

EINFACHES LI-ION BMS

mit AVR-Mikrocontrollern

Historie

Datum	
01.02.2012	Erste Version, grundsätzliche Struktur
28.02.2012	Topologien
11.03.2012	Strommessung, Master
25.05.2012	Ablaufplan Software
11.06.2012	Erste Gedanken zur Umsetzung des Masters
18.07.2012	Balancing deaktivieren beim Messen
07.08.2012	Diverse Modifikationen im internen Protokoll
24.08.2012	Konfiguration des Balancings und thermischen Managements
26.08.2012	Zähler für thermisches Management, Definitionen Fehler/Warnungen
01.09.2012	PackConnectionConfig
08.09.2012	CurrentADC Tabelle
12.10.2012	Messung der Stromaufnahme
01.11.2012	Korrekturen, Messdiagramme

Autor: Frank Bättermann

Inhaltsverzeichnis

<u>Abkürzungsverzeichnis.....</u>	<u>4</u>
<u>1 Vorwort.....</u>	<u>5</u>
<u>1.1 Verbesserungen.....</u>	<u>5</u>
<u>1.2 Lizenz</u>	<u>6</u>
<u>2 Anforderungen.....</u>	<u>7</u>
<u>2.1 Strommessung.....</u>	<u>8</u>
<u>2.2 Balancing.....</u>	<u>9</u>
<u>2.2.1 Passives Balancing.....</u>	<u>9</u>
<u>2.2.2 Aktives Balancing.....</u>	<u>9</u>
<u>3 Topologien.....</u>	<u>10</u>
<u>3.1.1 Einzelzellen.....</u>	<u>10</u>
<u>3.1.2 Kleine Batteriepacks.....</u>	<u>10</u>
<u>3.1.3 Große Batteriepacks.....</u>	<u>10</u>
<u>3.2 Serielle Schnittstelle am Mikrocontroller.....</u>	<u>11</u>
<u>4 Master.....</u>	<u>13</u>
<u>4.1 Aufgaben.....</u>	<u>13</u>
<u>4.2 Sicherheit.....</u>	<u>13</u>
<u>4.3 Datenaufzeichnung.....</u>	<u>13</u>
<u>4.4 Bestimmung des SOC.....</u>	<u>14</u>
<u>4.5 Hardware.....</u>	<u>14</u>
<u>4.5.1 Schnittstellen.....</u>	<u>15</u>
<u>4.5.2 Eingänge.....</u>	<u>15</u>
<u>4.5.3 Interne Ausgänge.....</u>	<u>15</u>
<u>4.5.4 Ausgänge.....</u>	<u>15</u>
<u>4.5.5 Anzeigen.....</u>	<u>16</u>
<u>4.6 Stromaufnahme (experimentell).....</u>	<u>16</u>
<u>4.7 Software.....</u>	<u>16</u>
<u>4.7.1 Masterstatus.....</u>	<u>17</u>
<u>4.7.2 Batterieverbindungsstatus.....</u>	<u>17</u>
<u>4.7.3 Slaveauslesezyklus-Status.....</u>	<u>17</u>
<u>5 Slave.....</u>	<u>18</u>
<u>5.1 Aufgaben.....</u>	<u>18</u>
<u>5.2 Hardware.....</u>	<u>18</u>
<u>5.2.1 Controller.....</u>	<u>18</u>
<u>5.2.1.1 Takterzeugung.....</u>	<u>18</u>
<u>5.2.1.2 Anforderungen für passives Balancing.....</u>	<u>19</u>
<u>5.2.1.3 Anforderungen für aktives Balancing.....</u>	<u>19</u>
<u>5.2.1.4 Mögliche AVR-Controller.....</u>	<u>20</u>
<u>5.2.2 Referenz.....</u>	<u>20</u>
<u>5.2.3 Spannungsmessung.....</u>	<u>20</u>
<u>5.2.3.1 Geschalteter Spannungsteiler.....</u>	<u>20</u>
<u>5.2.3.2 Interne Referenz gegen VCC.....</u>	<u>21</u>
<u>5.2.4 Temperaturmessung.....</u>	<u>22</u>
<u>5.2.4.1 KTY81-210.....</u>	<u>22</u>
<u>5.2.5 Stromversorgung.....</u>	<u>23</u>
<u>5.2.6 Stromaufnahme (experimentell).....</u>	<u>24</u>
<u>5.3 Software.....</u>	<u>25</u>

5.3.1 Laden der Konfiguration.....	25
5.3.2 Hauptschleife.....	25
5.3.3 Sleepmode.....	27
5.3.4 Aufwachen.....	28
5.3.5 Messung.....	28
5.3.6 Datenempfang.....	28
6 Master-Slave Bus.....	29
6.1 Adressen.....	29
6.2 Paketaufbau.....	29
6.3 Übertragungsformat.....	29
6.3.1 Zellspannung.....	30
6.3.2 Zelltemperatur.....	30
6.4 Standardwerte und Kalibrierung.....	30
6.5 Baudrate.....	30
6.5.1 Berechnungsvorschriften.....	31
6.5.2 Beispiele.....	31
6.6 Kommandos.....	31
6.6.1 Measure.....	32
6.6.2 Sleep.....	33
6.6.3 Result.....	33
7 Konfigurationsschnittstelle (seriell).....	34
7.1 Nachrichtenformat.....	34
7.2 Nachrichten.....	34
7.3 Konfigurationsmodus.....	40
7.4 Balancing.....	41
7.5 Thermisches Management.....	41
7.6 Zellspannungs-SOC-Tabelle.....	41
7.7 Stromkalibrierungs-Table.....	42
7.8 Rückmeldungen.....	42
8 Test.....	43
8.1 Aufbau.....	43
8.2 Diagramme von Testmessungen.....	43

Abkürzungsverzeichnis

ADC	Analog-Digital-Converter
BMS	Batterie Management System
C-Rate	Quotient aus Nennkapazität und tatsächlich fließendem Strom
SOC	State of Charge
SOE	State of Energy
SOH	State of Health
UART	Universal Asynchron Receiver Transmitter

1 Vorwort

Ich begann dieses BMS als Lernprojekt und aus der Erkenntnis heraus, dass kein skalierbares BMS mit offenem Code als Lernprojekt im Internet zu finden war. Stand heute ist ein nicht vollständig fertiger Prototyp aus dem Mai 2012. Die Veröffentlichung findet in der Hoffnung statt, dass das Material für jemanden hilfreich ist.

Folgende Hinweise bedürfen der unbedingten Beachtung:

1.1 Verbesserungen

- Abgesehen von der Schmelzsicherung und besteht kein Hardware-Kurschlusschutz. Der Master übernimmt zwar die Aufgabe durch die Strommessung, doch kann die Reaktionszeit von 10 bis 30 ms bereits ausreichen um die Hardware nachhaltig zu schädigen, dass ein Abschalten mehrerer hundert oder tausend Ampere nicht mehr möglich ist. Hier muss ein Hardware-Schutz durch einen Abgriff an den Strommess-Widerständen erfolgen und der Zustand z.B. in einem Flip-Flop zwischengespeichert werden.
- Im derzeitigen Aufbau mit 12 Slaves aus dem Prototypen-Platine kann es bei einem Leitungsdefekt zu einer Spannungaufteilung zwischen zwei Slaves kommen, die Controller nicht überleben. Hier können die Slaves mit einer 5,1V 1W Z-Diode anti-parallel zur Zelle die Slaves geschützt werden.
- Der Master speichert Laufzeitdaten derzeit lediglich so er in den powerdown-Modus versetzt wird. Zukünftig sollte dies zum Schutz vor (zu viel) Datenverlust auch zyklisch geschehen (unter Beachtung der EEPROM Lebensdauer).
- Die Firmware des Masters sollte, so sie aktiv in Entwicklung ist, bei jedem Start die Standardeinstellungen statt der gespeicherten Konfiguration laden. Anderenfalls könnten durch eine Änderung des Konfigurations-Strukturen Werte unbemerkt in falsche Variablen geladen werden. Geschieht dies z.B. mit einer Spannungsbegrenzung könnte dies zu Sicherheitsproblemen führen.
- Der Master-Mikrocontroller sollte deutlich mehr Pins haben um mehr Ein- und Ausgänge zur Verfügung stellen zu können.
- Viele Grenzwerte sollten mit einer kleinen Hysterese versehen werden, z.B. für die zu erreichende Differenzspannung zwischen den einzelnen Zellspannungen.
- Testzyklen haben ergeben, dass die Linearisierung der Temperaturmessung in der Firmware recht gut funktioniert. Dabei gibt es jedoch Sprünge, die nicht vorhanden

sein sollten, z.B. im Bereich von 0-3 °C oder 40-42 °C. Da diese die gleichen Linearisierungsfunktionen sind wie für andere Messungen z.B. Strom müssen alle Funktionen nach einer Veränderung erneut überprüft werden.

- Der Nullpunkt der Strommessung scheint sich beim Prototypen bei tiefen Temperaturen zu verschieben. Dies könnte mit den INA169 zu tun haben, da der A/D-Wandler eine recht gute Stabilität in der Referenz haben sollte. Die Messung des Entladestroms blieb hierbei jedoch unbeeinträchtigt.
- Die Spannungsmessung wird bei -20 °C deutlich verrauschter. Möglicherweise ist dies durch die interne Referenz begründet. Die Messfehler vergrößerten sich von +- 5 mV bei Zimmertemperatur in den Bereich +-50 mV. Als sicherheitsrelevant würde ich dieses Verhalten jedoch nicht einstufen, da selbst kommerzielle BMS-Chips diesen Fehler teilweise als Messauflösung zuzüglich Fehler angeben.
- Eine zukünftige Version des Masters sollte deutlich schneller getaktet sein oder direkt durch einen verhältnismäßig langsamen taktenden ARM-Controller ersetzt werden. Ein Hardware-CAN-Interface kann den Datendurchsatz deutlich erhöhen; das Protokoll muss dafür nicht verändert werden.
- Im Quelltext sowohl der Master- als auch der Slave Firmware sind viele Stellen mit „TODO“ gekennzeichnet. Hier sollten weitere Verbesserungen vorgenommen werden. Unbedingt dazugezählt werden sollte der Basisalgorithmus zur SOC-Bestimmung „Columb counting“

1.2 Lizenz

Diese Arbeit und ihre Teile dürfen unter der Lizenz Creative Commons NC-BY-SA v3 frei verwendet werden. Weitere Informationen zu den Lizenzbedingungen finden Sie unter <http://creativecommons.org/licenses/by-nc-sa/3.0/de/>.

2 Anforderungen

Da Lithium-Zellen sehr empfindlich auf Überladung, Tiefentladung und zu hohe Stromstärken reagieren wird ein Batterie Management System (BMS) zur Überwachung dieser Parameter eingesetzt. Dieses stellt sicher, dass unter allen Umständen ein sicherer Betrieb des Batteriespeichers möglich ist. Zwar verhalten sich verschiedene Zellchemieen unterschiedlich, doch sind die überwachten Parameter die selben und variieren lediglich in der Höhe der Grenzwerte. Darüber hinaus übernimmt ein BMS die Steuerung der Klimatisierung eines Batteriepacks und des Ladevorgangs. Die Bestimmung des Ladezustandes ist ebenfalls eine Kernfunktion und der für den Benutzer wichtigste Kennwert der über eine Schnittstelle der Außenwelt zur Verfügung gestellt wird.

Maximale und Minimale Zellspannung: Je nach Zellchemie und wie schonend man die Zellen verwenden möchte werden die Grenzwerte gewählt.

Maximaler Strom in Lade- und Entladerichtung: Die max. C-Rate ist in beiden Richtungen oft unterschiedlich definiert. Setzt sich eine logische Zelle eines Batteriepacks aus einer Parallelschaltung mehrerer einzelner baugleicher Zellen zusammen, so erhöht sich der maximale Strom dementsprechend.

Der einzige Effektive Schutz, den das BMS bzgl. der Spannungen und Ströme bieten kann ist die Trennung vom Ladegerät bzw. der Last beim Erreichen der Grenzwerte. Bei kleineren Packs mit Gesamtspannungen unterhalb der Schutzkleinspannung – die nicht galvanisch getrennt werden müssen – kann dies durch p-Kanals MosFets in der Plus-Leitung geschehen. Bei höheren Spannungen durch Relais mit entsprechender Stromtragfähigkeit in beiden Polen. Dennoch sollte möglichst nie unter Last geschaltet sondern versucht werden, Einfluss auf das Ladegerät oder die Last selbst zu nehmen um die Schalter nicht zu verschleifen.

Dem Schließen der Pluspol-Verbindung sollte eine Verbindung über einen Precharge-Widerstand vorausgehen. Das Verbinden des Pluspols ohne Vorwiderstand wird dann so lange abgewartet, bis der Ausgleichsstrom zwischen Batterie und Last zum Aufladen der dort evtl. vorhandenen Kapazitäten unter einen Schwellwert gesunken ist.

Temperaturen: Lithium-Batterien altern bei geringeren Temperaturen weniger als bei höheren, besitzen jedoch auch einen deutlich erhöhten Innenwiderstand durch die Herabsetzung der Reaktionsgeschwindigkeit der internen chemischen Vorgänge. Bei Überschreiten der Temperaturgrenzen, wie sie vom Hersteller für den sicheren Betrieb angegeben werden, kommt es zur Beschädigung der Zellen, z.B. durch Zersetzung des Elektrolyten. Ein vollständiges thermisches Management besteht aus der Überwachung der Temperaturen an repräsentativen Stellen, sprich den potentiell wärmsten und kältesten

Stellen, und die Möglichkeit der Einflussnahme durch Heizen oder Kühlen. Besonders bei hohen Strömen ist zu beachten, dass je nach Güte des thermischen Kontakts zwischen der Oberfläche einer Zelle ihrem Inneren Temperaturdifferenzen von einigen 10 K entstehen können.

In den allermeisten Fällen wird gegenwärtig auf eine wirklich Kühlung des Packs verzichtet und bei Übertemperatur der Betrieb eingestellt. Bei geringen Temperaturen erwärmen sich die Zellen durch den höheren Innenwiderstand relativ schnell selbst, jedoch sollte dieser Zustand vom Verbraucher beachtet werden. Eine aktive Heizung kann helfen schneller volle Betriebsbereitschaft her zu stellen.

Datenaufzeichnung: Zum Zweck der Wartung und Lebenszeitabschätzung (State of health, SOH) werden die verbrachten Zeiten, ge- und entladene Ladungsmengen [Ah] und Energiemengen [Wh] in jeweils den Zuständen „Leerlauf“ (Open circuit voltage, OCV), „Laden“ und „Entladen“ gespeichert.

Abschätzung des Ladezustands (State of charge, SOC): Die flache Kennlinien lithiumbasierter Akkumulatoren macht die Erfassung des aktuellen Ladezustand des Batteriepacks verhältnismäßig schwierig. Um dies dennoch zu ermöglichen, wird die geladene und entladene Ladungsmenge aufsummiert. Hieraus wird dann zusammen mit der tatsächlichen Spannung und dem Wissen über den SOH eine der SOC in Prozent errechnet. Die Kalibrierung auf 100% findet beim Erreichen der oberen Ladeschlussspannung aller Zellen statt.

Kommunikation: Es ist üblich Informationen über den Zustand der Batterie über verschiedene Protokolle an verbundene Systeme weiter zu geben um den Entlade- und Ladevorgang gezielter steuern zu können. Insbesondere ausschlaggebend für das Ladegerät ist der aktuelle maximale Ladestrom. Dieser ist durch die maximale Einzelzellspannung die derzeit im Pack vorliegt begrenzt.

2.1 Strommessung

Erfolgt im einfachsten Fall durch die Messung des Spannungsabfalls an einem Leistungswiderstand in der Minus-Leitung zwischen Batterie und Last (Shunt). Interessant wäre hier z.B. ein 60 mV Shunt, dessen Spannung über differentielle Eingänge des eines AD-Wandlers mit interner Verstärkung zu messen wäre.

Eine weitere Möglichkeit wäre die Nutzung eines Hallsensors, der eine Messung ohne Verlustleistung ermöglichen würde, jedoch teurer und schwieriger zu beschaffen ist.

2.2 Balancing

Durch Reihenschaltung von Zellen entsteht ein Pack, dessen Einzelzellen kontrolliert geladen werden müssen. Durch Toleranzen in der Fertigung und ungleichmäßige Alterung hat jede Zelle eine vom Nennwert abweichende Kapazität. Um ein Batteriepack möglichst voll laden und für eine gleichmäßige sorgen Alterung müssen die Zellen möglichst jederzeit einen gleichmäßigen Ladezustand haben.

2.2.1 Passives Balancing

Die meist verwendete Methode besteht darin, durch das Parallelschalten eines Leistungswiderstandes zu einer überdurchschnittlich vollen Zelle deren SOC zu verringern. Die Energie wird in Wärme umgesetzt. Dieses Verfahren ist am verbreitetsten, da der schaltungstechnische Aufwand übersichtlich ist und in den seltensten Fällen Defekte an aktiven Komponenten zu erwarten sind.

2.2.2 Aktives Balancing

Hier gibt es verschieden aufwendige und effiziente Verfahren. In jedem Fall laufen sie darauf hinaus, dass die Energie, die beim passiven Balancing in Wärme umgesetzt würde in schwächere bzw. leerere Zellen transferiert wird. Die kann z.B. durch das Multiplexen von Kondensatoren geschehen, durch das Umladen von und in benachbarte Zellen mit Hilfe von geschalteten Induktivitäten und Dioden oder auch durch Nutzung eines gemeinsamen Transformators. In letzterem Fall wird eine Zelle mit hoher Frequenz an die Primärwicklung mit geschaltet und das gesamte Batteriepack über Gleichrichter an der Sekundärwindung geladen.

Diese Verfahren werden hier nicht weiter besprochen.

3 Topologien

3.1.1 Einzelzellen

Zum sicheren Betrieb einer Einzelzelle werden alle Schutzeinrichtungen wie sie auch beim Batteriepack vorgesehen sind eingesetzt. Vorteile sind jedoch, dass durch das Vorhandensein lediglich eines Potentials keine besonderen Maßnahmen zur galvanischen Isolierung getroffen werden müssen. Für größere Leistungen müsste der Strom durch die begrenzte Zellspannung entsprechende Höhen erreichen weswegen meist nur kleinere Verbraucher wie Mobiltelefone an einer Zelle betrieben werden. In diesen Fällen werden die bereits mit einem eingebauten hochintegrierten BMS-Schaltkreis verkauft, sodass selbst beim Betrieb vieler Milliarden Zellen selten Probleme auftreten.

3.1.2 Kleine Batteriepacks

Wo die Größe für kleine Batteriepacks liegt ist nicht genau definiert. Meist liegt sie in dem Bereich der Schutzkleinspannung und der Anzahl Zellen, die sinnvoll mit einem Mikrocontroller zu kontrollieren sind. Aus diesem Grund sind Reihenschaltungen aus drei (Notebook) bis 12 Zellen (Pedelec) die Regel. Aufgrund der relativ kleinen Spannungen wird der Lastabwurf hier mit Leistungshalbleitern realisiert.

3.1.3 Große Batteriepacks

Bei höheren Energiemengen, wie sie zum Beispiel beim Betrieb eines Elektroautos benötigt werden, wird die Spannung des Batteriepacks im Normalfall zwischen 300 und 400V liegen. Besonderes Augenmerk muss bei einem solchen Aufbau auf die Isolationssicherheit und Spannungsfreiheit der Batteriepack-Pole im Fehlerfall gelegt werden. Auch müssen Gleichstromschütze in der Lage sein, die Verbindung zur Batterie bei deutlich mehr als dem nominellen Strom zuverlässig zu trennen.

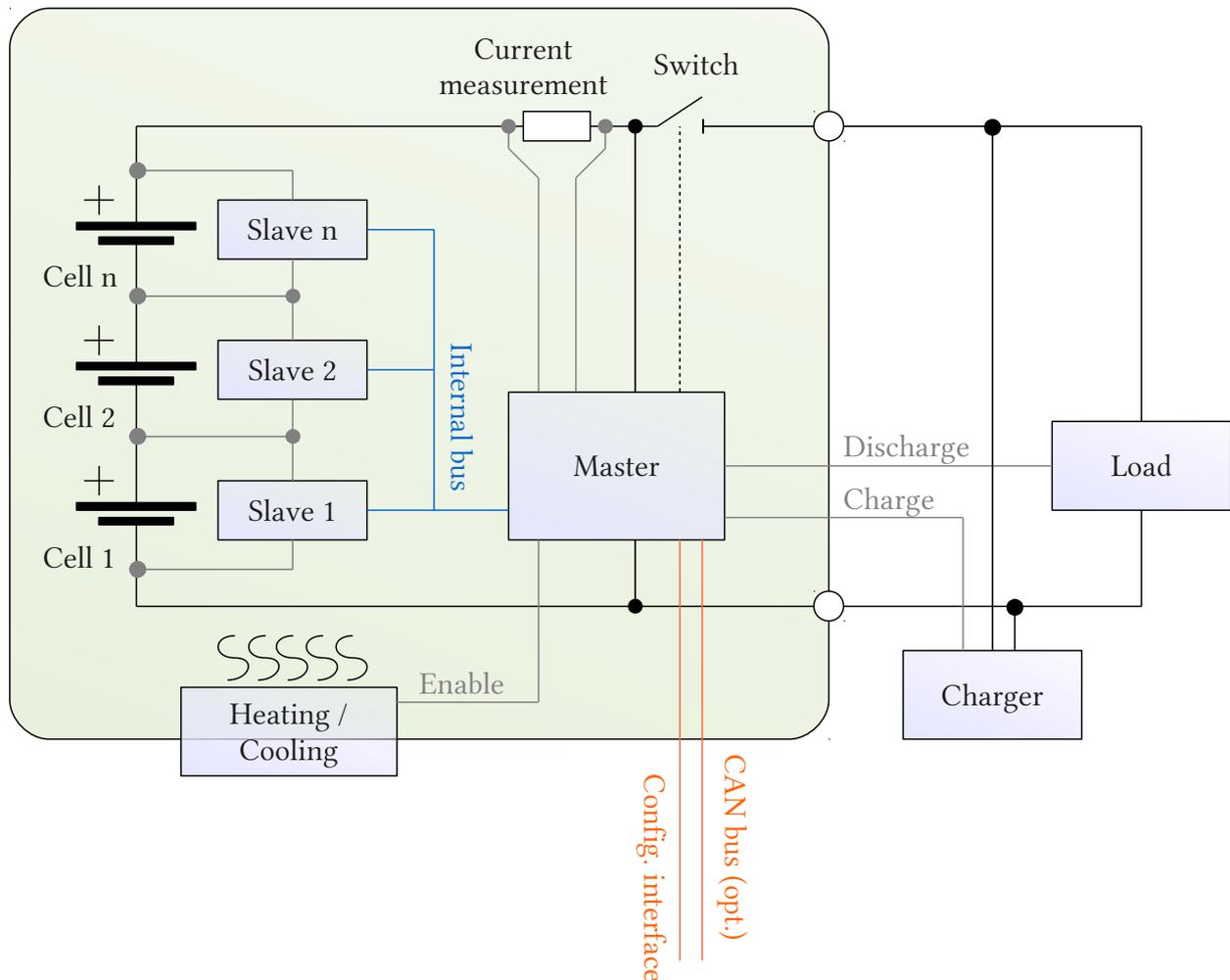


Abbildung 3.1: Schema eines Master-/Slave-BMS

Aufgrund der Größe des Batteriepacks wird hier verteilte Infrastruktur zur Kontrolle eingesetzt wie in Abbildung 3.1 dargestellt. So besteht ein solches System aus einer Anzahl von Slaves, die mit den Einzelzellen verbunden sind, und einem Master. Dieser kommuniziert mit den Slaves über einen BMS-internen Datenbus zur Datenabfrage und Steuerung des Balancierens. Ein Slave kann sowohl für mehrere als auch nur für eine Zelle „zuständig“ sein. Wobei die Kosten für Hardware bei letzterem Aufbau selbstverständlich höher sind. Dennoch ist dieses die elegantere Möglichkeit und auch der Verdrahtungsaufwand wird minimiert, denn diese Einzelzellslaves werden meist mechanisch auf der Zelle selbst montiert. So entfällt das zweifache Verdrahten: ein Mal zur Reihenschaltung der Zellen und ein weiteres mal zum BMS-Slave

3.2 Serielle Schnittstelle am Mikrocontroller

Unterschieden werden muss hier zwischen Controllern mit dezidierter UART-Hardware und Modellen ohne diese. Erschienen jüngst auch ATtiny-Controller mit Hardware-UART-

und ADC, so liegen die Preise in Bereichen, in denen auch z.B. der ATmega48PA liegt welcher besser verfügbar ist (Stand Mai 2012).

Verfügt ein Controller wohl über einen ADC jedoch keinen UART kann dieser per Software emuliert werden. Wichtig hierbei sind definierte Abtastzeiten möglichst ohne größere Jitter. Auch da die internen RC-Oszillatoren über einen weiteren Temperaturbereich nicht übermäßig stabil sind, ist die Baudrate in der Praxis relativ gering um eine korrekte Übertragung sicher stellen zu können. Dazu muss beachtet werden, dass die Kommunikation durch andere Vorgänge auf dem Controller, insbesondere andere Interrupts, beeinträchtigt werden kann. An Hardware benötigt ein Soft-UART mindestens einen Timer mit zwei Output-Compare-Interrupts zur Erzeugung der Abtastintervalle in TX-/RX-Richtung sowie einen externen Interrupt um beim Datenempfang das Startbit erkennen zu können. Besser ist ein Input-Capture-Interrupt mit dessen Hilfe der Abtastzeitpunkt, durch eine Kopie des Wertes den der Timer beim Beginn des Startbits hatte, exakt auf die Mitte der Bitzeit eingestellt werden kann.

4 Master

Der Master in einem BMS übernimmt die Kontrolle und Steuerung der Betriebsparameter sowie Aufzeichnung von Daten und Kommunikation mit anderen verbundenen Systemen.

4.1 Aufgaben

Die Messung der Zellspannungen sowie das Balancieren der Zellen, wie später unter 5.1 beschrieben, wird durch die Slaves übernommen. Die Steuerung jener wiederum muss durch den Master als eine Instanz erfolgen, welche Übersicht über alle aggregierten Daten besitzt. Es müssen also alle nach Kapitel 2 verbleibenden Aufgaben durchgeführt werden.

4.2 Sicherheit

Abwurf der Last bzw. des Ladegerätes erfolgt, wenn

- zulässige Spannungen, Ströme und Temperaturen über-/unterschritten werden,
- mindestens ein Slave einen Fehler meldet oder nicht innerhalb eines bestimmten Zeitraums antwortet oder
- Gehäuseschalter verbaut sind und unterbrochen wurden.

Nach dem Auftreten eines Fehlerzustands und der Behebung dessen darf das BMS keinen weiteren Betrieb, insbesondere kein Wiederanlauf, ohne Quittierung erlauben.

4.3 Datenaufzeichnung

Folgende Daten werden aufgezeichnet und im internen EEPROM des Masters abgelegt:

Data	Type	Unit	Maximum
Startups	uint32	1	4 Bill.
Runtime	uint32	s	$\frac{2^{32}-1}{60\text{ s}\cdot 60\text{ s}\cdot 24\text{ h}\cdot 365,24\text{ d}} \approx 136\text{ years}$
Runtime (charging)	uint32	s	(like above)
Runtime (discharging)	uint32	s	(like above)
Sum charge (charged)	uint64	mAs	$\frac{2^{64}-1}{1000\text{ mA}\cdot 3600\text{ s}} = 5,12\cdot 10^{12}\text{ Ah}$
Sum charge (discharged)	uint64	mAs	(like above)
Sum power (charged)	uint64	mWs	$\frac{2^{64}-1}{1\cdot 10^6\text{ mW}\cdot 3600\text{ s}} = 5,12\cdot 10^9\text{ kWh}$
Sum power (discharged)	uint64	mWs	(like above)
Shutdowns overcurr. (charge)	uint16	1	65535
Shutdowns overcurr. (discharge)	uint16	1	65535
Shutdowns undervoltage	uint16	1	65535
Shutdowns overvoltage	uint16	1	65535
Shutdowns undercooling	uint16	1	65535
Shutdowns overheating	uint16	1	65535

Tabelle 4.1: Datenformate und -aufzeichnung durch Master

4.4 Bestimmung des SOC

TODO

- Beginn der Ladung: Gesamtspannung, Einzelzellspannung (Min, Max), SOC
- Beginn der Entladung: Gesamtspannung, Einzelzellspannung (Min, Max), SOC

Weitere Varianten der SOC-Bestimmung setzen weiteres Wissen über die Eigenschaften der Batterien und deren SOH voraus. Mit deren Hilfe und einem Batteriemodell kann auch während des Betriebs die OCV der Einzelzellen errechnet werden.

4.5 Hardware

Mikrocontroller: ATmega328 mit 5V VCC, 14,7 MHz Baudratenquarz (oder 3,3V VCC und 7,3 MHz Quarz).

Optionen für zukünftige Versionen wären:

- ATmega324P: Sollte der SoftUART zu viel Laufzeit benötigen wäre der Einsatz zu erwägen, da hier zwei Hardware UARTs vorhanden sind.
- ATmega32U4: Wegfall der seriellen Konfigurationsschnittstelle zugunsten eines Hardware USB-Controllers.
- XMega: Deutlich schneller bei 3,3 V und mehr Hardwaremodule um den Rechenkern zu entlasten.
- Einsatz eines deutlich schnelleren Controllers z.B. aus der ARM-Familie: Um die Verwendung von Batteriemodellen zu ermöglichen

Versorgung des internen Busses: 5V über Buck-Coverter aus möglichst allen Zellen in Reihe um sie gleichmäßig zu belasten.

Strommessung: 16 Bit ADC (1,5mA Auflösung bei +- 50A) über ADS1110 oder MCP3425 (über I²C anbinden)

4.5.1 Schnittstellen

- **Interner Bus:** RS-232 mit TTL-Level; RS-485 bei verteilten Slaves
- **Diagnose Bus:** Diagnose und Konfiguration (RS-232 mit MAX3232, RS-485 mit MAX489)
- **Externer Bus:** (optional) CAN (MCP2515 und MCP2551 über SPI anbinden) zur Einbindung in automotive Umgebung oder Anbindung an andere externe Systeme

4.5.2 Eingänge

- Digitaleingänge
 - Ein-/Ausschalter (active low)
 - Gehäuseschalter (bei geschlossenem Gehäuse low)

4.5.3 Interne Ausgänge

- Digitalausgänge
 - Batterie Minuspol verbinden
 - Batterie Pluspol verbinden
 - Batterie precharge verbinden

4.5.4 Ausgänge

- Digitalausgänge
 - Laden möglich/nicht möglich

- Entladen möglich/nicht möglich
- Kühlung ein/aus
- Heizung ein/aus
- Analogausgänge (PWM)
 - SOC (100% duty cycle entsprechen 100% SOC)
 - Spannung (0% entsprechen U_{\min} , 100% U_{\max})
 - Strom (0% entsprechen $-I_{\max}$, 50% 0A, 100% $+I_{\max}$)

4.5.5 Anzeigen

Da kein Display verbaut wird, werden folgende LEDs vorgesehen:

- Normalbetrieb (grün)
- Fehler (rot)

4.6 Stromaufnahme (experimentell)

Das Testobjekt war ein Prototyp mit 10 Slaves, 1000 ms Zykluszeit, 3,3V VCC (MCP1703), 7,4 MHz Quarz und einer aktiven LED (2 mA). „Aktiv“: < 8 mA, „powerdown“: < 250 μ A.

Da der Spannungsregler mit lediglich 2 μ A Leerlaufstrom angegeben ist und Strommessung abgeschaltet wird wäre es interessant zu wissen, wo genau die Leckströme in dieser Größenordnung auftreten.

4.7 Software

Die Qualität der Firmware des Masters ist ausschlaggebend für die Betriebssicherheit des ganzen Batteriepacks. Dabei dabei wird eine Hauptschleife nach Abb. X durchlaufen.

TODO: Abbildung X

Nach dem Laden der Konfiguration und der Zähler beginnt der Master alle erforderlichen Daten für einen Durchlauf zu aggregieren. Dies betrifft die Zellspannungen, -Temperaturen, den Strom und alle weiteren Eingänge. Auch Konfigurationsnachrichten werden in diesem Zyklus empfangen und bearbeitet. Da es in der Realität zu Verzögerungen bei der Abfrage der Slaves kommen kann, trägt die tatsächliche Implementierung den veränderten Verhältnissen dadurch Rechnung, dass in je Schleifendurchlauf lediglich ein Slave abgefragt wird. Wurden alle konfigurierten Slaves ein mal abgefragt, wird das Balancing neu eingestellt als auch die Berechnung von Mittelwerten, SOC, etc ausgeführt. Die Erkennung von zu hohen Strömen findet während jedes Durchlaufs statt.

Die Verbindung der Batterie zur Last oder dem Ladegerät wird über den entsprechenden Status, wie in Kapitel 4.7.2 beschrieben, gesteuert. Dabei wird der Pluspol erst nach einer konfigurierten Mindestverzögerung und dem Unterschreiten einer eingestellten Stromgrenze verbunden.

4.7.1 Masterstatus

Value	State
0x00	Init
0x01	Active
0x02	PowerdownActive
0x03	Error
0x04	PowerdownError
0x05	ErrorAcknowledged

Tabelle 4.2: Masterstatus

4.7.2 Batterieverbindungsstatus

Value	State
0x0	DisconnectRequest
0x1	Disconnected
0x2	ConnectRequest
0x3	Precharging (minus & precharge connected)
0x4	Connected (minus & plus connected)

Tabelle 4.3: Master Batterieverbindungsstatus

4.7.3 Slaveauslesezyklus-Status

Value	State
0x0	(Re-)Start
0x1	Busy
0x2	Done (after reading out last slave)

5 Slave

Ein Slave überwacht lediglich eine einzelne Zelle, um die sonst nötige galvanische Entkopplung der Kanäle des ADCs zu vermeiden. Ebenfalls wird auf diese Weise die maximal mögliche Flexibilität bezüglich der Ausbaumöglichkeiten und der Größe des Batteriepacks erhalten. Durch die Herstellung der Slaves in großer Zahl vergünstigt sich deren Herstellung.

Die Nummerierung beginnt mit 1 am Plus-Pol des Packs. Diese Nummer entspricht gleichzeitig der Adresse über die der Slave auf dem internen Bus angesprochen wird.

Ein Slave ist so unselbstständig wie möglich und führt lediglich durch, was der Master befiehlt. Dies gilt für die Kommunikation ebenso wie für das Balancing. Der Ausgleichswiderstand wird nach dem Ablauf eines softwareseitig implementierten Watchdogs, der mit dem Doppelten der Zykluszeit aufgezogen wird, automatisch deaktiviert. Die Zykluszeit stellt der Slave selbst dadurch fest, welche Zeit zwischen zwei Anfragen vergeht. So wird ein unkontrolliertes Ausgleichen der Zellen verhindert.

5.1 Aufgaben

- Messung: Zellspannung, -temperatur
- Steuerung: Passives Balancing durch Parallelschaltung eines Widerstandes

5.2 Hardware

5.2.1 Controller

Sehr günstige und physisch kleine Modelle wie der ATtiny13A machen eine robuste Implementierung unter Beachtung der unter 3.2 aufgezeigten Anforderungen schwer. Aus diesem Grund wurde sich für einen ATtiny44A als Mikrocontroller für den Slave entschieden, der alle Hardwarevoraussetzungen erfüllt.

5.2.1.1 Takterzeugung

Eine asynchrone serielle Schnittstelle, ob hardware- oder softwarebasiert, benötigt einen möglichst konstanten Takt.

Der interne RC-Oszillator von AVR hat (Stand: Mai 2012) ein Drift von 6% über die Versorgungsspannung und hauptsächlich Temperatur (+-8% von -40 bis +80 °C). Eine mögliche Kalibrierung bezieht sich immer auf eine bestimmte Versorgungsspannung und Temperatur. Dadurch ist er für eine funktionssichere Implementierung ungeeignet.

Alternativ bietet sich die Nutzung eines Quarzes an. Wird dieses direkt an den Controller angeschlossen, so werden zwei Pins benötigt. Von einem Quarz ist ab zu sehen, da die Stromaufnahme verhältnismäßig hoch ist und sie auf mechanische Beanspruchungen wie Vibrationen empfindlich reagieren. Durch die hohe Güte ist die Einschwingdauer eines Quarzoszillators mit etwa 10 ms relativ lang.

Ein Keramikresonator ist mit etwa 0,5% initialer Abweichung für eine asynchrone Datenübertragung ausreichend genug. Ebenfalls schwingt er mit einer Millisekunde schnell an und steht kurz nach dem Aufwach-Signal sicher zur Verfügung.

5.2.1.2 Anforderungen für passives Balancing

Es werden 7 bis 8 Pins benötigt:

- 2 Pins: Versorgungsspannung (VCC, GND)
- 3 Pins: Messung der Zellspannung und Temperatur, Fet-Schalter für die Spannungsteiler (2 Pins bei Nutzung nach 5.2.3.2)
- 1 Pin: Balancing aktivieren
- 2 Pins: Serielle Schnittstelle (RX, TX)
- optional: 2 Pins für Quarz

Soll ein Quarz/Resonator für die Takterzeugung angeschlossen werden 10 Pins benötigt.

5.2.1.3 Anforderungen für aktives Balancing

Es werden 9 Pins benötigt:

- 2 Pins: Versorgungsspannung (VCC, GND)
- 3 Pins: Messung der Zellspannung und Temperatur, Fet-Schalter für die Spannungsteiler
- 2 Pin: PWM (mind. 50KHz) für Balancing (Verschieben von Energie in die obere Zelle oder von der oberen in die „eigene“)
- 2 Pins: Serielle Schnittstelle (RX, TX)
- optional: 2 Pins für Quarz

Soll ein Quarz/Resonator für die Takterzeugung angeschlossen werden 11 Pins benötigt.

Bei der Nutzung von aktivem Balancing und eines Soft-UARTs sollte der Controller mindestens 2 unabhängige Timer bieten. Die Auflösung der PWM ist aufgrund der geringen Taktfrequenz des Controllers und der minimal gewünschten PWM-Frequenz von 50kHz begrenzt.

$$F_{CPU} = 1 \text{ MHz}; F_{PWM} = \frac{F_{CPU}}{2^4} = 62,5 \text{ kHz (bei 4 Bit-PWM)}$$

5.2.1.4 Mögliche AVR-Controller

Typ	Pins	U _{Ref} [V]	Hardware	I [mA] (1 MHz, 3,3V)	Preis (18.05.2012)
AtTiny13A	8	1,1	ADC PWM	0,045 idle 0,38 active (1 MHz)	0,60€ (-DIP, guloshop) 0,99€ (-SU, Reichelt) 1,25€ (-SSU, CSD) 1,52€ (-SU, ab 10, Buerkl.)
AtTiny44A	14	1,1	ADC XTAL PWM	0,055 idle 0,42 active	1,00€ (-DIP, 84er, guloshop) 1,25€ (-SSU, Reichelt) 1,49€ (-SSU, CSD)
AtMega48PA	32	1,1	ADC XTAL PWM UART	0,06 idle 0,42 active	2,00€ (-DIP, 328er, golus.) 2,29€ (88er, CSD) 3,08€ (168er, ab 10, Buerkl.)

Tabelle 5.1: Übersicht möglicher Controller für einen Einzelzellslave

5.2.2 Referenz

Die internen Referenzen der 8-Bit-AVRs unterliegen Fertigungstoleranzen, die Atmel mit +-10% angibt. Man kann jedoch die Referenzspannung ausmessen, den realen Wert im EEPROM speichern und so eine ausreichend gute Messung realisieren. Das Ausmessen kann beispielsweise über eine Testfirmware funktionieren, die eine LED einschaltet sobald der zwei höchste ADC-Wert gemessen wird. Den mit Hilfe eines präzisen Multimeters gemessenen Spannungswert am ADC-Eingang in Millivolt kann man mit Hilfe einer eigens hierfür kompilierten Software in das EEPROM schreiben. Eine Übertragung per Soft-UART wäre auch denkbar.

5.2.3 Spannungsmessung

Während der Durchführung der Messung wird das möglicherweise aktivierte Balancing ausgesetzt um mögliche Spannungsabfälle auf dem Zuleitungen zum Slave zu verhindern.

5.2.3.1 Geschalteter Spannungsteiler

Bestehend aus 0,1% Widerständen, Messung gegen interne 1,1V Referenz:

$$U_{max} = 1,1 \text{ V} \cdot \frac{2,7\text{k} + 8,2\text{k}}{2,7\text{k}} = 4,441 \text{ V}, U_{Bit} = \frac{U_{max}}{2^{10}} = 4,34 \text{ mV (bei 10 Bit)}$$

Vorteil hierbei ist die gleichbleibende, recht gute Auflösung über den gesamten Messbereich. Man benötigt mindestens zwei Port-Pins (MOSFET, ADC).

Der Einfluss des Bahnwiderstandes eines Logic-Level-MOSFETs auf die gemessene Spannung liegt selbst bei geringen Zellspannungen unter 0,05% und ist damit vernachlässigbar.

5.2.3.2 Interne Referenz gegen VCC

So der Mikrocontroller die Möglichkeit bietet die Versorgungsspannung als Referenz und die interne Referenz als Eingang für den ADC zu wählen kann man ohne externe Bauteile auskommen und benötigt mindestens eine Port-Pin weniger als mit dem Spannungsteiler nach 5.2.3.1. Zusätzlich kann man dazu den Messfehler des Spannungsteilers umgehen:

$$VCC = \frac{2^{10}}{ADC} \cdot V_{Ref} \quad (\text{bei 10 Bit})$$

Nachteil hierbei ist jedoch, dass die Auflösung der Messung abhängig von der tatsächlichen Versorgungsspannung ist, siehe Abbildung 5.1.

$$U_{Bit} = \frac{VCC^2}{2^{10} \cdot V_{Ref}} \quad (\text{bei 10 Bit})$$

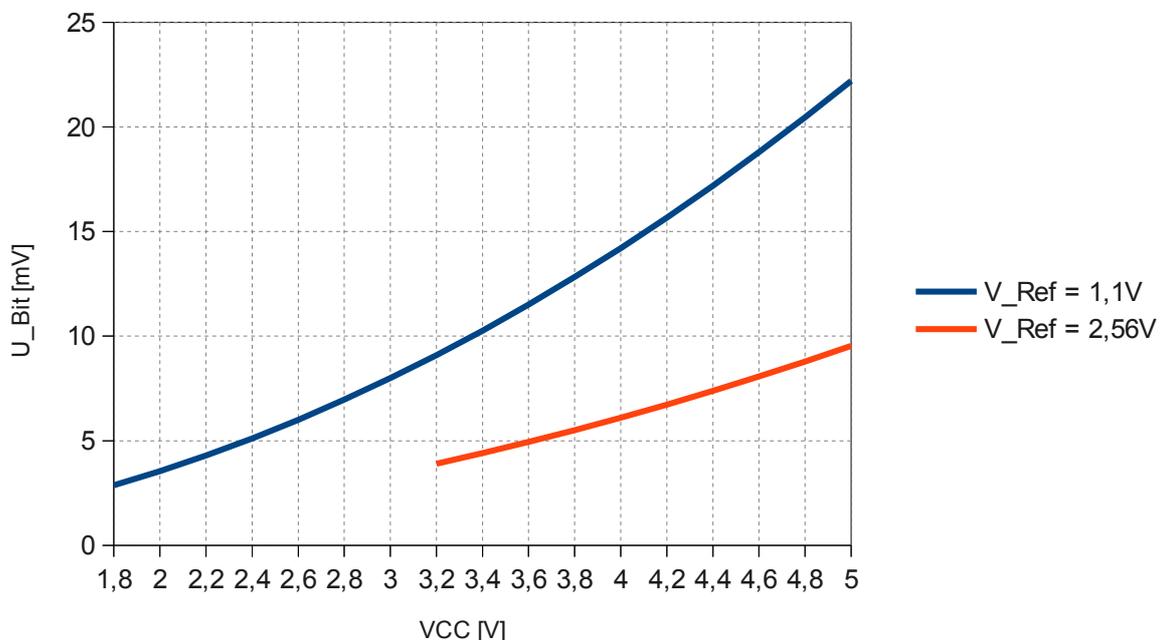


Abbildung 5.1: Auflösung der VCC-Messung gegen interne Referenz (10 Bit-ADC)

Die bedeutet, dass beim Balancieren im oberen Bereich der Zellspannung die Auflösung, dort wo sie wichtig ist, verhältnismäßig schlecht ist. Controller wie AtTin45V können jedoch zwischen zwei Referenzen wählen, so dass bei höherer Spannung auch eine größere Referenz gewählt werden kann.

5.2.4 Temperaturmessung

Die Messung der Zelltemperatur findet über einen Heiß- oder Kaltleiter in Verbindung mit dem ADC statt. Dieser Spannungsteiler sollte, wie unter 5.2.3.1 beschrieben, ebenfalls geschaltet werden um den Ruhestrom zu minimieren.

5.2.4.1 KTY81-210

Für einen KTY81-210 wäre eine konstante Spannung, ein Vorwiderstand von 4,7k und eine ADC-Referenz mit 2,5V ideal. Diese Beschaltung ergäbe über weite Temperaturbereiche eine nahezu lineare Kennlinie, so dass eine Messung ohne hohen Aufwand möglich ist.

Durch die schwankende Spannung am Spannungsteiler muss der gemessene Wert jedoch immer unterschiedlich interpretiert werden. Da der KTY81210 einen positiven Temperaturkoeffizienten besitzt muss darauf geachtet werden, dass bei hoher Temperatur die Referenzspannung nicht überschritten wird um den gemessenen Wert korrekt interpretieren zu können. Diese Abhängigkeit ist in Abbildung 5.2 in Verbindung mit einem 12kOhm Vorwiderstand dargestellt:

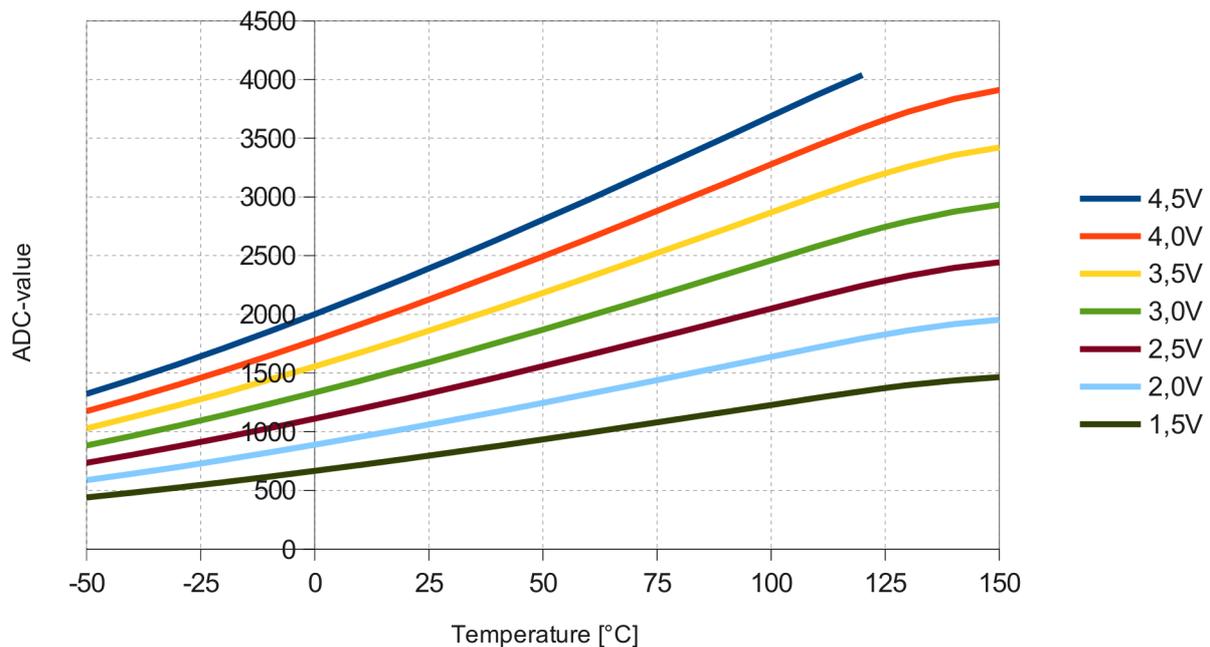


Abbildung 5.2: Abhängigkeit des ADC-Wertes von der Zellspannung

Da die Verschiebung der Kennlinie von der Zellspannung linear abhängig ist, lässt sich mit einer einzigen hinterlegten Kennlinie und dem Wissen, bei welcher Spannung der aktuelle Wert gemessen wurde, die aktuelle Temperatur bestimmen:

$$ADC_{ref} = \frac{ADC_{actual} \cdot U_{ref}}{U_{actual}}$$

Da diese Skalierung eine Multiplikation enthält kann man den Berechnungsvorgang erheblich beschleunigen, so man als Referenzspannung in Millivolt eine Potenz von zwei wählt. Hier bieten sich 4096mV an. Somit lässt sich die Multiplikation als Links-Shift um 12 Bits ausführen. Darüber hinaus umfasst die Kennlinie für 4,096V den gesamten Temperaturbereich des Sensors und bietet die größte Messwert-Dynamik. Da die mit dem ADC gemessenen Werte bei kleinerer Zellspannung ebenfalls kleiner werden verringert sich die Temporauflösung bei geringeren Zellspannungen etwas.

5.2.5 Stromversorgung

Die Versorgung des Controllers des Einzelzell-Slaves findet direkt mit der Zellspannung der statt. Das bedeutet, dass die gesamte Schaltung für den Direktbetrieb der Spannungen von entladenen LiFePO₄- und geladenen LiNiCoO₂-Zellen ausgelegt sein muss: 2,0V bis 4,2V plus eine Reserve nach unten und oben. Da die Versorgung des Mikrocontrollers nicht von der Zelle getrennt werden kann, darf dieser nur einen sehr geringen Ruhestrom aufnehmen.

Die Stromversorgung vorhandener in Richtung Master findet über durch selbigen mit +5V statt.

5.2.6 Stromaufnahme (experimentell)

2 V: < 130 μA /~ 550 μA (powerdown/active)

4,2 V: < 160 μA /~ 1150 μA (powerdown/active)

Ähnlich wie beim Master sollte hier noch etwas Forschung betrieben werden, wohin der Strom genau fließt. Der Mikrocontroller sollte nach 10 Sekunden weniger als 130 nA aufnehmen.

5.3 Software

TODO

5.3.1 Laden der Konfiguration

Werte wie die Adresse des Slaves sowie die Kalibrierungsdaten werden im EEPROM gespeichert und beim Starten des Mikrocontrollers von dort geladen. Der Ablauf dessen ist in Abbildung 5.3 dargestellt.

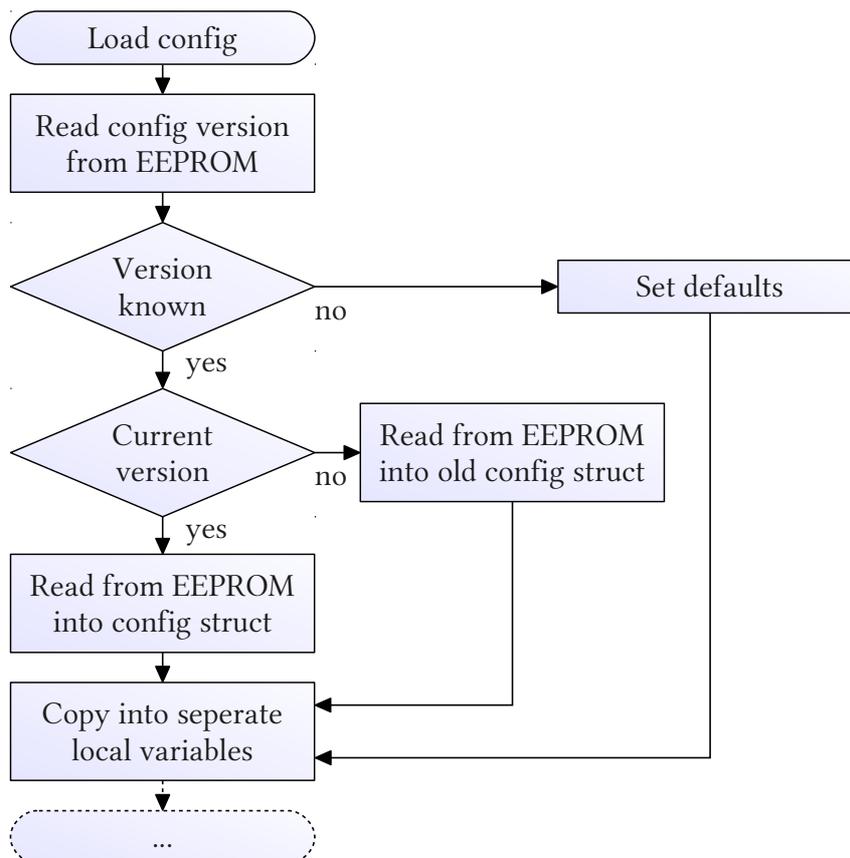


Abbildung 5.3: Flussdiagramm des Ladens der Konfiguration

5.3.2 Hauptschleife

Das Ziel der Software ist die zeitnahe und korrekte Übermittlung der angeforderten Messwerte bei möglichst geringer Leistungsaufnahme. Der Master weckt die Slaves zyklisch auf um Daten ab zu fragen. Um auch bei einem Verbindungsverlust bzw. Ausschalten des Masters sicher zu stellen, dass keine Zelle übermäßig entladen wird, wird vor dem Eintritt in den Energiesparmodus bei aktiviertem Balancing der Watchdog-Timer zum selbstständigen Aufwachen nach spätestens 8 Sekunden verwendet. Wird der Controller durch diesen aufgeweckt, wie in Fehler: Referenz nicht gefunden dargestellt, so wird

lediglich das Balancing deaktiviert und direkt wieder in den Energiesparmodus gewechselt. Dies ist der Zustand der minimalen Stromaufnahme des ganzen Slaves.

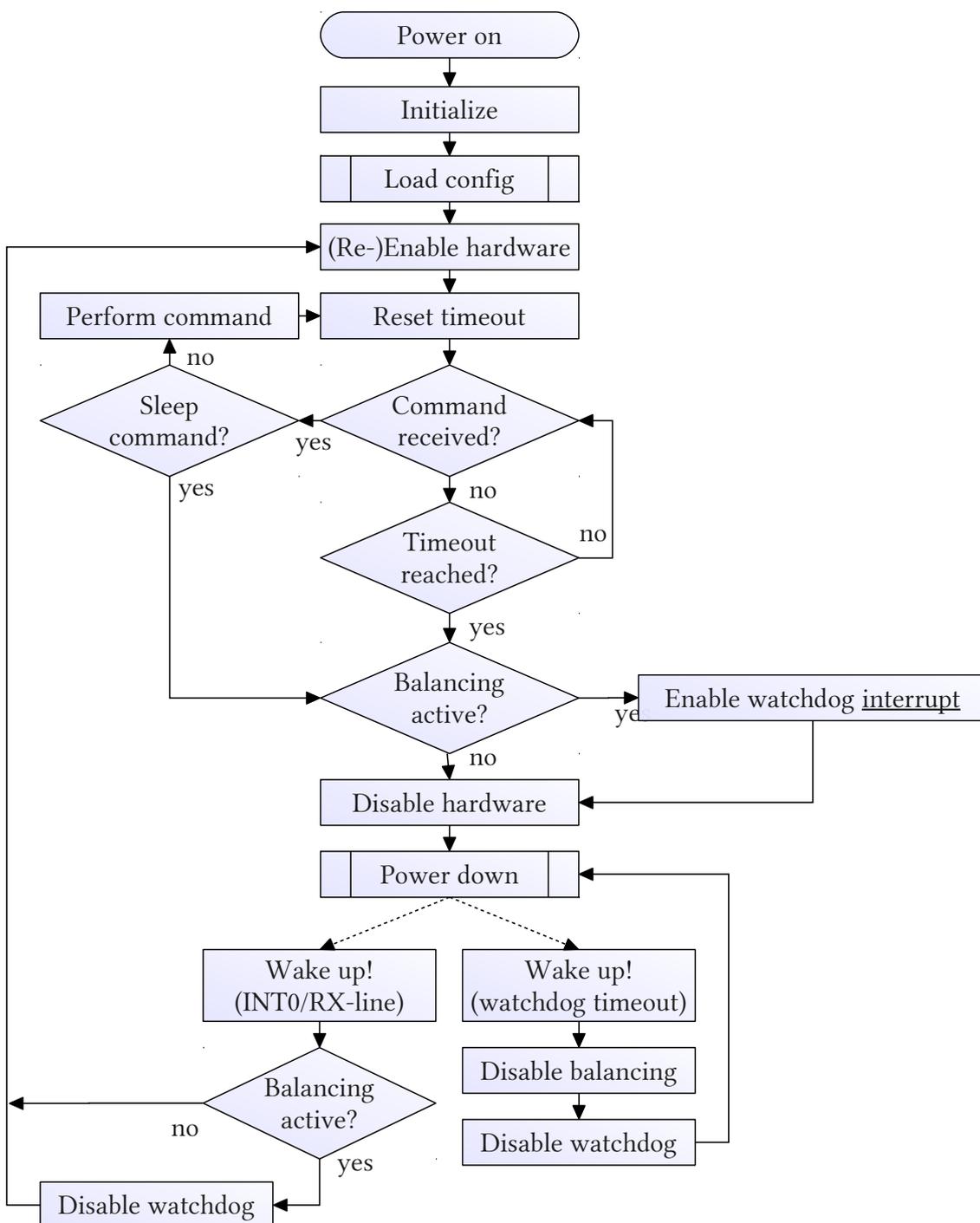


Abbildung 5.4: Flussdiagramm der Hauptschleife

5.3.3 Sleepmode

Nach der erstmaligen Verbindung mit einer Stromversorgung legt sich der Controller schlafen. Aus dem Modus „Power-Down“ kann er nur durch ein asynchrones Signal, wie

den INT0 oder den Watchdog, aufwachen. Der externe Interrupt 0 kann in diesem Schlafmodus ohne Takt keine Flanken detektieren und muss vor dem Schlafen auf „level“-Modus umkonfiguriert werden.

5.3.4 Aufwachen

Da der RX-Pegel des Slaves immer durch einen Pull-up auf high-Pegel liegt, schläft der Controller so lange bis der Master eingeschaltet wird und durch das „Herunterziehen“ des – von ihm aus TX-Pegels – auf low die mit dem Datenbus verbundenen Slaves aufweckt. Eine Möglichkeit dies um zu setzen ist das Senden eines Bytes mit dem Wert 0x00, welches einen low-Pegel definierter Länge erzeugt.

Die nötige Verzögerung zur Stabilisierung des Controllertaktes hängt von dessen Quelle ab und muss bei AVR Controllern in de Fuse-Bits definiert werden (Keramikkresonator ~1ms, Quarz ~10ms).

5.3.5 Messung

Sofort nach dem Aufwachen beginnt der Controller mit der Messung aller Werte und speichert diese bis sie versendet sind. Da alle Controller zum gleichen Zeitpunkt das Signal zum Aufwachen erhalten, werden alle Messungen gleichzeitig durchgeführt und es wird ein in sich konsistentes Abbild des Zustandes des Batteriepacks erfasst.

5.3.6 Datenempfang

Nachdem die Takte der Slaves sich stabilisiert haben und sie bereit für den Daten Empfang sind sendet der Master an jeden Slave Nachricht mit dem Balancing-Status (und optional anderen Daten). Der jeweilige Slave antwortet mit seinen Daten.

6 Master-Slave Bus

Für den Datenbus wird ein asynchroner serieller Bus verwendet. Dieser kann, je nach physikalischem Aufbau und Anzahl der Slaves, mit TTL-Pegel oder als RS485-Bus ausgeführt werden. Je Zeichen werden 8 Datenbits, keine Parität, ein Stop-Bit sowie ein Idle-Bit, also 11 Baud, übertragen (8N1).

Kommunikation wird immer durch den Master initiiert während Slaves lediglich in Richtung des Masters antworten.

6.1 Adressen

Die Slaves werden mit den Adressen 0x01 bis 0xfd angesprochen.

Die Adresse 0xff ist die Broadcast-Adresse. Nachrichten an diese Adresse werden von allen Slaves ausgewertet und im Normalfall nicht beantwortet.

Eine weitere Ausnahme stellt die Standardadresse nach 6.4 dar.

Antwortet ein Slave dem Master, so verwenden sie als Zieladresse 0x00.

6.2 Paketaufbau

Übertragene Daten werden mit der Struktur in Tabelle 7.1 übertragen.

Adresse	Kommando	Länge	Daten
1 Byte	1 Byte		n Bytes
8 Bit	4 Bit	4 Bit	8*n Bit

Tabelle 6.1: Aufbau eines Datenpakets

Adresse: Bestimmt den Empfänger des Datenpaketes.

Kommando: Definiert, welche Reaktion vom Empfänger erwartet wird. Dies kann z.B. die Übertragung von Daten und/oder das Herstellen von bestimmter Hardwarezustände beinhalten.

Länge: Die Anzahl der Bytes im Datenbereich.

Daten: Zu transportierende Nutzdaten. Ist die Länge zuvor mit Null angegeben entfällt dieses Feld.

6.3 Übertragungsformat

Die Byte-Reihenfolge ist „Big Endian“. Das bedeutet, dass bei Werten die mehr als ein Byte zu Repräsentation benötigen das Byte mit der höchsten Wertigkeit zuerst übertragen wird.

Die Ungültigkeit eines Wertes wird durch die Übertragung des maximal darstellbaren Wertes signalisiert.

6.3.1 Zellspannung

Übertragung findet als vorzeichenlose Ganzzahl mit 16 Bit statt. Der Wertebereich erstreckt sich von 0x0000 bis 0xffff, entsprechend 0 bis 65534mV mit einer Auflösung von 1mV.

6.3.2 Zelltemperatur

Übertragung findet als vorzeichenlose Ganzzahl mit 16 Bit statt. Der Wertebereich erstreckt sich von 0x0000 bis 0xffff, entsprechend 0°K bis 655,34°K (382,19°C) mit einer Auflösung von 0,01°K.

6.4 Standardwerte und Kalibrierung

Die Standardadresse eines frisch programmierten und nicht kalibrierten Slaves ist 0xfe. Dies erkennt die Firmware daran, dass im EEPROM des Controllers keine Konfiguration vorliegt. Somit kann, solange lediglich ein einziger unkonfigurierter Slave mit dem Datenbus verbunden ist, eine Adressvergabe durch den Master erfolgen. Erfolgt die Verbindung der Slaves sukzessiv so können alle Slaves mit identischer Firmware programmiert werden und im System durch den Master durchnummeriert und konfiguriert werden

Die Messung von Werten durch den Mikrocontroller unterliegt einer Vielzahl von linearen und nichtlinearen Einflüssen. Um alle fertigungsbedingten Toleranzen auszugleichen wird nach der Adressvergabe eine Kalibrierung der Messung der Versorgungsspannung durch den AD-Wandler durchgeführt. Hierfür wird der Slave mit einer genau bekannten Spannung versorgt und der aktuelle Messwert des AD-Wandlers in Kombination in Kombination mit dem bekannten Spannungswert im EEPROM gespeichert. Dieses Verfahren wird ein Mal für eine Spannung leicht über der niedrigsten erwarteten Spannung und für eine Spannung leicht unter der höchsten erwarteten Spannung durchgeführt. In Zukunft kann dann die wirklich anliegende Spannung interpoliert werden. Dieses Verfahren korrigiert jedoch keine nichtlinearen Fehler.

6.5 Baudrate

Wird (optimistisch) von einer 50% Auslastung des Datenbusses ausgegangen, bedingt durch Verzögerungen durch Verarbeitung empfangener und Vorbereitung zu sendener Daten, und ist die Anzahl der Slaves bekannt, so kann berechnet werden, welche minimale Baudrate benötigt wird. Hierzu muss ein vollständiger Abfragezyklus innerhalb eines gegebenen Abfrageintervalls betrachtet werden:

- Aufwachen: max. 10 ms
- je Slave:
 - Datenanfrage und Balance-Status: 3 Bytes
 - Übertragung der Daten: 6 Bytes
- Schlafen: 2 Bytes

Im Fall eines Software-UART kann eine höhere Baudrate durch Erhöhung der Taktfrequenz des Controllers erreicht werden. Zwar steigt die Stromaufnahme ebenfalls an, jedoch kann im Gegenzug eine längere Zeit im Stromsparmmodus, also gänzlich ohne Takt, verblieben werden.

6.5.1 Berechnungsvorschriften

$$\text{Datenvolumen} = (\text{Anzahl Slaves} \cdot (3+6) + 2) \text{ Bytes} \cdot \frac{100\%}{\text{Buslast}} \cdot \frac{1}{f}$$

$$\text{Baudrate}_{\min} = \text{Datenvolumen} \cdot \text{Baud/Byte} \cdot \frac{1000 \text{ ms}}{1000 \text{ ms} - \text{Aufwachverzögerung} \cdot \frac{1}{f}}$$

6.5.2 Beispiele

Im die Berechnung für zwei Anwendungsfälle eines kleinen und großen Batteriepacks. In beiden Fällen wird die Verzögerung durch das Aufwachen aus dem Energiesparmodus mit 10 ms angenommen.

- 12 Slaves, 50% Buslast, 1 Hz → 220 Bytes/s → min. 2444 Baud/s → 4800 Baud/s
- 100 Slaves, 50% Buslast, 0,5 Hz → 902 Bytes → min. 10022 Baud/s → 19200 Baud/s

6.6 Kommandos

Auf dem internen Bus werden die Anforderungen von Daten von den Daten durch die Zieladresse und die Anzahl der Datenbytes unterschieden.

Kommando	Richtung	Name	Länge	Daten
0x00	→ S	SlaveInfo	0	-
0x00	→ M	SlaveInfo	5	uint8: Address uint8: Model uint8: Firmware version uint16: Internal temperature
0x01	→ S	CellInfo	1	uint8: 0: Balance 1-7: reserved
0x01	→ M	CellInfo	4	uint16: Cell voltage [mV] uint16: Cell temperature
0x02	→ S	AdcMinMax	0	-
0x02	→ M	AdcMinMax	4	uint16: Adc minimum uint16: Adc maximum
0x03	→ S	AdcActual	0	-
0x03	→ M	AdcActual	6	uint16: Adc internal temperature uint16: Adc cell voltage uint16: Adc cell temperature
0x08	→ S	SetAddress	1	uint8: Address
0x09	→ S	CalibrationVoltage	0	-
0x09	→ S	CalibrationVoltage	8	uint16: Adc low uint16: Voltage low [mV] uint16: Adc high uint16: Voltage high [mV]
0x09	→ M	CalibrationVoltage	8	uint16: Adc low uint16: Voltage low [mV] uint16: Adc high uint16: Voltage high [mV]
0x0b	M→S	Measure	0	-
0x0c	M→S	Sleep	0	-
0x0f	S→M	Result	1	uint8: Result

Tabelle 6.2: Übersicht BMS-interner Kommandos

6.6.1 Measure

Anweisung einen Messvorgang durch zu führen. Um eine schnelle Messung zu ermöglichen sollte vom Master muss eine Verzögerung einkalkuliert werden, in der die Slaves keine Nachrichten beantwortet. Diese Nachricht muss *nicht* zur Datenabfrage nach dem Aufwecken der Slaves gesendet werden, da hier selbstständig eine Messung gestartet wird.

6.6.2 Sleep

Anweisung sofort in den Energiesparmodus einzutreten.

6.6.3 Result

Result-Nachrichten werden als Antwort auf Kommandos, die Parameter verändern, oder als Fehlermeldung bei Nichtlieferbarkeit angeforderter Daten versendet.

Code	Name
0x00	Ok
0x01	Unknown command
0x02	Parameter out of range
0x04	Wrong message format/length
0xff	Unknown error

Tabelle 6.3: Übersicht der Result-Codes auf internem Bus

7 Konfigurationsschnittstelle (seriell)

Zur Konfiguration wird ein asynchroner serieller Bus verwendet. Je Zeichen werden 8 Datenbits, keine Parität, ein Stop-Bit sowie ein Idle-Bit, also 11 Baud, übertragen (8N1). Die Baudrate beträgt 38400.

7.1 Nachrichtenformat

Diese Definition ist stark angelehnt an das Protokoll des CAN-Bus um eventuelle später Buserweiterungen ohne umfangreiche Neudefinitionen durchführen zu können.

Übertragene Daten werden mit folgender Struktur übertragen:

- Head: Byte 0 – 1:
 - Id (11 Bit)
 - RTR (1 Bit)
 - DLC (4 Bit)
- Body: Byte 2 – (n-2):
 - Data (0 – 64 Bit)
- Foot: Byte (n-1) – n:
 - CRC (16 Bit)

Am Paketende wird ein CRC16 (Generatorpolynom: $x^{16} + x^{15} + x^2 + 1$ (0xa001), Startwert 0xffff) angefügt um die Integrität der Kommunikation zu sichern. Der CRC schließt alle Bytes der Nachricht ein.

7.2 Nachrichten

In Tabelle 7.1 sind alle verwendeten Nachrichten Ids dargestellt. Nachrichten, die Daten anfordern werden mit gesetztem RTR-Bit und im Normalfall ohne Daten gesendet. Ist für die korrekte Erfüllung der Anfrage ein Parameter notwendig, z.B. die Zellnummer, so wird lediglich dieser Parameter übertragen und ist in hier mit einem ^R versehen. In der Regel befinden sich diese Datenfelder am Anfang der Nutzdaten, so dass die darauf folgende Daten nicht gefüllt werden müssen (kein Padding).

ID	Name	Length	Data
0x000	MasterInfo	4	uint8: Model uint8: Firmware version uint16: Internal temperature
0x001	SetMasterConfigMode (→ 7.3)	5	uint8: Level uint32: Passcode

7 Konfigurationsschnittstelle (seriell)

ID	Name	Length	Data
0x002	SetMasterConfigModePasscode (→ 7.3)	4	uint8: Level uint32: Passcode
0x003	MasterAdcIntMinMax	4	uint16: Min. ADC value uint16: Max. ADC value
0x004	MasterAdcIntTemperature	2	uint16: Actual ADC value
0x005	MasterAdcCurrentMinMax	8	int32: Min. ADC value int32: Max. ADC value
0x006	MasterAdcCurrent	4	int32: Actual ADC value
0x007	MasterDigitalInputs (→ 4.5.2)	1	uint8: 0: On (active low) 1: SecuritySwitch (active low) 2-7: Reserved
0x008	MasterDigitalOutputs (→ 4.5.4)	1	uint16: 0: StatusOn 1: StatusWarning 2: StatusError 3: (reserved) 4: BatteryMinus 5: BatteryPlus 6: BatteryPrecharge 7: (reserved) 8: ChargePossible 9: DischargePossible 10,11: (reserved) 12: Cooling 13: Heating 14,15: (reserved)
0x009	SetMasterDigitalOutputs		reserved
0x00a	MasterAnalogInputs		reserved
0x00b	MasterAnalogOutputs (→ 4.5.4)	3	uint8: SOC (duty cycle) uint8: Voltage (duty cycle) uint8: Current (duty cycle)
0x00c	SetMasterAnalogOutputs		reserved
0x00d	MasterPerformance	2	uint16: Cycle time [ms]
0x00e	MasterStatusWarningError	8	uint8: Status master uint8: 0-3: Status battery connection 4-7: Status slave readout cycle uint16: Warning 0: Cell voltage low 1: Cell voltage high

7 Konfigurationsschnittstelle (seriell)

ID	Name	Length	Data
			2: Cell voltage difference 3: Cell temperature low 4: Cell temperature high 5: Cell temperature difference 6: High charge current 7: High discharge current 8: SOC low 9: SOH low 10: Battery switch shutdowns 11-15: reserved uint16: Actual error 0: Cell voltage low 1: Cell voltage high 2: Cell voltage difference 3: Cell temperature low 4: Cell temperature high 5: Cell temperature difference 6: Overcurrent charge 7: Overcurrent discharge 8: SOC low 9: SOH low 10: Battery switch shutdowns 11: Slave communication 12: Slave temperature sensor 13: Security switch (reserved) 14: Isolation (reserved) 15: Other (reserved) uint16: Shutdown error (like actual error)
0x00f	MasterAcknowledge	1	uint8: 0: Error 1-7: Reserved
0x010	PackInfo	8	uint16: Manufacturer uint16: Model uint32: Serialnumber
0x011	SetPackInfo	8	(see above)
0x012	PackConfig	6	uint8: Cells in series uint8: Cell in parallel uint32: (Single) cell capacity [mAh]
0x013	SetPackConfig	6	(see above)
0x014	PackConnectionConfig	6	uint16_t: Precharge delay [ms] int16_t: Precharge release current [mA * 16]

7 Konfigurationsschnittstelle (seriell)

ID	Name	Length	Data
			int16_t: Battery switch zero current threshold [mA * 16]
0x015	SetPackConnectionConfig	6	(see above)
0x016	PackLimits	8	uint8_t: SOC low warning [% * 2] uint8_t: SOC low shutdown [% * 2] uint8_t: SOH low warning [% * 2] uint8_t: SOH low shutdown [% * 2] uint16_t: Battery switch shutdowns warning uint16_t: Battery switch shutdowns error
0x017	SetPackLimits	8	(see above)
0x018	BalancingConfig (→ 7.4)	8	uint8_t: 0: Enabled 1: RestrictSocRange 2: RestrictCurrentRange 3: reserved 4-7: Max cell voltage difference [mV] uint8_t: SOC range lower [% * 2] uint8_t: SOC range upper [% * 2] int16_t: Current range charge [-mA * 16] int16_t: Current Range discharge [mA * 16]
0x019	SetBalancingConfig	8	(see above)
0x01a	ThermalConfig (→ 7.5)	8	uint16_t: Cooling start temperature uint16_t: Cooling stop temperature uint16_t: Heating start temperature uint16_t: Heating stop temperature
0x01b	SetThermalConfig	8	(see above)
0x01c	CellTemperatureLimits	8	uint16_t: Cell temperature low warning uint16_t: Cell temperature low shutdown uint16_t: Cell temperature high warning uint16_t: Cell temperature high shutdown
0x01d	SetCellTemperatureLimits	8	(see above)
0x01e	CellDifferenceLimits	8	uint16_t: Cell voltage difference warning [mV] uint16_t: Cell voltage difference shutdown [mV] uint16_t: Cell temperature difference warning

7 Konfigurationsschnittstelle (seriell)

ID	Name	Length	Data
			uint16_t: Cell temperature difference shutdown
0x01f	SetCellDifferenceLimits	8	(see above)
0x020	CellVoltageLimits	8	uint16: Low warning [mV] uint16: Low shutdown [mV] uint16: High warning [mV] uint16: High shutdown [mV]
0x021	SetCellVoltageLimits	8	(see above)
0x022	CellVoltageSocTable (→ 7.6)	4	uint8: Table row uint8: SOC [% * 2] uint16: Cell voltage at SOC
0x023	SetCellVoltageSocTable	4	(see above)
0x024	CurrentLimitCharge	8	int32: Charge current warning [-mA] int32: Charge current shutdown [-mA]
0x025	SetCurrentLimitCharge	8	(see above)
0x026	CurrentLimitDischarge	8	int32: Discharge curr. warning [mA] int32: Discharge curr. shutdown [mA]
0x027	SetCurrentLimitDischarge	8	(see above)
0x028	CurrentAdcTableAdc (→ 7.7)	5	uint8: Table row int32: ADC value
0x029	SetCurrentAdcTableAdc	5	(see above)
0x02a	CurrentAdcTableCurrent	5	uint8: Table row int32: Current at ADC value [+-mA]
0x02b	SetCurrentAdcTableCurrent	5	(see above)
0x040	SlaveInfo	5	uint8: Address ^R uint8: Model uint8: Firmware version uint16: Internal temperature
0x041	SetSlaveAddress	2	uint8: Address uint8: New address
0x042	SlaveAdcMinMax	5	uint8: Address ^R uint16: Min. ADC value uint16: Max. ADC value
0x043	SlaveAdcActual	7	uint8: Address ^R uint16: Int. temperature ADC value uint16: Cell voltage ADC value uint16: Cell temperature ADC value
0x044	SlaveCalibrVoltageLow	5	uint8: Address ^R

7 Konfigurationsschnittstelle (seriell)

ID	Name	Length	Data
			uint16: ADC value uint16: Voltage [mV]
0x045	SetSlaveCalibrVoltageLow	5	(see above)
0x046	SlaveCalibrVoltageHigh	5	uint8: Address ^R uint16: ADC value uint16: Voltage [mV]
0x047	SetSlaveCalibrVoltageHigh	5	(see above)
0x060	CounterRuntime (→ 4.3)	8	uint32: Startups uint32: Runtime [s]
0x061	CounterRuntime2	8	uint32: Runtime charging [s] uint32: Runtime discharging [s]
0x062	CounterChargeCharged (→ 4.3)	8	uint64: Charge charged [mAs]
0x063	CounterChargeDischarged	8	uint64: Charge discharged [mAs]
0x064	CounterEnergyCharged (→ 4.3)	8	uint64: Energy charged [mWs]
0x065	CounterEnergyDischarged	8	uint64: Energy discharged [mWs]
0x066	CounterShutdowns (→ 4.3)	8	uint16: OvercurrentCharge uint16: OvercurrentDischarge uint16: NonZeroCurrent
0x067	CounterShutdowns2	4	uint16: Undervoltage uint16: Overvoltage uint16: Undercooling uint16: Overheating
0x06f	ResetCounters	1	uint8: 0: Runtime 1: Charge 2: Energy 3: Shutdowns 4-7: Reserved
0x070	PackStatus	8	uint16: SOC [% * 256] uint16: SOH [% * 256] uint16: Voltage [mV * 16] int16: Current [mA * 16]
0x071	PackStatusHighRes	8	uint32: Voltage [mV] int32: Current [mA]
0x072	Reserved		
0x073	Reserved		

ID	Name	Length	Data
0x074	CellVoltages	8	uint16: Min. cell voltage [mV] uint8: Min. voltage cell uint16: Max. cell voltage [mV] uint8: Max. voltage cell uint16: Average. cell voltage [mV]
0x075	CellTemperatures	8	uint16: Min. cell temperature uint8: Min. temperature cell uint16: Max. cell temperature uint8: Max. temperature cell uint16: Average cell temperature
0x076	CellInfo	6	uint8: Address ^R uint8: 0: Balance 1-5: reserved 6,7: requestErrors uint16: Cell voltage [mV] uint16: Cell temperature
0x0ff	Result (→ 7.8)	2	uint16: Result code

Tabelle 7.1: Nachrichten auf der Konfigurationsschnittstelle

7.3 Konfigurationsmodus

Um jegliche Konfiguration am Master oder einen der Slaves vor zu nehmen muss der Konfigurationsmodus betreten werden. Dies geschieht über die Nachricht „SetMasterConfigMode“ mit dem gewünschten Level und dem Zugangscode. In Tabelle 7.2 sind die verfügbaren Level dargestellt.

Code	Name	Description
0x00	None	Read-only access
0x01	Root	Read/write all parameters

Tabelle 7.2: Übersicht der Konfigurationslevel

TODO: In Zukunft sollten innerhalb einer Minute lediglich drei Zugangsversuche möglich sein um unberechtigte Zugriffe durch Ausprobieren sämtlicher Möglichkeiten zumindest zu erschweren.

Beim Betreten des Modus „None“ bzw. Verlassen eines anderen Modus in diesen ist kein Zugangscode nötig und wird nicht beachtet.

Die Zugangscodes für die Level können mit der Nachricht „SetMasterConfigModePasscode“ modifiziert werden. Hierfür muss der Master zuvor in diesen Modus versetzt werden, damit Zugangscodes nicht unberechtigt verändert werden können.

7.4 Balancing

Der Bereich in dem das Balancing aktiv ist kann durch Parameter beschränkt werden:

- SOC-Bereich: Balancing nur aktiv wenn der SOC im Bereich zwischen 0 % und der Untergrenze so wie der Obergrenze und 100 % SOC. Diese Einstellung verhindert das Balancing in einem möglicherweise sehr flachen Kennlinienbereich.
- Strom-Bereich: Balancing ist nur aktiv, wenn der aktuelle sich Strom im angegebenen Bereich befindet. Diese Einstellung kann verwendet werden um ein Balancing lediglich dann durch zu führen, sofern das Akkupack nicht geladen oder entladen wird.

7.5 Thermisches Management

Es können die Schwellwerte in denen die Kühlung oder Heizung aktiv wird konfiguriert werden:

- Falls eine Zelle die Einschalttemperatur für die Heizung unterschritten hat, wird diese aktiviert. So alle Zelltemperaturen die Ausschalttemperatur überschritten haben wird die Heizung wieder ausgeschaltet.
- Analog löst das Überschreiten der Einschalttemperatur für die Kühlung diese aus. Das Unterschreiten der Ausschalttemperatur stoppt die Heizung.

Zusätzlich können untere und obere Schwellwerte für die vollständige Außerbetriebnahme des Akkupacks definiert werden. Erst nach Überschreiten der Temperaturgrenzen für ein Wiedereinschalten ist ein erneuter Betrieb möglich.

7.6 Zellspannungs-SOC-Tabelle

Um eine genauere Bestimmung des SOC zu ermöglichen wird im Master eine Tabelle bestehend aus 12 Zellspannungs-SOC-Paaren hinterlegt. Während der SOC des ersten Eintrags bei 0% und der des letzten auf 100% fixiert sind, sollte diese Tabelle insbesondere die Zellspannungen im oberen als auch im Unteren Teil der Entlade-/Ladekennlinien berücksichtigen. Beispielhafte Werte für den SOC könnten folgende sein: 0%, 4%, 10%, 20%, 40%, 60%, 80%, 90%, 96%, 100%.

7.7 Stromkalibrierungs-Table

Analog zu 7.6 wird eine Tabelle konfiguriert, die Korrespondenzen zwischen Strom-A/D-Wandler und tatsächlichem Strom enthält. Die mittleren beiden Paare sollten Toleranzbereich für den Strom von Null definieren. Bei einer Tabellengröße von 8 sind dies die Zeilen 3 und 4, da 0-basiert.. So führen kleine Schwankungen der Messung nicht zu der Fehlerhaften Annahme, das aktuelle nennenswert Strom in oder aus dem Batteriepack fließt.

7.8 Rückmeldungen

Diese Result-Nachrichten werden als Antwort auf Kommandos, die Parameter verändern oder abfragen, versendet.

Code	Name
0x0000	Ok
0x0001	Access denied
0x0002	Parameter out of range
0x1000	Slave communication error
0x1001	Slave unknown command
0x1002	Slave parameter out of range
0x1004	Slave wrong message format/length
0x10ff	Slave unknown error
0x1100	Unexpected internal message command
0x1101	Unexpected internal message length
0xf000	Unknown error
0xf001	Unknown ID (for debugging purposes)
0xf002	Not implemented
0xf003	Wrong message format/length
0xf004	CRC error

Tabelle 7.3: Übersicht der Result-Codes des Masters

8 Test

Aus Zeitgründen wurde lediglich eine kurze Inbetriebnahme und eine Überprüfung der Sicherheitseinrichtungen durchgeführt. Folgende Entladungen wurden an einem 10s1p-Modul aus A123 LiFePO₄-Zellen mit 1100 mAh durchgeführt.

8.1 Aufbau

Die 4 mm Anschlussdosen, ein Sicherungshalter (32 A), der Shunt und die Zellen wurden auf einer Kunststoffplatte montiert. Auf jeder Zelle wurde ein Temperatursensor angebracht. Dabei ist sowohl die Spannungsmessung der für die Zellen als auch die Sensoren über einen Steckverbinder verbunden, damit eine Demontage schnell möglich ist. Für den Shunt wäre ähnliches ebenfalls überlegenswert gewesen. Da jedoch ursprünglich SMD-Shunts auf der Platine vorgesehen waren – die zum Aufbauzeitpunkt jedoch nicht lieferbar waren – war dies nicht mehr möglich..

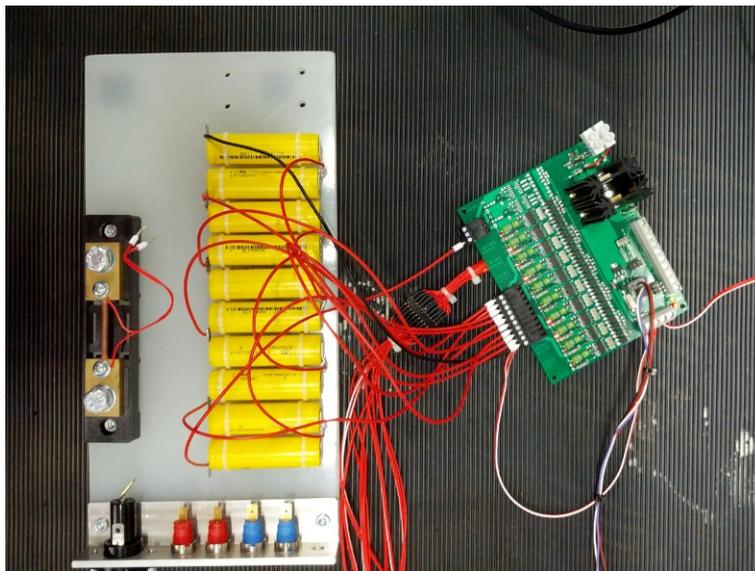


Abbildung 8.1: Unfertiger Testaufbau (Draufsicht)



Abbildung 8.2: Aufbau in Klimakammer

8.2 Testmessungen

Die Daten wurden mit Hilfe des python-Skripts über die serielle Schnittstelle aufgezeichnet.

I_{sum} vs U/T_{cell} ; 5C @ -10°C

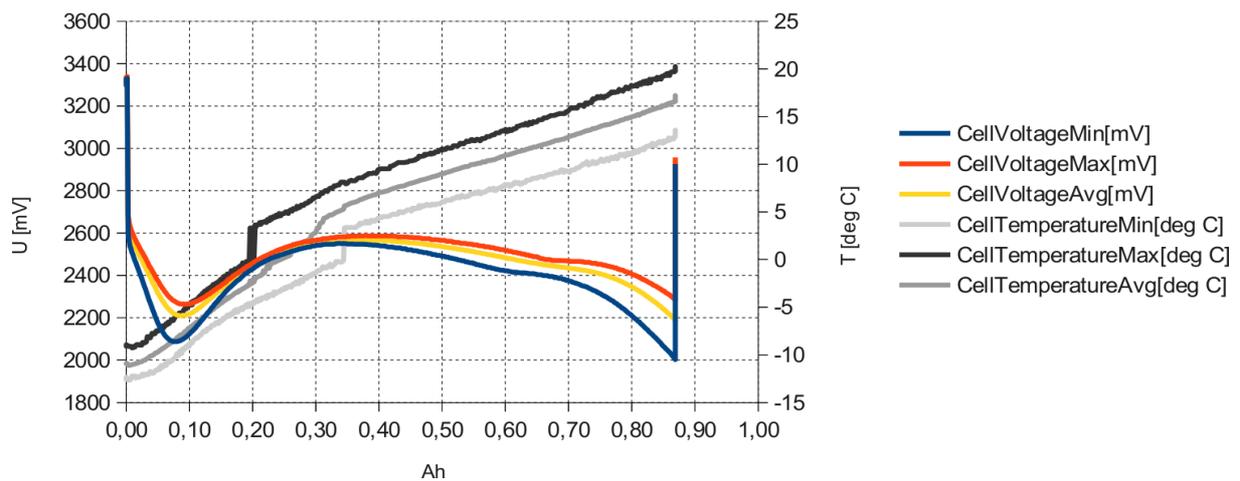


Abbildung 8.3: Testmessung 5C bei -10°C

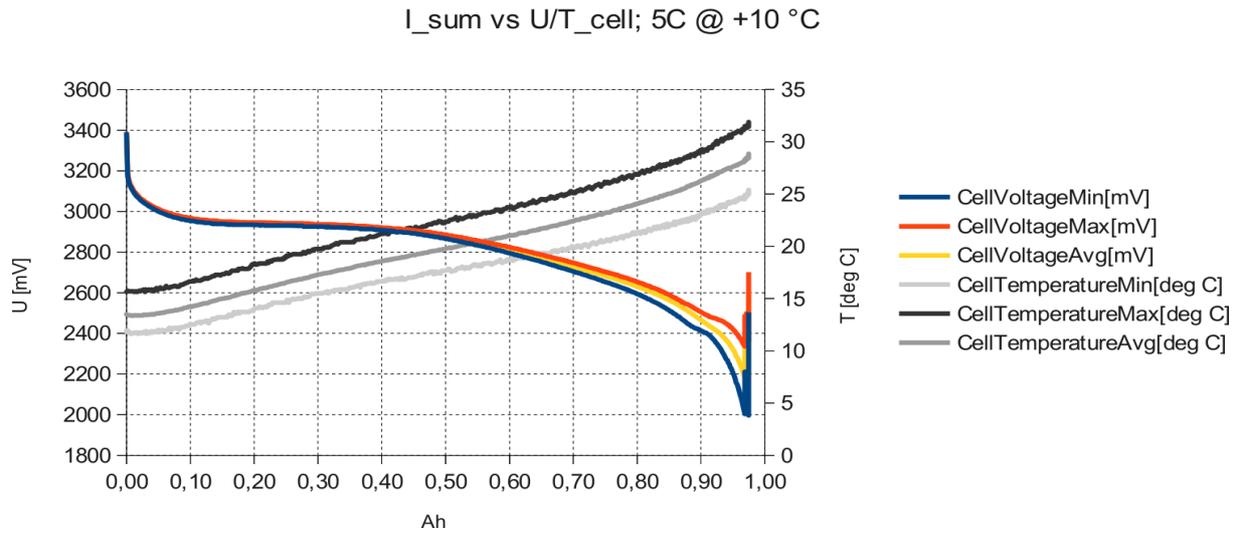


Abbildung 8.4: Testmessung 5C bei 10 °C

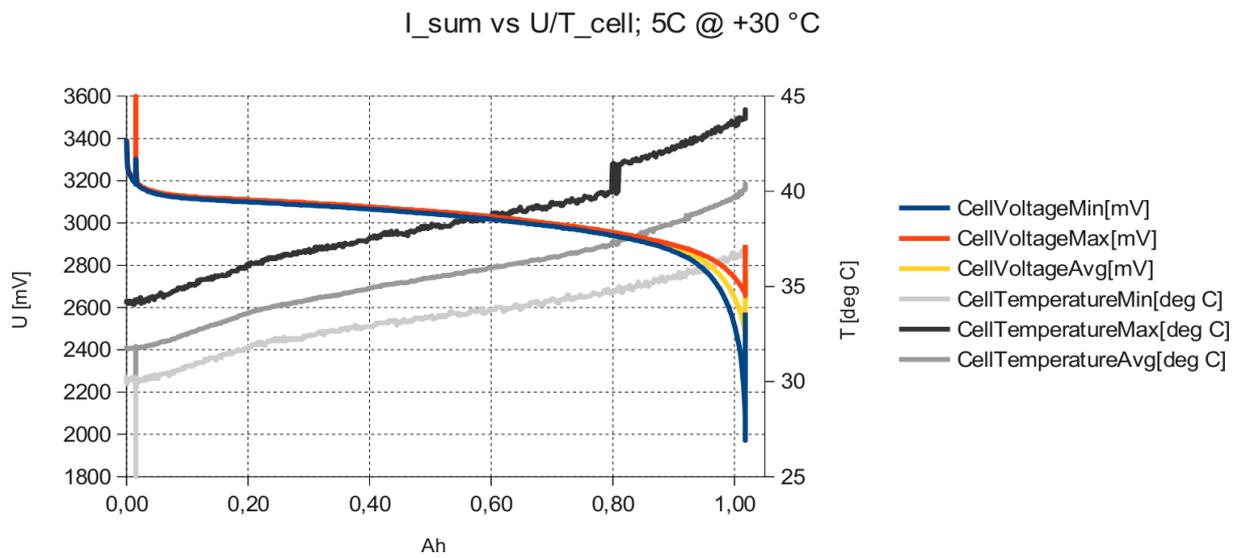


Abbildung 8.5: Testmessung 5C bei 30°C

Abbildungsverzeichnis

Abbildung 3.1: Schema eines Master-/Slave-BMS.....	11
Abbildung 5.1: Auflösung der VCC-Messung gegen interne Referenz (10 Bit-ADC).....	21
Abbildung 5.2: Abhängigkeit des ADC-Wertes von der Zellspannung.....	23
Abbildung 5.3: Flussdiagramm des Ladens der Konfiguration.....	25
Abbildung 5.4: Flussdiagramm der Hauptschleife.....	27
Abbildung 8.1: Testmessung 5C bei -10°C.....	43
Abbildung 8.2: Testmessung 5C bei 10 °C.....	44
Abbildung 8.3: Testmessung 5C bei 30°C.....	44

Tabellenverzeichnis

Tabelle 4.1: Datenformate und -aufzeichnung durch Master.....	14
Tabelle 4.2: Masterstatus.....	17
Tabelle 4.3: Master Batterieverbindungsstatus.....	17
Tabelle 5.1: Übersicht möglicher Controller für einen Einzelzellslave.....	20
Tabelle 6.1: Aufbau eines Datenpakets.....	29
Tabelle 6.2: Übersicht BMS-interner Kommandos.....	32
Tabelle 6.3: Übersicht der Result-Codes auf internem Bus.....	33
Tabelle 7.1: Nachrichten auf der Konfigurationsschnittstelle.....	40
Tabelle 7.2: Übersicht der Konfigurationslevel.....	40
Tabelle 7.3: Übersicht der Result-Codes des Masters.....	42