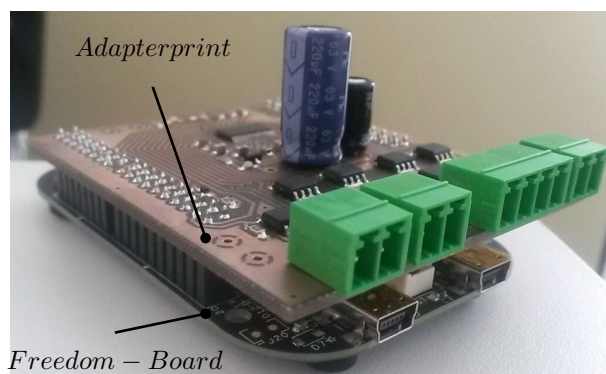


Abbildung 8: Freedomboard und Steppertreiberprint

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 Printdesignes zu finden.

5.2 Schema

Als Motorenkontroller 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 konnte aus dem Datenblatt (STMicroelectronics, 2012) 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 wurde die erste Möglichkeit, die Speisung mit nur der Motorenspannung gewählt. Als Motorenspeisung wurde 5V gewählt, da 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 5V Motorspannung 1.65V über dem Widerstand R4 anliegen müssen. Dimensioniert wurde für R4 10k und für R3 10k.

LED Speisung

Damit sofort ersichtlich ist, ob der Adapterprint gespeist ist, wurde 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 Motorenkontroller 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 wurde so beschaltet, dass eine LED den Fehlerfall auch optisch sichtbar macht.

Pinbelegung

Die Kommunikation mit dem Motorenkontroller 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 |
|--------------------------|---------------|------------|
| IO L6480 | | |
| \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 |
| SPI L6480 | | |
| \overline{CS} | J9 Pin 13 | PTE4 |
| CK | J9 Pin 9 | PTE2 |
| MOSI | J9 Pin 11 | PTE3 |
| MISO | J2 Pin 20 | PTE1 |
| Pneumatik | | |
| Ventil | J2 Pin 18 | PTE0 |

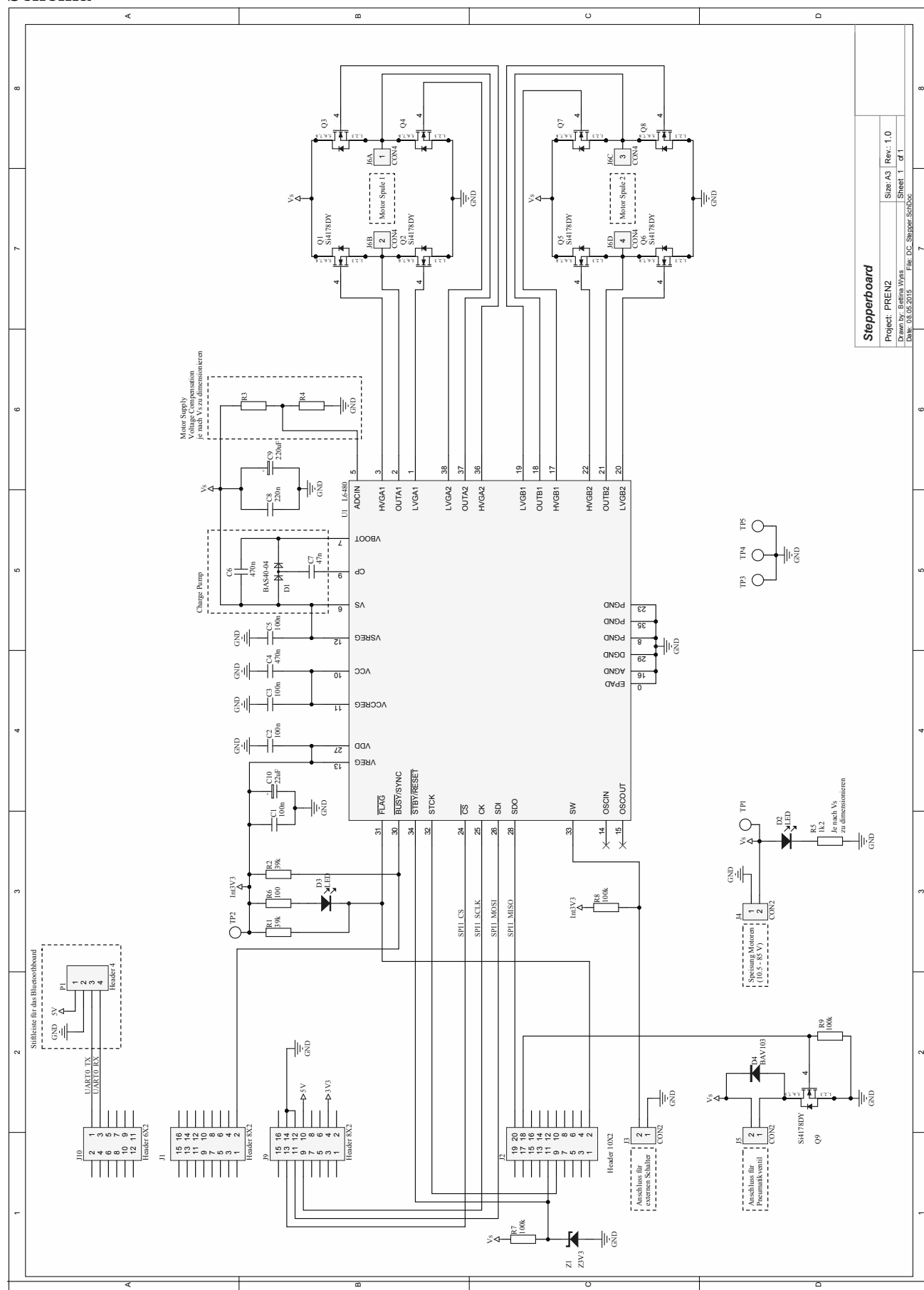
Tabelle 6: Pinbelegung

Zusätzliche Bestückung

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

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

Schema



Stepperboard

Project: PREN2

Drawn by: David Wyss

Date: 10.06.2015

Rev: 1.0

Sheet: 1 of 1

File: DC Stepper Schalts

Bestückungsdokumente

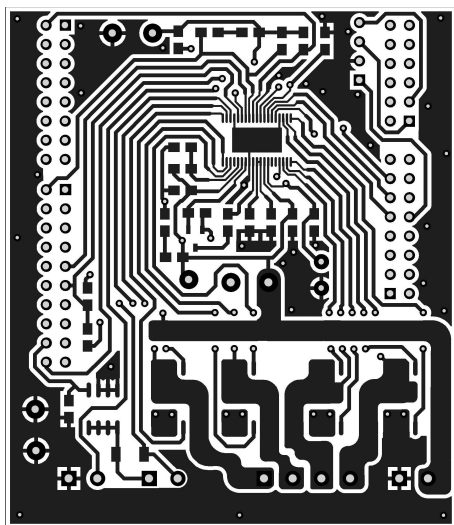


Abbildung 9: Top Layer

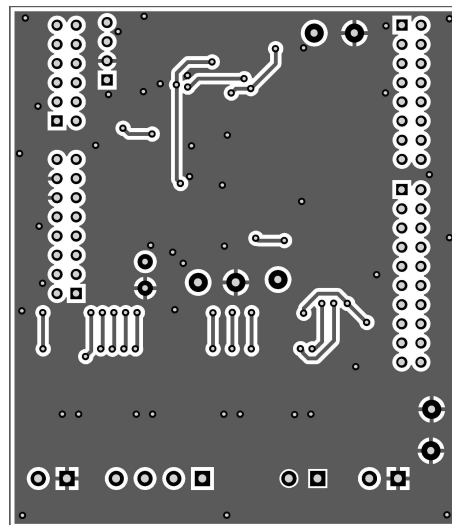


Abbildung 10: Bottom Layer

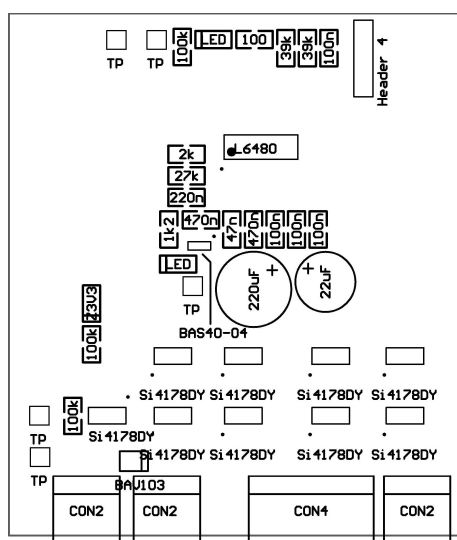


Abbildung 11: Werte

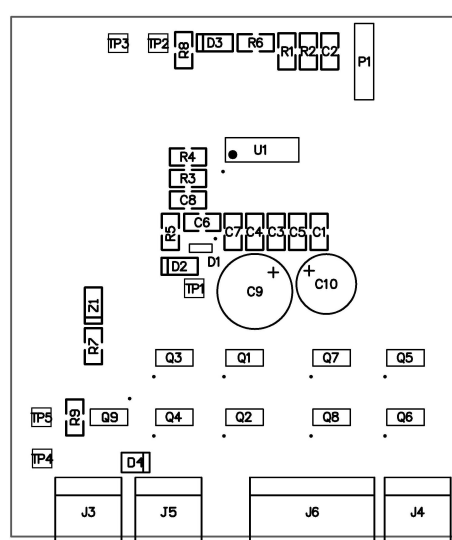


Abbildung 12: Bezeichnungen

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

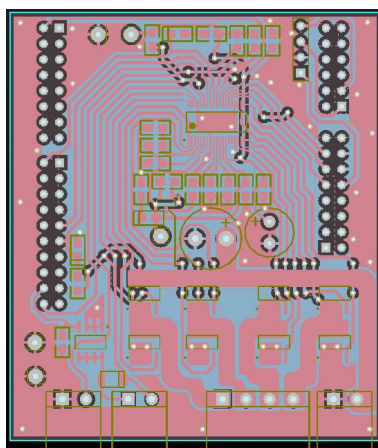


Abbildung 13: Printdesign mit Altium Designer

5.4 Software

Diese Software auf dem Freedom-Board dient als Schnittstellensoftware. Sie nimmt die Befehle der zentralen Recheneinheit über USB/UART an und sendet diese an den Motor Controller l6480 sowie das Pneumatikventil weiter. Das Software wurde mit der Entwicklungsumgebung Kinetis Design Studio von Freescale entwickelt. Diese Umgebung bietet ein Tool, welches Processor Expert heisst. Dieses erlaubt es, Komponenten wie z.B. einen Analog-Digital-Wandler in das Projekt zu integrieren. Die Konfiguration solcher Komponenten kann mit dem Component Inspector vorgenommen werden, ohne dass einzelne Register des Mikrokontrollers beschrieben werden müssen. Die Komponenten stammen von der Internetseite <http://steinerberg.com/EmbeddedComponents/> und können dort gratis gedownloadet werden.

Die Software auf dem Freedom-Board beinhaltet folgende fünf Komponenten mit ihren Aufgaben:

- FreeRTOS: Ein Open-Source-Echtzeitsystem für Mikroprozessoren und Mikrokontroller.
- Shell: Die Schnittstelle zur zentralen Recheneinheit. Sie vergleicht die von der zentralen Recheneinheit ankommenden Befehle und ruft die entsprechenden Funktionen auf.
- SynchroMaster: SPI- Komponente für die Kommunikation mit dem Schrittmotoren-IC L6480.
- LED: Anzeige auf dem Freedom-Board.
- Bit: Ansteuerung des Pneumatikventils.

Schrittmotorentreiber

Im Datenblatt des L6480 sind die verschiedenen Bitfolgen, welche einzelnen Kommandos entsprechen, dokumentiert. In der Abbildung 14 ist ein Beispiels eines solchen Kommandos aus dem Datenblatt des l6480 zu sehen, welches den Befehl RUN beschreibt. In diesem Beispiel werden 4 Bytes einzeln über die SPI- Schnittstelle gesendet:

1. Bitfolge für das Kommando RUN, sowie die Richtung (DIR) der Bewegung
2. Bit 16 bis 19 der Geschwindigkeit, mit welcher sich der Motor dreht
3. Bit 8 bis 15 der Geschwindigkeit
4. Bit 0 bis 7 der Geschwindigkeit

Run (DIR, SPD)

Table 52. Run command structure

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|--------------|-------|-------|-------|--------------|-------|-------|-------|-----------|
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | DIR | From host |
| X | X | X | X | SPD (Byte 2) | | | | From host |
| SPD (Byte 1) | | | | | | | | From host |
| SPD (Byte 0) | | | | | | | | From host |

Abbildung 14: Beispiel eines Kommandos aus dem Datenblatt des l6480

Der Treiber beinhaltet die Funktionen, welche die richtigen Bitfolgen über SPI an den L6480 senden. Fast alle dieser Funktionen wurden in Zusammenarbeit mit dem PREN-ET Team von Daniel Winz implementiert.

Shell

Für das Pneumatikventil und den Steppertreiber wurden Shellfunktionen implementiert. Erhält die Shell ein Kommando von der zentralen Recheneinheit, so wird die Eingabe ausgewertet und die entsprechende Shellfunktion ausgeführt. Die von der zentralen Recheneinheit gepassten Befehle sind:

- l6480 run [f/r] [speed]
- l6480 reset

- l6480 softstop
- l6480 hardstop
- l6480 soft Hiz

für die Steuerung des Motor Controllers, sowie

- vent shoot

für die Steuerung des Pneumatikventils.

Der Befehl "l6480 run" lässt den Motor in Vorwärtsrichtung [f] oder Rückwärtsrichtung [r] mit der in [speed] definierten Geschwindigkeit drehen. "l6480 reset" Konfiguriert den l6480 und bestromt den Motor, so dass dieser ein Haltemoment aufweist. Um den Motor ohne Haltemoment bewegen zu können, wird der Befehl "l6480 soft Hiz" verwendet, welcher die H-Brücke ausschaltet.

6 Ausblick

Eventuell hier noch etwas blabla.... Was hätte man besser machen können blabal...

Literatur- und Quellenverzeichnis

- Prof. Dr. Wolfgang Matthes. (2007). Schrittmotoren (Einführung) [Software-Handbuch]. Peukinger Weg 34, 59423 Unna.
- STMicroelectronics. (2012). *L6480, cSPINTM: microstepping motor controller with motion engine and SPI* (Datasheet). 39, Chemin du Champ des FillesPlan-Les-Ouates, CH1228 Geneva: Autor.
- Thomas Hopkins. (2012). *AN235 Application note, Stepper motor driving* (App. Note). 39, Chemin du Champ des FillesPlan-Les-Ouates, CH1228 Geneva: STMicroelectronics. (Doc ID 1679 Rev 2)

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 |
| 14 | Beispiel eines Kommandos aus dem Datenblatt des l6480 | 13 |

Tabellenverzeichnis

| | | |
|---|--|----|
| 1 | Übersicht der PREN-ET Projektgruppen | 2 |
| 2 | Übersicht der PREN-ET Repositories | 2 |
| 4 | Schnittstelle des Treibers L6480 | 6 |
| 6 | Pinbelegung | 9 |
| 8 | Stückliste (Bauteile nicht an Lager) | 12 |