

Fachbericht

Projekt 3 - Team 4

Windisch, 17. Januar 2019

Auftraggeber: Prof. Hans Gysin

Betreuung: Matthias Meier
Dr. Anita Gertiser
Bonnie Domenghino
Prof. Dr. Pascal Schleuniger

Projektleiter: Dominik Müller

Team: Pascal Fankhauser
Nico Canzani
Fabian Frey
Silvan Koch

Studiengang: Elektro- und Informationstechnik

Abstract

Today, many rechargeable lithium-ion cells are thrown away although they are still partially functional and can be reused in other applications. One such application is a home battery system capable of supplying an entire home with electricity. Used batteries have different capacities. For this reason, a special and efficient charging circuit is required. Such a system is the goal of this project. This presented solution includes various safety features, is easy to use and affordable at the same time. Especially since overcharged lithium-ion cells are dangerous and can burst into flames. A prototype was created to test the circuit design of the system. Hardware components from linear technologies were used because they provided two chips for battery management. The system monitors six blocks connected in series. Each block consists of 36 cells connected in parallel. Furthermore, efficient active cell balancing has been implemented to store the energy generated. To guarantee safety and reliability, a temperature sensor has been placed close to the whole battery pack. This allows the system to be shut down if the ambient temperature rises above 45 °C, thus reducing any fire hazard. The electrical circuit is designed modularly so that subsequent changes can be easily incorporated. This also allows the voltage to be as high as 150 volts. A master/slave model is used to implement the overall system. A Raspberry Pi Zero combined with an extension board for the communication acts as the master unit. The slave units are daisy-chained and are connected to the master unit by a twisted-pair cable. They measure the cell voltages, monitor the temperatures and are responsible for the active balancing. Theoretical calculations are verified by measurements in the laboratory with calibrated measuring tools. These tests show that an efficiency of approximately 81 % is reached. The temperature range is set by the cells and is between 0 °C and 45 °C.

Keywords: battery management system, lithium-ion cells, active balancing, voltage measurement, temperature monitoring, master/slave

Inhaltsverzeichnis

1 Einleitung	1
2 Technische Grundlagen	2
2.1 Balancing	2
2.1.1 Passiv	2
2.1.2 Aktiv	2
2.2 18650 Lithium-Akku	3
2.3 Technische Daten Sunny Boy Storage 2.5	3
3 Batteriemanagementsystem	4
3.1 Blockschaltbild	4
3.2 Schaltung Master	5
3.2.1 Raspberry Pi Zero W	5
3.2.2 EEPROM	5
3.2.3 LTC6820	6
3.2.4 Layout	6
3.3 Kommunikation	7
3.3.1 Master zu Slave	7
3.3.2 LTC6813 zu LTC3300	7
3.3.3 Master zu Sunny Boy Storage 2.5	8
3.4 Schaltung Slave	9
3.4.1 Umsetzung aktives Balancing	9
3.4.2 Spannungsmessung	10
3.4.3 Temperaturüberwachung	11
3.4.4 Temperaturmessung	12
3.4.5 Layout	13
4 Software	14
4.1 Hauptprogramm	14
4.2 Klassen	15
4.2.1 BMS	15
4.2.2 isoSPI	15
4.2.3 SunnyBoy	15
4.3 C++ Programm (isoSPI-Interface)	15

5 Web-App	16
6 Qualitätssicherung	17
7 Schluss	18
Literatur	19
Glossar	19
A Schema Master	20
B Layout Master	21
C Schema Slave	23
D Layout Slave	36
E Berechnungen Kommunikation	40
F Dimensionierung Slave	42
G Prüfanweisung	44
H Prüfprotokoll	56

1 Einleitung

Allein in der Schweiz wurden im Jahr 2017 129'000 Tonnen Elektroschrott gesammelt. Darunter befinden sich auch Geräte, in welchen noch brauchbare Akkus sind. Dies stellt ein erhebliches Problem für die ganze Umwelt dar. Um dem entgegenzuwirken, werden die gebrauchten Zellen aus vermeintlichen Elektroschrottgeräten ausgebaut. Die Zellen werden ausgemessen und nach deren Merkmalen wie Kapazität und Innenwiderstand bewertet. Zellen, die noch brauchbar sind, sollen für Energiespeicher-Anwendungen mithilfe eines Batteriemanagementsystems (BMS) wiederverwendet werden.

Für langfristige Energiespeicher-Anwendungen werden fast ausschliesslich Lithium-Ionen-Akkus eingesetzt. Ein BMS gewährleistet, dass eine Akkuzelle in einem sicheren Betriebsfenster betrieben wird. Die wichtigsten Parameter, die dabei eingehalten werden müssen sind die Spannungen der einzelnen Zellen, Ladeströme und Temperaturen. Des Weiteren werden die Funktionalitäten wie der Ladungsausgleich (Aktives Balancing) zwischen den Zellen verwirklicht.

Werden diese Parameter und Funktionen nicht beachtet, kann die volle Kapazität des Energiespeichers nicht vollumfänglich ausgenutzt werden. Ebenso kann eine Tiefentladung bzw. Überladung der Zellen oder der Betrieb bei inkorrekten Temperaturen den Energiespeicher zerstören. Im schlimmsten Fall resultiert die Zerstörung des Energiespeichers in einem Schadenfeuer, wovon auch umliegende Gegenstände oder das Gebäude betroffen sein können. Bei Lithium-Ionen-Zellen ist es aus Gründen der Langlebigkeit, Leistung und Sicherheit daher unerlässlich, ein Batteriemanagementsystem zu verwenden.

Das hier dokumentierte Projekt befasst sich mit der Entwicklung und Realisierung eines solchen Batteriemanagementsystems. Das Ziel des Endprodukts soll die Gewährleistung einer korrekten Ladung und Entladung, Balancing sowie einen sicheren Betrieb des Energiespeichers sein. Damit soll ermöglicht werden, alte, aber noch funktionstüchtige Akkuzellen weiterhin zu verwenden und die oben genannte Komplikationen zu vermeiden.

Zentrale Aufgabe des BMS ist es, die Spannungen aller Akkublöcke auszumessen und diese für die einzelnen Blöcke individuell anzupassen. Die Anpassung gewährleistet, dass alle Akkublöcke gleichzeitig maximal aufgeladen sowie bis zum Minimum entladen werden. Bei allfälligen Fehlern soll der Benutzer auf den Fehler hingewiesen und das System aus Sicherheitsgründen ausgeschaltet werden. Im Vergleich zu handelsüblichen BMS ist dieses speziell für den Auftraggeber konzipiert. Die Grösse der Prints, Dimensionierung der Bauteile sowie Anforderungen an Funktionen und Statusanzeige sind explizit dafür angepasst.

Dieser Fachbericht stellt die technische Dokumentation des BMS dar. In Kapitel 2 (Technische Grundlagen) werden alle relevanten elektrotechnischen Hintergründe erläutert. Kapitel 3 (Batteriemanagementsystem) beschreibt den Aufbau des Endproduktes. Weiter werden Funktionen, Möglichkeiten sowie Hintergrundinformationen erläutert. Im Kapitel 4 (Software für Masterprint) wird der Schwerpunkt auf die Programmierung des Masterprints gesetzt. Zusätzlich wird die Kommunikation zwischen den Prints und die Anbindung der Webapplikation erklärt. Das letzte Kapitel (Qualitätssicherung) erläutert das Testen des Endprodukts und die Validierung der Ergebnisse.

2 Technische Grundlagen

Die technischen Grundlagen beinhalten alle relevanten Hintergründe, welche zum Verständnis der Projektmaterei notwendig sind. Einerseits werden die Balancing-Methoden erläutert und andererseits wird Bezug auf die verwendeten Lithium-Akkumulatoren genommen.

2.1 Balancing

Bei in Serie geschalteten Akkumulatoren entstehen immer Kapazitätunterschiede, welche eine Verringerung der Gesamtkapazität zur Folge haben. Der so entstehende Ladungsunterschied verursacht, dass die einzelnen Zellen ihre Abschlussspannung zu unterschiedlichen Zeiten erreichen. Wenn vollgeladene Lithium-Ionen-Zellen weiterhin mit einer Ladespannung von mehr als 4.2 V geladen werden, ist es möglich, dass sie durch die Zersetzung ihres Elektrolyts beschädigt werden. Das Gleiche gilt bei der Tiefenentladung. Dabei ist zu beachten, dass eine Zelle nicht unter 2.8 V entladen werden sollte. Die Zelle, welche die niedrigste Kapazität aufweist, bestimmt in einer Serieschaltung die Gesamtkapazität. Um die Verringerung der Gesamtkapazität abzuschwächen wird Cell-Balancing angewandt. Dabei wird zwischen passivem und aktivem Balancing unterschieden [1].

2.1.1 Passiv

Das passive Balancing ist dabei die technisch einfache und dadurch auch weit verbreitete Methode. Die Stärken dieser Methode liegen vor allem im Ladeschlussbereich, also zu dem Zeitpunkt, bei dem die ersten Zellen eines Akkupacks fast vollständig geladen sind. Bei den Zellen, welche die Ladeschlussspannung erreicht haben, wird ein Widerstand parallelgeschaltet und somit deren Spannung begrenzt. So wird sichergestellt, dass die Zelle nur noch geringfügig weiter geladen wird. Die Zellen, welche die Ladeschlussspannung noch nicht erreicht haben, werden davon nicht beeinflusst und können somit weiterhin mit dem vollen Ladestrom versorgt werden. Hauptvorteil dieser Methode ist die kostengünstige und technisch leicht umsetzbare Realisierung. Dafür sind jedoch lange Ladevorgänge und grosse Ladekosten in Kauf zu nehmen. Der Ladevorgang hängt von der schwächsten Zelle des gesamten Systems ab. Dies hat zur Folge, dass die Zellen, welche bereits vollgeladen sind, so lange Energie in Form von Wärme über die Widerstände abgeben müssen, bis die letzte Zelle komplett geladen ist. Die dabei erzeugte Verlustwärme wirkt sich außerdem negativ auf die Lebensdauer der Zellen aus und kann eine mögliche Brandgefahr bedeuten [2].

2.1.2 Aktiv

Die komplexere, aber auch effizientere Variante ist das aktive Balancing. Hier wird ein Ladungstransfer unter den Zellen realisiert. Zellen mit höherer Ladung übertragen dabei Energie an die Zellen mit niedrigerer Ladung. Dieser Vorgang kann während dem ganzen Ladeprozess angewendet werden. Bei dieser Methode wird ein viel gröserer Wirkungsgrad erzielt, als beim passiven Balancing. Grund dafür ist, dass überschüssige Energie nur zu einem geringen Teil in Wärme umgewandelt wird. Die Umsetzung dieser Methode liefert einen höheren Schaltungsaufwand, jedoch kann durch die dazugehörende Laderegelungssoftware zwischen starken und schwachen Zellen unterschieden werden und die Lebensdauer der Akkumulatoren wird dabei deutlich verlängert [3].

2.2 18650 Lithium-Akku

Mit der Bezeichnung des 18650 Lithium-Ionen-Akku wird eine sogenannte Sekundärzelle beschrieben. Die Zahl 18650 bezeichnet die Dimensionen des Akkus. Der Durchmesser beträgt dabei 18 Millimeter und die Länge 65 Millimeter [4].

Sekundärzellen, oder auch Akkumulatoren genannt, sind wiederaufladbare elektrochemische Energiespeicher [5, S. 10]. Um leistungsintensiven Anwendungen gerecht zu werden, werden die Lithium-Ionen-Akkus als Parallel- und/oder als Serienschaltung verwendet. Durch die anwendungsspezifische Schaltung kann die verfügbare Kapazität durch ein BMS mit Balancing-Funktion optimiert werden. Alterungsbedingt können die Innenwiderstände und Kapazitäten der einzelnen Zellen variieren. Dies führt zu Tiefentladung oder Überladung einer Zelle, sofern keine Massnahmen getroffen werden. Die Kapazität bleibt dabei innerhalb der Lebensdauer eines Lithium-Ionen-Akkus länger erhalten.

2.3 Technische Daten Sunny Boy Storage 2.5

Für das Laden und Entladen wird der Batterie-Wechselrichter „Sunny Boy Storage 2.5“ eingesetzt. Beim Sunnyboy Storage 2.5 handelt es sich um den vorgegebenen Laderegler.

Wird der Sunnyboy Storage 2.5 über die Batterie versorgt, so verfügt er über eine maximale Gleichstromleistung von 2650 Watt bei einer maximalen Gleichspannung von 500 V am Eingang. Der Spannungsbereich liegt zwischen 100 V und 500 V und beträgt typischerweise 360V. Der maximale Strom beträgt 10 A (Nennstrom) oder 18 A kurzzeitig [6].

3 Batteriemanagementsystem

Die technische Umsetzung erfolgt über ein Master/Slave-Prinzip. Es wird somit eine Steuerung (Master) eingesetzt, welche bis zu zehn Überwachungsprints (Slaves) steuert. Mit jedem Slave können zwölf Blöcke überwacht und ausgeglichenen werden (siehe Abbildung 3.1). Bei insgesamt 120 Blöcken wird somit eine Ausgangsspannung von 336 V_{DC} bis 504 V_{DC} erreicht. Im Rahmen dieses Projektes wurde ein Prototyp entwickelt, welcher für sechs Blöcke ausgelegt ist. Die einzelnen Komponenten dieses Prototyps werden in den nachfolgenden Kapitel näher erläutert.

3.1 Blockschaltbild

Abbildung 3.1 zeigt den Aufbau (Blockschaltbild) des Systems.

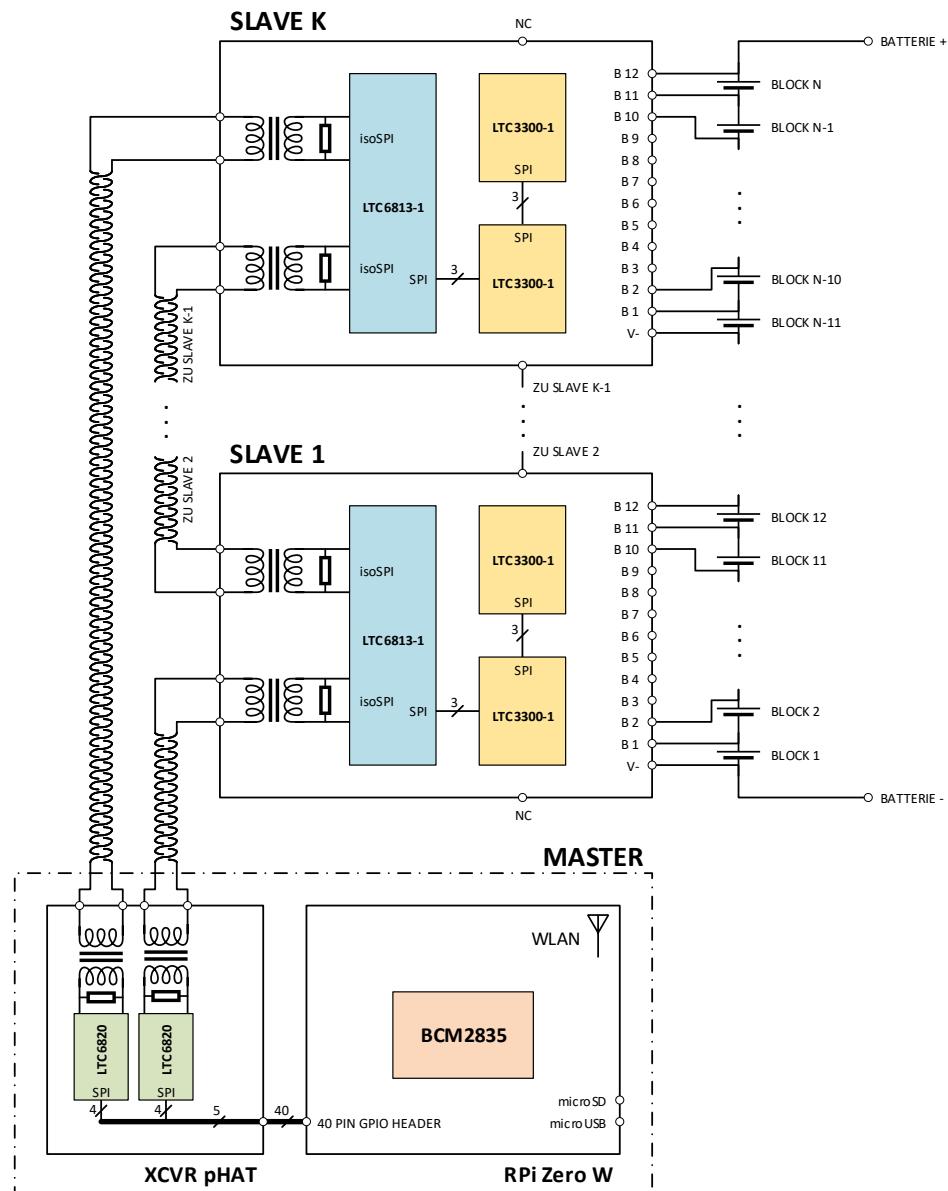


Abbildung 3.1: Blockschaltbild

3.2 Schaltung Master

Der Aufbau des Masters besteht einerseits aus einem Raspberry Pi Zero W und andererseits aus dem XCVR pHAT. Beim XCVR pHAT handelt es sich um eine eigens entwickelte Aufsteckplatine, welche als Kommunikationsschnittstelle zwischen dem Raspberry Pi Zero W und dem Überwachungsprint (Slave) dient. Dies ist notwendig, um die gewünschte Reichweite und die benötigte Isolation der Kommunikation zu gewährleisten.

3.2.1 Raspberry Pi Zero W

Der RPi Zero W verfügt über eine 40 Pin Stifteleiste (siehe Abbildung 3.2), welche als Schnittstelle für die Aufsteckplatine dient. Die notwendigen SPI-Signale gelangen so vom BCM2835 SoC zum XCVR pHAT.

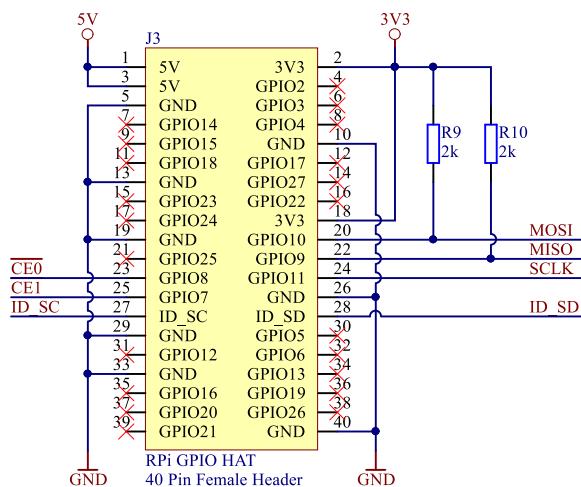


Abbildung 3.2: 40 Pin Stifteleiste

3.2.2 EEPROM

Die Aufsteckplatine verfügt über ein 32 kB EEPROM (siehe Abbildung 3.3). Dieses dient dazu, beim Aufstartvorgang des RPi die GPIOs richtig zu konfigurieren. Die Konfiguration wird zuvor auf dem EEPROM U3 gespeichert. Danach kann der Widerstand R11 bestückt werden, sodass der Schreibschutz aktiviert wird und die gespeicherte Konfiguration nicht ungewollt überschrieben werden kann. Des Weiteren ist das EEPROM U3 nötig, damit die Aufsteckplatine offiziell ein pHAT genannt werden darf [7].

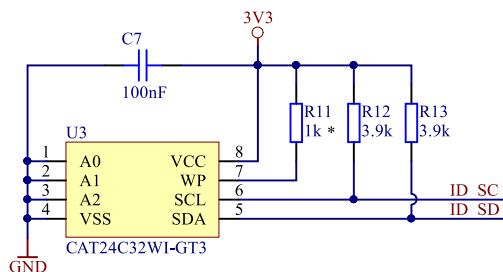


Abbildung 3.3: EEPROM CAT24C32 32 kB

3.2.3 LTC6820

Aufgrund der verschiedenen Spannungspotenzialen der Überwachungsprints muss die Kommunikation isoliert sein. Dies wird durch den Einsatz vom IC LTC6820 (Linear Technology) in Kombination mit einem Transformator erreicht (siehe Abbildung 3.4). Der IC U1 wandelt dabei die SPI-Signale in sogenannte isoSPI-Signale um. Diese ermöglichen die Kommunikation über eine verdrillte Zweidrahtleitung, welche Vollduplex ist. Die Amplitude der eben genannten isoSPI-Signale kann über die zwei Widerstände R3 und R4 eingestellt werden. Somit kann durch das Erhöhen des Stromverbrauchs die Reichweite vergrössert werden. Die Widerstände R3 und R4 wurden so dimensioniert, dass eine Reichweite von 50 m möglich ist (siehe Anhang E). Die in Abbildung 3.4 gezeigte Schaltung ist auf der Aufsteckplatine zwei Mal vorhanden. Somit können bei einem Kabelbruch der Kommunikationsleitung trotzdem noch alle Überwachungsprints erreicht werden (siehe Abbildung 3.1). Die Widerstände R1 und R2 dienen als Abschlusswiderstände um Reflexionen zu minimieren. Des Weiteren wird der Kondensator C1 eingesetzt, um Störungen auf Masse zu leiten. Die beiden Kondensatoren C2 und C3 sind Stützkondensatoren für die Speisung und den Ausgangstreiber des LTC6820. Der Transformator T1 verfügt über eine integrierte Gleichtaktdrossel, um Gleichtakt-Störungen zu unterdrücken [8].

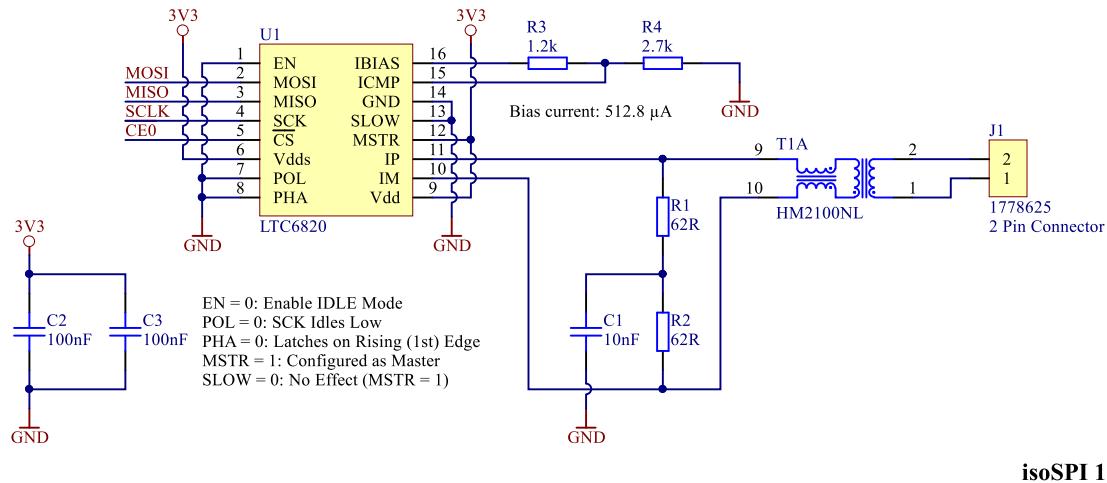


Abbildung 3.4: LTC6820 - Isoliertes Kommunikationsinterface (isoSPI)

3.2.4 Layout

Die Grösse der Aufsteckplatine ist gegeben durch die Grösse des RPi Zero W und den Spezifikationen für pHATs. Aufgrund der niedrigen Komponentenanzahl konnte das Layout mit nur zwei Kupferlagen realisiert werden (siehe Anhang B). Dabei wurde die Kupferlage auf der Unterseite als Massefläche verwendet. Dies hat den Vorteil, dass mittels Durchkontaktierungen einfach eine Verbindung auf Masse realisiert werden kann. Diese Massefläche wird nur von einer kurzen Leiterbahn (3.3 V Speisung) durchschnitten. Des Weiteren wurde die Massefläche um die THT-Stecker der Kommunikationsleitungen entfernt, sodass keine Störungen über die schnellen Kommunikationsleitungen (siehe Kapitel 3.3) eingekoppelt werden. Der Transformator wurde möglichst nahe an den zwei Kommunikationsstecker platziert, um Störungen direkt zu unterdrücken. Dies vergrössert gleichzeitig den Abstand zu den beiden LTC6820, was die Auswirkungen von Störungen weiter unterdrückt [7].

3.3 Kommunikation

Die Kommunikation lässt sich in drei unterschiedliche Bereiche unterteilen. Diese sind die Kommunikation zwischen dem Master- und dem Slaveprint, zwischen dem LTC6813 und LTC3300 sowie zwischen dem Masterprint und dem Sunny Boy Storage 2.5.

3.3.1 Master zu Slave

Die Kommunikation zwischen dem Master und den Slaveprints erfolgt über die proprietäre isoSPI Schnittstelle von Linear Technology. Es handelt sich dabei um eine isolierte SPI-Schnittstelle. Dazu werden die SPI-Signale vom Baustein LTC6820 über zwei verdrillte Leiter und zwei Transformatoren (siehe Abbildung 3.4) an die Überwachungsprints übermittelt. Damit bei einem Unterbruch der Kommunikationsleitung zwischen zwei Überwachungsprints die Kommunikation trotzdem noch gewährleistet ist, erfolgt diese umkehrbar mit zwei LTC6820. Diese werden auf der Transceiver-Erweiterung für den „RPi Zero W“ (XCVR pHAT) platziert.

Die Abbildung 3.4 zeigt den Schema Aufbau der Kommunikation auf dem pHAT. Die isoSPI-Signale welche vom LTC6820 ausgehen, bzw. empfangen werden, werden über den Baustein HM2100NL (T1A) gefiltert und sind dann über den Stecker J1 verdrillt zu den Slaveprints geführt.

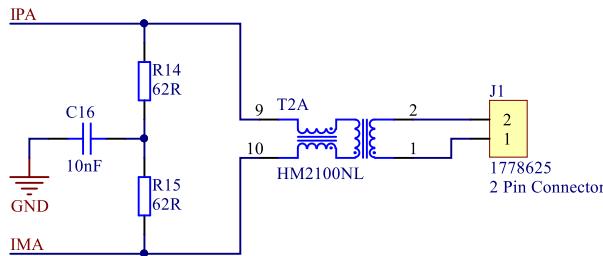


Abbildung 3.5: Schema Kommunikation Slave

Abbildung 3.5 zeigt die Beschriftungssituation auf dem Slaveprint auf. Dort werden die Signale nach dem Eintreffen wieder gefiltert und danach an den LTC6813 geführt.

3.3.2 LTC6813 zu LTC3300

In einem ersten Schritt wird über die isoSPI Schnittstelle und passendem Befehl ein Register des LTC6813 beschrieben. Dieses Register enthält Informationen für den LTC3300, welche die angeforderten Zustände der einzelnen Blöcke zum Balancing enthalten. Mit einem weiteren Kommando sendet der LTC6813 über eine SPI Schnittstelle diese Informationen an den LTC3300. Der LTC3300 kennt nun die geforderten Zustände der einzelnen Blöcke.

In einem zweiten Schritt erfolgt die Ausführung des Befehls. Der Vorgang ist der Selbe, wie beim ersten Schritt. Nun erfolgt aber die Aufforderung an den LTC3300, den zuvor erhaltenen Befehl auszuführen.

Mit dem gleichen Vorgehen ist es auch möglich, den momentanen Ausgleichstatus auszulesen. Die Antwort des LTC3300 wird in einem Register des LTC6813 gespeichert und kann von dort mit dem Raspberry Pi Zero W über die isoSPI-Schnittstelle ausgelesen werden.

3.3.3 Master zu Sunny Boy Storage 2.5

Die Kommunikation des Sunny Boy Storage 2.5 würde über einen Modbus erfolgen. Dieser Modbus wird in diesem Projekt jedoch nicht realisiert. Für die Projektarbeit wird die Kommunikation nur simuliert. Diese Schnittstelle würde dem Sunny Boy Storage 2.5 das Signal zum Laden und Anhalten geben.

3.4 Schaltung Slave

Auf dem Slaveprint findet die vom Master angeordnete Umsetzung des aktiven Balancing statt. Neben dem aktivem Balancing wird auch die Temperaturüberwachung, die Temperaturmessung sowie die Spannungsmessung ausgeführt.

3.4.1 Umsetzung aktives Balancing

Bei der Umverteilung der Energie zwischen den Blöcken wird die Energie von einem Block zu einem anderen weitergegeben, dadurch werden durchgehend kleine Potentialanpassungen realisiert. Der verwendete Baustein LTC3300-1 von Linear Technology kann bis zu sechs Blöcke balancieren. Um zwölf Blöcke gleichzeitig zu balancieren, werden auf dem Überwachungsprint (Slave) zwei LTC3300-1 platziert. Gemäss dem Datenblatt von Linear Technology wird mit geeigneter Beschaltung ein Wirkungsgrad von bis zu 90 % erreicht. Zu diesem Zweck wird dazu pro Block ein Transformator (hier ein 750312504 von Würth Elektronik) und zwei FETs (hier SiR882DP von Vishay) verwendet. Primärseitig wird durch das Durchschalten des FETs Energie auf die Spule des Transformators gegeben. Mit der korrekten Ansteuerung der sekundärseitigen FETs kann die Energie auf einen gewünschten Block weitergegeben werden. Die Aufgabe gibt folgende Ladesituation vor: Insgesamt sollen 40 Blöcke a 36 Zellen geladen werden können. Die Kapazität von einem Block beträgt mit $36 \cdot 2.4 \text{ Ah} = 86.4 \text{ Ah}$. Pro Stunde sollen 5 % von der Maximalkapazität durch das Balancing verschiebbar sein. Der Lade- und Entladestrom muss somit 4.3 A betragen, dadurch entstehen Spitzenströme von 10 A welche durch den Transformator fliessen. Um zu verhindern, dass die Spitzenströme Störsignale auf das ganze System übertragen und um zu garantieren, dass die Batterieblöcke mit einem konstanten Strom geladen und entladen werden, sind die Stützkondensatoren (C3 bis C6) verantwortlich. Aufgrund schneller Schaltzyklen muss die Gate-Kapazität oft umgeladen werden. Um den Ladestrom dieser Gate-Kapazität zu begrenzen wird ein Gate-Widerstand eingesetzt. Mit der Spannung über den Strommesswiderständen wird beim LTC3300 der Balancingstrom berechnet. Beim Erreichen der Grenzströme wird der dazugehörige FET sperrend bzw. leitend. Der benötigte Strommesswiderstand lässt sich entsprechend Formel 3.1 berechnen. Die weiteren Berechnungen für die passiven Komponenten der Beschaltung sind im Anhang, Kapitel F ersichtlich.

$$R_{Shunt} = \frac{50mV}{I_{peak}} \quad (3.1)$$

Die Abbildung 3.6 zeigt die Umsetzung des Balancing für einen Batterieblock. Auf dem Prototyp wurden sechs dieser Einheiten realisiert.

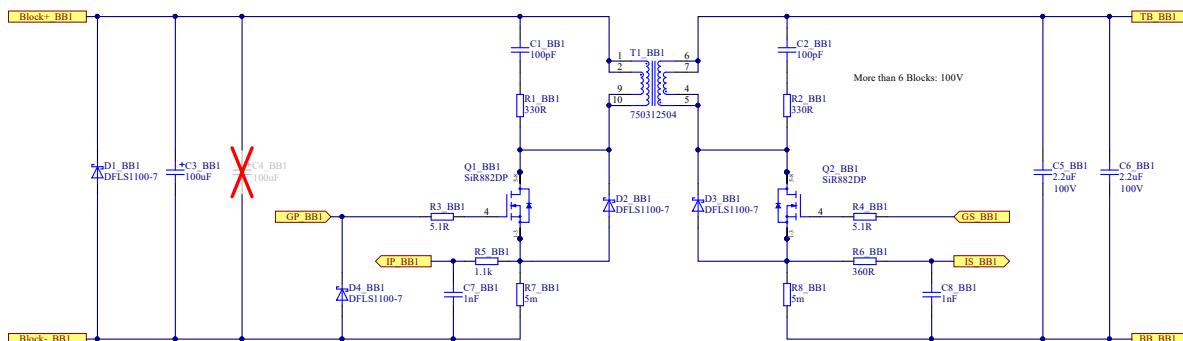


Abbildung 3.6: Schemaaufbau aktives Balancing

3.4.2 Spannungsmessung

Die Spannungsmessung wird mit dem LTC6813 von Linear Technology realisiert. Dabei handelt es sich um einen IC, welcher zur Spannungsmessung von bis zu 18 Zellen genutzt werden kann. Damit der Überwachungsprint nicht zu gross wird, werden allerdings nur zwölf Blöcke angeschlossen. Somit werden auch nur 12 von den 18 Messkanälen verwendet. Der benötigte Messbereich, welcher für eine Li-Ion-Akkuzelle relevant ist, liegt zwischen 2.8 VDC und 4.2 VDC. Der LTC6813 deckt einen Bereich von 0 VDC bis 5 VDC bei einem Messfehler von 2.2 mV ab. Die analogen Spannungswerte werden gewandelt und digital an den Master weitergeleitet. In der folgenden Abbildung wird der Schemaufbau aufgezeigt und im Anschluss erklärt.

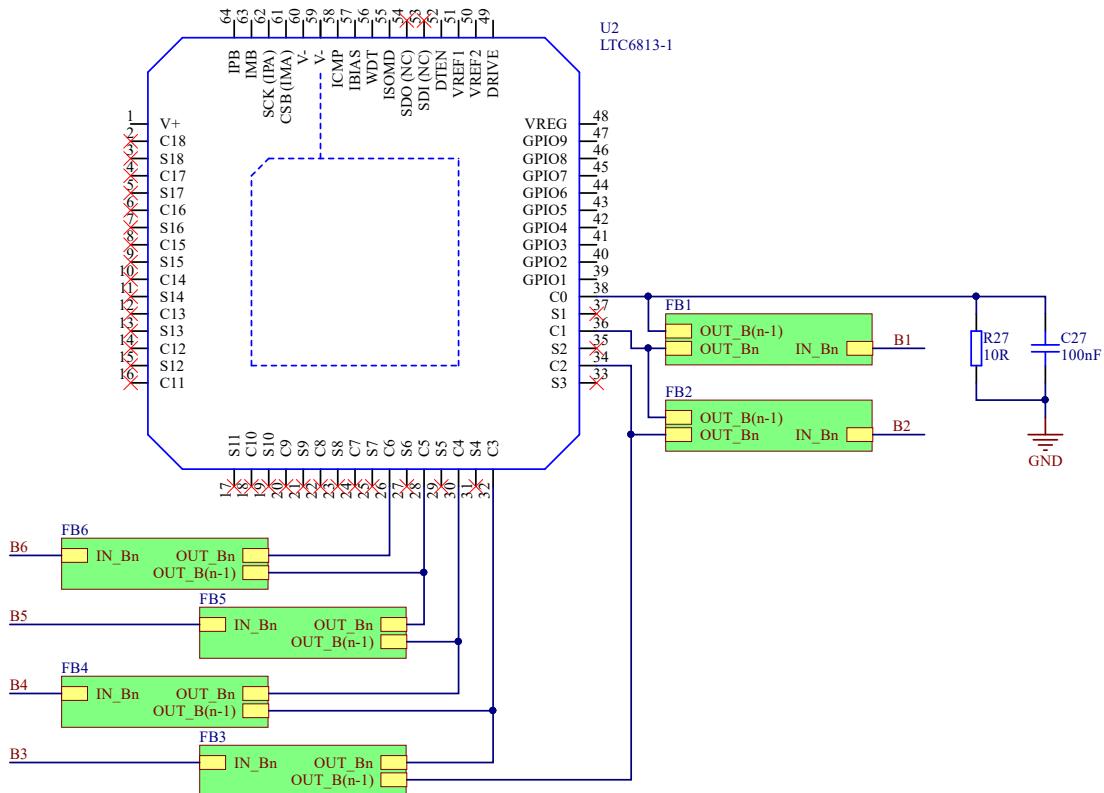


Abbildung 3.7: Schema Spannungsmessung

Gegenüber dem Originalprodukt ist der Prototyp für ein Balancing von sechs Blöcken entwickelt, somit wird auch die Spannungsmessung nur für diese Anzahl ausgelegt. Über die Leitungen B1 bis B6 sowie durch die Filterblöcke FB1 bis FB6 werden die Blockspannungen zum LTC6813 geführt. Wie in der Abbildung 3.8 ersichtlich ist, bestehen die Filterblöcke jeweils aus einem Widerstand und einem Kondensator. Die Filter wurden gemäss Datenblatt dimensioniert, sodass sie ein Tau von 1us aufweisen. Bei allfälligen Störungen kann pro Block ein weiterer Kondensator gegen Masse dazu geschaltet werden.

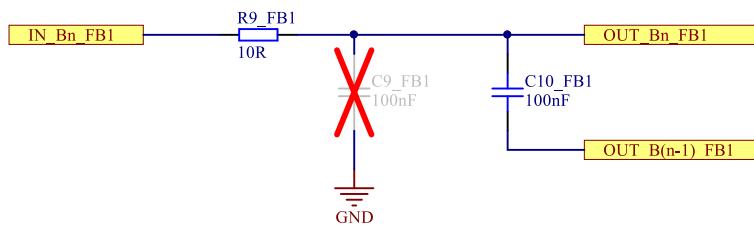


Abbildung 3.8: Schema Filterblock

3.4.3 Temperaturüberwachung

Die Temperaturüberwachung der einzelnen Zellen erfolgt über 18 in Serie geschaltete NC-Temperaturschalter mit einem Schaltwert von 50 °C (siehe Abbildung 3.9). Dabei werden im Abstand von zwei Zellen, Temperaturschalter zwischen den Zellen angebracht. Dies erfordert einen physischen Eingriff in die bestehenden Blöcke. Somit werden pro Block zwei Leiter auf den Überwachungsprint geführt. Da dieser dafür ausgelegt ist, zwölf Blöcke zu balancieren sowie zu überwachen, werden somit 24 Eingänge für die Temperaturüberwachung benötigt. Die erfassten Daten werden vom Master ausgewertet und bei einer Temperatur über 50 °C wird der Lade- oder Entladevorgang unterbrochen.

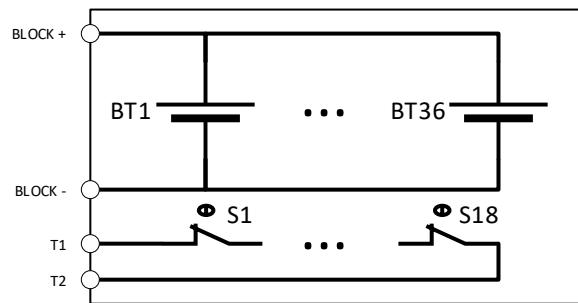


Abbildung 3.9: Schema Temperaturüberwachung

Bei dem entwickelten Prototyp wird die Temperaturüberwachung durch zwei Taster realisiert. Durch das Drücken einer der beiden Taster wird das Auslösen eines Temperaturschalters simuliert und somit der Überwachungskreis unterbrochen.

3.4.4 Temperaturmessung

Die Messung der Umgebungstemperatur wurde durch das Anbringen eines PT1000 an einem der Überwachungsprints realisiert. Dieser temperaturabhängige Widerstand hat einen positiven Temperaturkoeffizienten. Dies bedeutet, dass sein Leitwert bei Erhitzung abnimmt. Der Spannungsabfall über dem PT1000 wird gemessen und digital an den Master übermittelt. Dieser ermittelt dann den Widerstandswert des PT1000 und kann somit die Temperatur bestimmen. Dabei ist es wichtig, den Widerstand mit genügend grossem Abstand von verlustbehafteten Bau teilen zu platzieren. Dadurch können Messfehler verminder werden. Die nachfolgende Abbildung zeigt den Schaltungsaufbau der Temperaturmessung.

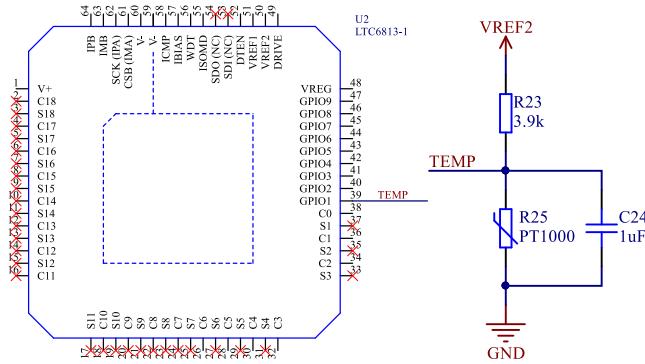


Abbildung 3.10: Schema Temperaturmessung

In der Abbildung 3.10 ist ersichtlich, wie der PT1000 mit dem LTC6813 verbunden ist. Dem PT1000 ist ein $3.9\text{ k}\Omega$ Widerstand als Spannungsteiler vorgeschaltet. Der LTC6813 legt die 3 V Referenzspannung am Spannungsteiler an (V_{Ref2}). Am Pin GPIO1 wird die Spannung über dem PT1000 gemessen. Dieser Spannungswert wird dem Master übermittelt und von diesem ausgewertet. Nachdem der Widerstandswert berechnet wurde, kann mittels der Formel 3.2 die Temperatur berechnet werden [9].

$$T = \frac{-R_0 \cdot \alpha + \sqrt{R_0^2 \cdot \alpha^2 - 4 \cdot R_0 \cdot \beta \cdot (R_0 - R)}}{2 \cdot R_0 \cdot \beta} \quad (3.2)$$

wobei:

$$\alpha = 3.9083\text{e-}3$$

$$\beta = -5.7750\text{e-}7$$

$$R_0 = 1\text{ k}\Omega$$

R = temperaturabhängiger Widerstand

3.4.5 Layout

Der Slaveprint setzt sich aus sechs Feldern fürs Balancing (analog), sowie dem Mittelteil mit den Mikrocontrollern und den Sensoren (digital) zusammen. Die Teile wurden räumlich getrennt. Um die Signalleitungen entsprechend Anforderungen des Datenblattes kurz zu halten, befindet sich der Digitalteil in der Mitte des Prints.

Die benötigten Verbindungen lassen sich grob kategorisieren: In der ersten Gruppe sind Leiterbahnen, welche die Balancingströme führen. Diese Leiterbahnen müssen breit genug für mindestens 4 A sein. Bei einer maximalen Kupfererwärmung von 10 K führt dies zu einer Leiterbahnbreite von 2 mm. Ausserdem können von diesen Leiterbahnen Störungen kommen. Deshalb liegt der Layer mit dem Balancing zu oberst und unterhalb wird die Massefläche als Abschirmung geführt. Die zweite Gruppe ist die Masse. Die Anforderung an das PCB sind viele und genaue Messungen. Genaue Messungen benötigen ein genaues, überall gleiches Bezugspotential. Dies kann durch die Massefläche erreicht werden. Wichtig ist, dass die Massefläche nicht mit Leiterbahnen durchzogen ist. Die Massefläche befindet sich auf dem zweiten Layer. Die dritte Gruppe sind die Signalleiterbahnen. Die Signale haben kleine Leistungen und deshalb nur eine Leiterbahnbreite von 0.2 mm. Die Leiterbahnen wurden ohne Schlaufen an die Bauteile geführt. Die Massefläche besitzt einen Ausschnitt um die isoSPI-Schnittstelle, um Einstreuungen der schnellen Kommunikation auf den Rest der Schaltung zu verhindern. Der unterste Layer wurde verwendet, um Brücken zu realisieren.

Da jeder Analogblock dieselbe Schaltung zum Balancen hat, ist die Bauteilanordnung identisch. Das Datenblatt weist darauf hin, dass der Primärbereich und der Sekundärbereich mit der zusätzlichen Beschaltung so nahe wie möglich beieinander liegen muss. Die FET-Transistoren werden mithilfe von Kühlflächen und Vias gekühlt. Dabei wurde darauf geachtet, dass die Kupferflächen grösser als 1.5 cm^2 sind. Bei den Shuntwiderständen wurde zwecks niederohmiger Verbindung zur Masse ebenfalls eine Kupferfläche mit Vias eingefügt. Die Stützkondensatoren wurden eingangsseitig beziehungsweise ausgangsseitig platziert, um die Spitzen des Balancings gegenüber den Klemmen und dem Rest des PCBs zu filtern. Da die Shuntwiderstände nur $5 \text{ m}\Omega$ sind, muss der Abgriff für die Messungen ohne zusätzliche Leiterbahn erfolgen.

Um keinen Schwingkreis zu bilden, sind die Gateansteuerungsleitungen so kurz wie möglich gehalten worden. Die Stützkondensatoren C14 bis C20 (siehe Schema im Anhang) sind entsprechend Datenblatt nahe am LTC3300 platziert. Dies verbessert die Genauigkeit der Messung. Auch die Filter für die Strommessung befinden sich dort. Die Filter für die Spannungsmessung (siehe Kapitel 3.4.2) befinden sich in unmittelbarer Umgebung des LTC8613, um allfällige Fehler, welche sich auf der Leiterbahn summieren, herauszufiltern.

Entsprechend Empfehlungen des Herstellers wurde außerdem darauf geachtet, dass die Groundpins des LTC3300 und LTC6813 direkt auf Masse geführt werden und nicht zuerst auf das Exposed Pad. Dieses Pad ist außerdem so niederohmig wie möglich an Masse angeschlossen, indem eine Fläche und viele Vias unter die Mikrocontroller platziert wurden. Die für die Speisung des Digitalteils verantwortliche Leiterbahn hat eine Breite von 0.6 mm. Dies würde reichen, um über ein Ampere zu leiten. Die Erwärmung der Leiterbahn beträgt unter diesen Umständen maximal 10 °C.

4 Software

Die Software wurde zum grössten Teil in Python geschrieben. Dies ist für die Entwicklungsphase extrem praktisch, da Änderungen sehr schnell umgesetzt werden können. Die Nachteile liegen allerdings in der Laufzeit. Dies liegt zum Teil auch am eher langsamem SoC BCM2835, welcher auf dem RPi Zero W verbaut ist. Der IC LTC6813 muss vor dem Senden eines Befehls aufgeweckt werden. Nach dem Aufwecken muss der Befehl innerhalb von 4.3 ms über die isoSPI-Schnittstelle gesendet werden. In der Praxis zeigte sich, dass diese Zeit mit dem Python Script nur selten eingehalten werden konnte. Um die Kommunikation verlässlich zu machen, wird aus diesem Grund ein C++ Programm verwendet. Nachfolgend wird das Hauptprogramm, welches den Ausgleich-Algorithmus implementiert, die Klassen und das C++ Programm näher erläutert.

4.1 Hauptprogramm

Das Hauptprogramm besteht aus einer Endlosschleife, welche den Ausgleich-Algorithmus implementiert. Des Weiteren werden die gemessenen Daten an eine MySQL-Datenbank gesendet, damit sie vom Web-App (siehe Kapitel 5) abgerufen und dargestellt werden können. Das Hauptprogramm importiert die BMS-Klasse (siehe Kapitel 4.2.1), sowie die folgenden Module:

- multiprocessing - Funktionen parallel ausführen
- time - Die aktuelle Uhrzeit (Unix-Timestamp)
- _mysql - MySQL Schnittstelle

Der Ausgleich-Algorithmus ist wie folgt aufgebaut:

1. Umgebungstemperatur messen
2. Temperaturüberwachung (mittels Temperaturschalter)
3. Ausgleich pausieren (Ausgleichsströme wurden Messfehler verursachen)
4. Spannungen messen
5. Ausgleich fortsetzen
6. Sofern alle Messungen innerhalb der Toleranzen sind:
 - Ausgleichbefehl bestimmen
 - Ausgleichbefehl ausführen
7. Ansonsten:
 - Ausgleich stoppen
 - Sunny Boy stoppen

Der Ausgleichbefehl wird bestimmt, indem zuerst der Mittelwert der Spannungen ermittelt wird. Dann wird mit einem Schwellwert bestimmt, was mit einem Block geschehen soll. Blöcke, welche über diesem Bereich liegen, werden durch das Ausgleichen entladen. Blöcke, welche unter diesem Bereich liegen, werden durch das Ausgleichen geladen.

Da es in der Praxis lange (bis zu 2 s) dauert, bis die gemessenen Daten an die MySQL-Datenbank übermittelt wurden, wird dies in einem separaten Prozess parallel ausgeführt. Somit wird der Ausgleich-Algorithmus davon geringer beeinflusst.

4.2 Klassen

Es werden insgesamt drei eigens geschriebene Klassen verwendet. Die erste modelliert das BMS, die zweite die isoSPI-Schnittstelle und die dritte den Sunny Boy.

4.2.1 BMS

Die BMS-Klasse enthält viele Klassenattribute, welche als Definitionen der vielen Befehle des LTC6813 dienen. Des Weiteren sind Methoden vorhanden, um die in Kapitel 4.1 beschriebenen Messungen durchzuführen. Die BMS-Klasse verfügt auch über einige Instanzattribute, um die gemessenen Werte zu speichern. Des Weiteren verfügt sie über ein Objekt der Klasse isoSPI und ein Objekt der Klasse SunnyBoy.

4.2.2 isoSPI

Die isoSPI-Klasse modelliert die Schnittstelle zwischen dem Master und den Slaves. Es sind Methoden enthalten, um Befehle zu senden und Daten zu empfangen. Des Weiteren ist eine Methode enthalten, um eine SPI-Transaktion zwischen dem LTC6813 (Spannungsmessungen) und dem LTC3300 (Ausgleichen) durchzuführen. Die effektiven Befehle und Daten werden allerdings nicht von dieser Klasse gesendet (aufgrund der langsamen Laufzeit). Diese werden als Argumente an das C++ Programm übergeben (siehe Kapitel 4.3). Weiter verfügt die Klasse über ein Instanzattribut, welches die Nummer der aktuell funktionierenden isoSPI-Schnittstelle (1 oder 2) speichert. Bei einer Fehlübertragung, erkannt durch eine ungültige Checksumme, wird dieses Attribut umgeschaltet.

4.2.3 SunnyBoy

Die Klasse SunnyBoy modelliert den Sunny Boy Storage 2.5. Diese Klasse dient nur als Proof of Concept und kommuniziert nicht wirklich mit dem Sunny Boy. Sie verfügt über den Zustand (gestartet oder gestoppt). Des Weiteren könnte der aktuelle Strom angefordert werden.

4.3 C++ Programm (isoSPI-Interface)

Das C++ Programm verwendet die Bibliothek bcm2835 von Mike McCauley. Mit dieser Bibliothek lässt sich einfach eine SPI-Transaktion auf dem RPi durchführen. Das Kommandozeilenprogramm liest die übermittelten Argumente vom Standardeingabe ein. Danach wird die angeforderte SPI-Transaktion mit dem übermittelten Befehl und den übermittelten Daten durchgeführt. Danach wird das Resultat auf der Standardausgabe ausgegeben, sodass dieses vom Python Script wieder eingelesen und geparsed werden kann.

5 Web-App

Das Web-App dient dazu, die gemessenen Daten für den Benutzer darzustellen (siehe Abbildung 5.1). Es wird auf einem gemieteten vServer mit dem Betriebssystem Ubuntu Server 18.04.1 LTS gehostet. Dieser verfügt über ein LAMP Softwarepacket, also über einen apache2-Webserver, eine MySQL-Datenbank und den PHP-Interpreter. Das Web-App verwendet AJAX (Asynchronous JavaScript And XML) und JavaScript, um die Daten zu aktualisieren, ohne die Website komplett neu zu laden. Damit die Daten möglichst ohne Verzögerung angezeigt werden, wird die MySQL-Datenbank alle 100 ms auf Änderungen überprüft. Falls die Daten für 15 s nicht mehr aktualisiert wurden, geht das Web-App davon aus, dass die Verbindung zum BMS unterbrochen wurde. Die Kommunikation des Web-Apps ist dabei stets einseitig. Dies bedeutet, dass das BMS die gemessenen Daten in die MySQL-Datenbank schreibt, aber nichts zurückliest.

The BMS - Online

Balancing: Aktiv

Umgebungstemperatur: 30.1 °C

Spannungen:

Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
3.524 V	3.562 V	3.628 V	4.008 V	3.003 V	3.548 V

Temperaturüberwachung:

Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
OK	OK	OK	OK	OK	OK

Abbildung 5.1: Web-App

6 Qualitätssicherung

Die praktische Validierung ist schlussendlich die Bestätigung der Funktionalität eines Produktes. Somit können auch die gesetzten Ziele optimal überprüft werden. Die Qualitätssicherung wurde in verschiedene Bereiche aufgeteilt und geprüft. Die Prüfanweisung im Anhang gibt Aufschluss über das genaue Vorgehen und die Anleitung zur Inbetriebnahme des BMS. Die Nachvollziehung der Ergebnisse ist darin ebenfalls ersichtlich.

Der erste Bereich beinhaltet die Bewertung des Balancing. Hierbei ist wichtig zu erwähnen, dass ein aktives Balancing vollführt wird. Mittels dem dafür programmierten Algorithmus wird die geeignete Lade- und Entladearakteristik eines jeweiligen Blocks gewährleistet. Somit können Ladungen von benachbarten Blöcken ausgetauscht werden. Der Transfer der Ladung vom höchsten zum niedrigsten Block im Stack ist durchführbar. Die Möglichkeit, Ladung aus einem in mehrere Blöcke oder Ladung aus mehreren Blöcken in einen zu transferieren besteht ebenfalls. Die Effizienz des Balancings erreichte dabei Spitzenwerte von 87%.

Die berechneten Balancingströme von 4.3 A wurden nicht erreicht. Die maximalen Werte der Balancingströme liegen bei 1.8 A. Die genaue Ursache konnte jedoch nicht festgestellt werden.

Alle sicherheitstechnischen Massnahmen sind gewährleistet. Das BMS ist in der Lage, benutzerrunabhängig Gefahren zu erkennen und den Lade-, bzw. Entladevorgang rechtzeitig zu beenden. Sollte ein Block Über- oder Unterspannung aufweisen, wird dies erkannt und der Balancing-Vorgang gestoppt. Dies geschieht ebenfalls bei Umgebungstemperaturen im Bereich über 45 °C oder unter 0 °C. Der dafür zuständige PT1000 wurde dabei mittels Wärmequelle über den genannten Wert erwärmt und das Stoppen des Balancings beobachtet. Genau gleich wurde dies mit Kältespray für Temperaturen unter dem genannten Wert erfolgreich getestet. Sollte die Betriebstemperatur der Zellen eines Blockes höher als erlaubt sein, wird das Balancing durch das Auslösen der Temperaturschalter ebenfalls gestoppt.

Eine bidirektionale Kommunikation ist gegeben. Sollte eine Kommunikationsleitung vom Master zum Slave unterbrochen sein, ist die Verbindung durch eine zweite Kommunikationsleitung gewährleistet. Somit kann die Ladung weiterhin ausgeglichen werden. Sollten ausversehen die Kommunikationsleitungen vertauscht angeschlossen werden, wird dies erkannt und der Betrieb kann trotzdem fortgeführt werden. Ein Defekt durch Verpolung wurde somit eliminiert. Getestet wurde dies durch simples Ausstecken und Vertauschen der jeweiligen Kommunikationsleitungen.

Durch die isolierte SPI Schnittstelle sind gegebene Potentialunterschiede, welche die Prints aufweisen werden, keine Beeinträchtigung.

Mittels Webapp kann der Status des Balancing-Vorgangs jederzeit auf der ganzen Welt beobachtet werden. Die vom Slave an den Master übermittelte Telemetriedaten werden per WLAN an eine Internetdomain geschickt. Somit kann der Benutzer via Laptop oder Mobiltelefon auf die Telemetriedaten zugreifen. Dies funktioniert zuverlässig und kann bei aktiver Internetverbindung jederzeit genutzt werden.

7 Schluss

Im Rahmen dieser Arbeit wurde «TheBMS» entwickelt und programmiert. Dieses Produkt dient zur Gewährleistung einer korrekten Ladung und Entladung, sowie einem sicheren Betrieb von rezyklierten Lithiumionen Akkumulatoren. Im Folgenden werden die im Projekt erreichten Ziele, die aufgetretenen Schwierigkeiten sowie einige Weiterentwicklungsmöglichkeiten erläutert.

Eine erste Schwierigkeit bestand darin, die Kommunikation zwischen dem Master- und Slaveprint herzustellen. Anstelle einer gültigen Checksumme gab der IC auf dem Slaveprint durchgehend eine Fehlermeldung zurück. Der Grund dieser Problematik wurde dann im Bereich der Software lokalisiert. Die ersten Befehle wurden in Python geschrieben. Dies erwies sich als gravierender Fehler, da der IC nur Signale bis zu einer Länge von 4.3 ms empfängt. Um eine passende Signallänge zu erreichen, wurde der Code nun in der Programmiersprache C++ geschrieben.

Die Berechnung der Umgebungstemperatur schien auf den ersten Blick eine kurze Angelegenheit zu sein. Dafür mussten die Widerstandswerte des PT1000 bestimmt und anschliessend anhand dieses Wertes die dazugehörige Temperatur berechnet werden. Die daraus berechnete Temperatur sollte dann über das UI ausgegeben werden. Beim Testen wurde jedoch bemerkt, dass die ausgegebene Temperatur zufällige Werte im Bereich von 10 °C bis 20 °C annahm und stetig wechselte. Bei der Fehlersuche wurde bemerkt, dass der eingeplante Filterkondensator zu gross dimensioniert war. Durch die falsche Dimensionierung fiel die Ladekurve des Kondensators zu lange aus und somit wurden unterschiedliche Spannungsabfälle über dem PT1000 gemessen. Obwohl beim Platzieren des PT1000 darauf geachtet wurde, dass dieser einen möglichst grossen Abstand von verlustbehafteten Bauteilen hat, fällt die gemessene Temperatur deutlich über der Raumtemperatur aus. Es empfiehlt sich bei einer Weiterentwicklung die Temperaturmessung auf dem XCVR pHAT zu platzieren.

Die Temperaturüberwachung, welche durch zwei Taster simuliert wird, konnte auf Anhieb umgesetzt werden. Die Taster erwiesen sich als gute Idee, weil diese beim Testen oft betätigt wurden, um einen Fehler zu simulieren.

Die bidirektionale Kommunikation konnte ebenfalls wie gewünscht umgesetzt werden. So spielt es keine Rolle, ob die Verbindung IsoSPI1 oder IsoSPI2 verwendet wird. Der Balancing Algorithmus überprüft selbstständig, welche der beiden Verbindungen funktioniert. Falls der Fall eintritt, dass keine der beiden Leitungen funktioniert, wird die Funktion wieder aufgenommen, sobald die Verbindung wieder besteht.

Zum Schluss wurde leider die Zeit für das Erstellen einer Webapplikation ein wenig knapp. Auf die grafische Darstellung des Balancingvorganges musste verzichtet werden. Jedoch sind dort alle notwendigen Informationen bezüglich des Balancings ersichtlich.

Weiterentwicklungen des Produktes gibt es an vielen Ecken. Der erstellte Prototyp wurde aus Preis- und Zeitgründen nur für sechs Blöcke realisiert. Die Bauteile würden jedoch optimal ausgenutzt werden, wenn pro Slaveprint zwölf Blöcke eingeplant wären. Des Weiteren wäre es von Vorteil, wenn auf dem Print visuell erkannt werden könnte, ob gerade Ladung ausgeglichen wird, oder ob die Software einen Fehler detektiert hat. Dies könnte durch eine LED-Anzeige realisiert werden und wäre auch beim Testen der Funktionen hilfreich gewesen. Ebenfalls wurde während dem Prüfen der Bedarf eines Reset-Buttons diskutiert, welcher den Raspberry-Pi Zero W zurücksetzen könnte.

Literatur

- [1] B. Korthals, *Optimale Ladung und Entladung einzelner Zellen*, 2014. Adresse: https://www.polyscope.ch/site/assets/files/28584/ps314_26_27.pdf.
- [2] M. Schmid, *Worauf es beim Battery Balancing von Lithium-Ionen-Akkus ankommt*, 2018. Adresse: <https://www.elektronikpraxis.vogel.de/worauf-es-beim-battery-balancing-von-lithium-ionen-akkus-ankommt-a-682080/>.
- [3] M. Schmid, *Worauf es beim Battery Balancing von Lithium-Ionen-Akkus ankommt*, 2018. Adresse: <https://www.elektronikpraxis.vogel.de/worauf-es-beim-battery-balancing-von-lithium-ionen-akkus-ankommt-a-682080/index2.html>.
- [4] I. Buchmann, *How do Lithium Batteries Work?*, 2019. Adresse: https://batteryuniversity.com/learn/article/lithium_based_batteries.
- [5] E. Rahimzei, *Kompendium: Li-Ionen-Batterien*. VDE Verband der Elektrotechnik, 2015.
- [6] SMA, *SUNNY BOY STORAGE 2.5 Datenblatt*, 2017. Adresse: <https://www.sma.de/produkte/batterie-wechselrichter/sunny-boy-storage-25.html>.
- [7] Raspberry Pi Foundation, *Introducing Raspberry Pi HATs*, 2014. Adresse: <https://www.raspberrypi.org/blog/introducing-raspberry-pi-hats/>.
- [8] Linear Technology, *LTC6820 Datenblatt*, 2012. Adresse: <https://www.analog.com/media/en/technical-documentation/data-sheets/6820fb.pdf>.
- [9] Code 10, *Platinum resistance thermometers*, 2011. Adresse: http://www.code10.info/index.php?option=com_content&view=article&id=82:measuring-temperature-platinum-resistance-thermometers&catid=60:temperature&Itemid=83.

Glossar

BMS Batteriemanagementsystem. 16

GPIO General Purpose Input Output - Dabei handelt es sich um einen Kontakt, welcher als Ein- oder Ausgang konfiguriert werden kann. 5

IC Integrated Circuit (Integrierter Schaltkreis). 6, 14

pHAT Partial Hardware Attached on Top (PCB zur Erweiterung des RPi Zero). 5, 6

RPi Abkürzung für Raspberry Pi. 5, 6, 14, 15

Slave Der Slave erhält Anweisungen vom Master und führt diese aus. 5

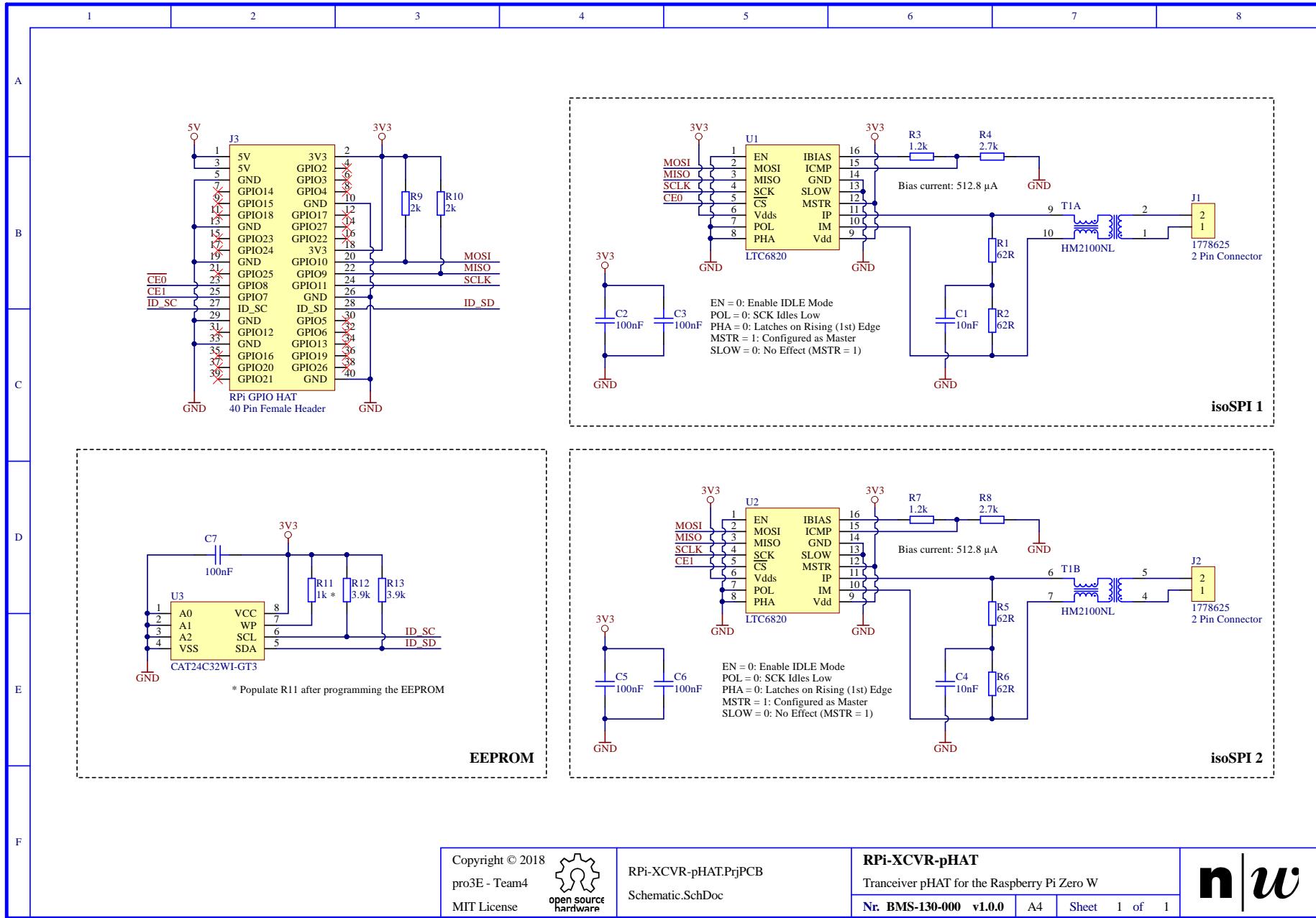
SoC System-on-a-Chip (Ein-Chip-System). 5, 14

THT Through-hole technology (Durchsteckmontage) - Dabei handelt es sich um bedrahtete elektronische Bauelemente. 6

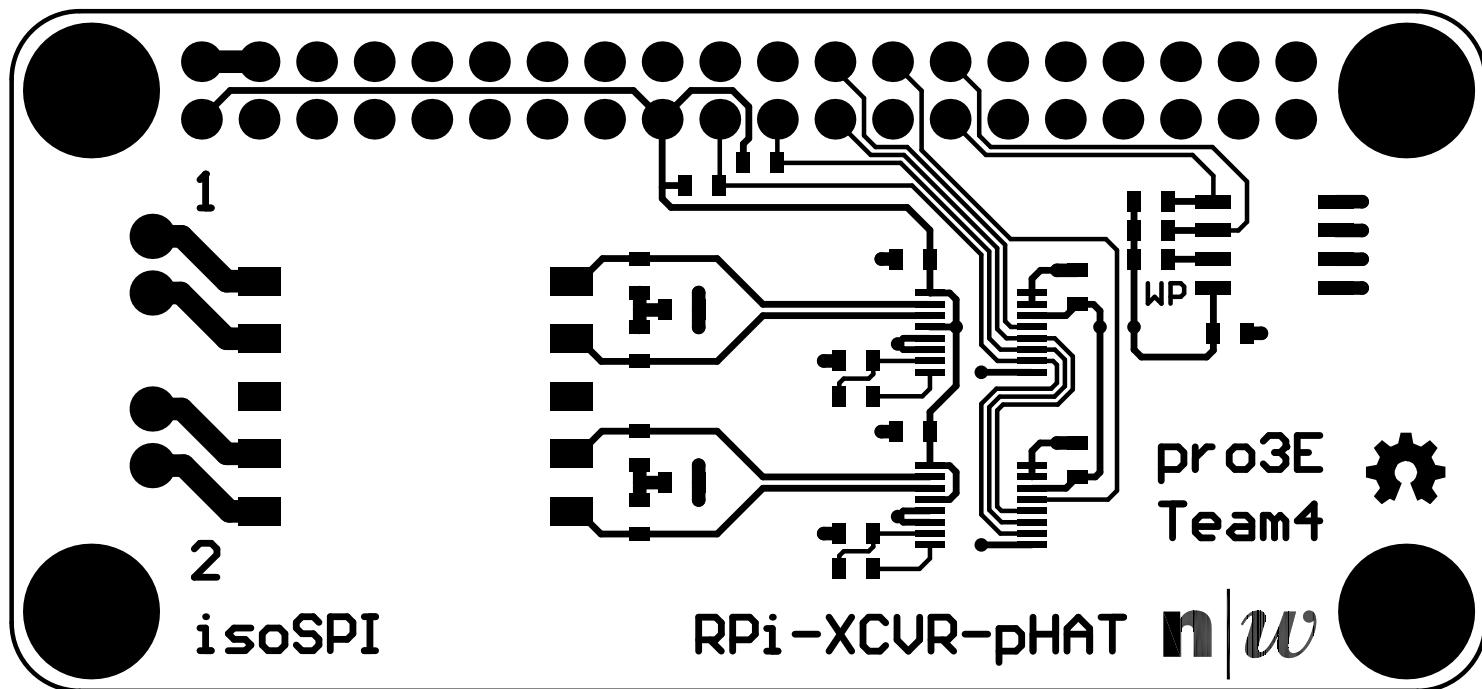
XCVR Transceiver: Sender und Empfänger. 5

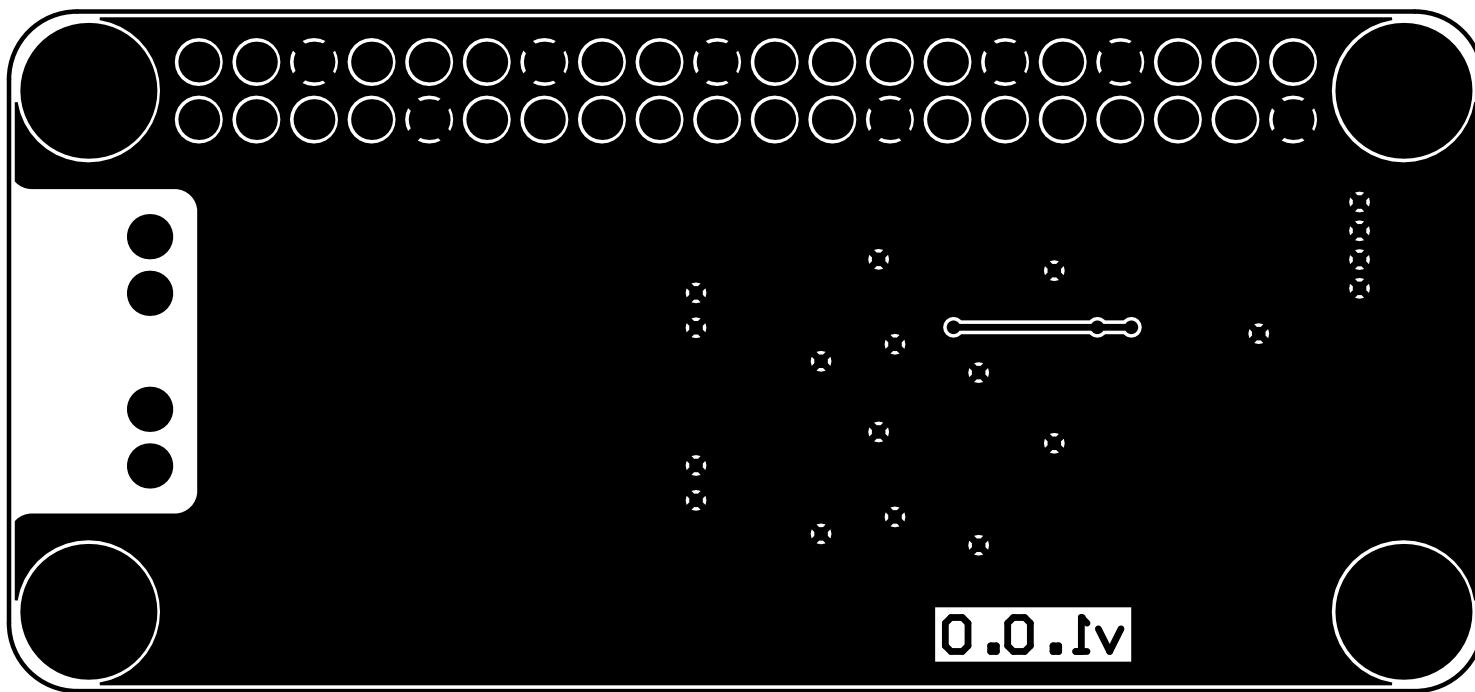
A Schema Master

A SCHEMA MASTER

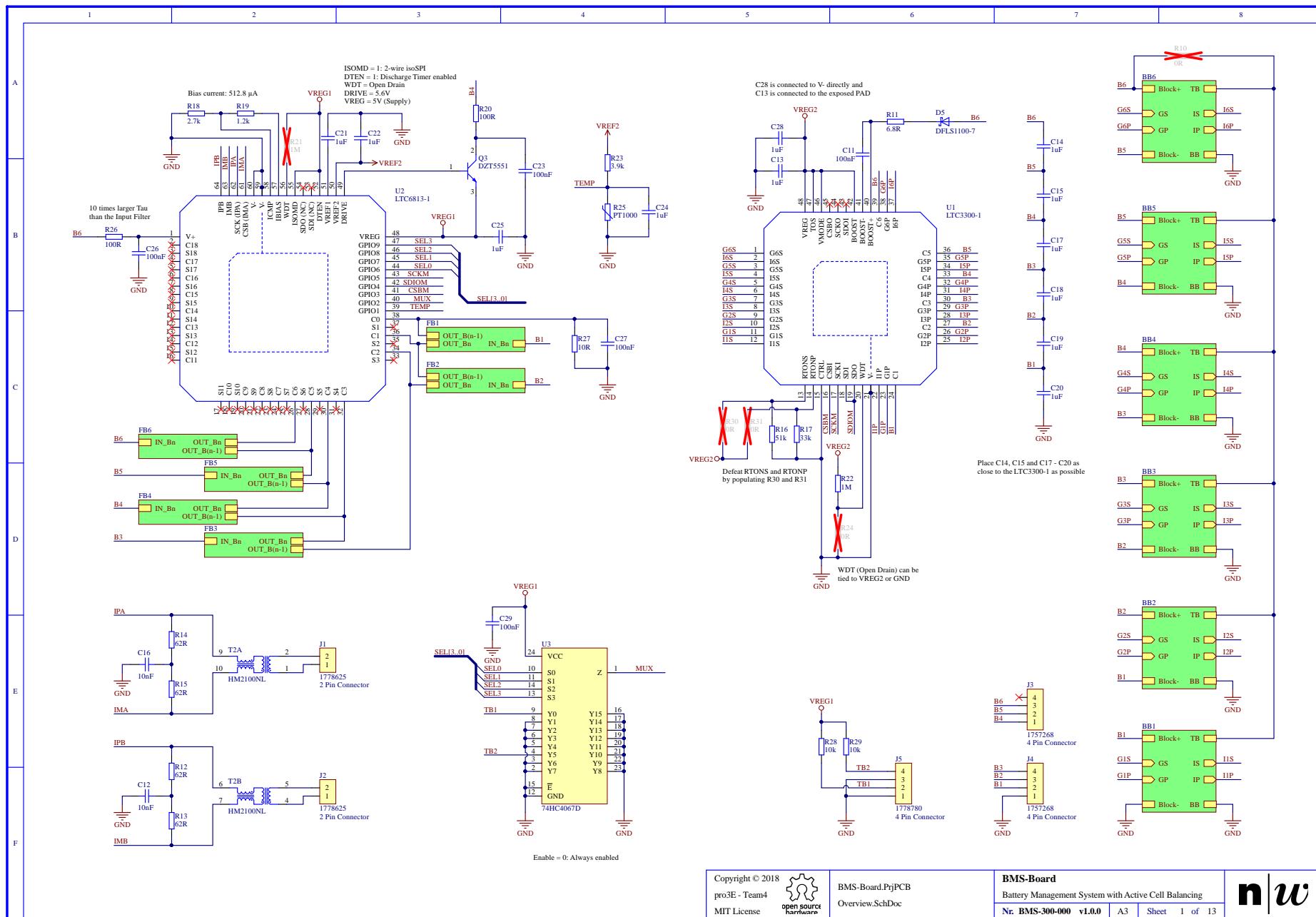


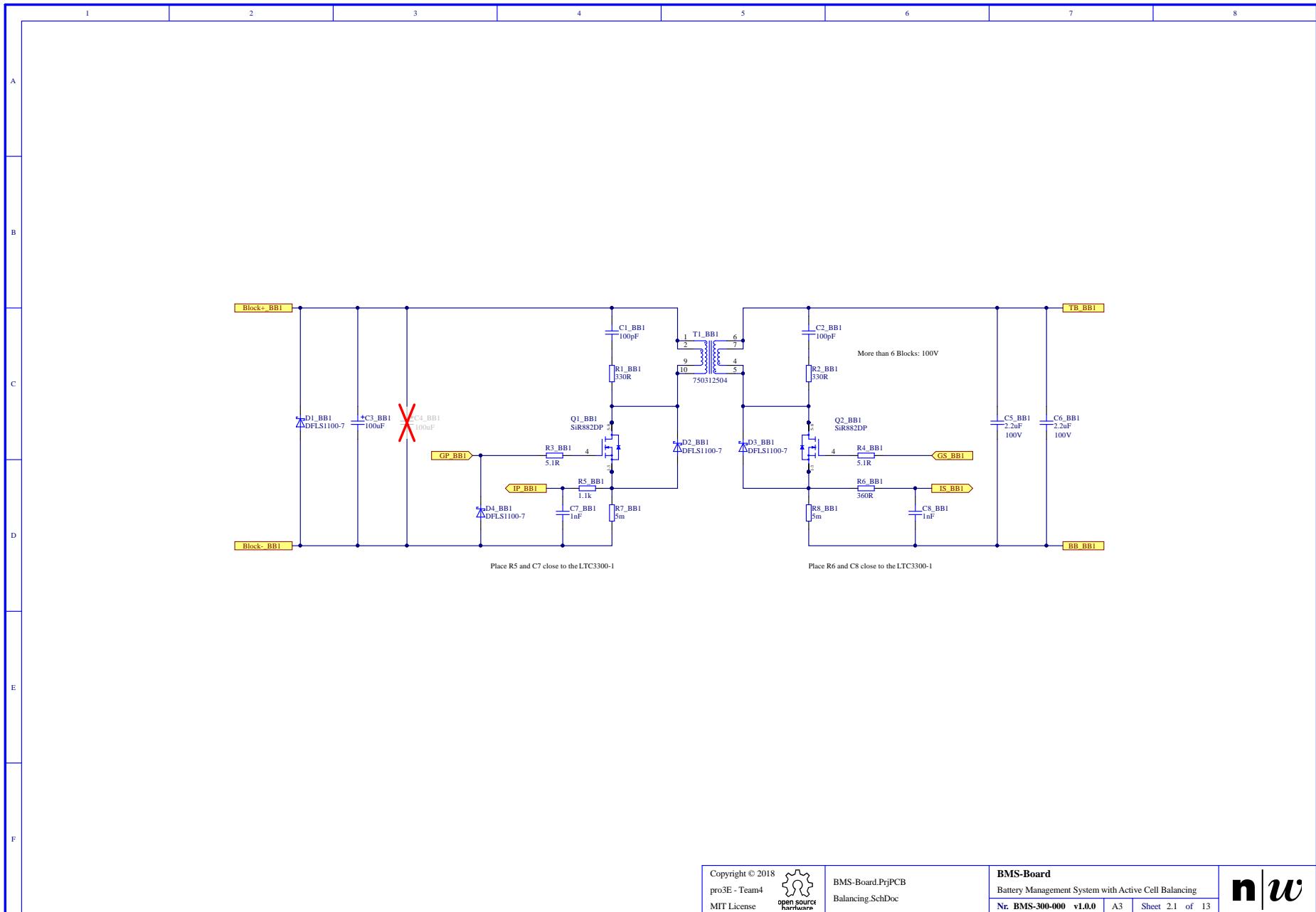
B Layout Master

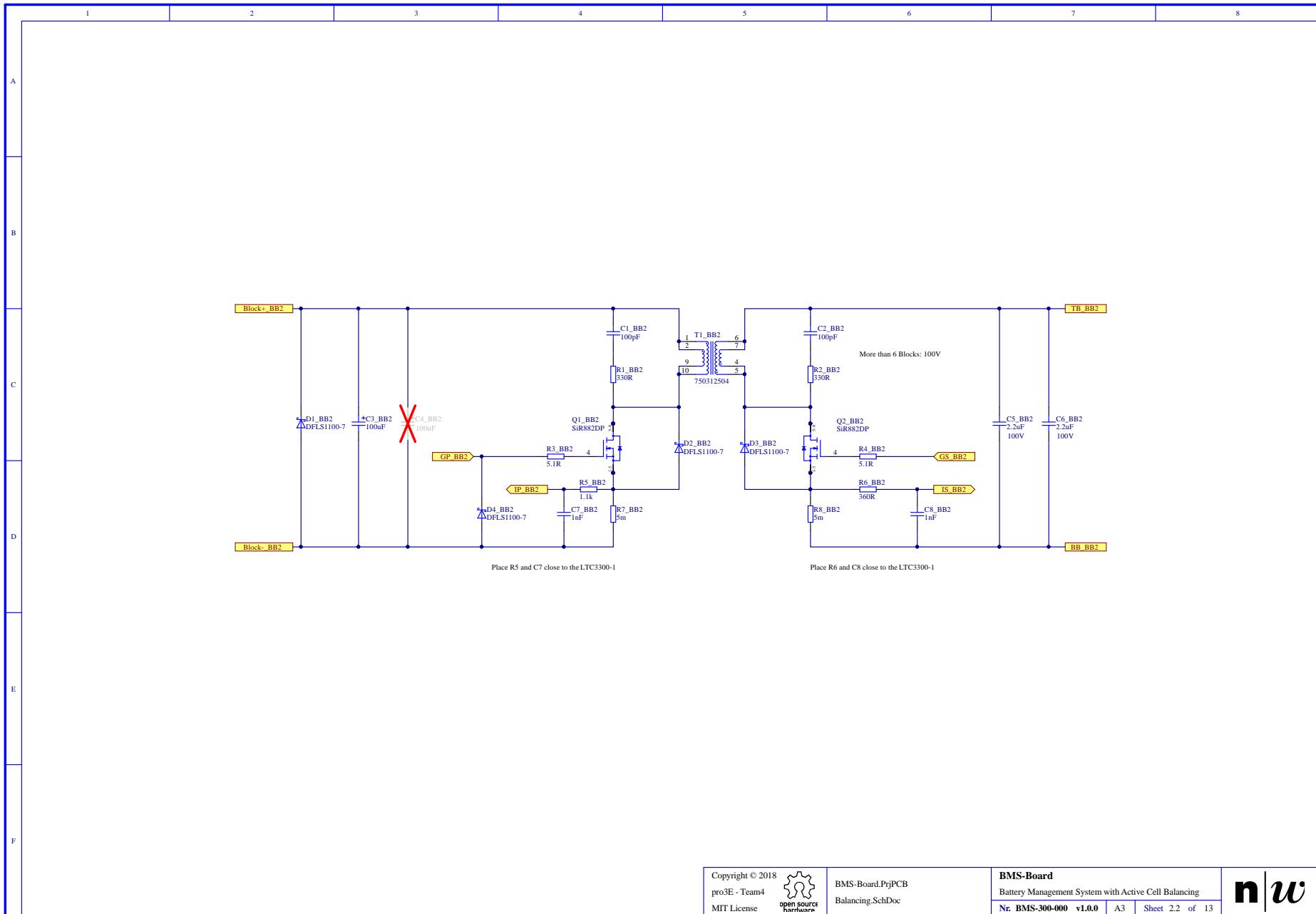


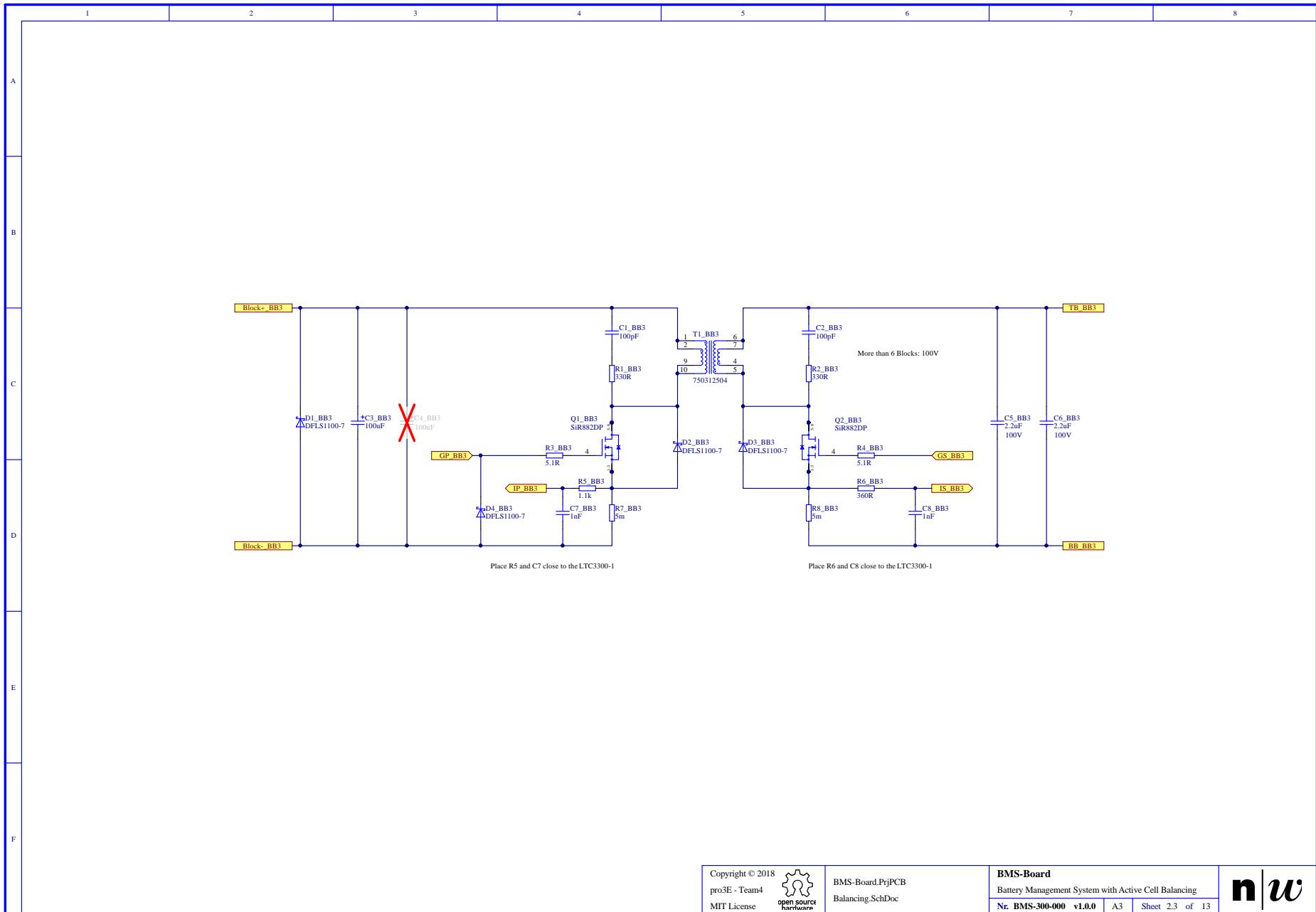


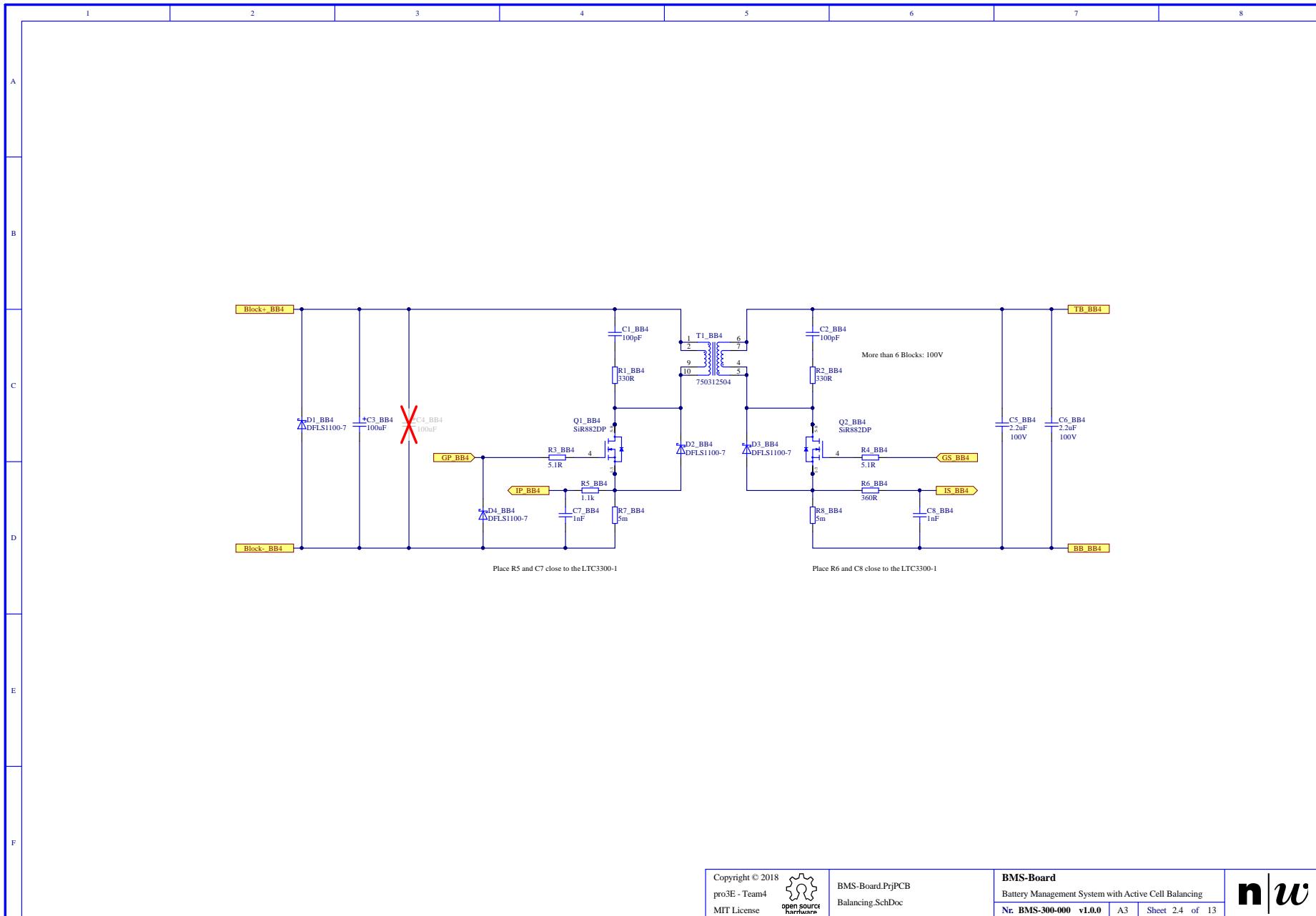
C Schema Slave

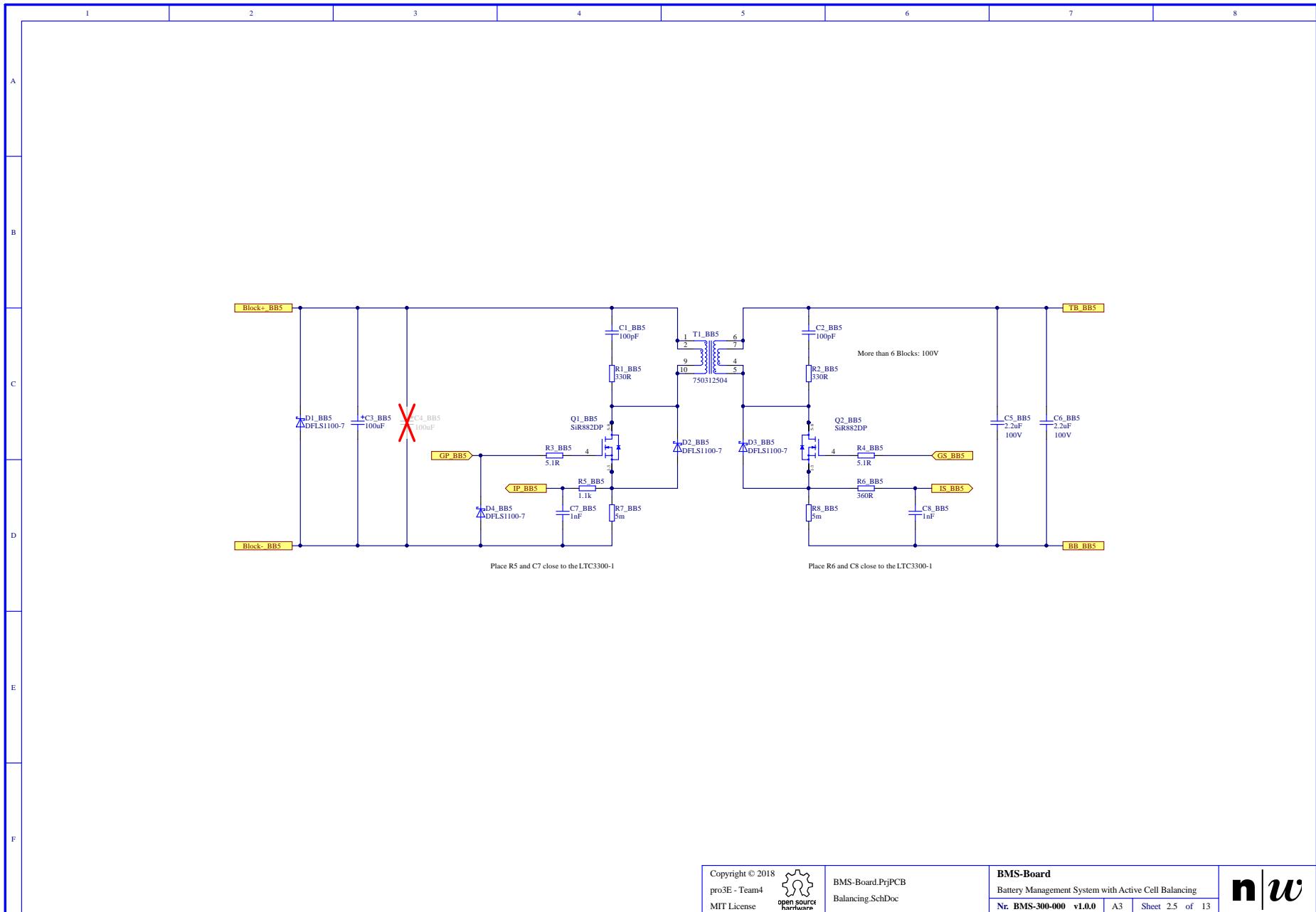


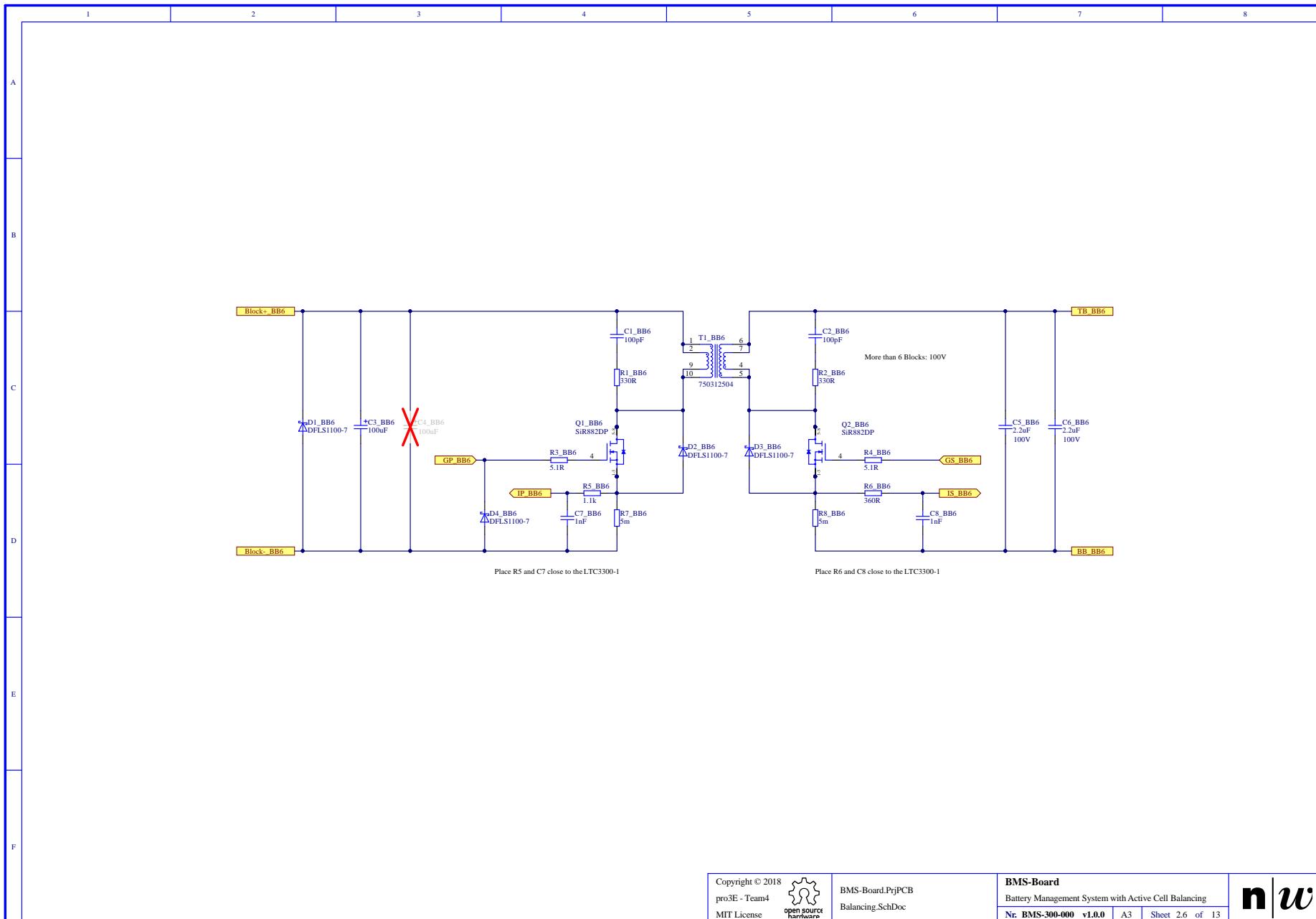


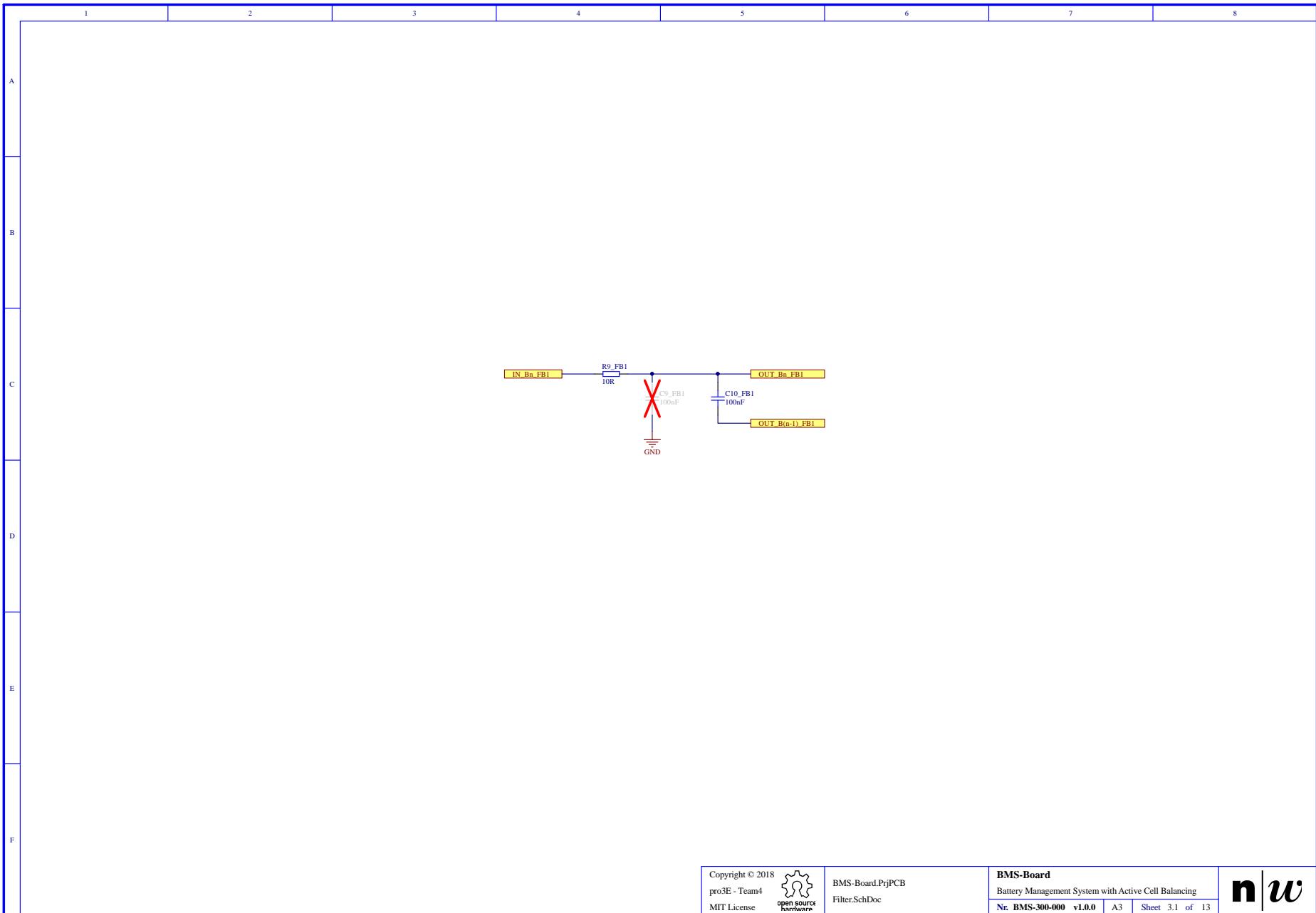


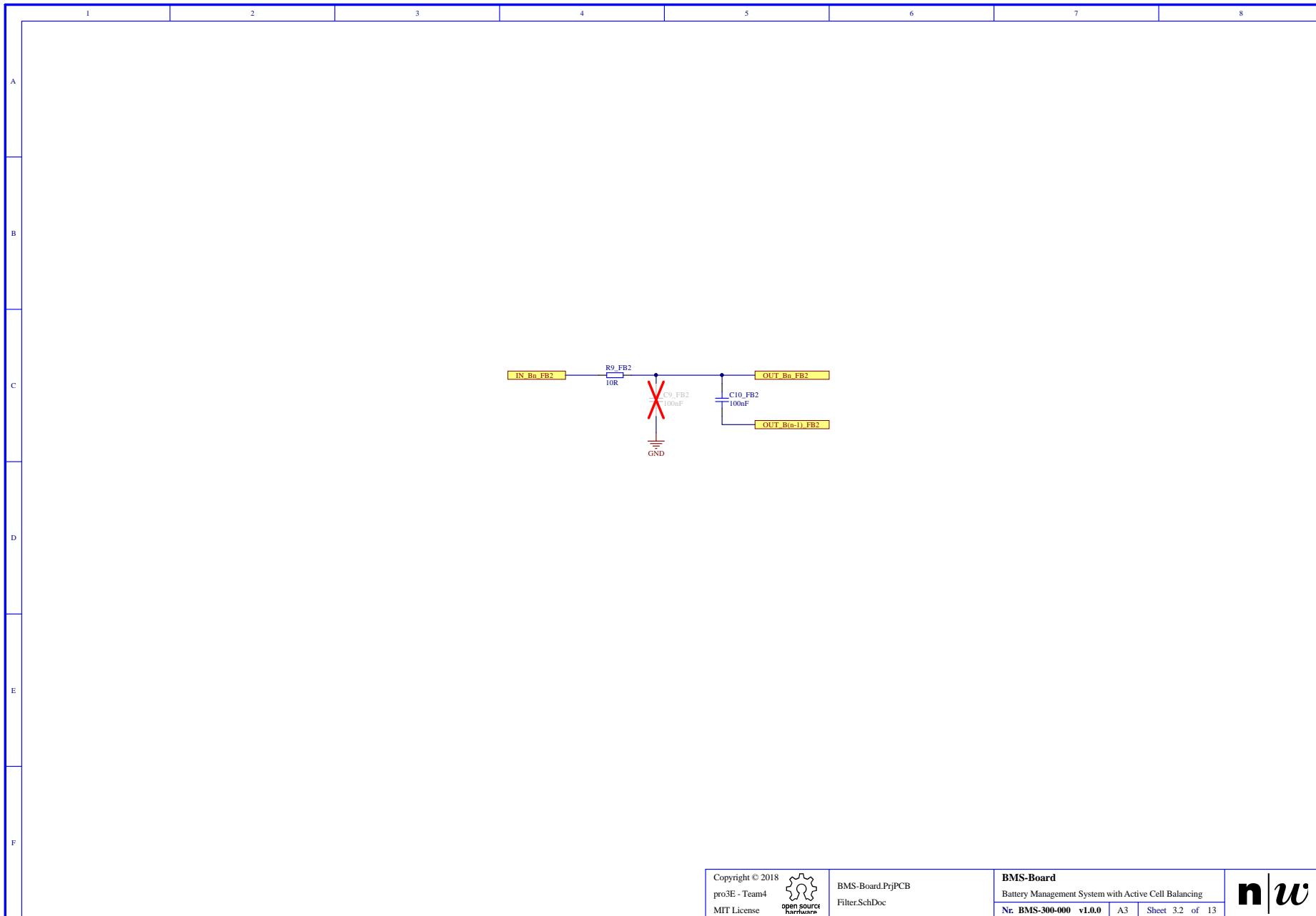


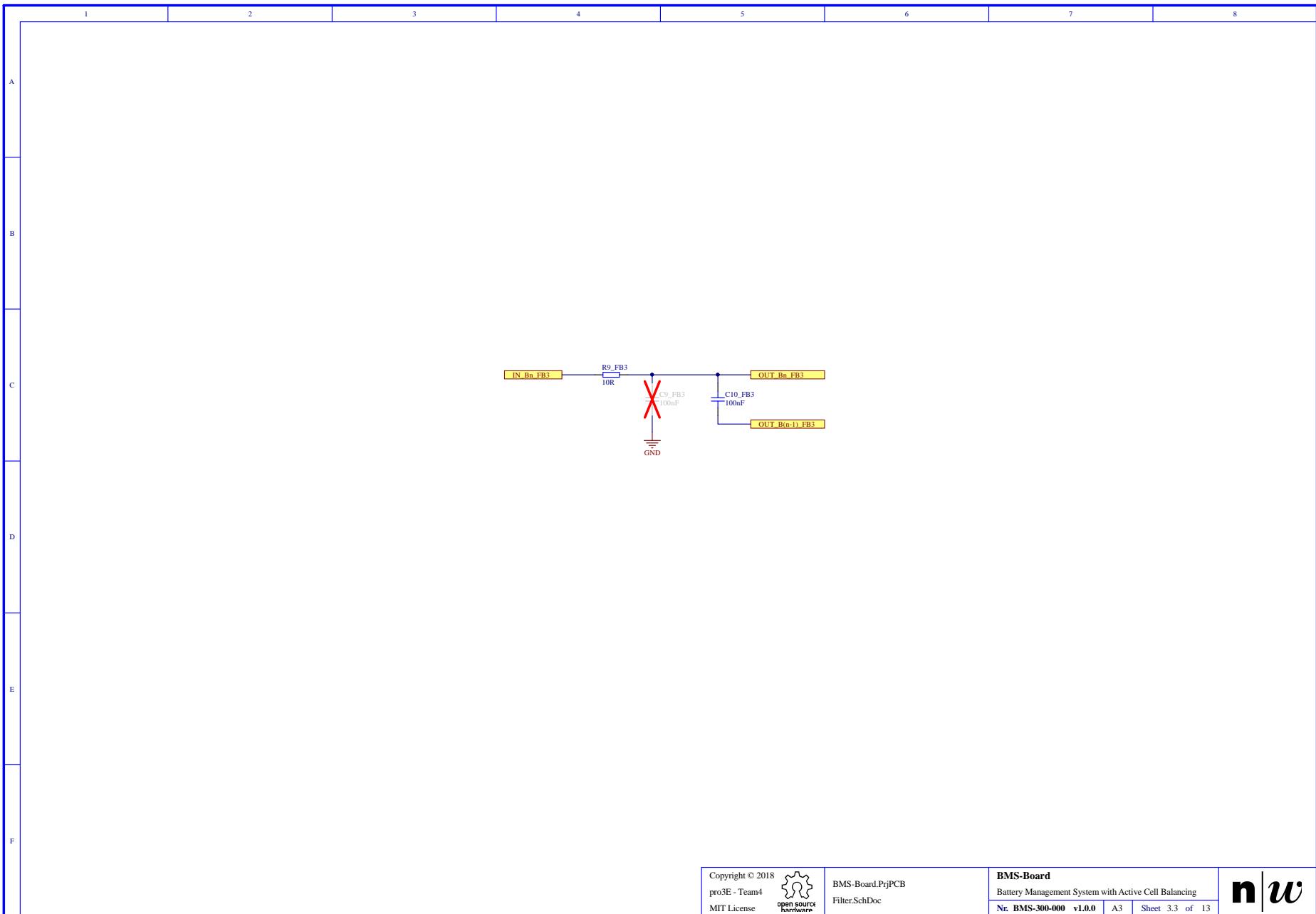


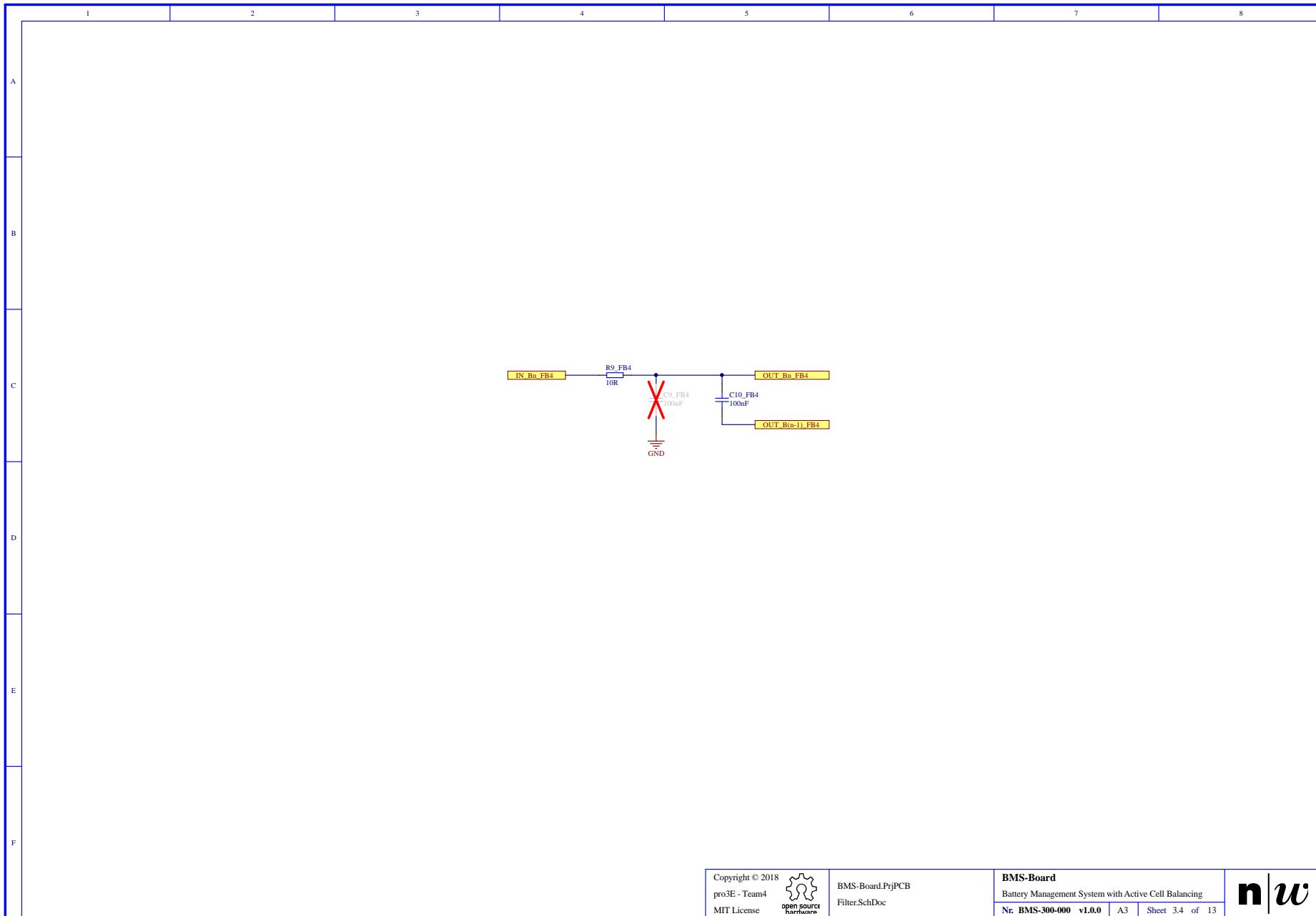


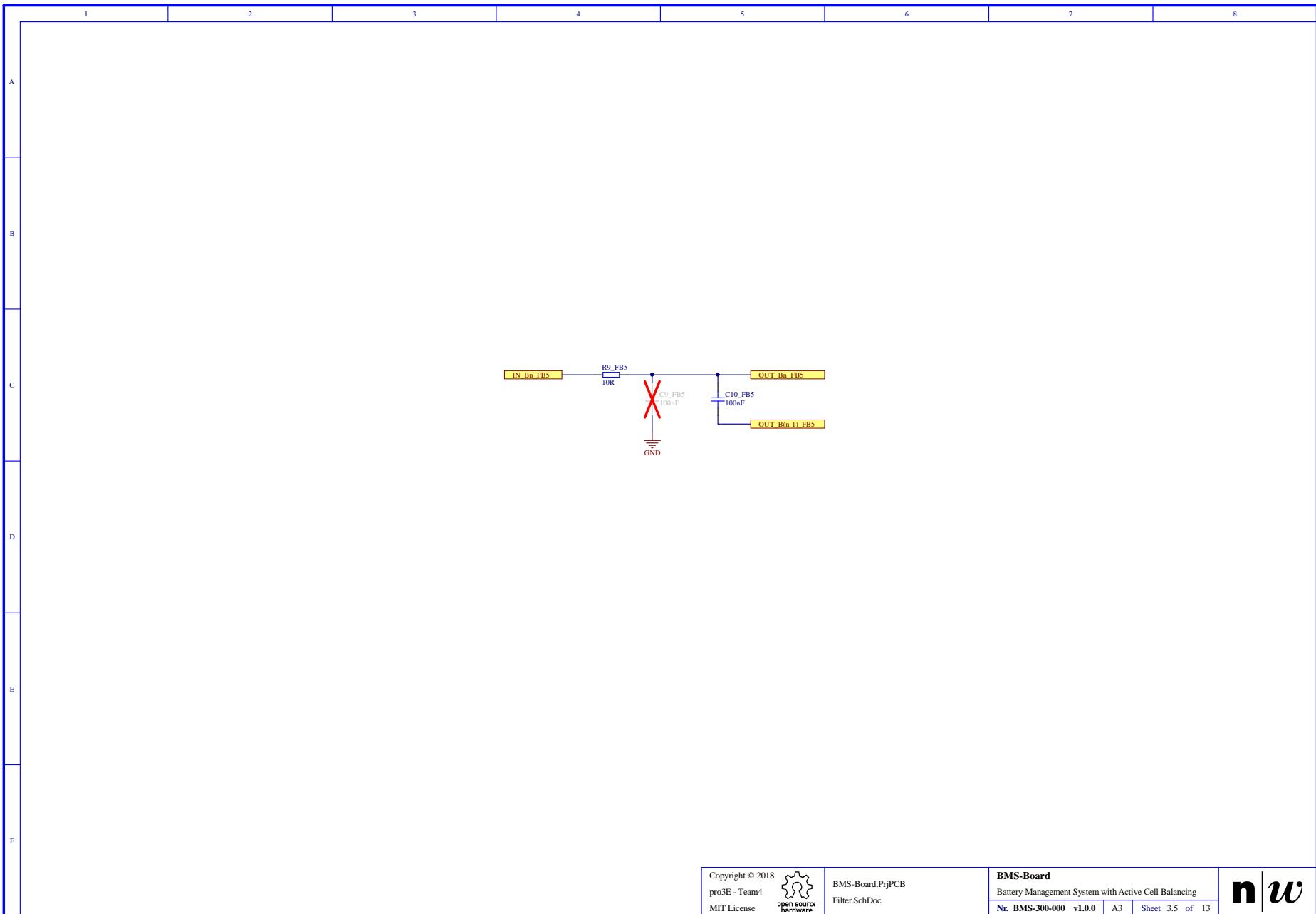


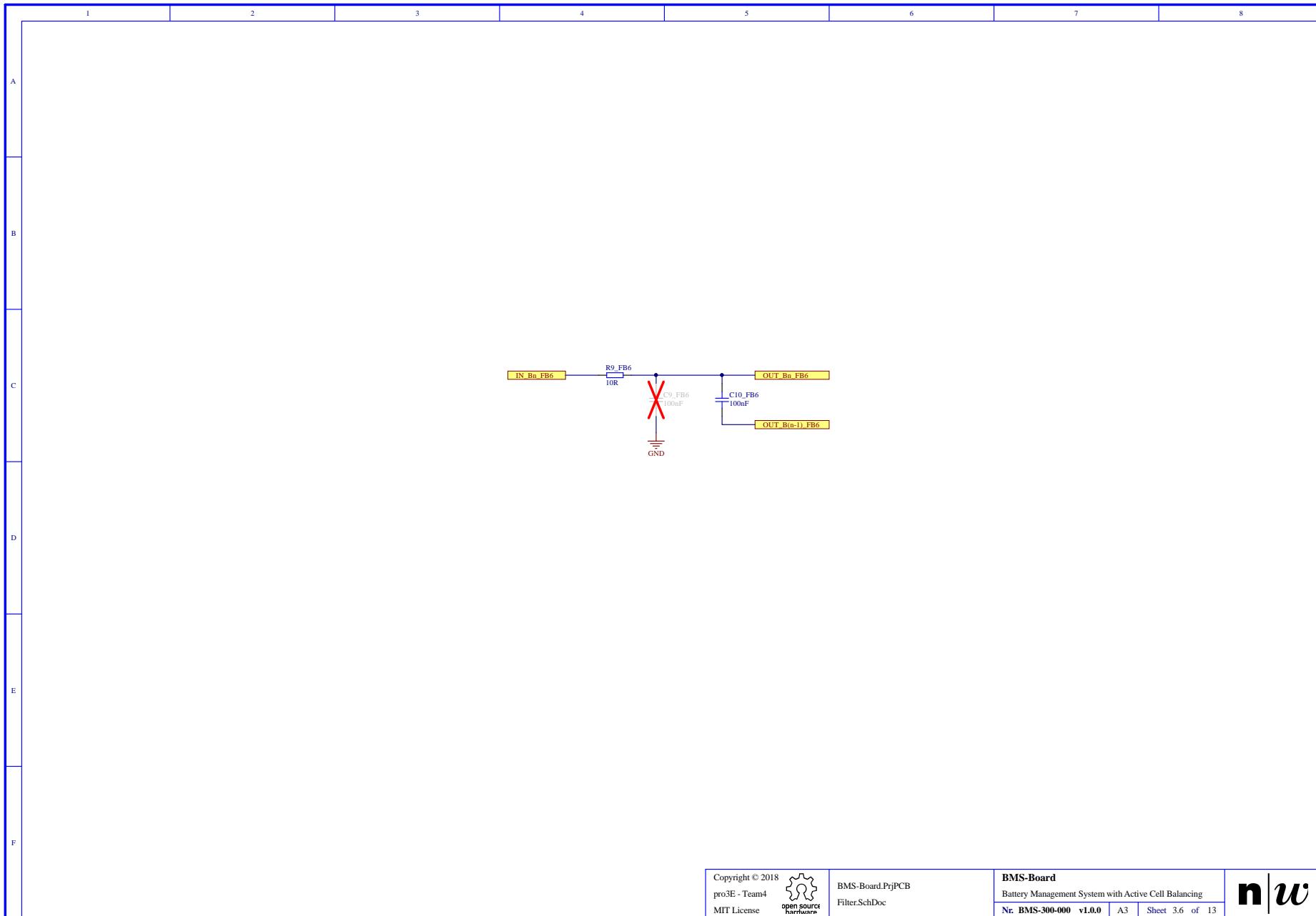


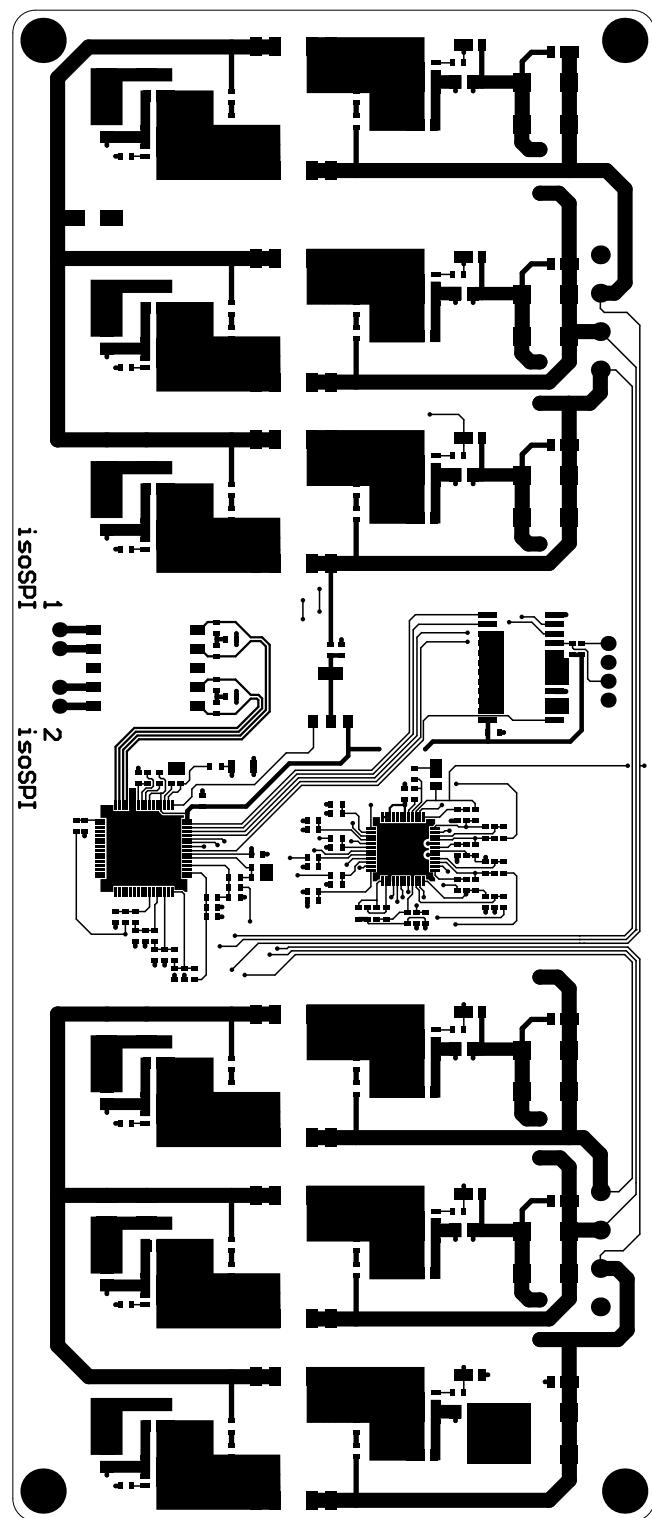


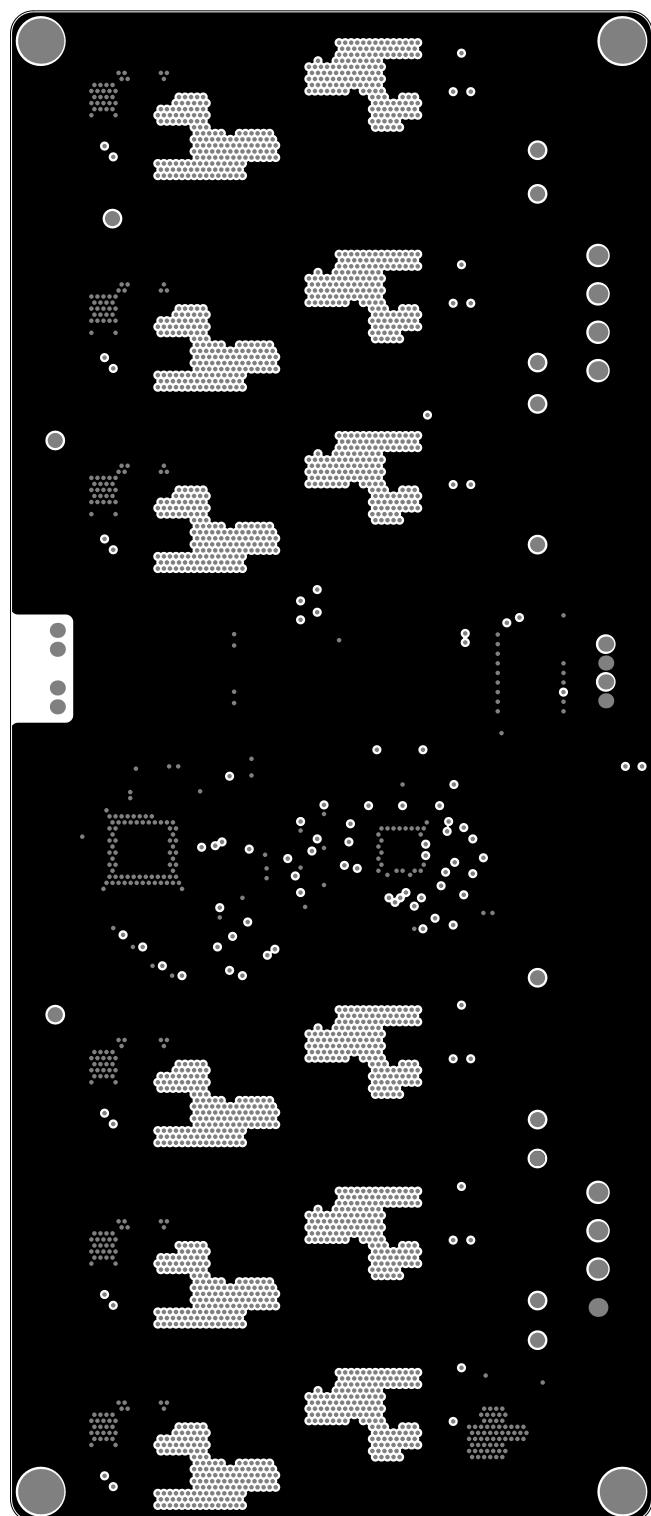


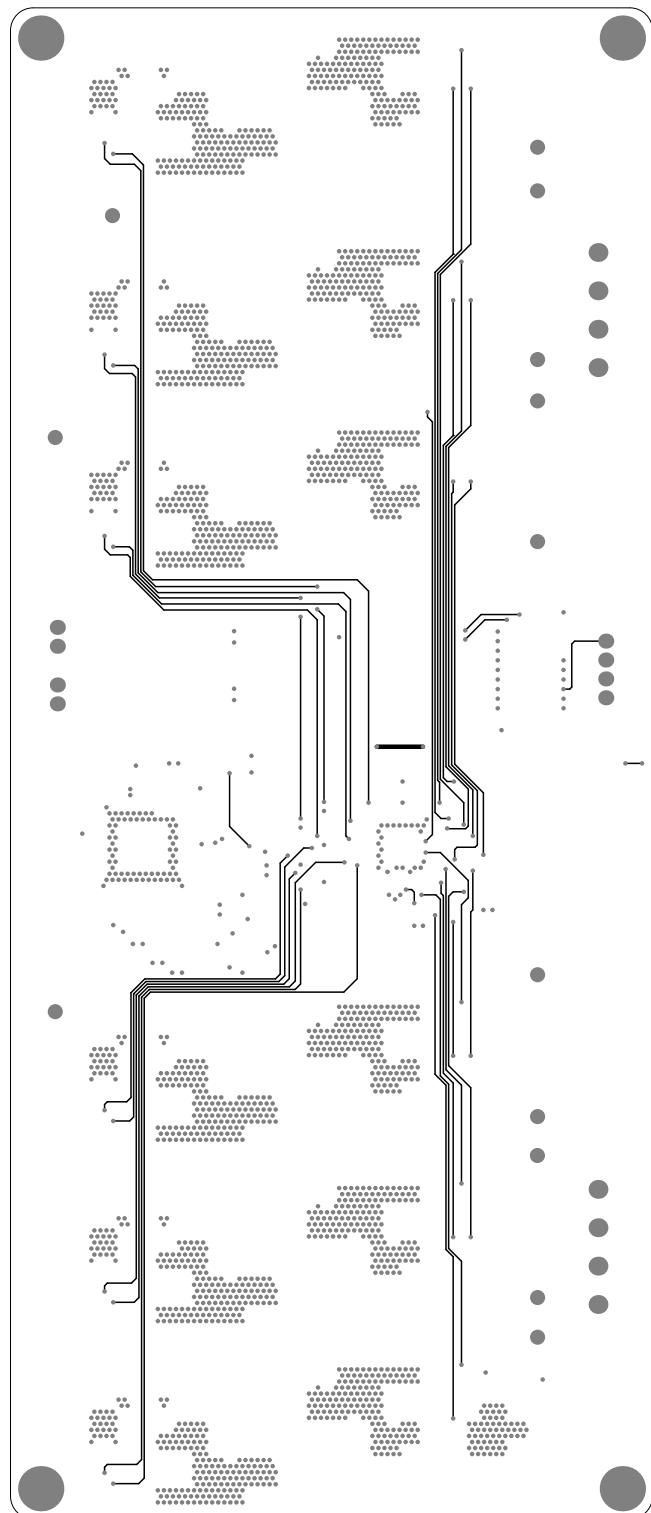


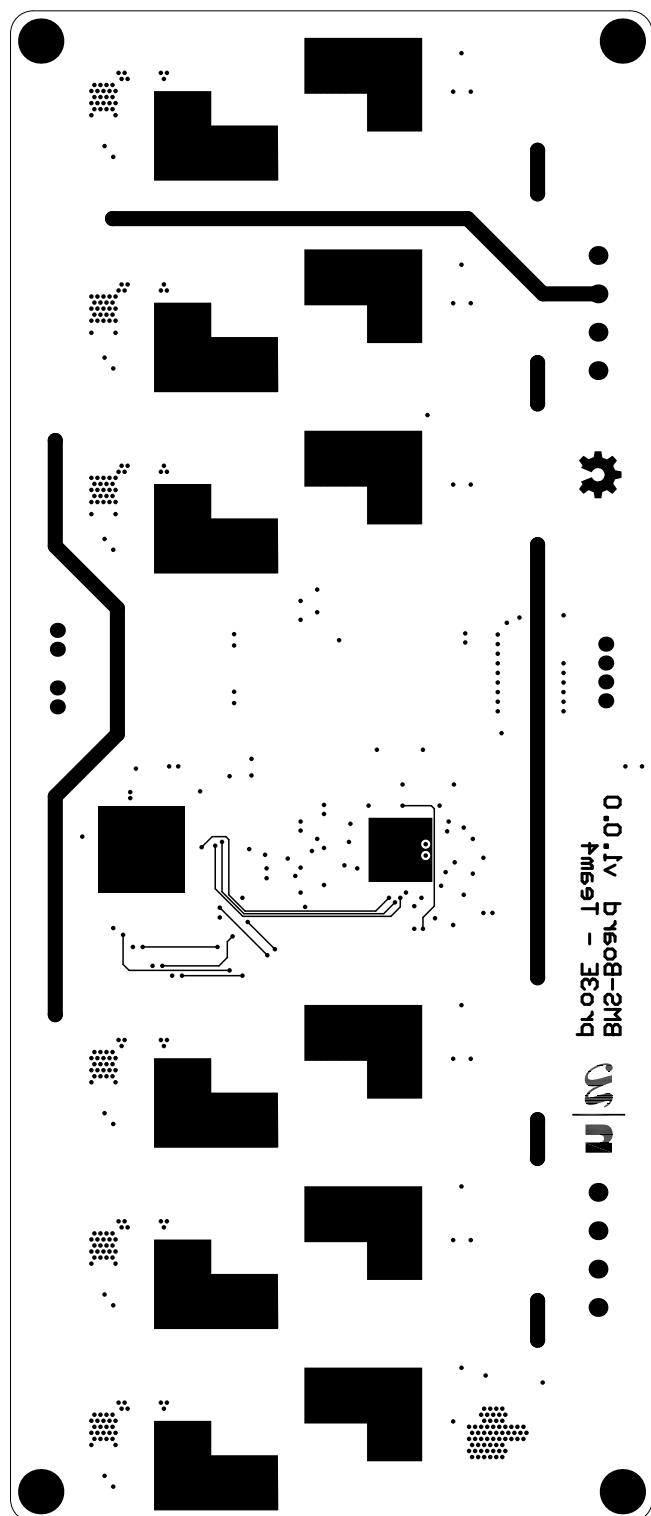




D Layout Slave







E Berechnungen Kommunikation

Master

Berechnet:

R_M	124 Ohm
I_B	500.0E-6 A
V_A	620.0E-3 V
V_TCMP	310.0E-3 V
V_ICMP	620.0E-3 V
R_B2	1240 Ohm
R_B1	2760 Ohm

Gewählt:

R_B2	1200 Ohm
R_B1	2700 Ohm
I_B	512.8E-6 A
I_DRV	10.3E-3 A
I_IP	10.3E-3 A
I_IM	10.3E-3 A
V_ICMP	615.4E-3 V
V_TCMP	307.7E-3 V
V_A	635.9E-3 V

For cables less than 10m:

SPI clock 1 MHz

For cables 50 meters or less:

$$I_B = 0.5 \text{mA}$$

$$V_A = (20 \cdot I_B) \cdot (R_M/2)$$

$$V_{TCMP} = 1/2 \cdot V_A$$

$$V_{ICMP} = 2 \cdot V_{TCMP}$$

$$R_{B2} = V_{ICMP}/I_B$$

$$R_{B1} = \left(\frac{2V}{I_B} \right) - R_{B2}$$

Slave

Berechnet:

R_M	124 Ohm
I_B	500.0E-6 A
K	0.5
V_A	620.0E-3 V
V_TCMP	310.0E-3 V
V_ICMP	620.0E-3 V
R_B2	1240 Ohm
R_B1	2760 Ohm

Gewählt:

R_B2	1200 Ohm
R_B1	2700 Ohm
I_B	512.8E-6 A
I_DRV	10.3E-3 A
I_IP	10.3E-3 A
I_IM	10.3E-3 A
V_ICMP	615.4E-3 V
V_TCMP	307.7E-3 V
V_A	635.9E-3 V

 R_M = Transmission Line Characteristic Impedance Z_0 $V_A = (20 \cdot I_B) \cdot (R_M/2)$ V_{TCMP} (Receiver Comparator Threshold) = $K \cdot V_A$ V_{ICMP} (voltage on ICMP pin) = $2 \cdot V_{TCMP}$

$$R_{B2} = V_{ICMP}/I_B$$

$$R_{B1} = (2/I_B) - R_{B2}$$

For lower power links: $I_B = 0.5\text{mA}$ and $K = 0.5$ For full power links: $I_B = 1\text{mA}$ and $K = 0.5$ For long links (>50m): $I_B = 1\text{mA}$ and $K = 0.25$

F Dimensionierung Slave

Fixe Werte			
Parameter	Wert	Einheit	Bemerkung
S	6		Secondary Cells in Stack
T	1		1:T transform ratio from prim. to sec.
V cell	3.6	V	
V cell max	4.2	V	
V cell min	3.2	V	
n charge	90%		Transfer efficiency sec to prim
n discharge	90%		Transfer efficiency prim to sec
L pri = L sec	3.64E-06	H	Datenblatt
C Nmosgate	2.00E-09	F	

Bauteile (Datenblatt)			
Parameter	Wert	Einheit	Bemerkung
C bypass Vreg zu V-	1.00E-6	F	16V
R-Parallel prim/sec	330.00E+0	Ohm	Trial and error
C-Parallel prim/sec	100.00E-12	F	Trial and error
Resistor+Schottky R:	6.80E+0	Ohm	
C für Filter pri	1.00E-9	F	
C für Filter sec	1.00E-9	F	

Annahmen für Berechnung			
Parameter	Wert	Einheit	Bemerkung
I charge	4.29	A	
I discharge	4.29	A	

Berechnungen			
Parameter	Wert	Einheit	Bemerkung
I peakprim	10.01E+0	A	
I peaksec	10.01E+0	A	
f charge	84.69E+3	Hz	
f discharge	84.69E+3	Hz	
t onmaxpri	11.39E-6	s	
t onmaxsec	1.90E-6	s	
RC Sensefilter pri	1.14E-6	OhmF = s	
RC Sensefilter sec	189.77E-9	OhmF = s	

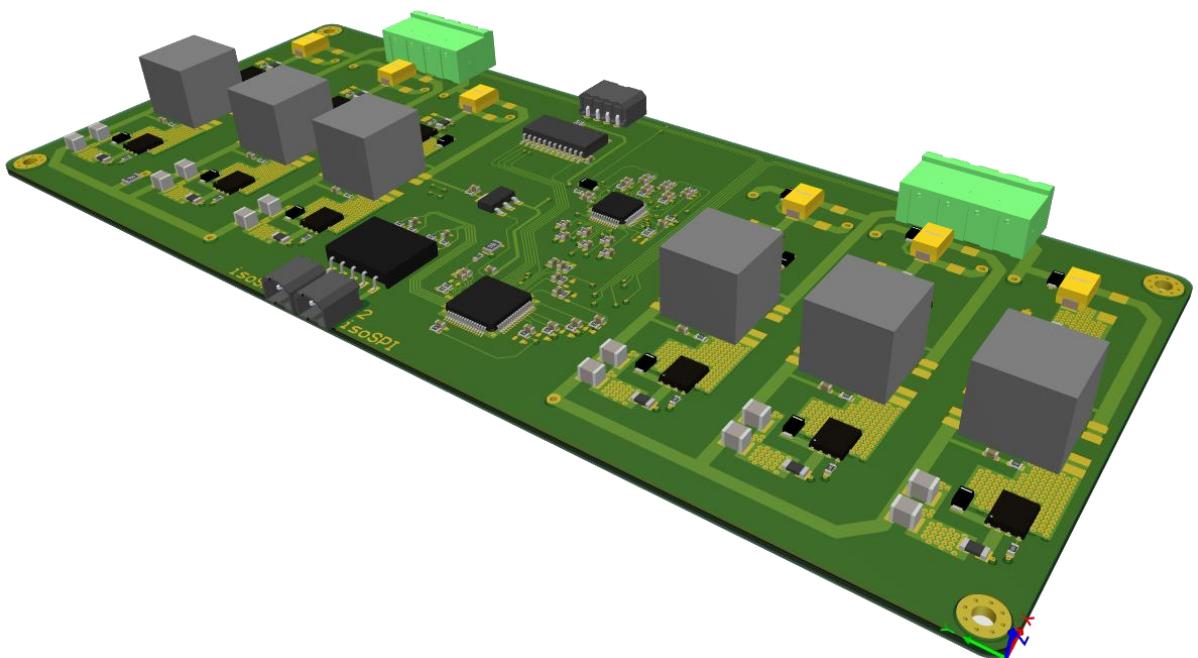
Bauteile berechnet				
Parameter	Wert	Einheit	Bemerkung	
R sensepri	5.00E-3	Ohm	0.25	W
R senssec	5.00E-3	Ohm	0.25	W
R tonp >	31.63E+3	Ohm	Bei Problemen,	
R tons >	23.72E+3	Ohm	R tons und p auf Vreg kurzschl.	
C bypass Cellmin	51.48E-6	F	10	V
C bypass Boost+/-	200.00E-9	F	10	V
R für Filter pri	1.14E+3	Ohm		
R für Filter sec	189.77E+0	Ohm		

Bauteile gewählt				
Parameter	Wert	Einheit	Bemerkung	
R sensepri	5.00E-3	Ohm	0.50	W
R senssec	5.00E-3	Ohm	0.50	W
R tonp >	33.00E+3	Ohm		
R tons >	51.00E+3	Ohm		
C bypass Cellmin	100.00E-6	F	10	V
C bypass Boost+/-	100.00E-9	F	10	V
R für Filter pri	1.10E+3	Ohm		
R für Filter sec	360.00E+0	Ohm		

G Prüfanweisung

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 1 von 12
Bearbeitet: FrF	Datum: 13.01.2019	
Freigabe: MuD	Datum: 13.01.2019	

TheBMS
Pro3E-Team4



n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 2 von 12

1 PRÜFAUFBAU

Prüfaufbau

Benötigte Instrumente:

- 6 Netzgeräte mind. 5A falls Spannung für Funktionsprüfung auf 3.0 VDC bzw. 4.0 VDC eingestellt werden muss. Ansonsten min. 1 A
Bsp.: EPS EPM-1865
- Leistungswiderstand Culatti (4.8 R / 16 A)
- Laptop als Hotspot
- TheBMS
- Multimeter

2 ABKÜRZUNGEN

Abkürzung	Beschreibung
BMS	Batterie-Management-System
isoSPI	Isolated Serial Peripheral Interface
UI	User Interface

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 3 von 12

3 PRÜFAUFBAU ERSTELLEN

Step	Prüfanweisung
	<p><u>Hinweis:</u> Die Balancingfunktion soll für jeden Block geprüft werden. Um einen Block zu simulieren, werden Netzgeräte verwendet und dienen somit als Blocksimulator. Dem jeweils zu ladendem Block wird ein Leistungswiderstand parallel geschaltet. Dies wird aufgrund der zu verarbeiteten Leistung getan.</p>
3.1	Prüfaufbau gemäss Beilage (Abbildung 3) erstellen. Die ausgegrauten Elemente werden benötigt, um die Balancing-Funktion für den jeweiligen Block zu testen. Leistungswiderstand und Multimeter werden für die jeweiligen Blockmessungen neu angeschlossen.
3.2	Blocksimulator 1 bis 6 auf 3.5 VDC einstellen und der Reihe nach (Blocksimulator 1, Blocksimulator 2,...) einschalten.
3.3	Micro-USB- Kabel am Master an rechtem Port einstecken und anschliessend Stecker bei 230 VAC anschliessen (siehe Beilage Abbildung 1)
3.4	Hotspot auf eigenem Gerät mit dem Namen thebms und Passwort pro3ETeam4 erstellen.
	<p><u>Hinweis:</u> Der Master verbindet sich automatisch mit dem Hotspot. Sobald der Master mit dem Hotspot verbunden ist, startet dieser die Übertragung der Telemetriedaten auf die Website.</p>
3.5	Auf dem eigenen Laptop im Browser TheBMS.ch öffnen
3.6	Auf dem User Interface sind nun alle angeschlossenen Spannungen, die Raumtemperatur sowie der Balancing-Status ablesbar.

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 4 von 12

4 FUNKTIONSPRÜFUNG

Step	Funktionsprüfung für Balancing von Block_1	
4.1	Leerlaufstrommessung von $I_{GND} = 3.5 \text{ mA} \pm 1 \text{ mA}$	<input type="checkbox"/>
4.2	Leistungswiderstand auf 1Ω eingestellt an Blocksimulator_1 anschliessen.	<input type="checkbox"/>
4.3	Um einen Balancing-Vorgang auszulösen wird der Blocksimulator_1 auf 3 VDC eingestellt. Blocksimulator_2 auf 4 VDC einstellen.	<input type="checkbox"/>
<p><u>Hinweis:</u> Der Balancing-Algorithmus ermittelt den Mittelwert der angeschlossenen Blocksimulatoren (dieser wäre in diesem Fall bei 3.5 VDC. Sobald ein Blocksimulator um 0.1 VDC davon abweicht wird das Balancing ausgelöst. Dabei wird die Energie der Blocksimulatoren, welche über dem Mittelwert liegen auf die Blocksimulatoren, welche unterhalb des Mittelwerts liegen transferiert.</p>		
4.4	Blocksimulator_2 liefert nun einen Ausgleichsstrom von 1 A (Blocksimulator_2 balanced Blocksimulator_1)	<input type="checkbox"/>
4.5	Strom von Blocksimulator_1 verringert sich nun auf ca. 3 A, da dieser von Blocksimulator_2 ge-balanced wird.	<input type="checkbox"/>
4.6	Blockspannungen sowie Temperaturüberwachung, sowie Temperaturmessung sind auf dem Userinterface ersichtlich.	<input type="checkbox"/>
4.7	Um eine Überspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_2 erhöhen. Balancing-Vorgang wird ab 4.2 VDC gestoppt.	<input type="checkbox"/>
4.8	Spannung an Blocksimulator_1 wieder auf 3VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.9	Um eine Unterspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_1 verringern. Balancing-Vorgang wird unter 2.8 VDC gestoppt.	<input type="checkbox"/>
4.10	Spannung an Blocksimulator_1 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.11	Strommessung während Balancing-Vorgang $I_{GND} = 10 \text{ mA} \pm 2 \text{ mA}$	<input type="checkbox"/>

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 5 von 12

Step	Funktionsprüfung der Sicherheitsfunktionen des Slaveprints	
4.12	Um die Bidirektionalität der Verbindung zu testen werden folgende Schritte durchgeführt: Grünes Kabel (isoSPI1) an Slaveprint ausstecken Balancing Vorgang wird weiterhin ausgeführt, da Kommunikation über oranges Kabel (isoSPI2) gegeben ist.	<input type="checkbox"/>
4.13	Grünes Kabel (isoSPI1) an Slaveprint wieder einstecken und oranges Kabel (isoSPI2) an Slaveprint ausstecken. Balancing Vorgang wird weiterhin ausgeführt, da Kommunikation über grünes Kabel (isoSPI1) gegeben ist.	<input type="checkbox"/>
4.14	Grünes Kabel (isoSPI1) ausstecken Balancing-Vorgang wird angehalten, da keine Kommunikation zum Master stattfinden kann.	<input type="checkbox"/>
4.15	Grünes Kabel (isoSPI1) an isoSPI2-Port auf Slaveprint anschliessen. Balancing wird erneut ausgeführt, da Kommunikation auch von isoSPI1 (Master) via isoSPI2 (Slave) gewährleistet werden kann.	<input type="checkbox"/>
4.16	Um eine Übertemperatur auf den Li-Ion 18650 Zellen zu simulieren, kann einer bzw. beide der Taster betätigt werden. Diese simulieren die Temperaturschalter, welche auf den Blocks angebracht werden. → Taster drücken und halten. Balancing-Vorgang wird gestoppt. Auf dem Userinterface wird ein Fehler ausgegeben.	<input type="checkbox"/>
4.17	Taster wieder entlasten. Blancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.18	Um eine Übertemperatur der Umgebung zu simulieren, mittels Wärmequelle den PT1000 (siehe Beilage Abbildung 2) erwärmen. Mittels Userinterface Umgebungstemperatur beobachten. Sobald 45°C erreicht bzw. überschritten werden, wird der Balancing-Vorgang gestoppt.	<input type="checkbox"/>
4.19	Sobald die Temperatur des PT1000 unter 45°C sinkt, wird der Balancing-Vorgang wieder ausgeführt.	<input type="checkbox"/>
4.20	Um eine zu kalte Umgebung zu simulieren, mittels Kältespray den PT1000 (siehe Beilage Abbildung 2) abkühlen. Mittels Userinterface Umgebungstemperatur beobachten. Sobald 0°C erreicht bzw. unterschritten werden, wird der Balancing-Vorgang gestoppt.	<input type="checkbox"/>
4.21	Sobald die Temperatur des PT1000 über 0°C steigt, wird der Balancing-Vorgang wieder ausgeführt.	<input type="checkbox"/>
Ab hier werden die Schritte um die Funktionalität der Blöcke 2, 3, 4, 5 und 6 zu prüfen gleich durchgeführt wie für Block 1		

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 6 von 12

Step	Funktionsprüfung für Balancing von Block_2	
4.22	Alle Blocksimulatoren auf 3.5 VDC einstellen. Leerlaufstrommessung von $I_{GND} = 3 \text{ mA} \pm 1 \text{ mA}$	<input type="checkbox"/>
4.23	Leistungswiderstand auf 1Ω eingestellt an Blocksimulator_2 anschliessen.	<input type="checkbox"/>
4.24	Um einen Balancing-Vorgang auszulösen wird der Blocksimulator_2 auf 3 VDC eingestellt. Blocksimulator_3 auf 4 VDC einstellen.	<input type="checkbox"/>
4.25	Blocksimulator_3 liefert nun einen Ausgleichsstrom von 1 A (Blocksimulator_3 balanced Blocksimulator_2)	<input type="checkbox"/>
4.26	Strom von Blocksimulator_2 verringert sich nun auf ca. 3 A, da dieser von Blocksimulator_3 ge-balanced wird.	<input type="checkbox"/>
4.27	Blockspannungen sowie Temperaturüberwachung, sowie Temperaturmessung sind auf dem Userinterface ersichtlich.	<input type="checkbox"/>
4.28	Um eine Überspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_3 erhöhen. Balancing-Vorgang wird ab 4.2 VDC gestoppt.	<input type="checkbox"/>
4.29	Spannung an Blocksimulator_2 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.30	Um eine Unterspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_2 verringern. Balancing-Vorgang wird unter 2.8 VDC gestoppt.	<input type="checkbox"/>
4.31	Spannung an Blocksimulator_2 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.32	Strommessung während Balancing-Vorgang $I_{GND} = 10 \text{ mA} \pm 2 \text{ mA}$	<input type="checkbox"/>

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 7 von 12

Step	Funktionsprüfung für Balancing von Block_3	
4.33	Alle Blocksimulatoren auf 3.5 VDC einstellen. Leerlaufstrommessung von $I_{GND} = 3 \text{ mA} \pm 1 \text{ mA}$	<input type="checkbox"/>
4.34	Leistungswiderstand auf 1Ω eingestellt an Blocksimulator_3 anschliessen.	<input type="checkbox"/>
4.35	Um einen Balancing-Vorgang auszulösen wird der Blocksimulator_3 auf 3 VDC eingestellt. Blocksimulator_4 auf 4 VDC einstellen.	<input type="checkbox"/>
4.36	Blocksimulator_4 liefert nun einen Ausgleichsstrom von 1 A (Blocksimulator_4 balanced Blocksimulator_2)	<input type="checkbox"/>
4.37	Strom von Blocksimulator_3 verringert sich nun auf ca. 3 A, da dieser von Blocksimulator_4 ge-balanced wird.	<input type="checkbox"/>
4.38	Blockspannungen sowie Temperaturüberwachung, sowie Temperaturmessung sind auf dem Userinterface ersichtlich.	<input type="checkbox"/>
4.39	Um eine Überspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_4 erhöhen. Balancing-Vorgang wird ab 4.2 VDC gestoppt.	<input type="checkbox"/>
4.40	Spannung an Blocksimulator_3 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.41	Um eine Unterspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_3 verringern. Balancing-Vorgang wird unter 2.8 VDC gestoppt.	<input type="checkbox"/>
4.42	Spannung an Blocksimulator_3 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.43	Strommessung während Balancing-Vorgang $I_{GND} = 10 \text{ mA} \pm 2 \text{ mA}$	<input type="checkbox"/>

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 8 von 12

Step	Funktionsprüfung für Balancing von Block_4	
4.44	Alle Blocksimulatoren auf 3.5 VDC einstellen. Leerlaufstrommessung von $I_{GND} = 3 \text{ mA} \pm 1 \text{ mA}$	<input type="checkbox"/>
4.45	Leistungswiderstand auf 1Ω eingestellt an Blocksimulator_4 anschliessen.	<input type="checkbox"/>
4.46	Um einen Balancing-Vorgang auszulösen wird der Blocksimulator_4 auf 3 VDC eingestellt. Blocksimulator_5 auf 4VDC einstellen.	<input type="checkbox"/>
4.47	Blocksimulator_5 liefert nun einen Ausgleichsstrom von 1 A (Blocksimulator_5 balanced Blocksimulator_4)	<input type="checkbox"/>
4.48	Strom von Blocksimulator_4 verringert sich nun auf ca. 3 A, da dieser von Blocksimulator_5 ge-balanced wird.	<input type="checkbox"/>
4.49	Blockspannungen sowie Temperaturüberwachung, sowie Temperaturmessung sind auf dem Userinterface ersichtlich.	<input type="checkbox"/>
4.50	Um eine Überspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_5 erhöhen. Balancing-Vorgang wird ab 4.2 VDC gestoppt.	<input type="checkbox"/>
4.51	Spannung an Blocksimulator_4 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.52	Um eine Unterspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_4 verringern. Balancing-Vorgang wird unter 2.8 VDC gestoppt.	<input type="checkbox"/>
4.53	Spannung an Blocksimulator_4 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.54	Strommessung während Balancing-Vorgang $I_{GND} = 10 \text{ mA} \pm 2 \text{ mA}$	<input type="checkbox"/>

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 9 von 12

Step	Funktionsprüfung für Balancing von Block_5	
4.55	Alle Blocksimulatoren auf 3.5 VDC einstellen. Leerlaufstrommessung von $I_{GND} = 3 \text{ mA} \pm 1 \text{ mA}$	<input type="checkbox"/>
4.56	Leistungswiderstand auf 1Ω eingestellt an Blocksimulator_5 anschliessen.	<input type="checkbox"/>
4.57	Um einen Balancing-Vorgang auszulösen wird der Blocksimulator_5 auf 3 VDC eingestellt. Blocksimulator_6 auf 4VDC einstellen.	<input type="checkbox"/>
4.58	Blocksimulator_6 liefert nun einen Ausgleichsstrom von 1A (Blocksimulator_6 balanced Blocksimulator_5)	<input type="checkbox"/>
4.59	Strom von Blocksimulator_5 verringert sich nun auf ca. 3 A, da dieser von Blocksimulator_6 ge-balanced wird.	<input type="checkbox"/>
4.60	Blockspannungen sowie Temperaturüberwachung, sowie Temperaturmessung sind auf dem Userinterface ersichtlich.	<input type="checkbox"/>
4.61	Um eine Überspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_6 erhöhen. Balancing-Vorgang wird ab 4.2 VDC gestoppt.	<input type="checkbox"/>
4.62	Spannung an Blocksimulator_5 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.63	Um eine Unterspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_5 verringern. Balancing-Vorgang wird unter 2.8 VDC gestoppt.	<input type="checkbox"/>
4.64	Spannung an Blocksimulator_5 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.65	Strommessung während Balancing-Vorgang $I_{GND} = 10 \text{ mA} \pm 2 \text{ mA}$	<input type="checkbox"/>

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 10 von 12

Step	Funktionsprüfung für Balancing von Block_6	
4.66	Alle Blocksimulatoren auf 3.5 VDC einstellen. Leerlaufstrommessung von $I_{GND} = 3 \text{ mA} \pm 1 \text{ mA}$	<input type="checkbox"/>
4.67	Leistungswiderstand auf 1Ω eingestellt an Blocksimulator_6 anschliessen.	<input type="checkbox"/>
4.68	Um einen Balancing-Vorgang auszulösen wird der Blocksimulator_6 auf 3 VDC eingestellt. Blocksimulator_1 auf 4VDC einstellen.	<input type="checkbox"/>
4.69	Blocksimulator_1 liefert nun einen Ausgleichsstrom von 1 A (Blocksimulator_1 balanced Blocksimulator_6)	<input type="checkbox"/>
4.70	Strom von Blocksimulator_6 verringert sich nun auf ca. 3 A, da dieser von Blocksimulator_1 ge-balanced wird.	<input type="checkbox"/>
4.71	Blockspannungen sowie Temperaturüberwachung, sowie Temperaturmessung sind auf dem Userinterface ersichtlich.	<input type="checkbox"/>
4.72	Um eine Überspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_1 erhöhen. Balancing-Vorgang wird ab 4.2 VDC gestoppt.	<input type="checkbox"/>
4.73	Spannung an Blocksimulator_6 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.74	Um eine Unterspannung der zu balancenden Zelle zu simulieren, Spannung an Blocksimulator_6 verringern. Balancing-Vorgang wird unter 2.8 VDC gestoppt.	<input type="checkbox"/>
4.75	Spannung an Blocksimulator_6 wieder auf 3 VDC einstellen. Balancing-Vorgang wird erneut ausgeführt.	<input type="checkbox"/>
4.76	Strommessung während Balancing-Vorgang $I_{GND} = 10 \text{ mA} \pm 2 \text{ mA}$	<input type="checkbox"/>

5 REVISIONSKONTROLLE

Rev	History – Grund der Änderung	Datum	Verantwortlich
1.0	Dokument erstellt	11.01.2019	P. Fankhauser
2.0	Dokument überarbeitet. Zudem wurden Ergänzungen und Erweiterungen zur effizienteren und qualitativeren Prüfung hinzugefügt.	13.01.2019	F. Frey

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 11 von 12

6 BEILAGE

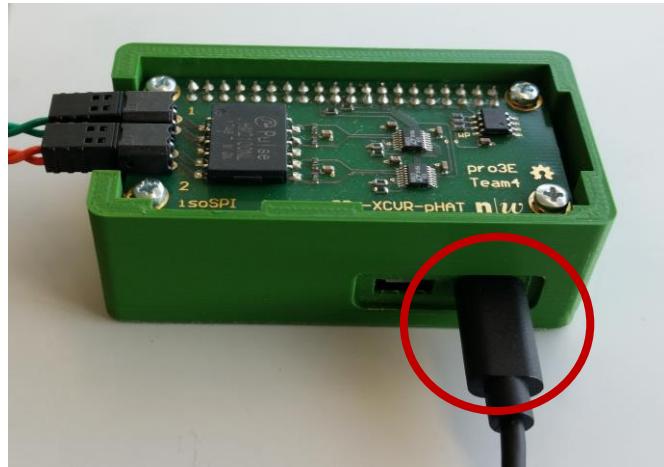


Abbildung 1: Masterprint

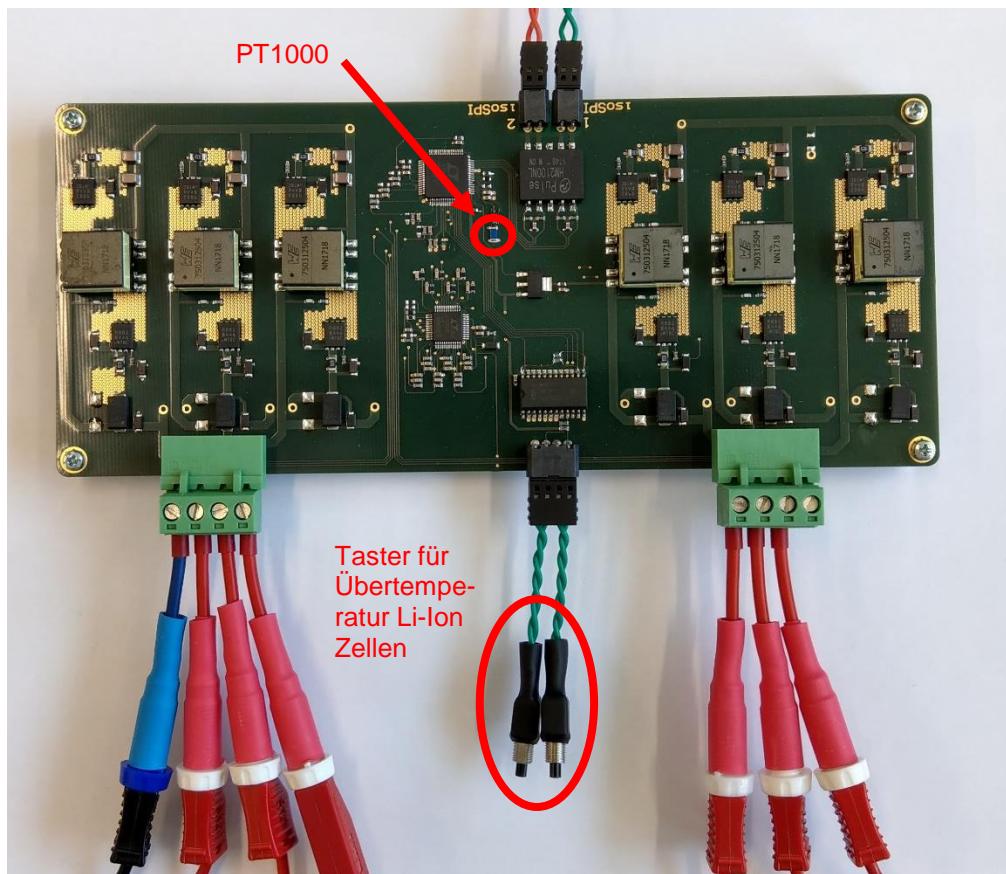


Abbildung 2: Slaveprint

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfanweisung TheBMS	Pro3ETeam4
Verfasst: FaP	Datum: 11.01.2019	Seite: 12 von 12

BLOCKSIMULATOREN

mit Leistungswiderständen für Funktionsprüfung

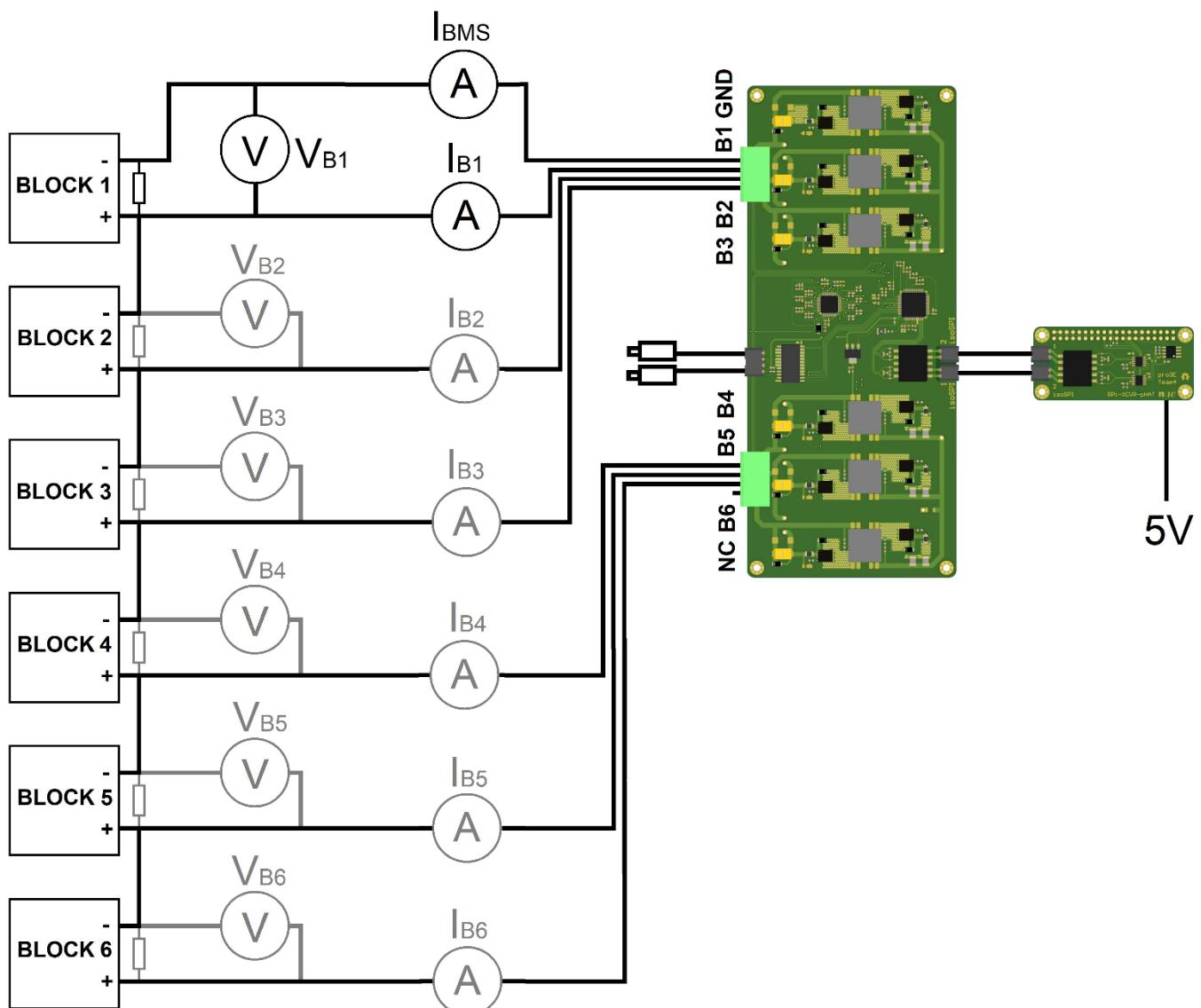
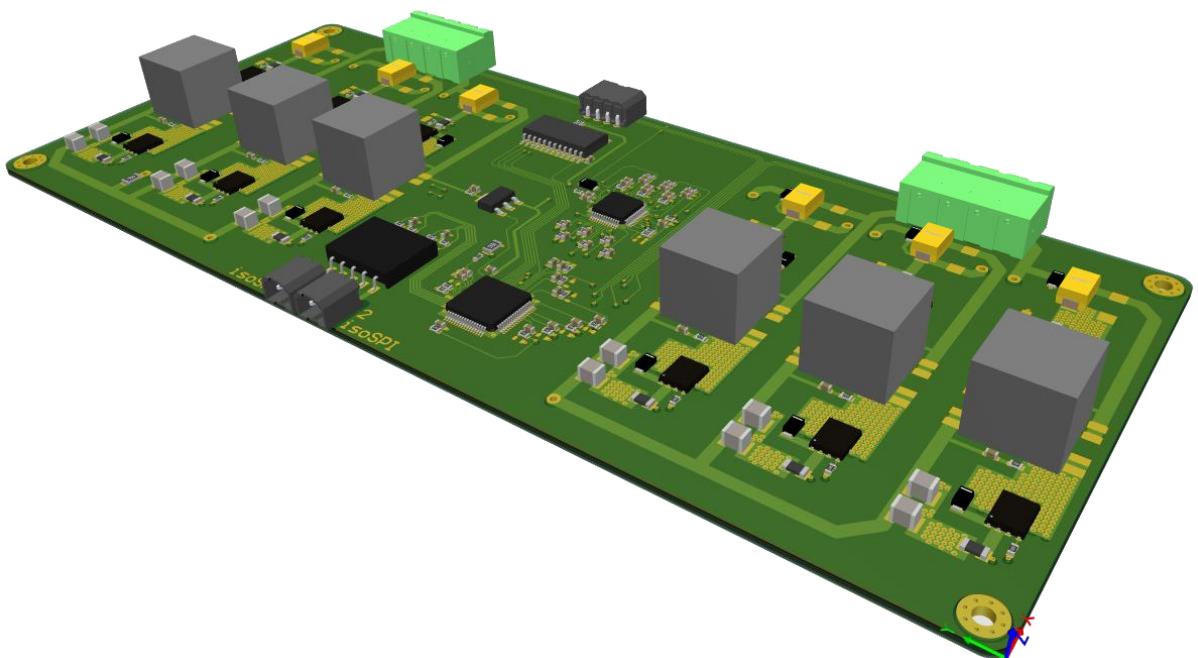


Abbildung 3: Prüfschema

H Prüfprotokoll

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfprotokoll TheBMS	Pro3ETeam4
Verfasst: FrF	Datum: 12.01.2019	Seite: 1 von 4
Bearbeitet: FrF	Datum: 12.01.2019	
Freigabe: FrF	Datum: 12.01.2019	

TheBMS
Pro3E-Team4



n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfprotokoll TheBMS	Pro3ETeam4
Verfasst: FrF	Datum: 12.01.2019	Seite: 2 von 4

1 PRÜFPROTOKOLL

Effizienzmessung

Um die Effizienz bestimmen zu können wurde der Wirkungsgrad mit gemessenen Werten berechnet.

Dazu wurden Messungen von jeweils einem Block zum anderen getätigt. Die Leistung, welche von dem zu ladenden Block aufgenommen wird, wurde durch die Leistung der abgegebenen Blöcke geteilt.

Folgende Ergebnisse wurden erzielt:

Von Block 4 zu Block 5:

Leistung abgegeben von Block 4 $P_4 = 3.898 \text{ V} \cdot 1.86 \text{ A} = 7.25 \text{ W}$

Leistung aufgenommen von Block 5 $P_5 = 3.065 \text{ V} \cdot 0.80 \text{ A} = 2.45 \text{ W}$

Leistung ebenfalls aufgenommen von Block 3 (aufgrund Balancing-Algorithmus)

$P_3 = 3.533 \text{ V} \cdot 1.1 \text{ A} = 3.89 \text{ W}$

$$\text{Wirkungsgrad } \eta = \frac{P_{\text{aufgenommen}}}{P_{\text{abgegeben}}} = \frac{P_5 + P_3}{P_4} = \frac{2.45 \text{ W} + 3.89 \text{ W}}{7.25 \text{ W}} = \frac{6.34 \text{ W}}{7.25 \text{ W}} = \underline{\underline{0.87}}$$

Von Block 6 zu Block 5:

Leistung abgegeben von Block 6 $P_6 = 3.95 \text{ V} \cdot 0.97 \text{ A} = 3.83 \text{ W}$

Leistung ebenfalls abgegeben von Block 4 (aufgrund Balancing-Algorithmus) $P_4 = 3.46 \text{ V} \cdot 0.62 \text{ A} = 2.15 \text{ W}$

Leistung aufgenommen von Block 5 $P_5 = 3.1 \text{ V} \cdot 1.58 \text{ A} = 4.89 \text{ W}$

$$\text{Wirkungsgrad } \eta = \frac{P_{\text{aufgenommen}}}{P_{\text{abgegeben}}} = \frac{P_5}{P_6 + P_4} = \frac{4.89 \text{ W}}{3.83 \text{ W} + 2.15 \text{ W}} = \frac{4.89 \text{ W}}{5.98 \text{ W}} = \underline{\underline{0.81}}$$

Von Block 4 und 6 zu Block 3 und 5:

Leistung abgegeben von Block 4 $P_4 = 3.88 \text{ V} \cdot 1.6 \text{ A} = 6.2 \text{ W}$

Leistung abgegeben von Block 6 $P_6 = 3.95 \text{ V} \cdot 0.93 \text{ A} = 3.67 \text{ W}$

Leistung ebenfalls abgegeben von Block 2 (aufgrund Balancing-Algorithmus) $P_2 = 3.50 \text{ V} \cdot 0.60 \text{ A} = 2.1 \text{ W}$

Leistung aufgenommen von Block 3 $P_3 = 3.06 \text{ V} \cdot 1.65 \text{ A} = 5.05 \text{ W}$

Leistung aufgenommen von Block 5 $P_5 = 3.07 \text{ V} \cdot 1.48 \text{ A} = 4.54 \text{ W}$

$$\text{Wirkungsgrad } \eta = \frac{P_{\text{aufgenommen}}}{P_{\text{abgegeben}}} = \frac{P_4 + P_6 + P_2}{P_3 + P_5} = \frac{3.83 \text{ W} + 2.15 \text{ W} + 2.1 \text{ W}}{5.05 \text{ W} + 4.54 \text{ W}} = \frac{8.08 \text{ W}}{9.59 \text{ W}} = \underline{\underline{0.84}}$$

Von Block 4 zu Block 3 und 5:

Leistung abgegeben von Block 4 $P_4 = 3.90 \text{ V} \cdot 1.1 \text{ A} = 4.29 \text{ W}$

Leistung ebenfalls abgegeben von Block 2 (aufgrund Balancing-Algorithmus) $P_2 = 3.39 \text{ V} \cdot 0.24 \text{ A} = 0.81 \text{ W}$

Leistung aufgenommen von Block 3 $P_3 = 3.03 \text{ V} \cdot 1.14 \text{ A} = 3.45 \text{ W}$

Leistung aufgenommen von Block 5 $P_5 = 3.03 \text{ V} \cdot 0.21 \text{ A} = 0.63 \text{ W}$

$$\text{Wirkungsgrad } \eta = \frac{P_{\text{aufgenommen}}}{P_{\text{abgegeben}}} = \frac{P_3 + P_5}{P_4 + P_2} = \frac{3.45 \text{ W} + 0.63 \text{ W}}{4.29 \text{ W} + 0.81 \text{ W}} = \frac{4.08 \text{ W}}{5.1 \text{ W}} = \underline{\underline{0.8}}$$

Somit haben wir auf ein Spitzenergebnis des Wirkungsgrades von 87.4% erreicht.

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfprotokoll TheBMS	Pro3ETeam4
Verfasst: FrF	Datum: 12.01.2019	Seite: 3 von 4

Bidirektionale Kommunikation

Damit im Fall eines defekten Leiters die Kommunikation mit dem Masterprint erhalten bleibt, wird eine bidirektionale Kommunikation eingesetzt.

Um diese zu testen, können die grünen und orangen isoSPI-Kabel beliebig getauscht und einzeln ausgesteckt werden. Der Balancing-Vorgang wird nach wie vor ausgeführt, solange mindestens eine Verbindung zum Masterprint besteht.

Stromaufnahme Slaveprint

Die Stromaufnahme des Slaveprint wurde über die GND Leitung gemessen, da diese keine Balancingströme führt. Es fließen lediglich der Strom des Slaveprints.

Die Stromaufnahme, wenn kein aktives Balancing durchgeführt wird liegt bei 3.5 mA

Die Stromaufnahme, wenn aktives Balancing durchgeführt wird liegt bei 10.5 mA

2 REVISIONSKONTROLLE

Rev	History – Grund der Änderung	Datum	Verantwortlich
1.0	Dokument erstellt	12.01.2019	F. Frey

n w Fachhochschule Nordwestschweiz Hochschule für Technik	Prüfprotokoll TheBMS	Pro3ETeam4
Verfasst: FrF	Datum: 12.01.2019	Seite: 4 von 4

3 BEILAGE

BLOCKSIMULATOREN mit Leistungswiderständen für Funktionsprüfung

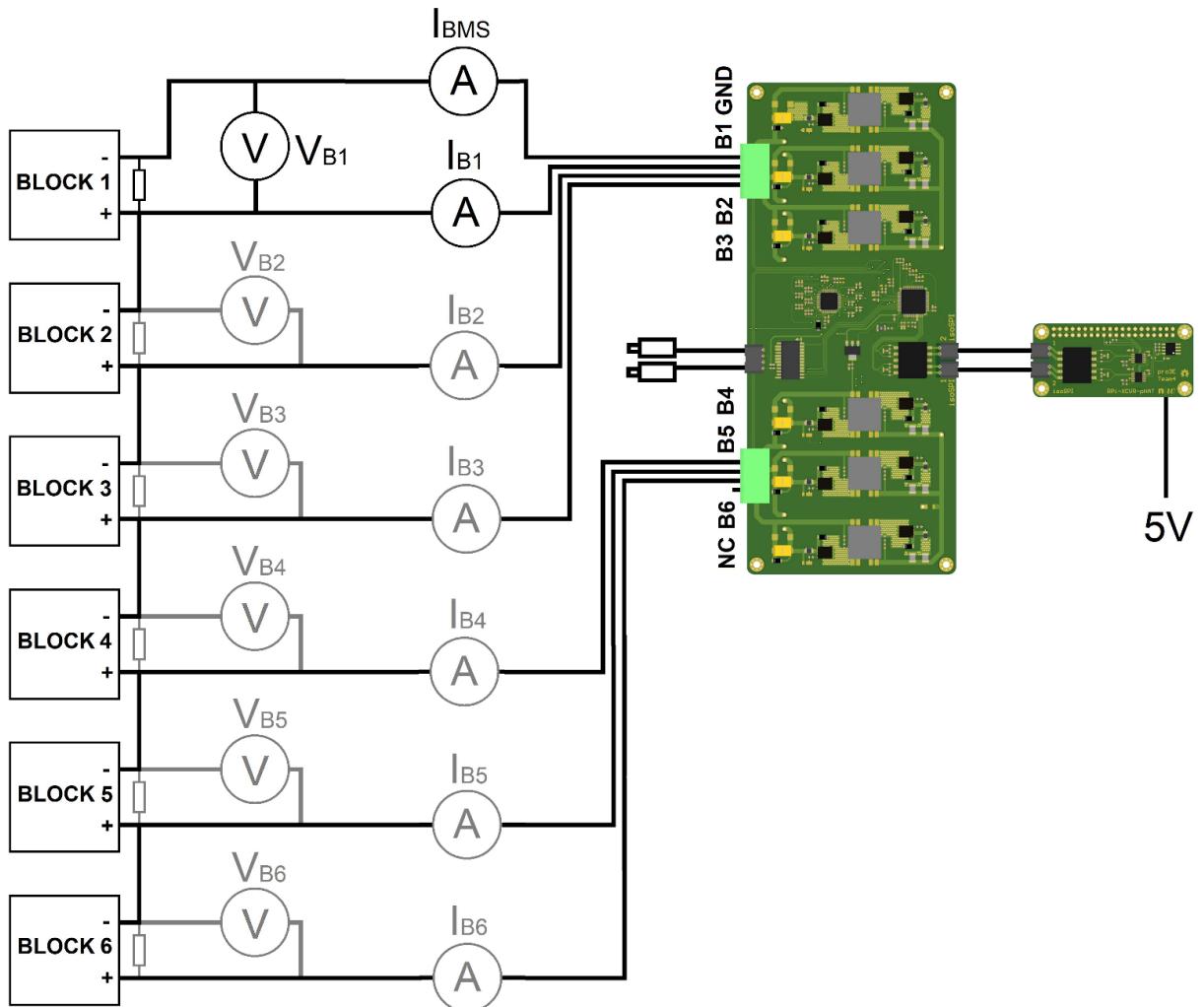


Abbildung 1: Prüfschema