Battery Monitor

Final Report

AS-0.3200

Automatio- ja systeemitekniikan projektityöt

Spring 2014

Otso Jousimaa

Aleksi Salonen

Johan Holmberg

25.5.2014

Table of Contents

Abbreviations

ADC - Analog to Digital Converter

BOM - Bill Of Materials

CS - SPI Chip Select

ESD - Electrostatic Discharge

GPIO - General Purpose Input/Output

KCL - Kirchoff's Current Law

LPP - Low-Pass Filtering

MBPS - Megabits Per Second

MCU - Microcontroller Unit

MISO - SPI Master In, Slave Out

MOSI - SPI Master Out, Slave In

NTC - Negative Temperature Coefficient

SCLK - SPI Clock

SMPS - Switch Mode Power Supply

SPI - Serial Peripheral Interface bus

TTL - Transistor-Transistor logic

USB - Universal Serial Bus

1 Introduction

The original project task was to create a battery balancer solution for the batteries in an electric car. The solution would consist of a separate battery balancer board, battery monitor board and a controller that would command the circuits. The battery balancer was designed and produced in another project during the Autumn of 2013. This project was concentrated on designing and producing the monitor part and integrating all the components into a working system.

Due to the fact that the balancer boards from the previous project were broken, the project scope was reduced to designing and producing the monitor board and controlling the monitor with a microcontroller. This document describes the final solution to the updated task. We also thought about making our own balancer board, but as there were no blueprints from the previous group, we deemed it to be way too time consuming. Also, the ordering process would've taken over 2 weeks, and there wouldn't have been time to write any software to use the balancer board.

The monitor board was designed using KiCad [1] for schematic edition and layout. LTSpice 4 [2] was used to simulate analog sections of board. There was enough components to fit on the board to motivate the use of a 4 layer board and KiCad capabilities to design multilayered boards. Also, as the boards were ordered from China instead of producing them on our own, we had no real reason not to use the 4 layer board.

Biggest risk in making the board was an unrecoverable mistake in hardware design. By the time a mistake would be found, there would not be time left for creating and assembling a new board. This risk was mitigated by designing a board with functional blocks that have minimal interactions and some free space in between. This way components could be added later in between testpoints and vias and signals could be injected to simulate proper behaviour of blocks. This separation of functionality and preparedness for later additions was a very good decision, since when the initial design failed we were able to isolate causes of errors and test fixes for them.

To make the task easier, we chose to use a Arduino Nano [3] board with its ATmega328 [4] microcontroller to run the necessary monitoring logic. The Arduino environment is easy to set up, which allowed us to put more focus on designing the monitor board and writing software for controlling the board.

2 Workflow

2.1 Study phase

Immediately at the beginning of the project we stumbled upon problems. We could not get the balancer board from the previous project to work. It took a while before we got to the root of the problem. Some pins of the main circuit on the board were short circuited. We were unable to fix the short circuited pins which lead to the updated project scope as mentioned in the introduction. Also, even if we could've removed the short circuits, the board probably wouldn't have worked. It's possible, that during the short circuit removal attempts the board overheated and broke some components. The other board was broken before we got it, as there were visible burn marks on some of it's components.

We also had to study on how to use KiCad. Luckily, Otso had prior experience and was willing to teach us about the software.

2.2 Initial design

A first prototype was made following the datasheet of the LTC6803-3 rather closely. Because there was a short time frame to get the design complete to get board ready for software development, there were a few mistakes that should have been caught in design review phase.

Most of these mistakes could be fixed afterwards by connecting components between test points and vias on board, and the schematic diagram of circuit has been updated to include these fixes.

2.3 Producing the boards

Because it is too difficult to produce your own 4 layered board, the monitor boards were ordered from China. The components on the board were soldered by hand. We also created a stripped version of the board as a single layer design and tried to solder it, but soon realized that the quality was not good enough for precision soldering.

Elecrow [5] was selected to supply the boards because of cheap prices, good manufacturing capabilities and fast delivery.

2.4 Software development

The Arduino development environment was tested early on and some test programs were written to ensure that the Arduino with its ATmega328 was suited for the job.

Serious software development began as late as in the beginning of April. This was already far

behind schedule. Software development was luckily initially rapid and quite straight forward, with a few major problems faced only later on.

First, the limited resources on the Arduino Nano board set an upper limit for the amount of code to write. Secondly, the SRAM on the Atmega328 circuit is limited to 2KB of data. Thirdly, the Arduino development environment set limitations on how to structure the program.

The first issue never really proved to be a problem. This is mainly because other difficulties delayed adding new features to the point where several part of the software had to be left undeveloped. In any case a complete battery balancer implementation should fit in the 32KB memory available.

The second issue resulted in a problem that took serious time and effort to solve. What filled the 2KB memory space was all the constant, static strings hardcoded into the code, used for debug messaging and printing usage instructions. The symptoms appeared random as the Arduino would crash and reboot, or just stop communicating with any computer over the USB-serial bus. This could happen when adding a few lines of code to the project, but the controller could crash without even running the added code which was confusing. Significant time was lost in finding the real issue.

The last point forced us to restructure the program several times. An example of the restrictions faced was that although functions are automatically shared and can be called between files within the same Arduino project, the same does not apparently apply for global variables. The order in which the files appear in the project matters, but then again the order of the files cannot be chosen freely but is always in alphabetical order. These kinds of strange restrictions slowed down software development.

2.5 Documentation

Most of documentation was generated “on the go”, as design choices were made. This document gathers the most relevant parts of documentation generated during the course. If you're a future group tasked to integrate the board as a part of some larger system, please feel free to contact us with any questions.

3 Project results

3.1 Hardware

3.1.1 LTC6803

LTC6803 [6] is a *Multicell Battery Stack Monitor* integrated circuit. It features 12 differential ADCs which can be used to measure voltages of individual batteries as well as 2 ADCs which can be used for measuring external voltages, for example NTC-resistors.

Communication between ICs in system is done through SPI-like current signalling bus. Current mode signalling allows stacking of boards without level shifters in communication lines. Communication to MCU uses same bus, but communication must be isolated to avoid destroying attached control logic in case of some electrical fault in battery stack.

The circuit has also a watchdog timer and 2 GPIO pins which can be used to display status information or communicate with other digital devices in circuit. Leds were connected to these pins.

3.1.2 Printed circuit board

The circuit board was implemented using a 4-layer design which allows unbroken ground and power planes. Digital and analog sections of board are kept separate, and careful attention is paid to keep clearance distances in accordance of IPC 2221 [7].

The board design allows three separate functions: A bottom of stack board which communicates with the MCU and above boards, middle boards which pass data up- and downwards and top of stack. Having different roles on same PCB cuts down design time and cost.

All project files can be found at project's Github repository [8], PDF printouts should have accompanied this document. In case they were not, files are available at a http://ojousima.net/as03200/pdf [9] until 2015

Ten PCBs and parts for four assemblies were ordered. Two assemblies are complete with fixes, one with isolator. One assembly is complete, but not fixed. Fourth assembly went horribly wrong, twice. On the fourth assembly the LTC6803-3 had to be extracted out of the PCB using hot air, and it's potentially damaged. Since design is known to be defective, other components were not populated. In addition, two boards were destroyed while preparing for assembly, so only three bare PCBs remain intact. Our assembly process had a yield of approximately ~ 50 % for PCBs and ~75 % for components. This is comparable to experiences of research staff at laboratory.

3.1.3 Power supply

LTC6803 offers a possibility of using power supply completely isolated from batteries. In this case,

ripple rejection resistors must be omitted and ferrite bead and polarity / over voltage protection zener can be omitted.

An isolated PSU should be connected to V+ and DGND pins. Please note that this power connection does not have reverse polarity protection.

In battery powered mode PSU bypass circuit must be installed completely. Zener diode offers protection against ESD spikes and voltage transients as well as limited protection against over voltage and reverse polarity. In case of over voltage, power dissipated in zener will burn out the connection quickly and zener will fail, hopefully as a short.

In case of a reverse voltage, reverse voltage is clamped to forward voltage of zener. Bypass resistors limit the current to LTC6803 to under 10 mA as long as reverse voltage is clamped below 2 V, which is survivable. [6 p. 31]

To reject power supply ripple, bypassing is required. Since power supply and lowest battery measurement are isolated from each other in LTC6803-3, resistors can be added to both supply and ground lines. Differential bypassing suggested by LTC6803 datasheet (Figure 1) can be used.

This supply bypassing method also offers an easy way to measure current consumption of the LTC-6803. A multimeter probing across resistor shows voltage across resistor, current can be calculated from equation I = U*R.

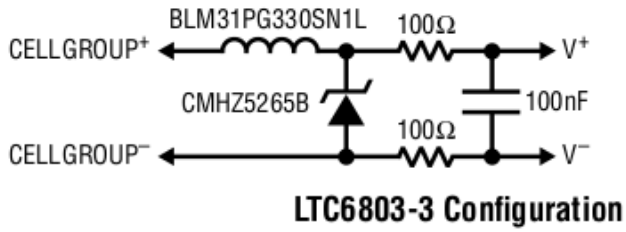

Figure 1: voltage protection and filtering

3.1.4 Hardware fault conditions

Table 15 of LTC6803 datasheet [6 p.30] lists a number of fault conditions. Two first cases are mitigated by LPF circuit with protection zeners and internal clamps at LTC6803.

There is no redundant power supply in case of a power loss, if communication is lost battery charging / discharging should be driven into a safe state.

A break in battery stack would force reverse stack potential on communication link during discharge, high voltage series diodes protect monitor circuits and digital isolator protects controller circuit. During charge redundant current path can handle up to 10 A of current.

Two last cases are not accounted for. In case of a break within a stack, current carried by 0.5 mm

trace between stacks. This trace can carry at most 2.5 A of a current according to IPC-2152 [9]. If this trace burns out, voltage gets applied to discharge circuit and LPF resistors, leading to destruction of discharge circuit. Discharging voltage then is applied across the LPF resistors, 2 * 10 k ohms. Maximum survivable voltage is limited by power dissipation of resistors,

$$P = U^2 * R \rightarrow U = \sqrt{PR} = \sqrt{2 * 0.25 * (10000 + 10000)} = 100\,V$$

This is not a practical value in actual car battery stack, but it will do in laboratory testing.

Disconnect inside stack while charging can also lead to a catastrophic failure if voltage and current are not limited.

3.1.5 Passive balancing

LTC6803 has internal mosfets that can be used for modest passive balancing. Internal mosfets are connected directly into lower cell. LPF's series resistance would raise the voltage at ADC if current would be returned on LTC6803 side of filter. Figure 2 clarifies issue. Discharge current should be returned to input connector of battery rather than to ADC input pin to avoid rising voltage at lower ADC over spec.

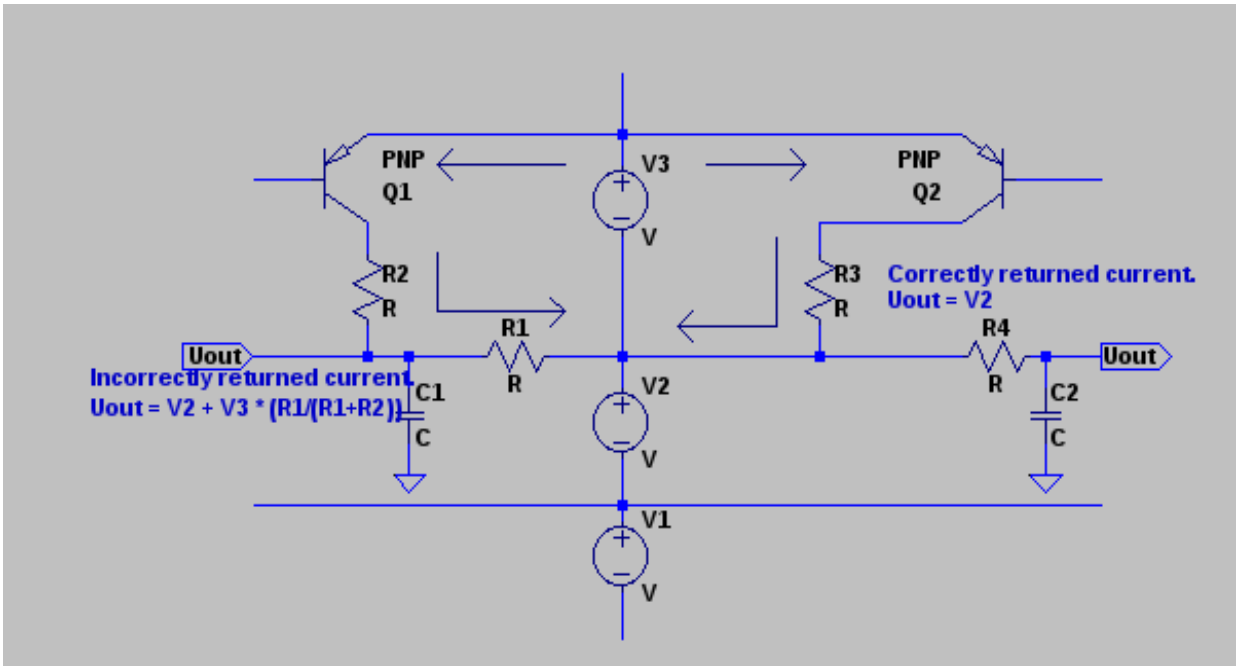

Figure 2: Always, always, always consider KCL.

This is done by using external transistor which can direct most of the current back to connector input.

Datasheet of LTC6803 suggests using Si2351DS PMOS [10] with a protection zener from gate to source. Using a PNP transistor avoids the need for a separate protection zener, since zener of LPF couples the voltage of lower cell to next cell in series, and this voltage gets coupled into base of transistor through emitter-base junction.

Voltage and current ratings should be comparable to Si2352DS: -20 Vce and 3 A of collector

current. ZUMT718 [11] is a decent match with Vce of -20 V, Ic of 1 A (3 A pulsed). It also has suitable package and it is readily available.

ZUMT718 has Hfe of at least 300 at collector current of 100 mA. The base resistor can have 100 * resistance of shunt resistor while still keeping transistor in saturated condition.

Since passive balancing can only turn excess charge into heat, heat management must be considered. Since external transistor is used to return bulk of current back to connectors, power dissipation at LTC6803 is negligible.

Most of the power is dissipated in shunt resistors. Since actual balancing is going to be implemented at a separate circuit, these resistors do not need to be scaled for useful powers. A 100 ohm 0805 resistor with power handling capacity of ¼ W can be to simplify the BOM, since PSU bypassing circuit already uses these.

Approximately 40 mA can be pulled from battery through this resistor. A LED is inserted in current path to show diagnostics information. This can be used for “battery full” indication. If a led is used, current must be limited to ~20 mA. This is conveniently achieved with 100 ohms shunt resistor together with forward drop of led.

Alternatively footprint of led could be replaced with another 0805 resistor for doubled power handling capacity. In that case 27 ohms would be a reasonable choice, as total series shunt resistance would be ~ 50 ohms leading to a current of ~ 80 mA when discharging a battery at 4.2 V (Typical lithium-ion cell at full charge). Power dissipated in a single resistor would be ~ 170 mW.

Every mosfet has a property of drain-source leakage current. Internal mosfets of LTC6803 are surely not an exception. While leakage current of a mosfet would not usually be a problem, any leakage is amplified by external PNP transistor.

LTC6803 has internal pull-up mosfets which connects the base of transistor to potential of upper cell. This will keep transistor strictly in cut off region, limiting leakage to it's own leakage current. Therefore, it is important to ensure in software that transistors are pulled up. External LEDs will light up if there is significant leakage current.

Lithium-ion batteries have ability of discharging at rather extreme currents, possibly causing fire or explosion. In case where PNP-transistor's base gets shorted to ground transistor will act as diode through emitter to base. Semiconductors have a habit of failing as short circuits unless internal connections of chip burn out.

This scenario is mitigated by laying base resistor before LPF's via to ground. In case of a short the resistors will limit the current through transistor and zener diodes will provide a path to ground for other cells. Resistors of LPF's will limit the current through these connections to survivable level.

3.1.6 Hardware configuration

Board has two jumpers, VMODE and TOS. Jumpers can be connected to VREG by placing jumper in above position and to V- by placing jumper to lower position.

Case: Standalone circuit. TOS is connected to VREG. VMODE is connected to VREG.

Case: Multiple circuits, bottommost circuit. TOS is connected to V-. VMODE is connected to VREG.

Case: Multiple circuits, middle circuit. TOS is connected to V-. VMODE is connected to V-.

Case: Multiple circuits, topmost circuit. TOS is connected to VREG. VMODE is connected to V-.

Only bottommost circuit requires isolator and assorted capacitors and connector to MCU. Bottommost circuit does not need lower chain connector and topmost circuit does not need upper chain connector.

GPIO and Watchdog leds and resistors can be omitted if not used. If temperature sensing is not used, precision resistors and NTC connectors can be left out.

3.1.7 Finished board and used set up

The finished board can be seen below in Figure 3.

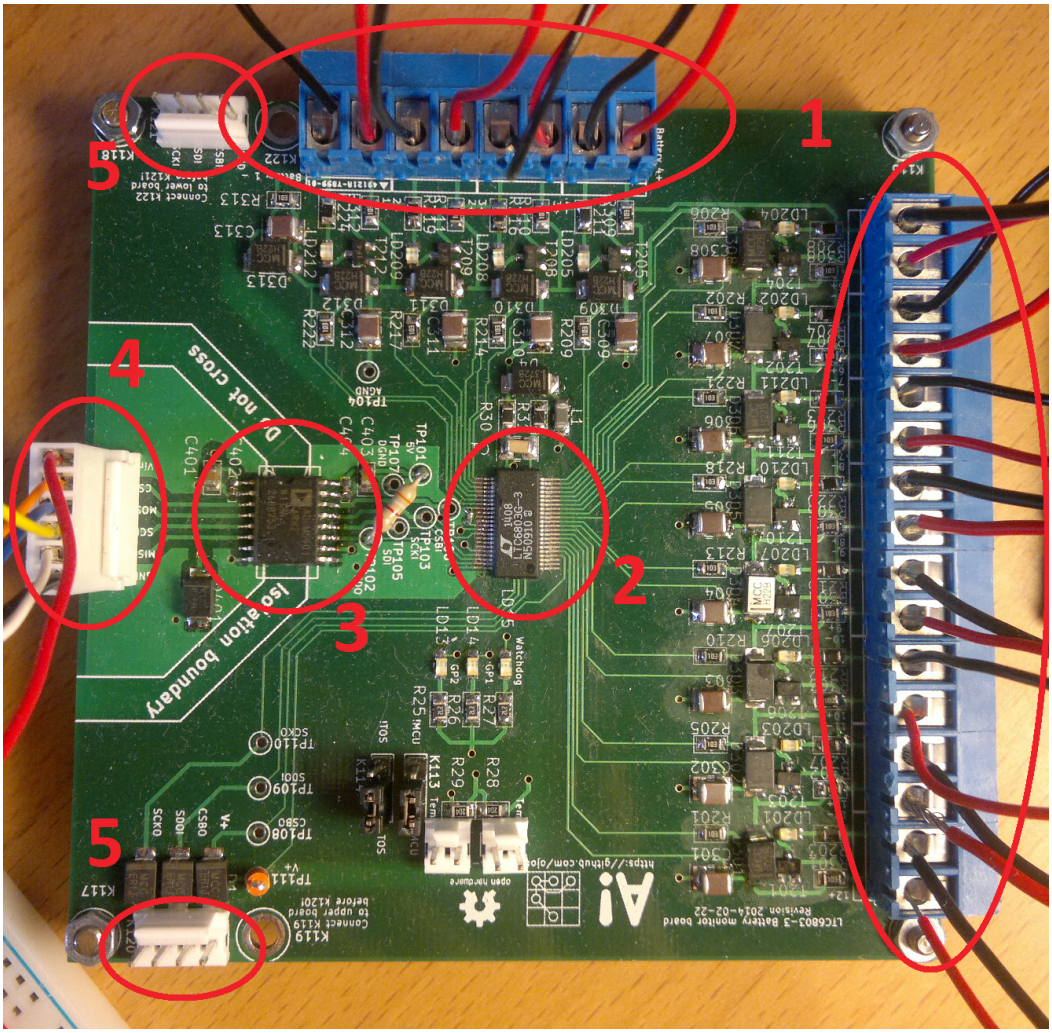

Figure 3: The monitor board

The numbered parts seen in Figure 3 are:

1. The battery or cell interface
2. The LTC6803 circuit
3. The isolator
4. The isolated SPI connection to the MCU
5. The SPI ports to other stacked monitor boards.

The setup used for testing can be seen below in Figure 4.

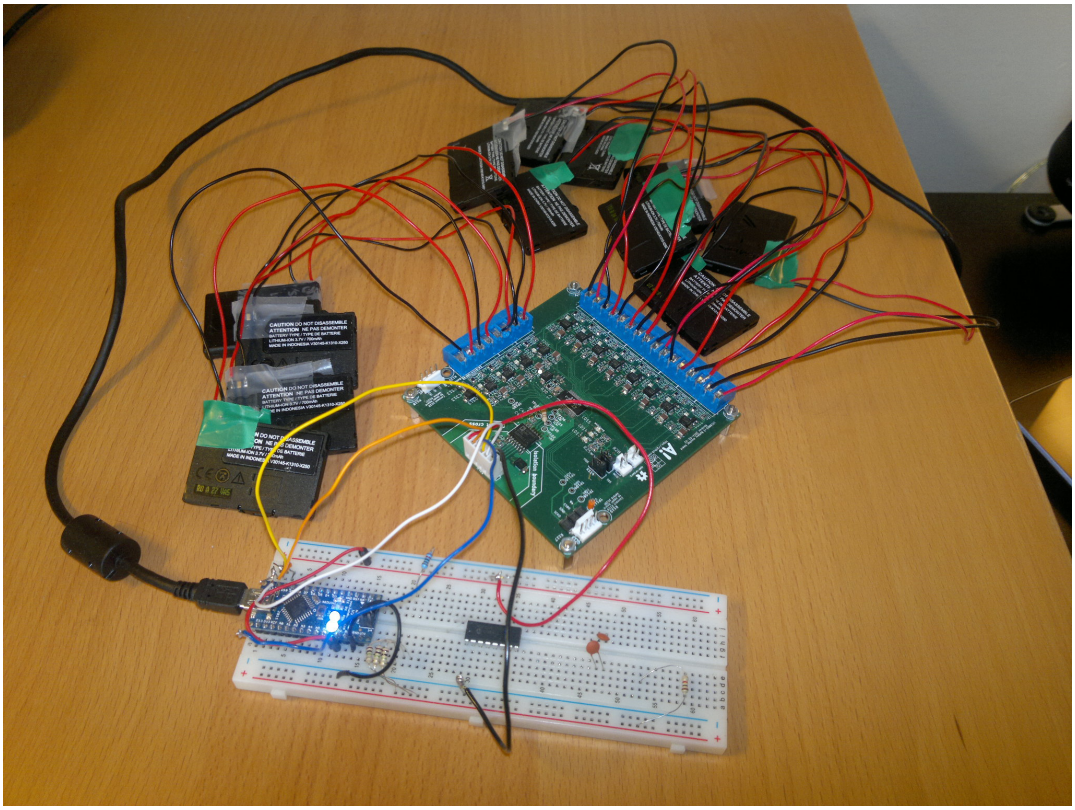

Figure 4: The testing setup

3.2 Software

The project software is available at GitHub. [8]

3.2.1 Software structure and interfaces

The intended structure of the software is depicted in Figure 5 below.

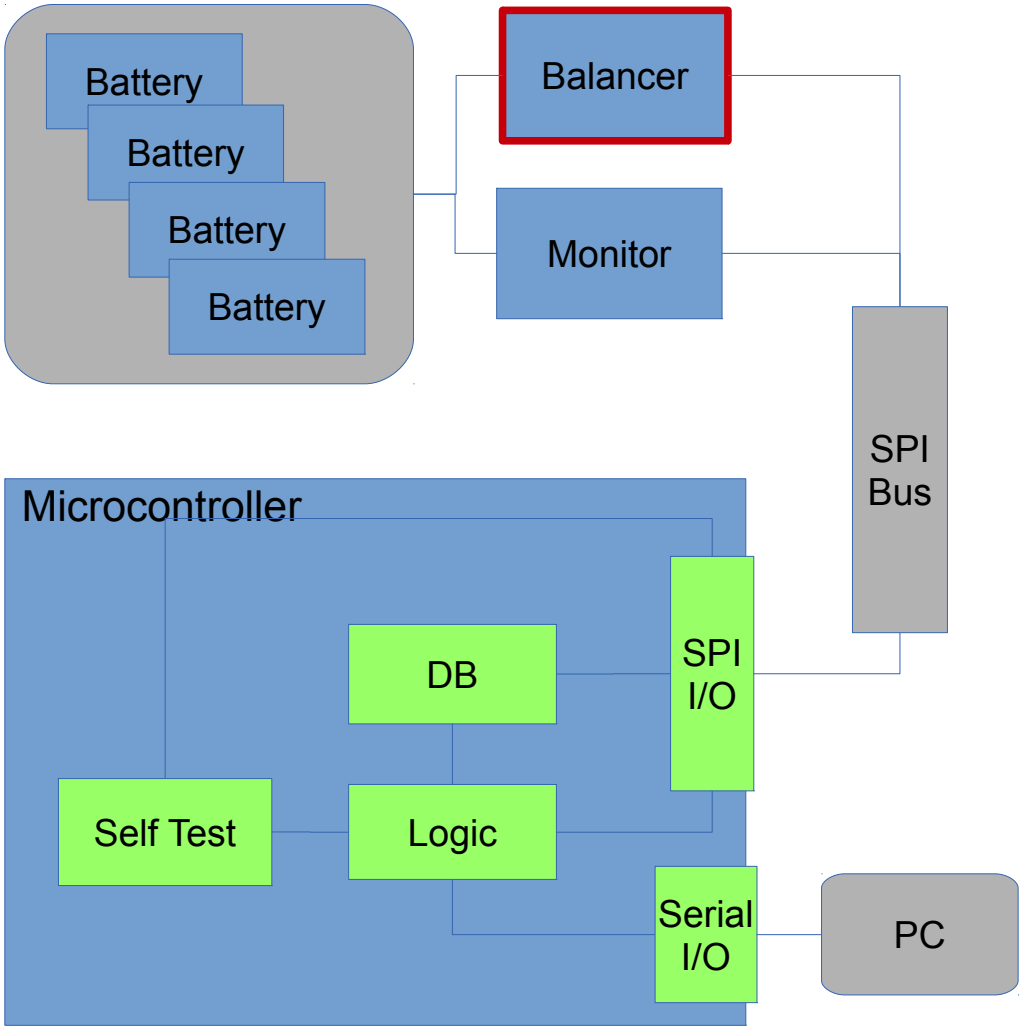

Figure 5: Software components and interfaces

As seen in Figure 5 above, the software is split into several modules to keep the code structure clear. The Logic module consists of the main loop, which makes various checks and schedules the microcontrollers operation. The Serial I/O module is responsible for handling user input to the system. The SPI I/O module handles sending and receiving data to and from the SPI bus. The Database (DB) module is intended to store data from the monitor and balancer boards in meaningful structures for the Logic modules purposes. Lastly, the Self Test module checks incoming data from the SPI bus as well as the microcontrollers internal data and checks that the entire system operates properly. It raises error flags which the Logic module can use in case of invalid operation.

The software is configured from a laptop connected to the MCU over a USB-serial port. In the case of the Arduino Nano board we used, this simply means installing the Arduino development environment and connecting the computer to the Arduino over USB. Using the Arduino Serial Monitor feature, the user can send commands to the microcontroller.

The microcontroller controls the monitor board over a SPI bus. This is done by writing to control registers in the monitors LTC6803 circuit or by sending direct commands to the LTC6803. Data is read from the circuit by reading control or diagnostics registers on the LTC6803.

The balancer board has not been done yet and is therefore marked with red in Figure 5. The balancer board should in be connected to the same SPI bus as the monitor, since the SPI bus supports several slave nodes on the same bus.

3.2.2 Software design

The software is designed to be modular so that it can be modified and improved later. The design is illustrated below in Figure 6.

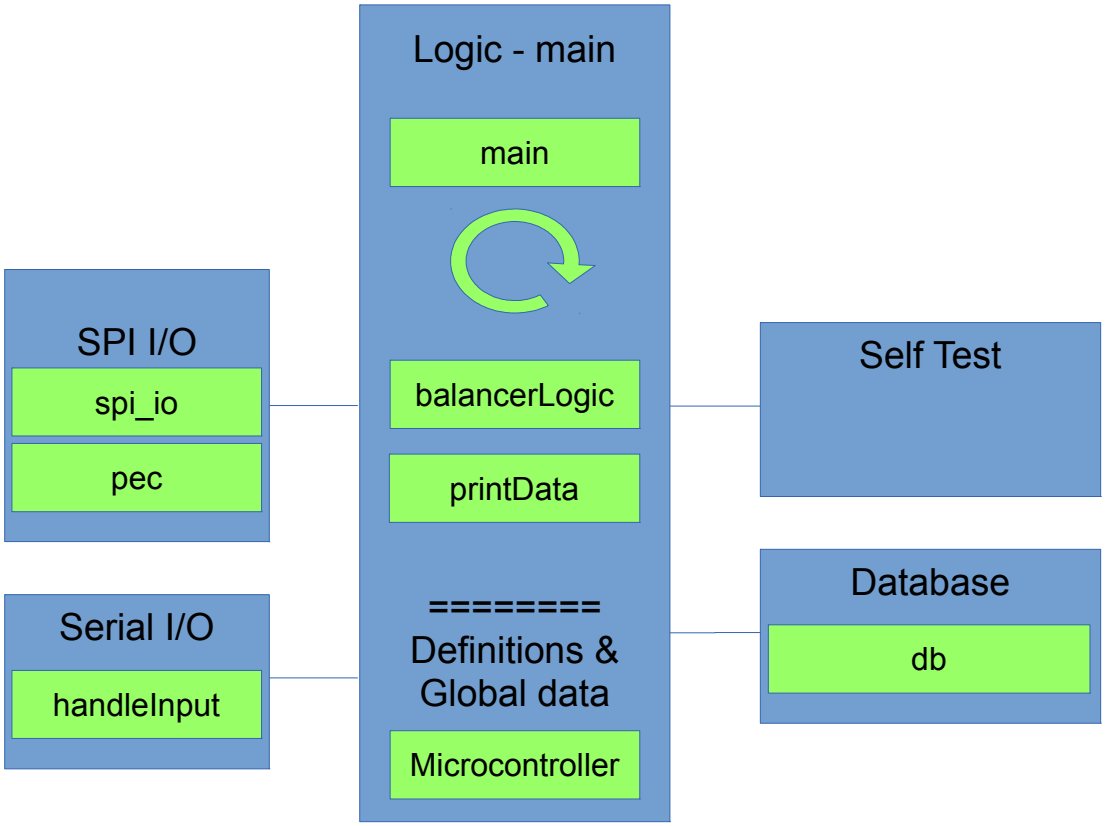

Figure 6: Software design

Figure 6 above shows how the different software modules are split into separate files. At startup the Arduino first calls the setup function one time. Then the loop function in the main file is continuously called until power loss. The setup function is used to setup the system,

communications and other one time tasks and is called once at boot. The main loop runs in the loop function. The loop function checks for user input in every cycle and calls the input handler routine. It then checks if adc conversions should be on or off and sends the appropriate command to the monitor board. Lastly, all diagnostics and configuration data is read from the monitor board registers and copied to the microcontroller memory on every cycle.

The spi_io file contains all the functions related to transferring and receiving data over the SPI bus. For this purpose, checksums need to be calculated for the transferred data as described in the LTC6803 datasheet [6]. This functionality is implemented in the pec file. The monitor board behaviour is modified in two ways. Either by sending direct command codes to the LTC6803 circuit, or by modifying the LTC6803 circuits configuration registers. The command codes and the content of the configuration registers can be looked up in the LTC6803 datasheet [6].

User input is handled in the handleInput file as seen in Figure 6. To make the main loop cleaner, all actions in response to user input are called directly from the handleInput file. Currently, user input is restricted to single characters. Enabling and disabling discharging is a 2 phase, 2 character input. First the user enters the command to start or stop discharging, after which the user specifies which individual cell or alternatively all cells to discharge or stop discharging.

The self test and database modules are currently incomplete. Work on these has not even begun yet.

3.2.3 Implemented software features

The logic for the battery monitor functionalities is not very complicated and should have been, as such, easy to implement. However, as mentioned in 2.4 Software development, the problems encountered slowed down development.

1. Currently it is possible to read configuration the LTC 6803 configuration and diagnostics registers.
2. ADC conversions, i.e. checking battery voltages can be turned on and off.
3. Passive balancing can be enabled and disabled per individual battery, or all batteries at the same time.
4. Communication integrity between the LTC 6803 and the Atmega328 MCU is verified, i.e. checksums are checked and error messages printed in case of failure.
5. Printing debug messages is available using the Arduino Serial Monitor. A help message describing the features is printed with the 'H' character.

3.2.4 Known software limitations

1. There is currently no automatic battery balancing logic implemented. This was not a high priority since the adc conversions often proved to be unreliable. Setting low and high

voltage limits is pointless until battery voltage measurements are truly stable. In the future, this logic can be either done completely on the MCU, comparing battery voltage measurements to some limits specified by the user. The other option is to use the LTC6803 circuits low-voltage and high-voltage flags. The user sets the low and high limits for the adc conversions to the LTC6803 configuration register. The LTC6803 then reports if a adc conversion value is outside is permitted range. Setting the high and low limits to the configuration register is not implemented so far.

2. In order to avoid error messages flooding the PC-Atmega328 USB-serial connection, error messages are only allowed to be printed at certain intervals. But no message queue has been implemented, which means that error messages not occurring constantly can be missed completely if relying on this feature.
3. Passive balancing should not be active while performing adc conversions. The balancing operation disturbes the battery voltage measurements and gives bad results. The software should in the future be implemented so that the two actions seem simultaneous to the user, but are in reality activated and deactivated in intervals so that they are not active at the same time. This would naturally mean that the adc conversions would be done at a lower frequency than normal. Currently, balancing and adc operations are simultaneous if the user gives those commands to the MCU.
4. Support for several monitor boards connected in a stack is missing. According to our understanding of the LTC6803 datasheet [6], adding this is relatively straight forward. For instance when asking for all the stacked monitor boards diagnostics registers, all the data is received in one single, simply bigger packet. What has to be done in software is extract the data from the different boards from the big message and verify the monitor boards checksums separately. The inverse applies when sending data, i.e. writing several configuration registers at one time. Refer to the LTC6803 datasheet for more information [6].

3.3 Interfaces

3.3.1 Monitor – Microcontroller

According to LTC6803 datasheet, SPI bus communication to MCU should be isolated to avoid destruction of controller in case of a battery stack connection failure. Since high-speed digital isolators need more current than integrated regulator of LTC6803 can source, a separate 5V source is required. [6]

Isolator needs 4 channels, 3 outputs from MCU side and 1 input to MCU side. Inbound signals are CS, SCLK and MOSI. Return channel carries MISO. At least 1MBPS data rate is required. Analog Devices offers *iso*Power -product line which has integrated isolated SMPS. Using such chip decreases overall complexity of design, as separate SMPS is not needed.

Based on these criteria, ADuM5401 [12] is chosen as isolator. It has versions for 1 MBPS (A) and 25 MBPS (C) communication, A version is chosen to lower costs. SPI is clocked by master, so in case of data corruption speed of communication can be decreased.

SMPS of ADuM5401 operates without ferrite core to contain the stray magnetic field, which is why the chip is practically guaranteed to generate RF noise which would take highly specialized PCB layout to keep at EMC / FCC -compatible levels. If the product is to be commercialized, the layout has to be reviewed by an expert.

Test points are placed on isolated side to allow verifying data transmission across isolation boundary. In case of isolator design failure, these test points can be used to inject signals into target board.

3.3.2 Monitor – Batteries

All battery connectors have screw terminals. This allows a defined power connection sequence. Screw terminals are also resistant against vibration and shock, which is important consideration in an automotive environment.

The end goal is to integrate monitor board with balancer board, so connectors are 5.00 mm pitched with current carrying capacity of 10 A.

Every battery has their own connection. Batteries are connected in series on board, but this connection should not be used as only connection between batteries.

While batteries can be connected in any sequence without endangering the boards and even reverse voltage is survivable if source is current limited, it's recommended to connect batteries from bottom to top, negative terminal first.

Electrical connection to ADC has a RC low-pass filter shown in Figure 7 to reject noise coupling into measurement. A 7.5 V zener diode gives over voltage and limited reverse polarity protection for the measurement cell.

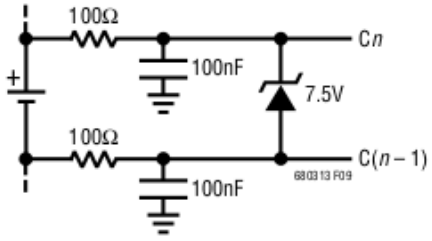

Figure 7: LPF + protection zener

Series resistance of 100 ohms in parallel with 100 nF capacitance would introduce an 1 mV DC

measurement error to ADC. On the other hand, LPF offers 30 DB noise rejection and ESD protection with zener. Since 1 mV is smaller than LSB of ADC, this error is negligible. [6 p. 27]

On the other hand, ground connection of capacitor creates hazardous voltage differentials on board. In case where capacitor gets shorted grounded, up to 50 V (topmost cell, while charging) could be imposed on filtering resistor. This could be a catastrophic failure, as

$$P = \frac{U^2}{R} \Rightarrow P = 25\,\mathrm{W}$$

While current of 0.5 A is not too much, if the resistor fails as short entire battery stack gets shorted. As a mitigation measure LPF of 10k ohms and 3.3 uF is be used. 10 k series resistance limits current to 5 mA, and power dissipated is 250 mW. Capacitor that can handle at least 75 V (50% margin) is notably larger (at least 1206) and more expensive. DC error of such a circuit is smaller at 0.5 mV. [6 p. 27] Figure 8 shows the interface.

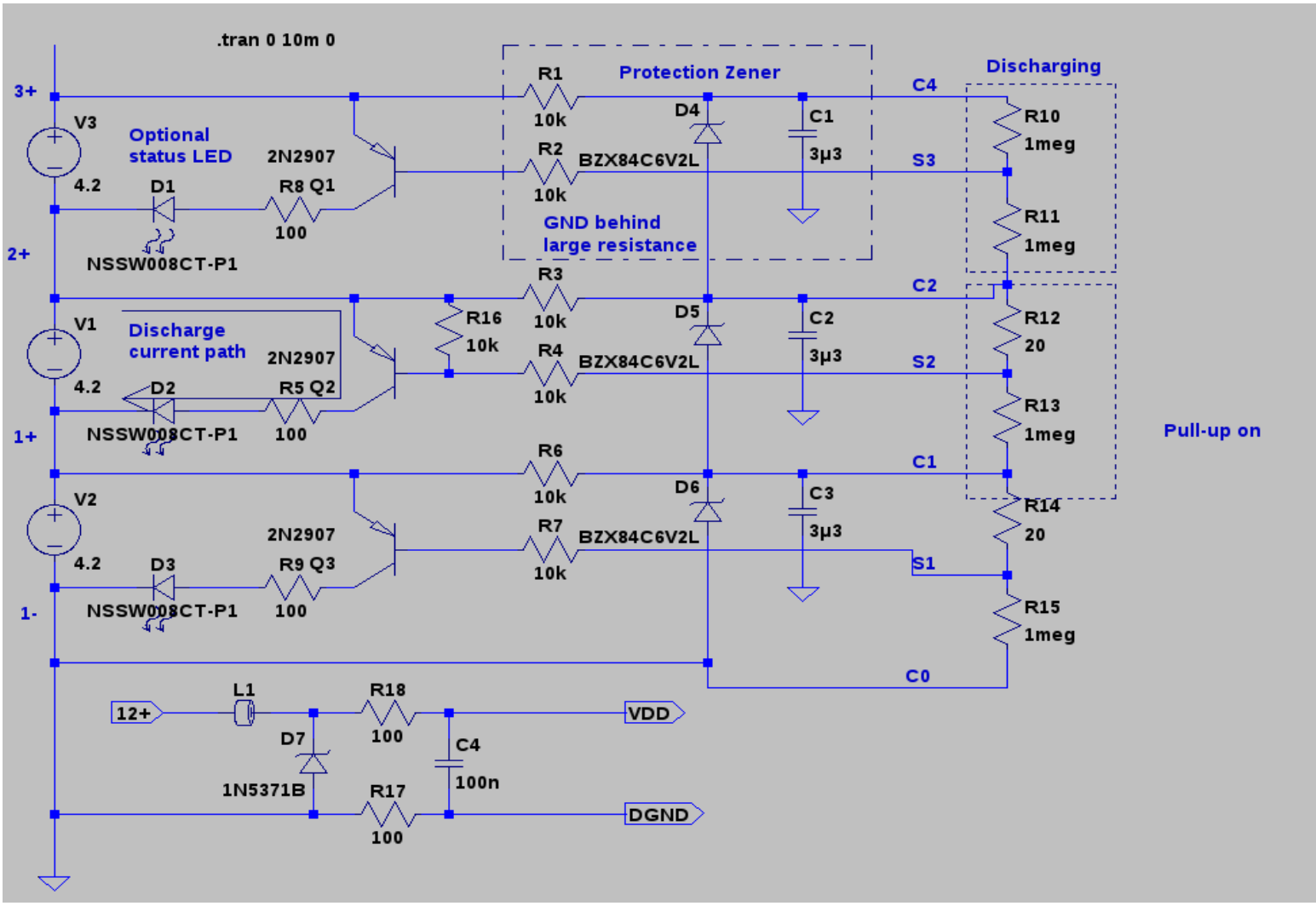

Figure 8: Battery interface of circuit

3.3.3 Microcontroller – PC

The ATmega328 on the Arduino Nano communicates with a PC over a USB-Serial Port emulator with speed 9600bps.

3.4 Implemented features and usage

3.4.1 Completed features

Currently the circuitry and software supports isolated communication to MCU. Monitor circuit can measure voltages of individual cells and activate passive discharging of any combination of cells. GPIO leds can be used for giving user information about status of circuit.

Measurement accuracy and precision are “reasonable”, and probably within the specified 10 mV range of LTC6803. However, this has not been thoroughly tested, especially not over the full operating temperature and voltage ranges.

Self diagnostics of LTC6803 can be verified and give out specified values.

3.4.2 Future features

Temperature sensing has not been implemented in software. Stacked communication is supported by hardware but only partly in software.

Level polling and setting alarm limits in LTC6803 for cells can be implemented in software. After setting the level limits in LTC6803 MCU doesn't have to poll the board continuously, LTC6803 will report anomalies.

Open connection detection to batteries could be implemented in software by appropriate usage of discharge circuitry. However, discharge circuitry is unreliable without external pull-up resistors and a glitch could lead to false results.

3.4.3 Usage

Connections should be made in following order:

1) Redundant current paths should be established with toothwashers and screws connecting to a heavy-duty wire (at least 10 A continuous current carrying capability) between boards in stack.
2) Data connections between boards should be established.
3) Jumpers should be set to appropriate positions, see 3.1.6 Hardware configuration for details.

4) Batteries should be connected, negative terminal first, from lowest to highest.
5) Microcontroller should be connected to isolated connector in bottommost board.

If Arduino is used, code should run as is. Commands are sent through “serial monitor” in Arduino environment. Send the 'H' character command to print usage instructions.

4 Time usage and schedule

The original schedule for the project can be seen below in Figure 9.

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Opening lecture	x																		
Prestudy of project	x	x	x																
Plan and initial presentation (29.1)		x	x																
KiCad training			x	x															
Study previous project and its results, i.e. balancer board			x	x	x														
Design monitor board			x	x	x	x													
Test connection between balancer and microcontroller				x	x					(x)	(x)								
Sof tware design description					x	x	x		(x)	(x)	(x)								
Production and assembly of monitor						x	x	x											
Intermediate report and presentation (5.3.)							x	x											
Integration and coding of control logic									x	x	x (x)	x (x)	 (x)	 (x)					
Testing										x	x	x (x)	x (x)	x (x)	 (x)	 (x)			
Final documentation and presentation (7.5)															x	x	x		

Plan B – The balancer board non functional																			
Design new balancer board					(x)	(x)	(x)	(x)											
Production and assembly of new balancer board								(x)	(x)	(x)									

Figure 9: Original schedule

We were able to stick to the project schedule the first weeks, almost up until the intermediate report. Due to other courses, we fell significantly behind schedule after the intermediate report. The hardware part of the project progressed far more smoothly than the software. The software was developed mainly during the last 4 weeks of the project. Team member time usage per week can be seen below in Figure 10 and total time usage in Figure 11.

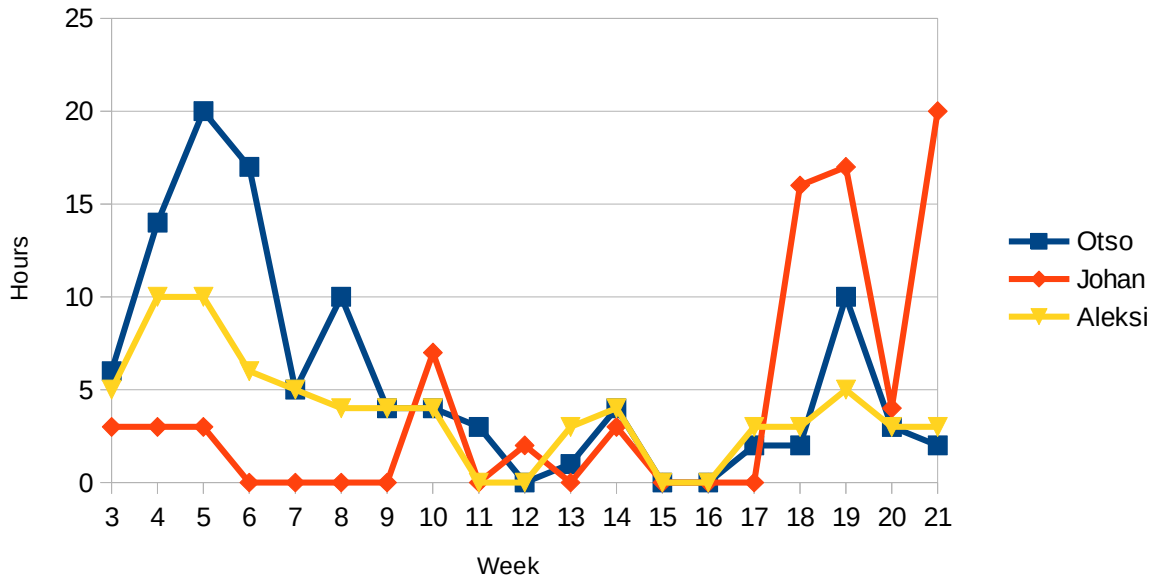

Figure 10: Time usage per team member per week

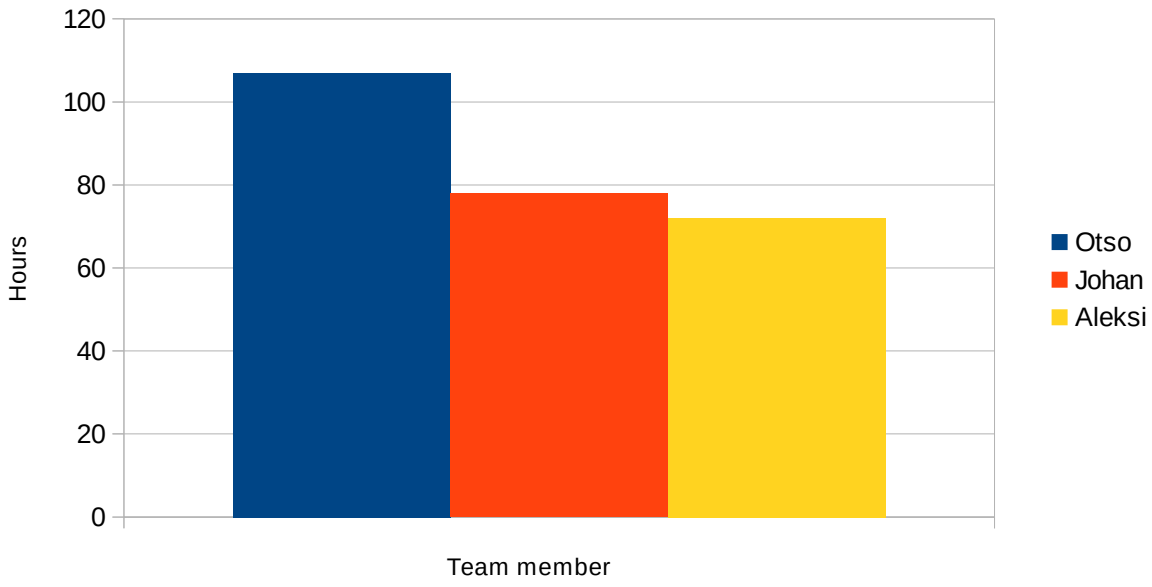

Figure 11: Total team member time usage

5 Future work and improvement suggestions

5.1 PCB revision 2014-02-22 errata

PCB had a few significant faults:

1) SDO-line used for relaying data to MCU required pull-up resistor, which was omitted. Fix: 4k7 resistor can be connected between test points for 5V and SDO. This has been tested and verified.
2) VREG should have an 1 uF bypass capacitor. Fix: Suitable capacitor can be added in isolator bypass footprint, even thought placement is far from optimal. This has been tested and verified.
3) VREF should have 1uF bypass capacitor. Fix: Capacitor can be connected between VREF- and DGND vias near LTC6803. This has been tested and verified.
4) External discharge transistors turn on seemingly at random. This might have something to do with internal pull-ups of LTC6803 behaving in an unexpected manner or some leakage current (maybe through zeners or capacitors?) getting amplified. Possible fixes: Either add external pull-up resistor (10 k) or remove discharge transistors to avoid bleeding batteries / reducing accuracy of ADC readings. Neither has been tested.

PCB also had some cosmetic faults, such as silkscreen texts being covered by components and MOSI / SCL lines have swapped labels near connector.

These mistakes have been incorporated in revised schematic available at [GITHUB].If another prototype seriers is made, these fixes should be included in PCB layout.

5.1 PCB revision 2014-02-22 improvements

In addition to previously mentioned errors, the circuit has some room for improvement, especially for more robust protection against faults. The bottom side of PCB is almost empty, protection circuitry for battery disconnection – and detecting faults – could be added there.

Power supply section could also benefit from a replaceable fuse which would limit the stress on protection zener in case of over voltage and reverse voltage.

References

[1] *KiCad EDA Software Suite*. Kicad-pcb.org. Retrieved 24-5-2014. Available at www.kicad-pcb.org/display/KICAD/KiCad+EDA+Software+Suite.

[2] *Design Simulation and Device Models*. Linear.com. Retrieved 24-5-2014. Available at http://www.linear.com/designtools/software/#LTspice.

[3] *Arduino Nano.* Arduino.cc. Retrieved 24-5-2014. Available at http://arduino.cc/en/Main/arduinoBoardNano.

[4] *ATmega48PA/88PA/168PA/328P.* Datasheet. Retrieved 24-5-2014. Available at http://www.atmel.com/Images/doc8161.pdf.

[5] *4 Layer 10 * 10cm Max – 10pcs*. Elecrow.com. Retrieved 24-5-2014. Available at http://www.elecrow.com/4-layer-10-10cm-max-10pcs-p-365.html

[6] *LTC6803-1 and -3 - Multicell Battery Stack Monitor.* Datasheet. Retrieved 24-5-2014. Available at http://cds.linear.com/docs/en/datasheet/680313fa.pdf.

[7] *Generic Standard on Printed Board Design.* Standard. Retrieved 24-5-2014. Available at http://sisko.colorado.edu/CRIA/FILES/REFS/Electronics/IPC_2221A.pdf.

[8] *ojousima/as-03200*. Github repository. Retrieved 24-5-2014. Available at: https://github.com/ojousima/as-03200.

[9] *Standard for Determining Current-Carrying Capacity In Printed Board Design* . Standard. Retrieved 24-5-2014. Available at http://people.senecac.on.ca/john.ebden/aed/IPC2152.pdf.

[10] *P-Channel 20-V (D-S) MOSFET.* Datasheet. Retrieved 24-5-2014. Available at http://www.vishay.com/docs/73702/si2351ds.pdf.

[11] *Super323 SOT323 PNP SILICON POWER (SWITCHING) TRANSISTOR* Retrieved 24-5-2014. Available at http://www.diodes.com/datasheets/ZUMT718.pdf.

[12] *Quad-Channel, 2.5 kV Isolators with Integrated DC-to-DC Converter.* Datasheet. Available at http://www.analog.com/static/imported-files/data_sheets/ADuM5401_5402_5403_5404.pdf Retrieved on 24-5-2014.