

IO-Link Master Arduino Shield

Machbarkeitsstudie: Arduino-Shield als IO-Link Master

Dozent:

T. Mähne

Auftraggeber:

Balluff AG, P. Gaggero

Autor:

L. Kropf

Zusammenfassung

In dieser Projektarbeit geht es darum einen IO-Link Master für das Arduino zu erstellen. Da IO-Link noch nicht so bekannt ist, wie z.B. SPI, wurde Anfangs eine kleine Übersicht erstellt, was IO-Link eigentlich ist. Es wurde auch etwas auf die Geschichte der Automation eingegangen, um das Ganze besser zu sehen. In der Theorie wurden einige technische Grundlagen von IO-Link angeschaut.

In einem weiteren Schritt wurde dann mit der Planung des IO-Link Shields begonnen. Dafür wurde erstmals ein Arduino Board gewählt sowie der IO-Link Master festgelegt werden. Da IO-Link für den Betrieb eine Spannung von 24V benötigt, brauchte es eine Speisung, die eine grösstmögliche Flexibilität gewährleistet. Diese wurde berechnet und simuliert bis die Ergebnisse zufriedenstellend waren.

Das eigentliche Shield wurde nun nach den Vorgaben der BFH und der Firma gezeichnet. Nachdem das Design den Ansprüchen genügte, wurde es zur Produktion freigegeben und bestückt.

Es wurden Messungen durchgeführt, die die Funktion der belasteten Speisung prüfen.

Um das ganze Shield abzurunden, wurde damit begonnen eine Software zu schreiben, die den IO-Link Master auf dem Shield vom Arduino aus ansteuerbar macht. Einige Ergebnisse dieser Software sind hier auch noch beschrieben.

Zum Ende hin wurde noch resümiert, was gut war und was verbessert werden muss.

Eigenständigkeitserklärung

Hiermit versichern wir, dass wir den vorliegenden Bericht aufgrund der Besprechungen mit dem betreuenden Dozenten und der in diesem Bericht vollständig angegebenen Informationsquellen selbständig angefertigt haben. Zitate sind als solche gekennzeichnet.

Burgdorf, 19.01.2018

Kropf Lukas

Inhalt

	1
1 IO-Link, was ist das?	5
1.1 Geschichte	5
1.2 Motivation zum Erstellen des IO-Link Master Shields	8
2 Theorie	9
2.1 Kommunikation	9
2.1.1 Grundlagen	9
2.1.2 Codierung	9
2.1.3 Kommunikationsaufbau	9
2.1.4 Anschluss	10
3 Planung	12
3.1 Zeitplan	12
3.2 Vorgaben	12
3.3 Auswahl des Arduinos	12
3.4 Auswahl des IO-Link Master	13
3.4.1 Erneue Auswahl des IO-Link Master	13
4 Speisung	14
4.1 Konzept	14
4.2 Simulationen	14
5 PCB	17
6 Umsetzung	20
6.1 Hardware	20
6.1.1 Löten und Inbetriebnahme IO-Link Master Shield	20
6.1.2 Messungen Buck-Converter	20
6.1.3 Messungen Boost Converter	22
6.2 Software	23
6.3 Verwendete Software	26
6.1 Verwendete Messmittel	26
7 Erfolgskontrolle	27
7.1 Zeitmanagement	27
7.2 Anforderungen und Vorgaben	27
8 Offene Punkte und weiterführende Arbeiten	28
9 Schluss	29
10 Abbildungsverzeichnis	30
11 Tabellenverzeichnis	30
12 Glossar	31
13 Quellen	31
Anhang I Zeitplan	I

1 IO-Link, was ist das?

1.1 Geschichte

Wir beginnen von vorne: Anfangs der industriellen Revolution wurde alles noch mit Relais-schaltungen realisiert. Die Realisierung solcher Steuerungen war recht aufwändig, da jedes Signal, jede Leitung auf das gewünschte Relais geführt werden musste. Dies führte zu grossen Schaltschränken mit vielen Metern an Kabel und dies war teuer. Die Funktion einer solchen Steuerung war auf die verdrahtete Programmierung beschränkt und Änderungen waren recht teuer.

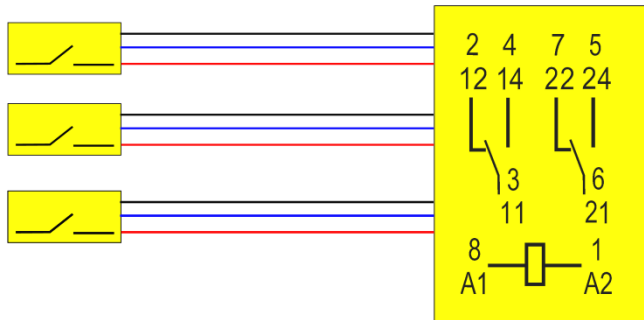


Abbildung 1 Relaissteuerung

Abgelöst wurden die Relaissteuerungen von SPS (Speicher-Programmierbare Steuerung). Sie bieten eine sehr viel kompaktere Verdrahtung der einzelnen Signale hin zur Steuerung. So konnte die gleiche Anzahl an Sensoren und Aktoren mit viel weniger Platzverbrauch angeschlossen werden. Auch waren die SPS's, wie es ihr Name schon sagt, programmierbar. So konnten auch die Kosten für die Entwicklung deutlich reduziert werden. Nachträgliche Änderungen am Programm waren ohne grosse Probleme möglich.



Abbildung 2 SPS Speuerung

Mit der Zeit wurden die SPS immer leistungsfähiger und ihr Funktionsumfang wuchs ständig. Jedoch hatten die SPS dasselbe Problem wie bis anhin. Die Verkabelung musste vom Sensor / Aktor teils über längere Strecken bis zum Anschluss der SPS verlegt werden. Diesem Problem wurde unter anderem mit zwei Verdrahtungsebenen begegnet. In der Nähe der Sensoren wurde ein dezentraler Klemmenkasten montiert, wo die Speisung der einzelnen Sensoren aufgeteilt wurde. Die Signale wurden noch immer einzeln bis zur SPS gezogen. Diese Verdrahtung ist relativ fehleranfällig, da es viele Klemmstellen geben kann.

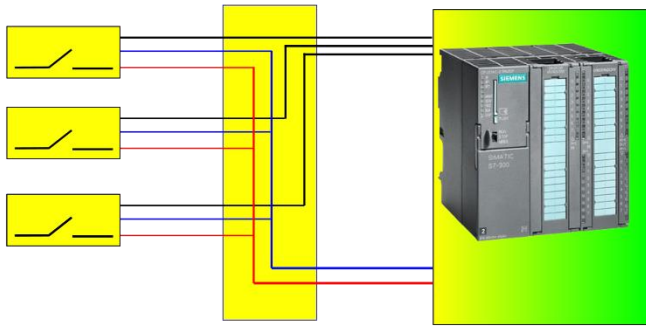


Abbildung 3 SPS Steuerung mit Verteilerbox

Diesem Problem wurde mit normierten Steckverbindern begegnet, oftmals M5, M8 oder M12. Je nach Distanz wurden die Stecker direkt bei der Steuerung angeschlossen, oder an einen dezentralen passiven Verteiler, der die Funktion des Klemmkasten einnimmt «Abbildung 4». Durch die Steckverbinder können sich kaum noch Verdrahtungsfehler einschleichen. Problematisch sind hier vor allem die analogen Signale. Diese werden durch parallel verlegte Leitungen gestört. Somit sind hier oftmals teurere, geschirmte Kabel notwendig.

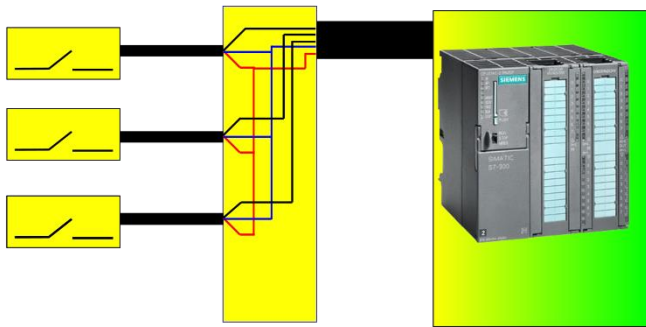


Abbildung 4 SPS mit Kabel und Verteiler

Der nächste grosse Schritt war die Einführung der Feldbusse wie Profibus, CAN, RS485 usw. Durch die Einführung der Busse konnte nun eine zentrale leistungsstarke Steuerung erstellt und die einzelnen Baugruppen über grössere Distanzen angesprochen werden. Und das alles nur über ein Kabel. Der Vorteil liegt auf der Hand. Es ist mit relativ geringem Verdrahtungsaufwand zu rechnen. Auch die Kosten halten sich im Rahmen, da deutlich weniger Zeit für die Montage gebraucht wurde. Die Feldbus-Empfänger sind nun jedoch nicht mehr passiv wie eine Verteilerbox. Sie wurden zu intelligenten Baugruppen, die vor dem Betrieb konfiguriert werden mussten.

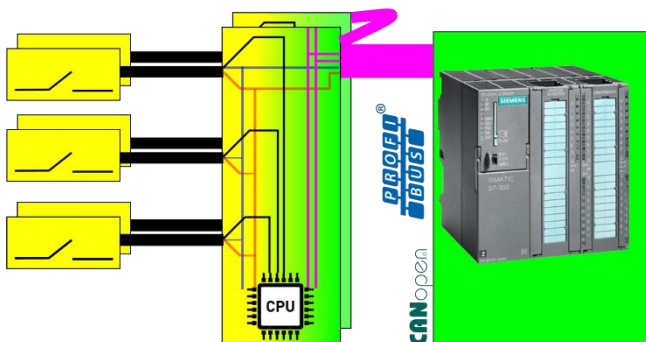


Abbildung 5 SPS mit BUS

Die Anbindung der Sensoren an die Baugruppe erfolgte jedoch immer noch auf dieselbe Art wie bis anhin. Die meisten Sensoren verfügen über genormte Ausgänge, z.B. 0-20mA, 4-20mA, 0-10V oder 0-

5V. Diese Werte müssen aber in der Baugruppe zu einem reellen Wert umgerechnet werden, damit sie weiter verrechnet werden können. Die Baugruppe kann keine Aussage über den Zustand des Sensors machen z.B. ob der Sensor zu heiss hat oder verschmutzt ist.

Eine Lösung wäre, den Sensor auch an den Feldbus anzuschliessen. Dies würde die Sensoren deutlich teurer machen. Gewisse Feldbusse sind auch auf eine Anzahl Teilnehmer begrenzt. So können je nach Anlagegrösse nicht genügend Sensoren angeschlossen werden.

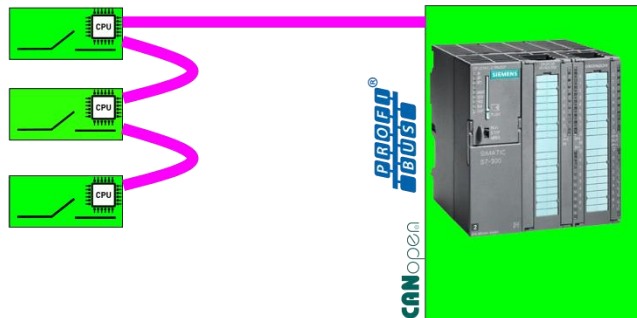


Abbildung 6 SPS mit Sensoren am Bus

Hier kommt IO-Link ins Spiel.

Es ermöglicht die Anbindung von Sensoren an eine Baugruppe. Dies mit günstigen ungeschirmten 3-Leiter Kabel, die bis anhin auch schon verwendet wurden. Es ermöglicht nun Sensoren über ein digitales Interface anzusprechen. Dabei ist die Kommunikation nicht nur auf den eigentlichen Messwert z.B. Distanz beschränkt. Es können weitere Informationen wie Temperatur oder Verschmutzungsgrad der Linse übertragen werden. Da der Datenaustausch bidirektional erfolgt, kann der Sensor während dem Betrieb neu eingestellt werden, wenn z.B. eine Schaltschwelle von 2m auf 1m geändert werden soll. Oder man spricht keinen Sensor an, sondern einen Aktor, z.B. eine Leuchte. Dank der bidirektionalen Kommunikation ist dies nun möglich.

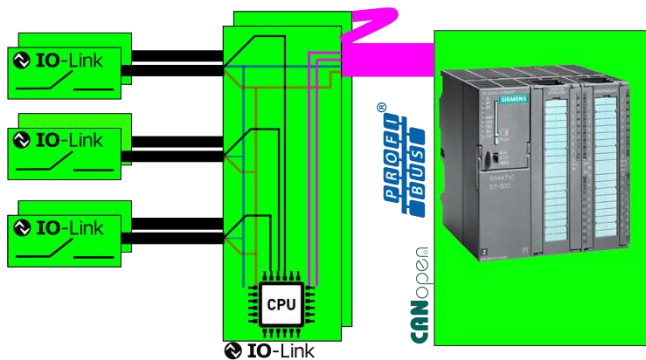


Abbildung 7 SPS mit IO-Link Master am Bus

Ein weiterer Vorteil der digitalen Kommunikation über IO-Link ist, dass der gemessene analoge Wert, z.B. Distanz, nicht mehr in ein normiertes analoges Signal umgewandelt werden muss. Die Distanz wird erfasst und sogleich Digital weitergeschickt, ohne weitere D/A Wandlung. Bei der Baugruppe entfällt die Analog-Digital Wandlung. Somit ist für die Genauigkeit der Messung nur noch der Sensor zuständig, der Rest kann vernachlässigt werden. [1]

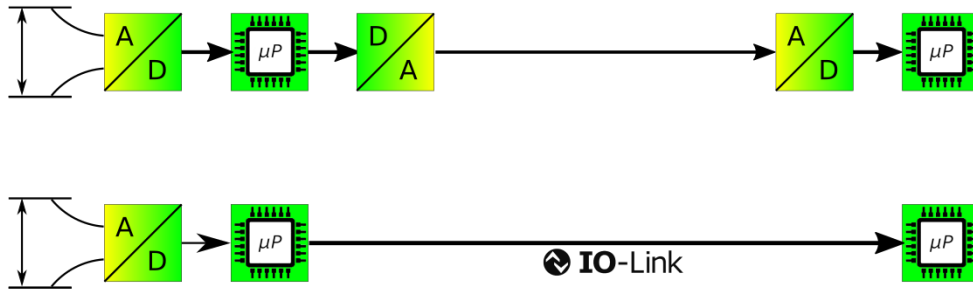


Abbildung 8 Distanzmessung konventionell vs. IO-Link

1.2 Motivation zum Erstellen des IO-Link Master Shields

Für eine Präsentation oder eine kurze Vorführung wäre es sehr praktisch, wenn man mit einem einfachen Aufbau die Möglichkeiten von IO-Link aufzeigen könnte.

Auch im Bereich des Studiums oder kleineren Forschungsaufgaben wäre es sehr angenehm, wenn die IO-Link Sensoren auf eine einfache Art und Weise angesprochen werden könnten. Heute werden aus Gründen der Einfachheit Sensoren mit analogen Ausgängen verwendet. [5]

Auch für die Ausbildung gäbe es Vorteile. Anstatt gleich eine ganze SPS mit Speisung und IO-Link Interface zu benutzen, kann hier mit relativ kleiner Hardware gearbeitet werden, die preislich auch interessanter ist.

Da sich in der letzten Zeit Arduino zu einer recht verbreiteten Plattform für Rapid Prototyping und Hobbyisten gemausert hat, lag der Wunsch nahe, ein IO-Link Master Shield für Arduino zu erstellen.

Dieses Shield würde allerdings nicht in die Produktion der Firma eingehen und dient nur zu Demonstrations- und Schulungszwecken.

2 Theorie

2.1 Kommunikation

2.1.1 Grundlagen

IO-Link ist kein BUS!

IO-Link ist eine «Punkt zu Punkt Verbindung», die zwischen einem Device und einem Master besteht. Der Master an sich hat dann in der Regel einen Feldbusanschluss, siehe «Abbildung 7».

Die Kommunikation vom IO-Link Master zum Device wird über ein 3, 4 oder 5-adriges Kabel sichergestellt. Für Port Class B muss ein 5-adriges Kabel verwendet werden, für Port Class A mindestens ein 3-adriges Kabel. Das Kabel darf eine maximale Länge von 20m nicht überschreiten.

Da IO-Link mit einem Spannungspegel von 24V arbeitet, ist es ein sehr robustes Kommunikationssystem. Kurz nachgerechnet: Unsere Zufallsbitsequenz von 24V wird durch ein Sinussignal von 5V gestört.

$$SNR = 20 * \log \left(\frac{\frac{24V}{\sqrt{2}}}{\frac{5V}{\sqrt{2}}} \right) \approx 13,6 \text{ dB}$$

Bei einer solchen SNR sollte eine störungsfreie Kommunikation problemlos möglich sein.

2.1.2 Codierung

Die IO-Link Kommunikation zwischen Master und Device basiert auf dem Übertragungsformat von UART. Eine Sequenz besteht aus einem Start Bit, 8 Datenbits gefolgt von einer geraden Parität und einem Stoppbit, auch bekannt als 8-E-1.

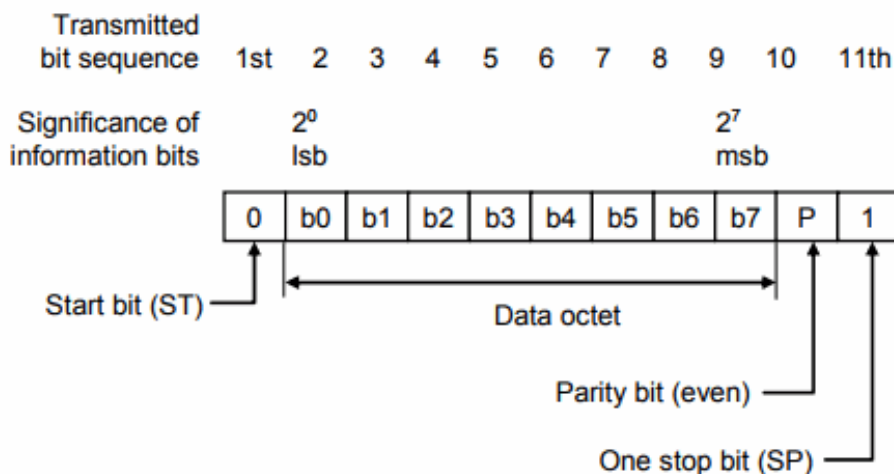


Abbildung 9 Format eines SDCl UART Frames [3]

Die Geschwindigkeit ist auf 3 Modi festgelegt worden. [4]

COM1: 4'800 Baud
COM2: 38'400 Baud
COM3: 230'400 Baud

Dem einen oder anderen kommen diese Geschwindigkeiten sicher bekannt vor, es handelt sich hierbei um Standard Serial Baudraten.

2.1.3 Kommunikationsaufbau

Ein Sensor befindet sich beim Einschalten immer im SIO Modus. In diesem Modus wird nur eine Schaltschwelle 0 oder 1 übertragen und es besteht keine IO-Link Kommunikation.

Um mit dem Sensor zu kommunizieren, muss erst der Master ein Wake-Up Impuls WURQ erzeugen. Dieser kann als einen kurz andauernden Kurzschluss angesehen werden. Wenn Q/C low ist, wird die Leitung auf Hight getrieben und umgekehrt. Das Device erkennt dies und wechselt sofort in den hochohmigen Zustand.

Er wartet T_{DMT} lang. Anschliessend versucht der Master erst mit der Baudrate von COM3 eine Verbindung aufzubauen indem er eine «Master message» sendet. Ohne Antwort versucht er das mit der Baudrate von COM2 und anschliessend COM1. Ein Device unterstützt jeweils nur eine Übertragungsgeschwindigkeit.

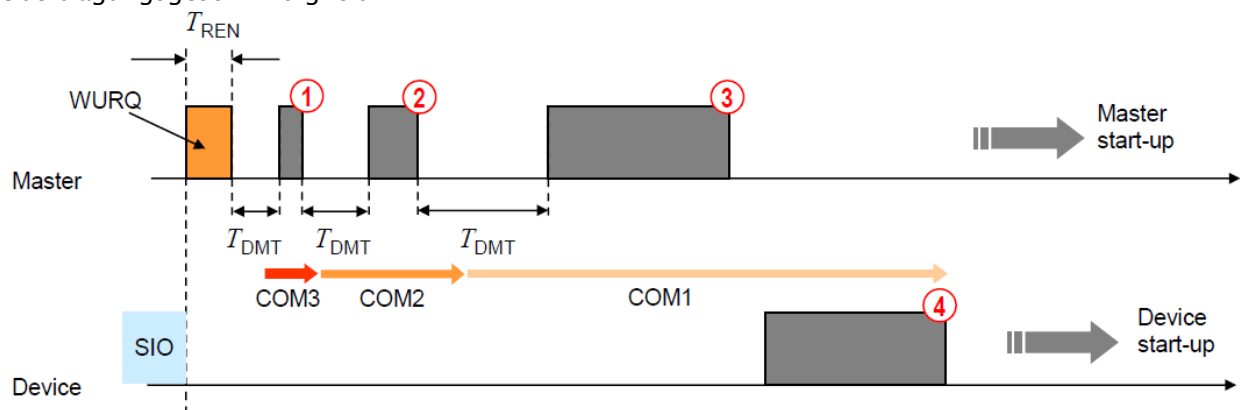


Abbildung 10 Verbindungsaufbau IO-Link [3]

2.1.4 Anschluss

Für die Anschlussstechnik in IP65/67 wurde unter Anderem der M12 Steckverbinder definiert. In der Regel haben Sensoren einen 4-Poligen Stecker. Master müssen über einen 5-poligen M12 Stecker verfügen.

Beispiel hierfür ist «Balluff BNI EIP-502-105-Z015». Dieser verfügt über 16 digitale Ein-/Ausgänge, wovon vier als IO-Link Master Schnittstellen gebraucht werden können.

Der M12 als Steckverbinder ist eine schlaue Wahl, da dieser bei den Sensoren recht verbreitet ist. So sind viele Sensoren, die IO-Link nicht unterstützen, an den Master anschliessbar.

Wird die Anschlussstechnik in IP20 gewählt, so können Schraub- oder Steckverbinder verwendet werden. Ein Beispiel hierfür ist «WAGO I/O-System – Serie 750/753», welcher auch 4 IO-Link Master Schnittstellen hat.



Abbildung 11 WAGO 750/753



Abbildung 12 Balluff BNI EIP-502-105-Z015

Es existieren zwei Portklassen, Port Class A (Typ A) und Port Class B (Typ B).
 Typ B kann über das selbe Kabel auch noch eine weitere Speisung mitführen. So kann das Stromlimit von 200mA [2]S8 vom Typ A überwunden werden.
 Typ A kann dafür den Pin 2 als digitales Signal gebrauchen, entweder als Eingang oder Ausgang.

In der folgenden Tabelle wird die Pinbelegung der Ports dargestellt.

Pin	Port Class A	Port Class B
1	L+, Speisung	L+, Speisung
2	Nicht vorgegeben, meist ein weiteres digitales Signal	2L+, zweite (galvanisch getrennte) Speisung
3	L-, Speisung	L-, Speisung
4	C/Q, Kommunikation	C/Q, Kommunikation
5	Nicht vorgegeben	2M, Masse zweite Speisung

Tabelle 1 Vergleich Port Class A & B

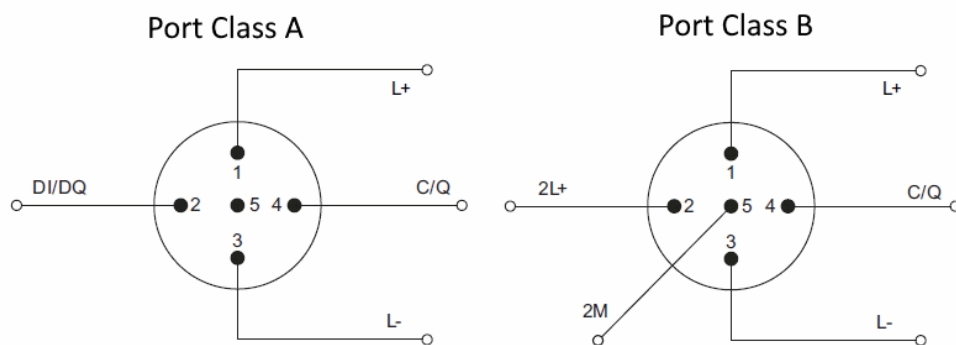


Abbildung 13 Vergleich Port Class A & B

IO-Link ist auch für die Steckverbinder M5 und M8 definiert. Diese werden jedoch selten gebraucht.

3 Planung

3.1 Zeitplan

Der Ursprüngliche Zeitplan wurde erstellt und es wurden folgende Meilensteine gesetzt:

- 10.11.2017 PCB bestellen
- 17.12.2017 Library geschrieben und getestet
- 19.01.2018 Projektabgabe

Der komplette Zeitplan ist im Anhang zu finden.

3.2 Vorgaben

Die Vorgaben wurden im Laufe der Arbeit festgelegt. In einem ersten Schritt wurde ein Pflichtenheft erstellt, welches den groben Umfang und Standpunkte festgelegt hatte. Weitere Vorgaben und Details zu den Anforderungen des Pflichtenhefts wurden im Gitlab der BFH¹ festgehalten. Eine Auflistung der Anforderungen ist in der «Tabelle 10 Anforderungen» zu finden.

3.3 Auswahl des Arduinos

Als einer der ersten offenen Punkte aus dem Pflichtenheft wurde eine Arduino Plattform festgelegt. Kriterien waren neben Preis auch die Konnektivität und die Zukunftssicherheit. Weiter wurde der Wunsch durch die BFH geäußert, dass das Board mit der GECKO4-Education Plattform kompatibel ist.

Board	Footprint	Spannung	SPI	UART	I2C	Kosten ²	Notiz
Arduino UNO	UNO	5V	1	1	1	23Fr.	
Arduino Yun	UNO	5V	1	(1) ³	1	102Fr.	Wegen der geteilten Kommunikation ist dieses Board eher nicht geeignet. 3 Devices auf einem UART kommt nicht gut.
Arduino Leonardo	UNO	5V	1	1	1	21Fr.	
Genuino Zero ***	UNO	3.3V	1	2 ⁴	1	36Fr.	Hat eine 12CH-DMA
Arduino Due	MEGA	3.3V	1	4	2	42Fr.	Hat eine 17/6CH DMA [6]
Arduino Mega 2560	MEGA	5V	1	4	1	43Fr.	

Tabelle 2 Vergleich Arduino Plattformen

Die Wahl fiel auf den Arduino Due. Ausschlaggebend dafür war, dass er den selben Footprint wie der Arduino MEGA 2560 und das GECKO4-Education hat (mit Ausnahme des SPI-Headers bei der GECKO4 Plattform) sowie das Vorhandensein von 4 Hardware Serial-Ports.

Weiter verfügt der DUE mit einem Cortex M3 Microprozessor über einen grösseren Funktionsumfang als der MEGA. Hierzu gehören:

- eine leistungsstarke DMA,
- Interruptfähigkeit auf allen Eingängen

¹ https://gitlab.ti.bfh.ch/balluff/bte5511_kropl2/issues

² Alle Preise stammen von <https://www.distrelec.ch>, am 16.10.2017

³ Wird als Kommunikation zwischen dem uController und dem uProzessor gebraucht.

⁴ Nur einer ist auf den Pins verfügbar.

- 32-Bit Prozessor Architektur
- Mehr Flash und RAM,
- Höherer Clock von 84MHz anstatt 16MHz.

Der Preis des DUEs ist sogar tiefer als der des MEGAs, wenn auch nicht viel.

3.4 Auswahl des IO-Link Master

Als nächster Schritt wurde ein IO-Link Master IC ausgewählt. Hier gab es vorgängig keine Vorgaben.

IO-Link Master	Kommunikation	Spannung Logik	Spannung Versorgung	Anz. IO-Link Ports	Preis	Notiz
LTC2874IUHF #PBF-ND	SPI + 4x UART	2.9-5.5V	8-34VDC	4	17Fr.	4x UART auf den uC
MAX14827 AATG+-ND	SPI + UART	5V + 3.3V	9-60VDC	1	4Fr.	Möglichkeit für Multiplexing des UARTS und dem SPI
MC34CM3120 EPR2TR-ND	SPI + 2x UART	3.3V	8-32V	2	6Fr.	Mit 2x UART auf den uC

Tabelle 3 Vergleich IO-Link Master

Die Wahl fiel auf den MAX14827 [7]. Ausschlaggebend dafür war der relativ günstige Preis und die Möglichkeit den UART zu multiplexen. Weiter ist er kompatibel zu 3.3V und 5V und hat eine sehr hohe Eingangsspannung von bis zu 60V.

3.4.1 Erneue Auswahl des IO-Link Master

Bei der Prüfung des fast fertig entworfen PCB wurde festgestellt, dass es sich beim MAX14827 nicht um einen IO-Link Master handelt, sondern um einen IO-Link Node. Dies führte dazu, dass eine erneute Wahl getroffen werden musste.

Die Wahl fiel auf den MAX MAX14824 [8], da er den gleichen Footprint und sehr ähnliche elektrische Spezifikationen wie der MAX14827 aufweist. Die Pinbelegung war jedoch komplett anders, was dazu führte, dass der gesamte Teil des IO-Link nochmals gelayoutet werden musste.

4 Speisung

4.1 Konzept

Da für den Betrieb von IO-Link Sensoren 24V benötigt werden, entstand die Idee einen Boost-Converter einzubauen. So ist man in der Lage, Sensoren mit geringer Leistungsaufnahme direkt über die 5V Speisung vom USB zu betreiben.

Die IO-Link Master haben einen eingebauten Linearregler um intern 5V und 3.3V zu erstellen.

Um die IO-Link Master thermisch zu entlasten, gibt es die Möglichkeit, den 5V Regler auszuschalten und einen externen Spannungsregler zu gebrauchen. Dies wurde so umgesetzt, dass mit einem 24V Spannungseingang auch das Arduino mit 5V versorgt werden kann. Damit wird eine fast maximale Flexibilität der Spannungsversorgung erreicht.

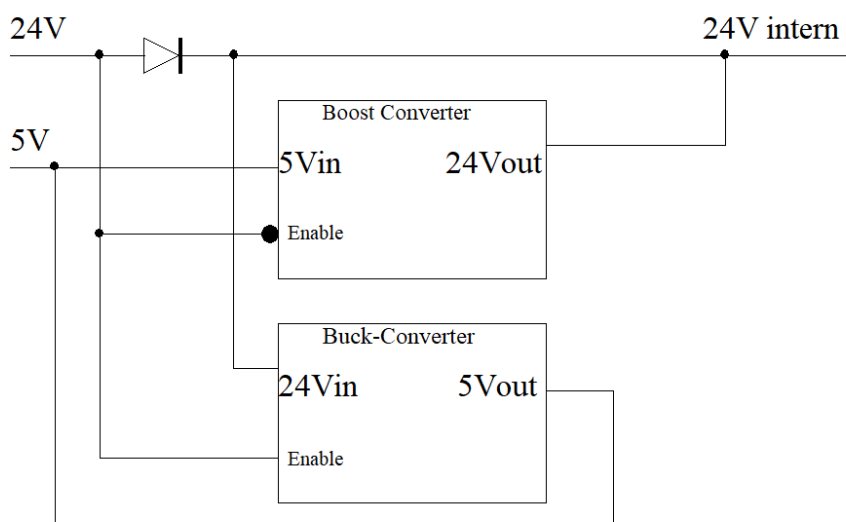


Abbildung 14 Blockdiagramm Speisung

4.2 Simulationen

Um die Möglichkeit einer Fehlfunktion des Buck und Boost-Converters auszuschliessen wurde viel Zeit in die Simulation der Speisung investiert. Auf der Internetseite von LT-Spice Wiki konnte ein Modell des verwendeten Buck-Converter MC34063 heruntergeladen werden. [9] So konnte die Möglichkeit einer Fehlfunktion des Buck und Boost-Converters minimiert werden. In den folgenden zwei Grafiken wird die Simulationsschaltung dargestellt.

Für die Simulation wurden nur «reelle» Bauteile verwendet. Einzig das Model MySW wurde als annähernd ideal angesehen. Dies wird gebauht um eine offene Verbindung zu simulieren.

Da der MC34063 von Werk aus nicht ausschaltbar ist, musste diese Funktion nachgeliefert werden. Dies wird erreicht, indem der Ausgang des Spannungsteilers (R3&R8, respektive R5&R7) mit einem FET auf ein höheres Niveau gezogen wird. Der MC34063 versucht die Spannung am Inv Pin auf 1.25V zu halten. Liegt eine höhere Spannung an, so schaltet er nicht mehr ein.

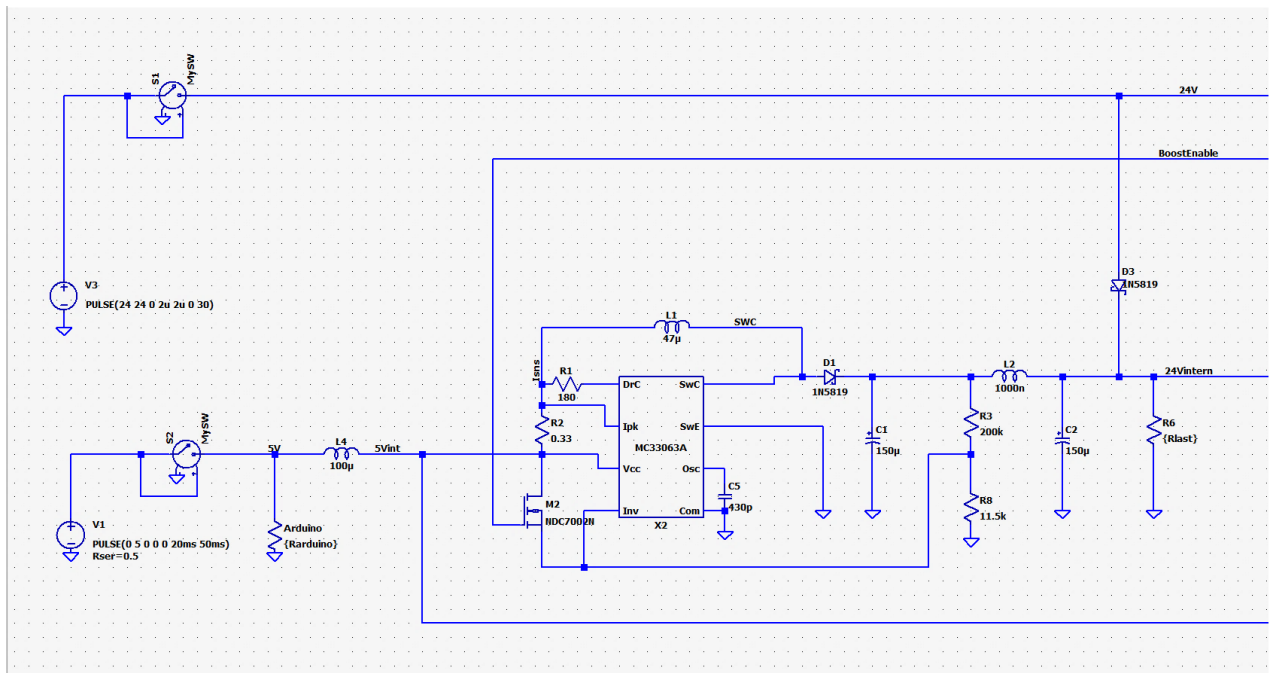


Abbildung 15 Simulationsaufbau Boost-Converter

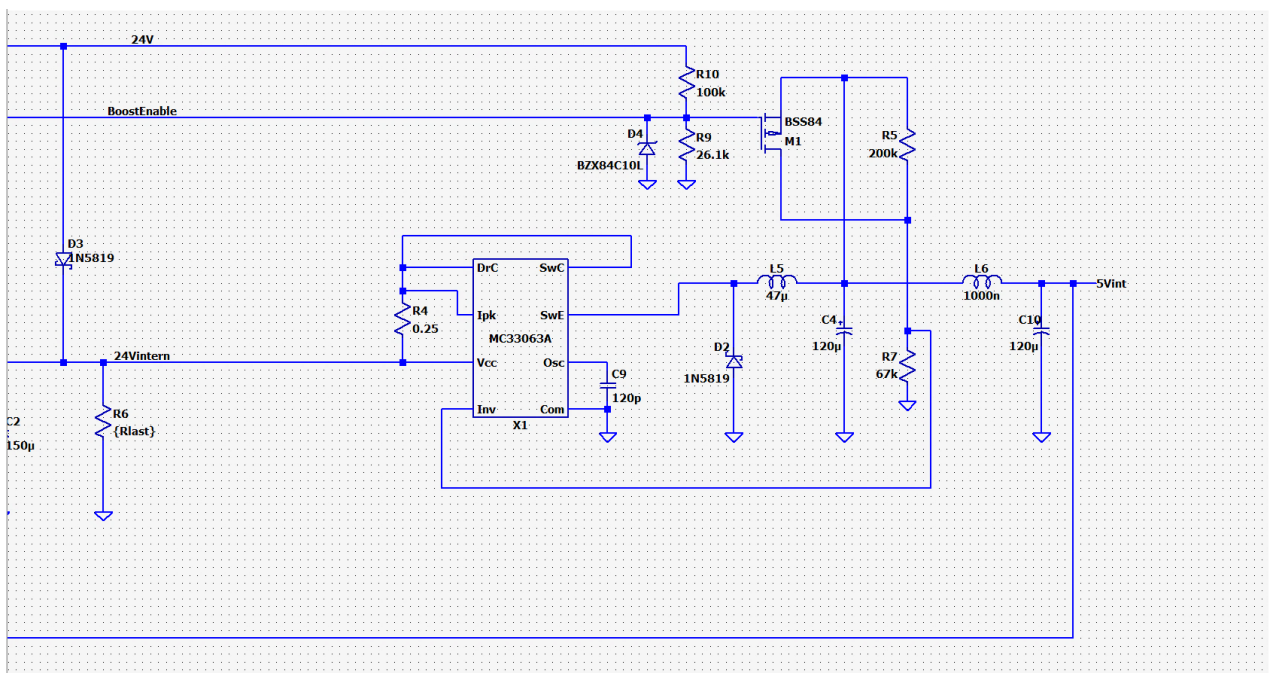


Abbildung 16 Simulationsaufbau Buck-Converter

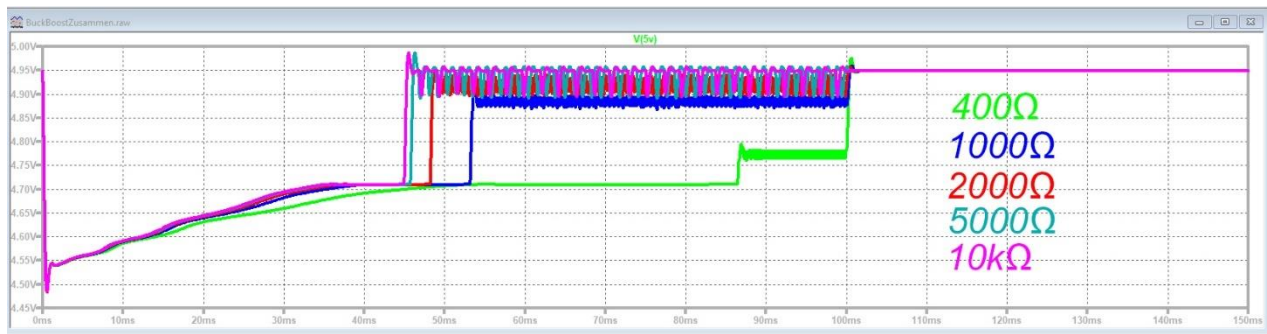


Abbildung 17 5V im Boost-Modus

In «Abbildung 17» ist zu sehen, wie der Boost-Konverter die 5 V vom Arduino belasten. Die Widerstände wurden an die 24 V des Boost-Ausgangs angeschlossen (R6)

- Als Quellenwiderstand wurde 0.5Ω angenommen.
- Ab 100ms wird die externe 24 V Speisung eingeschaltet.

5 PCB

Das PCB wurde mit dem Open-Source Layout Programm KiCAD erstellt.

Für das Grund-Design wurden drei Varianten vorgestellt.

1. Variante 1 mit vier IO-Link Master, vier M12 Steckern und vier M8 Steckern
2. Variante 2 mit vier IO-Link Master, vier M12 Steckern und zwei M8 Steckern
3. Variante 3 mit vier IO-Link Master, vier M12 Steckern

Variante 3 wurde gewählt, da der Firma keine M8 Master bekannt waren. Die Kosten für M8 Stecker waren ungefähr gleich hoch wie die der M12 Stecker, was ebenfalls gegen die anderen zwei Varianten sprach.

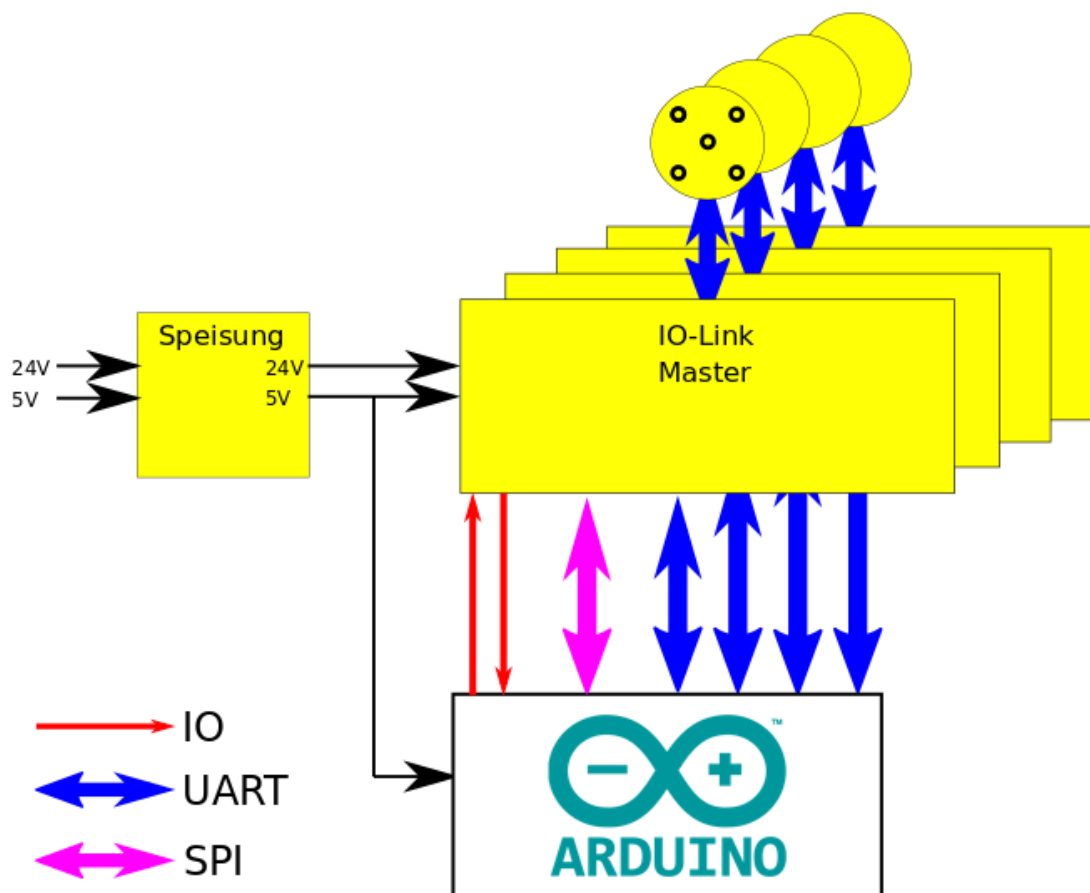
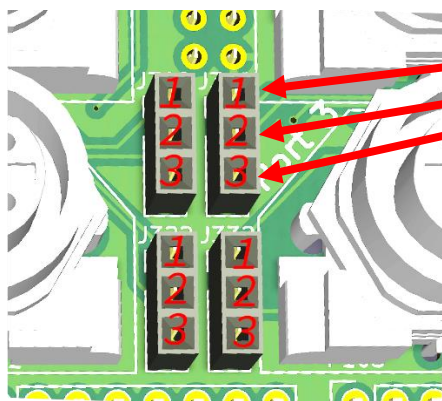


Abbildung 18 Blockschaltbild IO-Link Master Shield

In Abbildung 18 Blockschaltbild IO-Link Master Shield ist die Übersicht über das IO-Link Master Shield zu sehen. Die Speisung stellt jederzeit sicher, dass für das Arduino und die IO-Link Master 5 V und 24 V zur Verfügung stehen. Jeder IO-Link Master ist über das SPI mit dem Arduino verbunden. Jeder IO-Link Master verfügt über eine gewisse Anzahl an digitalen Ein und Ausgängen, die auf das Arduino geführt werden. Jeder Arduino verfügt über eine eigene UART Kommunikation zum Arduino. Der Anschluss an die Sensoren wird über einen M12-5 Stecker sichergestellt.

Um einen einfachen Abgriff für die Signale auf den IO-Link Leitungen zu haben, wurde entschieden, die Signale über eine Buchsenleiste und einen Spannungsteiler zugänglich zu machen. Der Spannungsteiler wurde von 24 V auf 3.3 V eingestellt.



Die Belegung wiederholt sich für jeden Port

1. DI
2. C/Q (IO-Link Kommunikation)
3. GND

Abbildung 19 Messanschluss Port 0-3

Um Messungen zu vereinfachen wurde ein zusätzlicher Massenanschluss hinzugefügt, siehe «Abbildung 20», J122.

Die 24 V Speisung wurde mit 4 mm Laborbuchsen angeschlossen.

Der Print wurde 2-Lagig geplant und gefertigt.

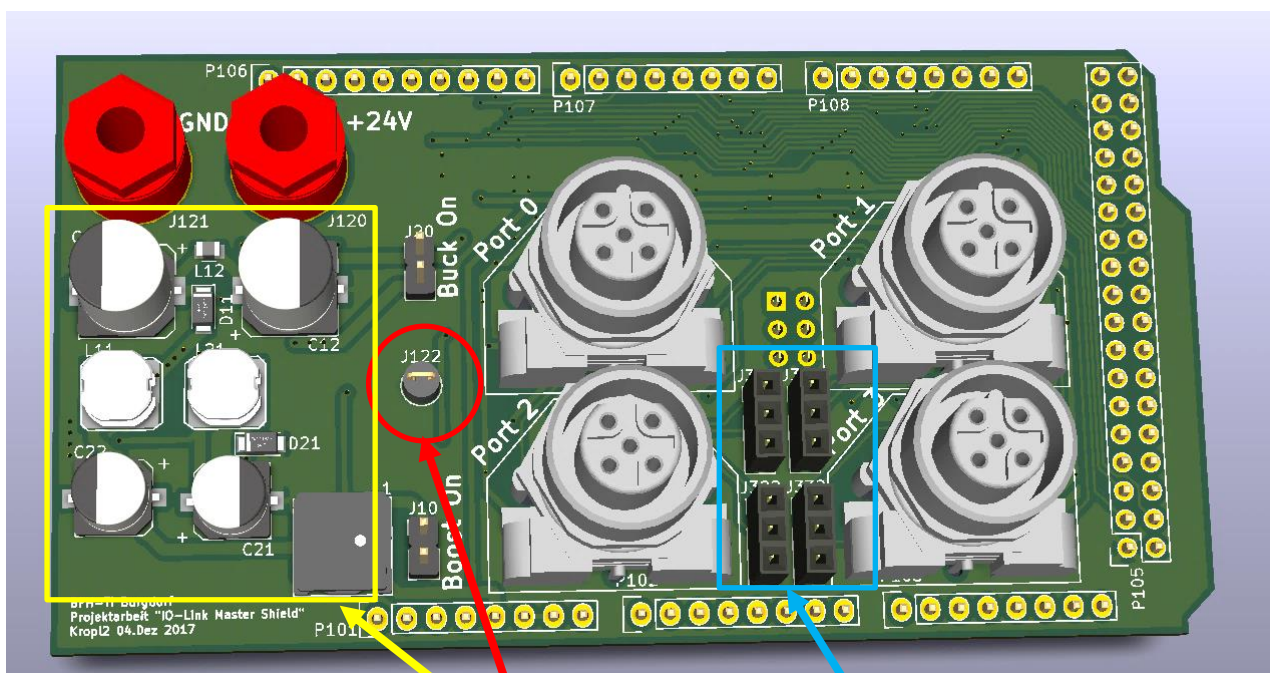


Abbildung 20 Vorschau von oben

Masseanschluss
Buck- Boost Konverter

Messanschlüsse für DI /
C/Q

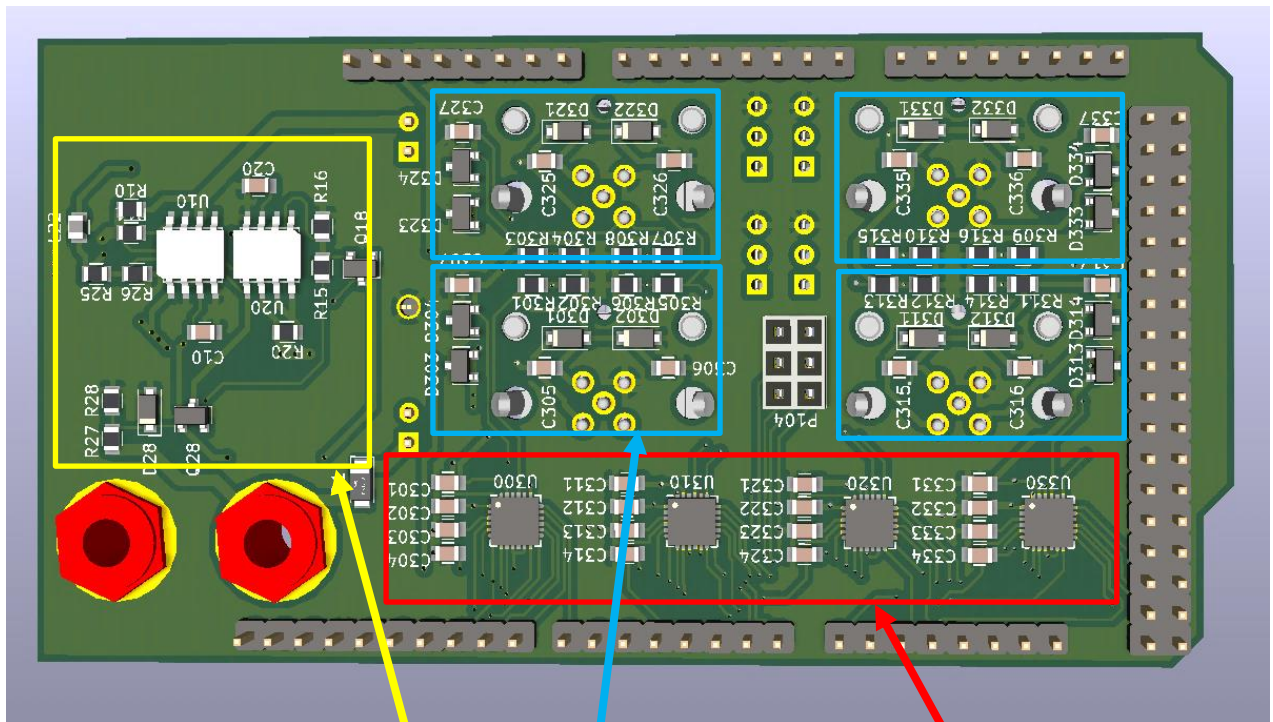


Abbildung 21 Vorschau von unten

EMV Schutzschaltungen
und Spannungsteiler

IO-Link Master

Steuerung Buck-Boost
Konverter

6 Umsetzung

6.1 Hardware

6.1.1 Löten und Inbetriebnahme IO-Link Master Shield

Die PCB's wurden an der BFH bestückt mit einem Reflow-Ofen gelötet. Bei einer ersten Inbetriebnahme wurde das IO-Link Master Shield ohne Arduino an 5V und an 24V angeschlossen. Es wurde kontrolliert, ob der Buck- und der Boost-Konverter funktionieren. Bei beiden gelöteten Boards konnten keine Defekte festgestellt werden. Der Boost-Konverter lieferte wie geplant 23V, der Buck-Konverter saubere 5V. Dies wurde auch bei den späteren Messungen bestätigt.

Bei der Inbetriebnahme der IO-Link Master wurde festgestellt, dass bei jedem Board jeweils ein anderer IO-Link Master nicht über das SPI antwortet. Dies wird auf das Löten zurückgeführt. Da kein Stencil bestellt wurde, musste alles von Hand aufgetragen werden. Hierbei kam es wohl zu fehlerhaften Lötstellen, jedoch nirgends zu einem Kurzschluss.

Es wurde festgestellt, dass der Boost Konverter nicht laufen darf, wenn das Arduino programmiert wird. Es funktioniert einfach nicht. Während dem Betrieb scheint es keine Einschränkungen zu geben.

6.1.2 Messungen Buck-Converter

Gemessen wurde an den Punkten im Schema «Energieversorgung, 2/3», auf Schema Sektor A1: 5V, A2: 5Vint und A2: 24Vintern.

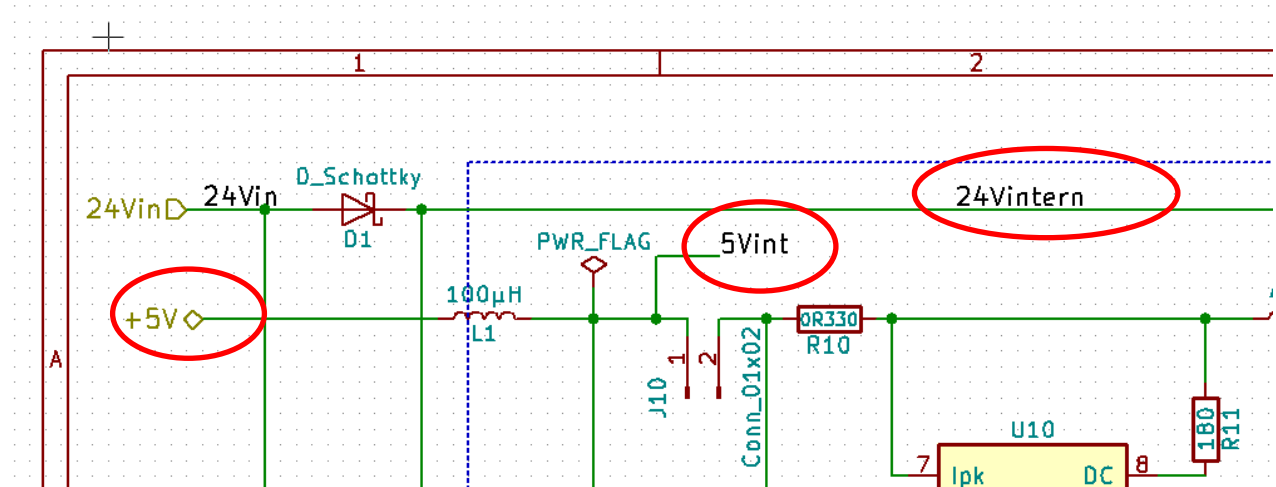


Abbildung 22 Messpunkte

Messaufbau:

Für die 5 V Speisung wie auch für die 24 V Speisung wurde ein Labornetzgerät verwendet.

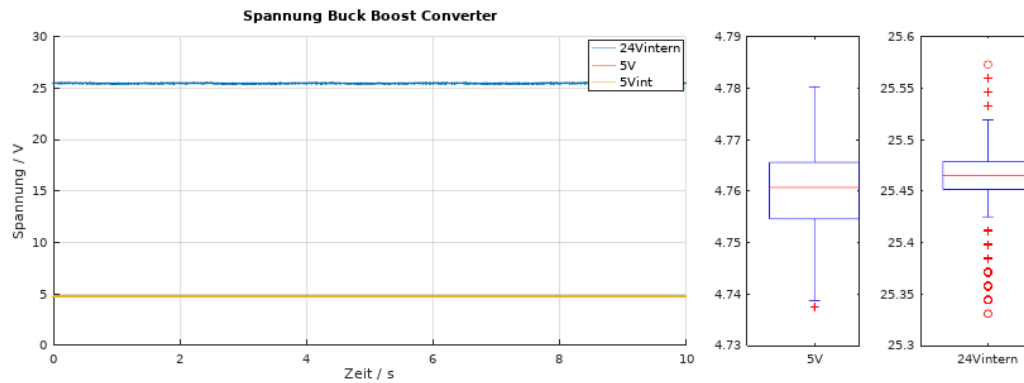


Abbildung 23 Spannung Buck Converter über 10s

In der obigen Abbildung ist gut zu sehen, dass der Buck-Converter die Spannung auch über eine Zeit von ~10s auf 4.76V im Mittel halten kann. Hier ist auch der Rippel im Boxplot gut sichtbar, er bewegt sich zwischen 4.78 und 4.74, also 40mV. Dies deckt sich auch mit der nächsten Abbildung.

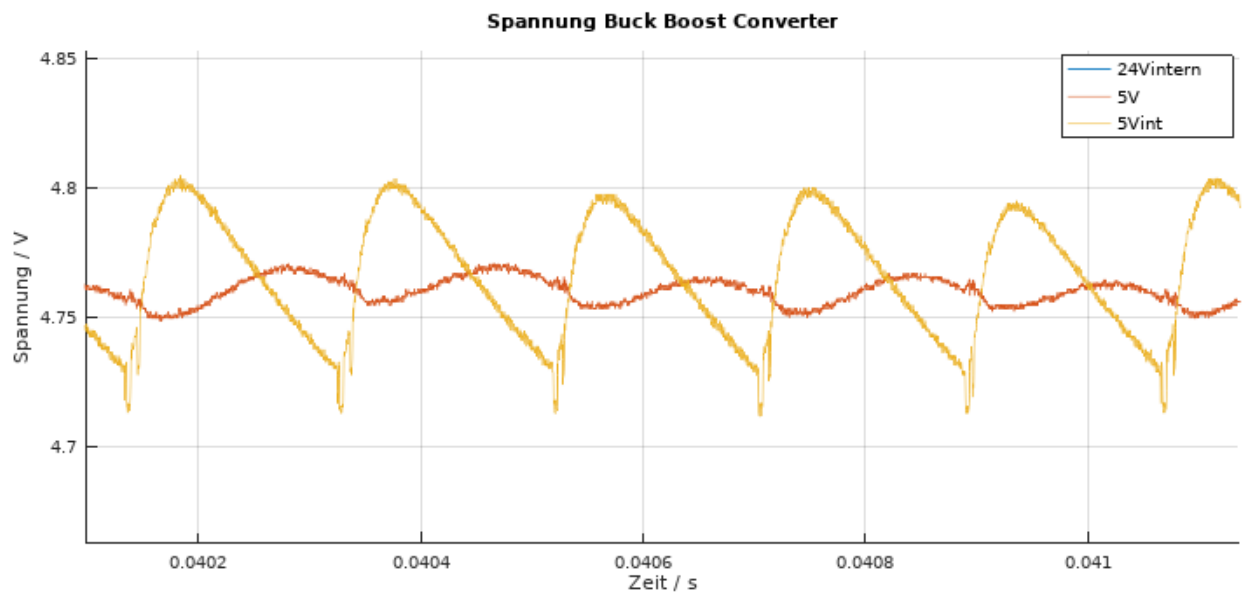


Abbildung 24 Rippel Ausgang Buck-Converter nach 10s Betrieb

Es ist ein Rippel von weniger 40mV Peak Peak zu sehen.

6.1.3 Messungen Boost Converter

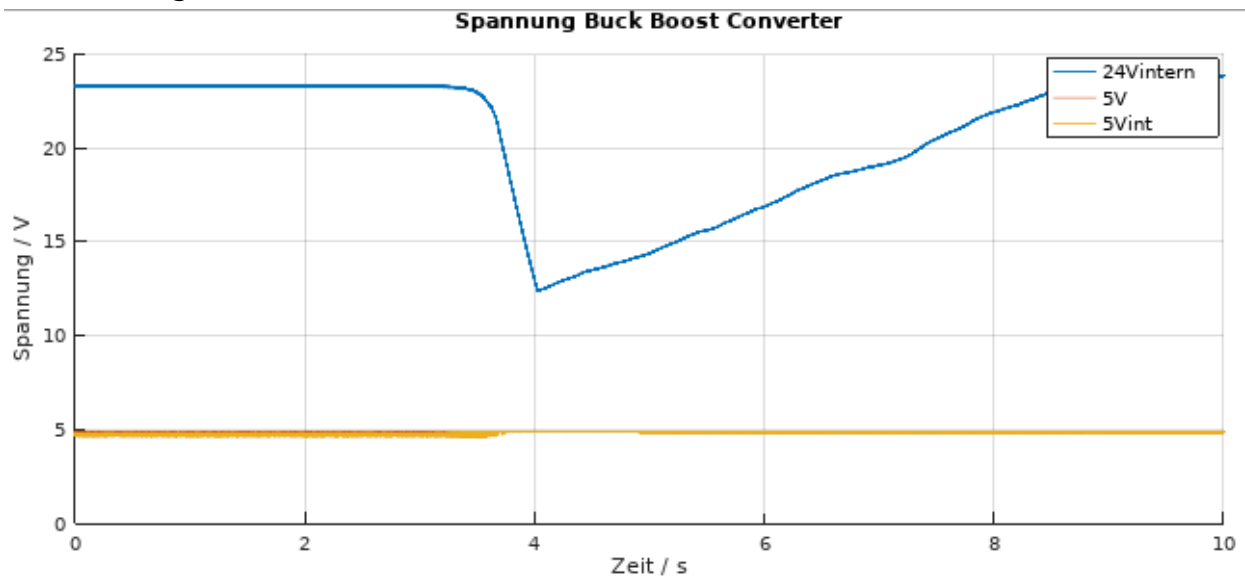


Abbildung 25 Wechsel Boost-Buck Mode

Die Spannung wird langsam von 0 V auf 24 V erhöht. Bei ca. 11 V schaltet der Boost-Konverter aus und bei ca. 12 V schaltet der Boost dann ein. Ein rascher Wechsel würde die 24Vintern nicht einbrechen lassen.

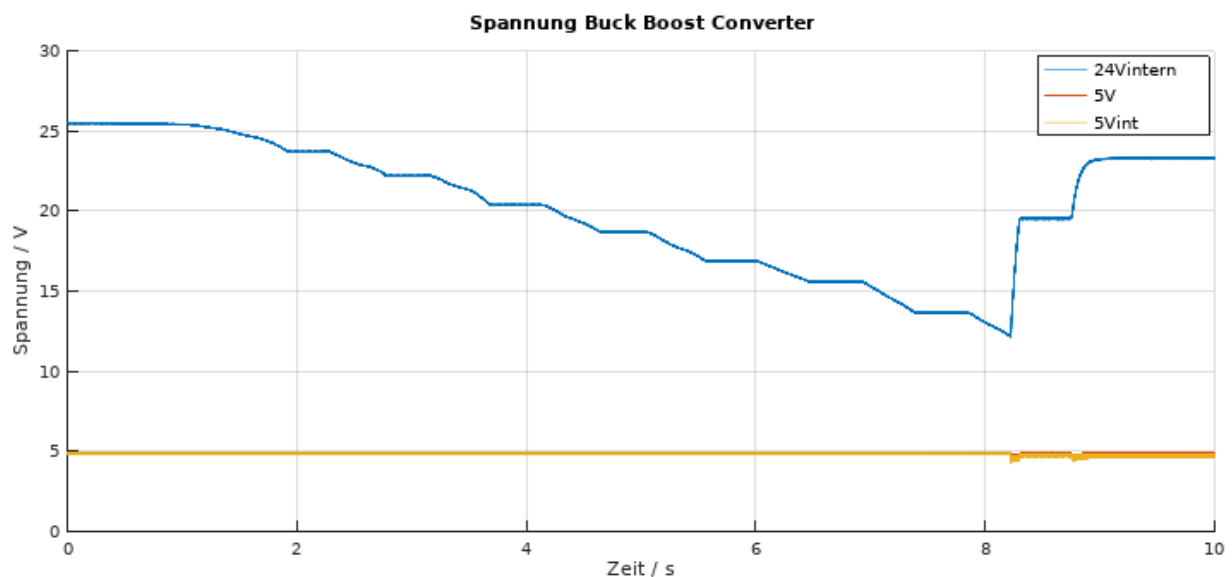


Abbildung 26 Wechsel Buck-Boost Mode

Das langsame Abklingen von 24 V lässt die Schaltschwelle vom Boost- Konverter sehen. Diese beträgt ca. 12 V. Bei einem abrupten Abbruch der 24 V Speisung würde es nicht zu einem unterschreiten der 20 V Grenze der 24Vintern Spannung führen.

Gut zu sehen ist auch, dass der Boost Konverter kein Überspringen hat.

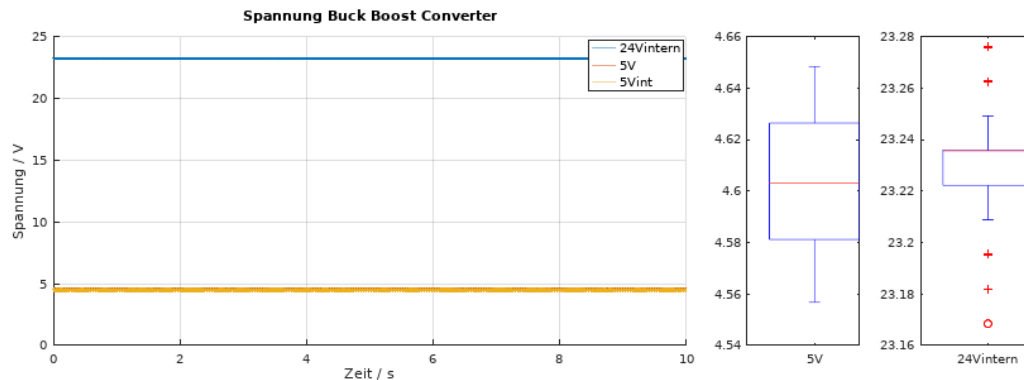


Abbildung 27 Spannungen Boost-Converter

Die 24 V wurden mit 820Ω belastet. Dies ergibt einen Strom von $\sim 30\text{mA}$ und eine Leistung von 0.7W .

Die 5V Speisung wird vom Boost-Converter recht stark beeinflusst. Der Mittelwert der Spannung liegt jetzt bei 4.6V . Der Rippel liegt bei ca. 0.1V . Diese Werte decken sich auch mit «Abbildung 28».

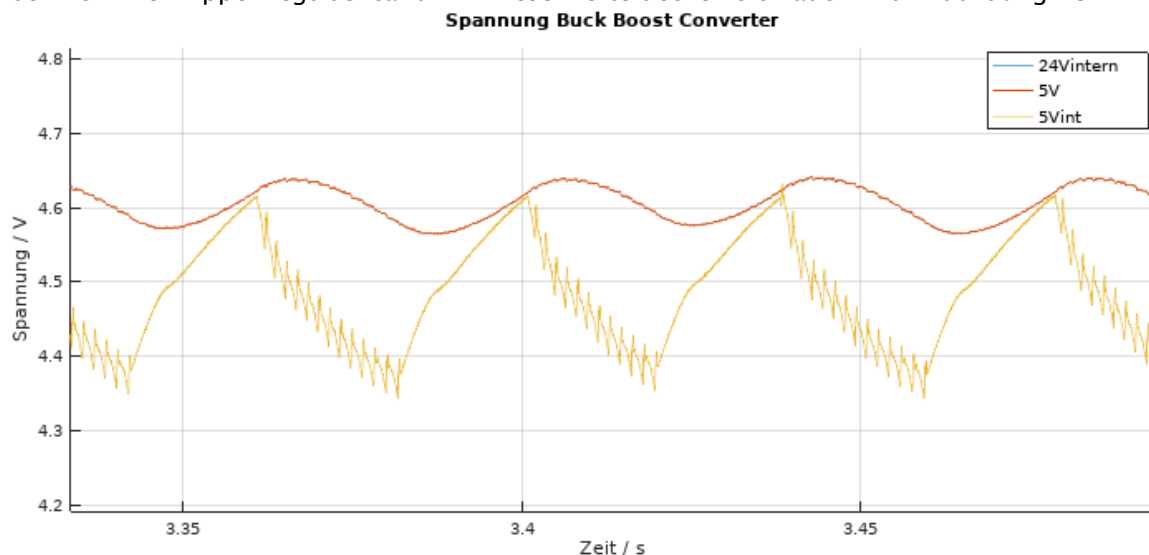


Abbildung 28 Rippel 5 V im Boostmodus

Hier ist ein Rippel von ca. 100mV zu sehen.

Erwähnt werden muss allerdings auch, dass der Rippel und auch das Spannungsniveau stark von der verwendeten Speisung und dem verwendeten MicroUSB Kabel abhängig ist.

6.2 Software

Wegen der fortgeschrittenen Zeit konnte keine umfassende Software / Library geschrieben werden. Aus diesem Grund wurde der Fokus auf die Funktionen des IO-Link Masters gelegt, ohne die Implementierung der IO-Link Kommunikation. Denn bevor die IO-Link Kommunikation aufgebaut werden kann, muss mit dem IO-Link Master kommuniziert werden.

Es wurde die Hardware der IO-Link Master Shields in einer Klasse abgebildet und die Möglichkeit geschaffen, die SPI-Register der einzelnen IO-Link Master anzusprechen. Ebenfalls können alle direkten Signale des IO-Link Masters vom Arduino DUE gelesen oder gesetzt werden.

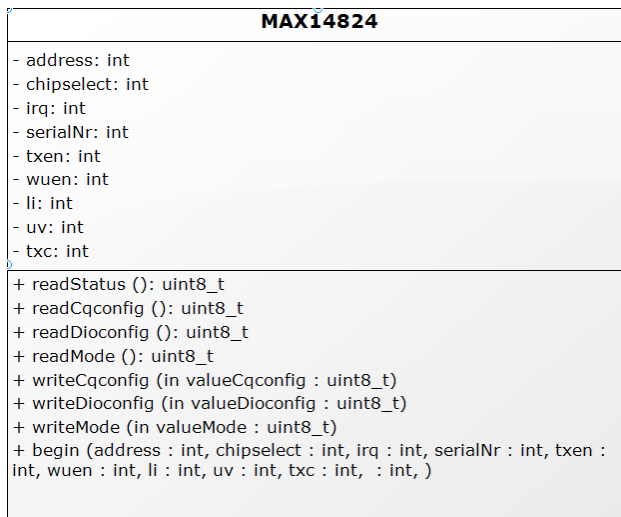


Abbildung 29 Klassendiagramm MAX14824

Zu Beginn der Softwareerstellung funktionierte nicht viel. Die Pegel der SPI-Signale, (Chip Select (CS) und MOSI) lagen oft im Bereich von $V_{cc}/2$ und V_{cc} , siehe «Abbildung 30» Channel 2. Es wurde festgestellt, dass Pin 10 sich aus irgendeinem Grund nicht als Slave-Select verwenden lässt, obwohl es erfolgreiche Projekte auch mit CS=10 gibt. Die Lösung war dann, dass Pin 11 als CS genommen wurde. Die Messungen bestätigten dann, dass diese Lösung funktioniert, siehe «Abbildung 31».

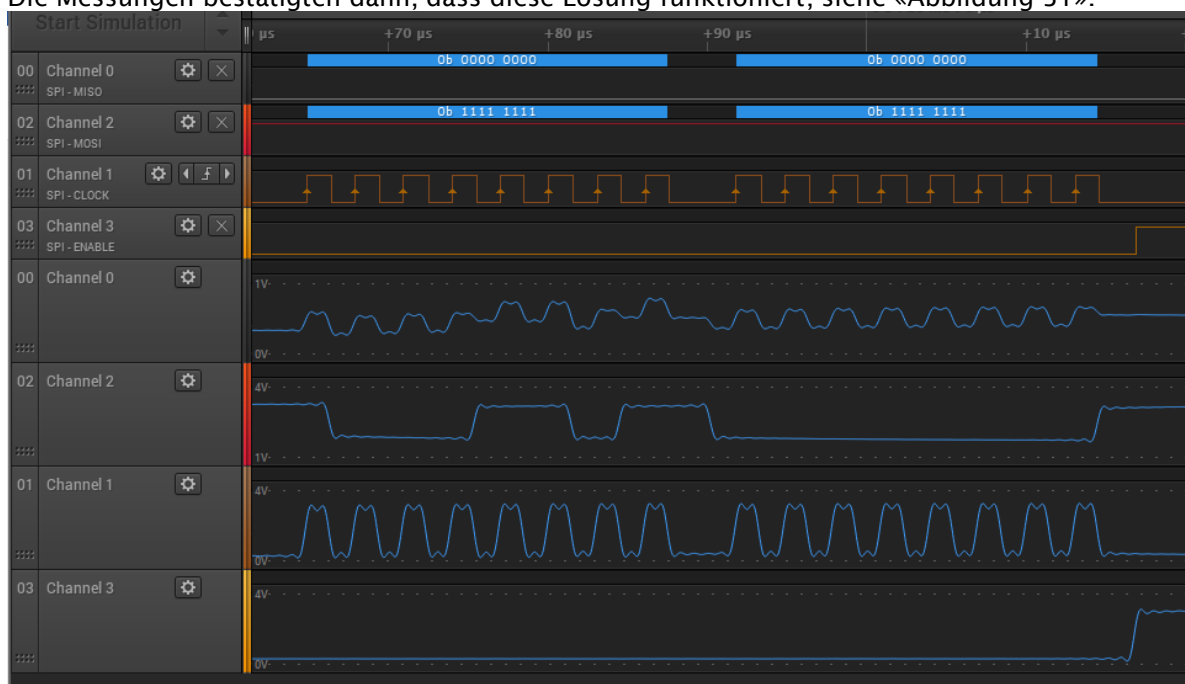


Abbildung 30 Fehlerhafte SPI Kommunikation

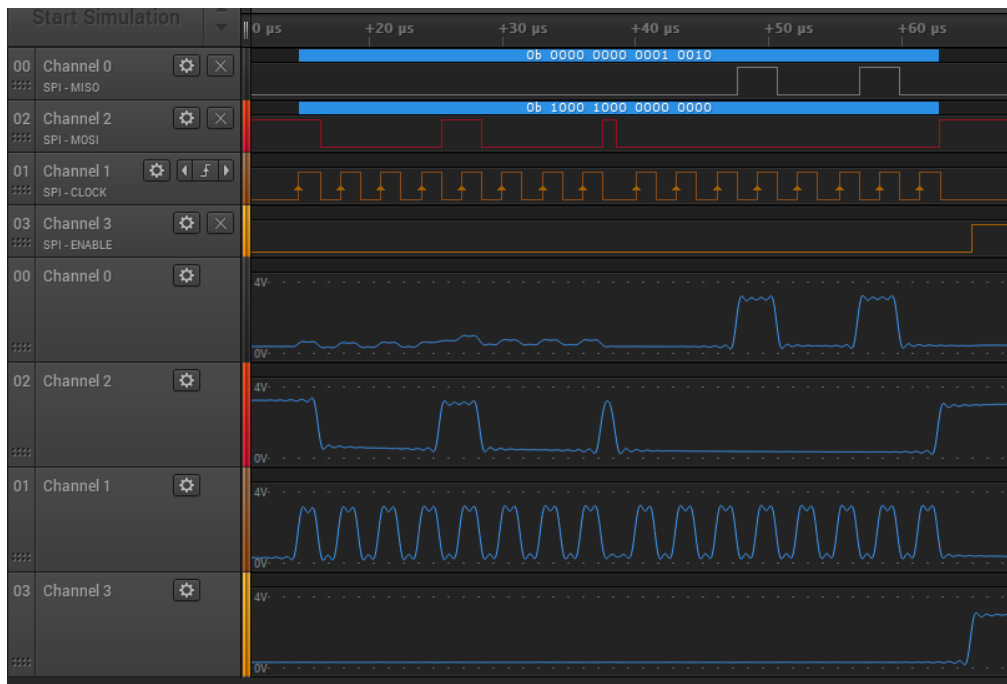


Abbildung 31 Funktionierende SPI Kommunikation mit Antwort

Da die SPI-Kommunikation nun funktionierte, konnten die Register der Master erfolgreich mit den geschriebenen Funktionen ausgelesen werden.

Wird der IO-Link Master ausgelesen, so gibt das Programm folgendes für Port 3 aus, auf welchem ein IO-Link Sensor steckt.

Zustand	Status	Mode	DIO Config	C/Q Config
24V Speisung aus	0001 0000	0000 1000	0000 0000	0000 0000
24V Speisung ein, kein Sensor	0001 0000	0000 0000	0000 0000	0000 0000
24V Speisung ein, Sensor unbetätigt	0001 1000	0000 0000	0000 0000	0000 0000
24V Speisung ein, Sensor betätigt	0010 1000	0000 0000	0000 0000	0000 0000

Tabelle 4 Resultate SPI Register ausgelesen

Hier werden nur die Register «Mode» und «Status» behandelt, für die restlichen Register wird auf das Datenblatt [8]S18ff verwiesen.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
x	x	DiLvl	~QLvl	C/QFault Int	UV33 Int	UV24 Int	OTemp Int

Tabelle 5 Register «Status»

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RST	WuEnBit	X	C/QFault	UV24	OTemp	UV33En	LDO33Dis

Tabelle 6 Register «Mode»

Die Register zeigen den korrekten Status an. Der verwendete Sensor gibt auf der DI Leitung ein analoges 0-10V Signal aus. Dies spiegelt sich in Bit5 des Statusregisters wieder. Weiter ist er genau das Inverse vom Status Register Bit4, welche das Niveau von C/Q Anschluss wiedergibt.

Bit 3 im Mode Register zeigt eine Unterspannung an, ist auch korrekt, da keine 24V Speisung anliegt.

Zustand	IRQ	Unterspannung	LI
24V Speisung aus	1	1	1
24V Speisung ein, kein Sensor	1	0	1
24V Speisung ein, Sensor unbetätigt	1	0	1
24V Speisung ein, Sensor betätigt	1	0	0

Tabelle 7 Resultate digitale Eingänge

Die Funktion der digitalen Eingänge ist korrekt. Der IRQ ist low Activ, somit hat der Master kein Fehler gespeichert. Die Unterspannung ist auch korrekt. Li ist das Invertierte von Di, was heisst, dass auch diese Funktion korrekt ist.

6.3 Verwendete Software

Hier werden alle Softwares aufgelistet, die während der Projektarbeit gebraucht wurden.

Software	Verwendung	URL
KiCAD	Erstellen des PCB Layout	kicad-pcb.org
Eclipse mit Arduino Plugin "Sloeber"	Programmieren des Arduinos	eclipse.baeyens.it
LT-Spice	Simulation der Schaltung	linear.com/designtools/software/
Octave	Darstellung der Messergebnisse	gnu.org/software/octave/
Inkscape	Erstellen der Grafiken	inkscape.org
Software Ideas Modeler	Erstellen des Klassendiagramms	softwareideas.net
Word	Erstellen des Berichtes	

Tabelle 8 Verwendete Software

Zu KiCAD wurden noch einige Plugins verwendet:

- KiCad StepUp, 3D Plugin für FreeCAD. Ermöglicht den Datenaustausch zwischen FreeCAD und KiCAD
- KiCost zum Erstellen von interaktiven BOM's
- HashDefineElectronics/KiCad_BOM_Wizard zum Erstellen statischer BOM's

6.1 Verwendete Messmittel

Messmittel	Info
Saleae Logic 8	Acht Kanal Logic Analyser. Jeder Kanal analogfähig
Multimeter	Diverse Messungen

Tabelle 9 Messmittel

7 Erfolgskontrolle

7.1 Zeitmanagement

Die Planung konnte eigentlich nur bis zum Punkt 2 eingehalten werden. Ein Grund dafür war, dass zu wenig Zeit in die ersten Schritte gelegt wurde. Das Einarbeiten ging noch nach Plan, auch das Pflichtenheft konnte zur rechten Zeit fertig gestellt werden.

Die grosse Schwierigkeit war das Schema.

Die nicht ganz alltägliche Buck-Boost-Konverter Schaltung war so nicht eingeplant und hat dementsprechend auch deutlich mehr Zeit in Anspruch genommen als gedacht. Die passenden Komponenten auf Digikey zu finden war nicht ganz so leicht, so wurde immer wieder Zeit mit der Suche nach Dioden oder Drosseln verbraucht. So kam ca. ein Mehraufwand von einer Woche zustande

Der zweite grosse zeitraubende Fehler war der falsche IO-Link Master MAX14827. Das PCB-Layout war im Großen und Ganzen schon fertig, als herauskam, dass das falsche IC ausgewählt wurde. Dies hat dann zu einer Verzögerung von ca. drei Wochen geführt.

Das PCB konnte dann erst am 7.12.2017 bestellt werden, anstatt wie geplant am 10.11.2017

Das Löten und die Inbetriebnahme der IO-Link Shields war auch nicht eingeplant. Dies ging bei der Planung vergessen. (ca 10-15h, inkl. Messungen)

Dies war denn auch der Grund, weshalb nicht so viel Software geschrieben wurde.

7.2 Anforderungen und Vorgaben

Anforderung	Priorität	Umgesetzt
M12-5 Anschlüsse	M	Ja
Der Master ist in der Port Class A ausgeführt	M	ja
Multimaster, Mehr als 1 IO-Link Master	M	Ja
Messanschlüsse für DI/DQ und C/Q mit Spannungsteiler	1	Ja
Betrieb mit 24V und 5V	1	ja
Dedizierter Massenpin	2	Ja
GECKO4-Educatoir ⁵ Kompatibilität	3	Ja
Grösstmögliche Kompatibilität zu anderen Arduino Boards	2	ja
Library für das IO-Link Master Shield	1	Nein
Eine IO-Link LED Leuchte Ansprechen	M	Nein
Gehäuse für den IO-Link Master	3	Nein
IOT Anbindung ermöglichen	3	Nein

Tabelle 10 Anforderungen

⁵ <https://www.microlab.ti.bfh.ch/wiki/huce:microlab:projects:internal:gecko:gecko4-education:start> FPGA Education Board

8 Offene Punkte und weiterführende Arbeiten

Es sollten Lötjumper oder 0Ω Widerstände für die Verbindung vom SPI-Headers zu Pin 50 52 53. Die Pins wurden wegen der SPI Kompatibilität mit dem GECKO4-Education herausgeführt
Die Löcher für die 4mm Laborstecker Anschlüsse müssen auf 6mm verkleinert werden, oder eine andere Lösung gefunden werden. Die momentane Lösung verhindert, dass das Arduino MEGA gebraucht werden kann. Der USB-B Anschluss ist im Weg
Trotzdem durchgängige Steckverbinder für die Pins brauchen, z.B. SAM1206-50-ND. Dies erleichtert die Fehlersuche ungemein. Ermöglicht auch das einfache Anschliessen irgendwelcher Peripherie.
Buck-Boost Konverter mit höherer Schaltfrequenz redesignen. Dies ergibt kleinere und günstigere Bauteile.

Bei einer erneuten Bestellung den Stencil nicht vergessen

Vor einer allfälligen grösseren Bestellung die Funktion der IO-Link Funktion prüfen.

Der grösste Punkt ist das Erstellen einer IO-Link Master Software z.B. nach http://www.io-link.com/share/Downloads/Spec-IODD/IO_Device_Description_V1.1_Specification.zip

9 Schluss

Meine Motivation für dieses Projekt war, dass es sich um das Erstellen von einem PCB handelte. Da ich in diesem Bereich wenig bis keine Erfahrungen gemacht habe, dachte ich mir, es wäre eine gute Möglichkeit hier eigene Erkenntnisse zu erlangen. Als Automatisierer hatte ich oft mit Sensoren und SPS zu tun. Deshalb dachte ich mir, dass diese Projektstudie eine gute Möglichkeit wäre, mein Wissen im Bereich IO-Link zu vergrössern.

Ich würde sicher nicht mehr alles gleich machen. Diverse weiterführende Arbeiten und Verbesserungen sind schon in den offenen Punkten beschrieben. Das Datenblatt von IC's werde ich zukünftig besser studieren, respektive auch einmal die Webseite des Herstellers anschauen.

IO-Link wird noch weiter an Bedeutung gewinnen, unter anderem weil es immer wichtiger wird den Ausfall von Sensoren oder Aktoren im Voraus zu erkennen und dadurch Betriebsausfälle zu vermeiden. Hier kann IO-Link als offener Standard einen Teil dazu beitragen.

Von KiCAD war ich positiv überrascht. Nach einer gewissen Einarbeitungszeit, die relativ kurz war, gelang es mir recht gut mit der Software umzugehen. KiCAD kann vielleicht nicht mit Altium Designer mithalten, aber für kleine bis mittlere Projekte ist KiCAD definitiv eine sehr gute alternative.

Mit dem Ergebnis bin ich zufrieden. Dieses IO-Link Shield gibt gerade auch wegen der aufwändigen Speisung eine sehr angenehme Flexibilität, die ich während dem Testen immer wieder schätzte.

Ich konnte in dieser Projektarbeit viele Erfahrungen im Design und Erstellen von PCB's sammeln. Hier kann ich auch auf das Modul «Advanced PCB Layout» hinweisen, welches einen sehr guten Einblick in die Thematik gibt. Ohne dieses Modul wäre ich nicht in der Lage gewesen, diese Projektarbeit zu schreiben.

10 Abbildungsverzeichnis

Abbildung 1 Relaissteuerung	5
Abbildung 2 SPS Steuerung	5
Abbildung 3 SPS Steuerung mit Verteilerbox	6
Abbildung 4 SPS mit Kabel und Verteiler	6
Abbildung 5 SPS mit BUS	6
Abbildung 6 SPS mit Sensoren am Bus	7
Abbildung 7 SPS mit IO-Link Master am Bus	7
Abbildung 8 Distanzmessung konventionell vs. IO-Link	8
Abbildung 9 Format eines SDCI UART Frames [3]	9
Abbildung 10 Verbindungsaufbau IO-Link [3]	10
Abbildung 11 WAGO 750/753	10
Abbildung 12 Balluff BNI EIP-502-105-Z015	10
Abbildung 13 Vergleich Port Class A & B	11
Abbildung 14 Blockdiagramm Speisung	14
Abbildung 15 Simulationsaufbau Boost-Converter	15
Abbildung 16 Simulationsaufbau Buck-Converter	15
Abbildung 17 5V im Boost-Modus	16
Abbildung 18 Messanschluss Port 0-3	18
Abbildung 19 Vorschau von oben	18
Abbildung 20 Vorschau von unten	19
Abbildung 21 Messpunkte	20
Abbildung 22 Spannung Buck Converter über 10s	21
Abbildung 23 Rippel Ausgang Buck-Converter nach 10s Betrieb	21
Abbildung 24 Wechsel Boost-Buck Mode	22
Abbildung 25 Wechsel Buck-Boost Mode	22
Abbildung 26 Spannungen Boost-Converter	23
Abbildung 27 Rippel 5V im Boostmodus	23
Abbildung 28 Klassendiagramm MAX14824	24
Abbildung 29 Fehlerhafte SPI Kommunikation	24
Abbildung 30 Funktionierende SPI Kommunikation mit Antwort	25

11 Tabellenverzeichnis

Tabelle 1 Vergleich Port Class A & B	11
Tabelle 2 Vergleich Arduino Plattformen	12
Tabelle 3 Vergleich IO-Link Master	13
Tabelle 4 Resultate SPI Register ausgelesen	25
Tabelle 5 Register «Status»	25
Tabelle 6 Register «Mode»	25
Tabelle 7 Resultate digitale Eingänge	26
Tabelle 8 Verwendete Software	26
Tabelle 9 Messmittel	26
Tabelle 10 Anforderungen	27

12 Glossar

B

BFH BFH: Berner Fachhochschule, Technik und Informatik (BFH-Ti) vertreten durch Torsten Mähne, Thomas Schott, Max Felser

F

Firma Balluff AG Kletenstrasse 49, 2564 Bellmund, Pascal Gaggero

S

SDCI Single-Drop Digital Communication Interface

SIO Standard Input / Output

T

Tdmt Master message delay

W

WURQ Wake Up Request

13 Quellen

- [1] *Was ist IO-Link*. [Performance]. Balluff AG Bellmund, 2016.
- [2] T. Mähne, «Ausschreibung Projektarbeit,» Okt 2017. [Online]. Available: <https://moodle.bfh.ch/mod/data/view.php?d=2789&rid=37819&filter=1>.
- [3] IO-Link, «IO Link Interface ans System Specification V1.1.1,» Juli 2013. [Online]. Available: http://www.io-link.com/share/Downloads/Spec-Interface/IOL-Interface-Spec_10002_V112_Jul13.pdf.
- [4] E. H. J. G. S. (Hrsg.), «Sensoren in Wissenschaft und Technik,» Deutschland, Vieweg+Teubner Verlag, 2012, pp. 649-651.
- [5] IO-Link.com, «IO-Link Planungsrichtlinie 10911,» Sept 2016. [Online]. Available: <http://www.io-link.com/share/Downloads/Planung/IO-Link-Planungsrichtlinie-10911-V10-Sep16.pdf>. [Zugriff am 08 Januar 2018].
- [6] Atmel, «Atmel.com,» Atmel, 2015. [Online]. Available: http://www.atmel.com/Images/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf. [Zugriff am 06 Oktober 2017].
- [7] M. Integrated, «MAX14827A,» Oktober 2017. [Online]. Available: <https://www.maximintegrated.com/en/products/interface/transceivers/MAX14827A.html>.
- [8] Maxim Integrated, «MAX14824,» Oktober 2017. [Online]. Available: <https://www.maximintegrated.com/en/products/interface/transceivers/MAX14824.html>.
- [9] «ltwiki.org,» 13 Mai 2013. [Online]. Available: http://www.ltwiki.org/?title=Simulating_the_MC34063_in_Inverter_Configuration_with_an_Accurate_TL431A_Model. [Zugriff am 24 Okt 2017].

Anhang I Zeitplan

[illegible]