UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

# Battery Pack Cells Monitoring for Intelligent Charging

**Miguel Duarte Beirão**

Dissertation to obtain Master’s degree in
**Engenharia Eletrotécnica e de Computadores**

Advisor: Professora Doutora Maria do Rosário Alves Calado

**Covilhã, June 2016**

# Acknowledgements

I would like to thank my teacher Maria do Rosário Alves Calado, for accepting and supporting me on this challenge to get me to be a master in engineering. Despite the countless amount of obstacles, not a single one made this project fall apart due to the teacher’s availability and help.

Then I would like to thank the person that deserves the most out of everyone, Engineer José Pombo. I could never have gotten this far without him, his knowledge and teachings. Even though he had no obligation, he was able to give me the strength to work more and more every day. An incredible person that always knew when to get mad and when to be supportive, being patient no matter how many times I needed his help. A humble and intelligent person that no doubt would be an incredible acquisition to any university as a teacher.

To my old friends, who I would have pages and pages filled with stories that got me to be the person who I am today. No matter what, no matter when, they were always there to lend me a hand rising me back up after tripping on any obstacle. Hoping I won’t forget any of them, I want to thank Carolina, Sofia, Anabela, Raquel, Inês Jorge, Pedro S., Henrique, Pedro A., who patiently listened, supported and commented on my explanations on this project, making me feel better every time.

To all my university colleagues, that made this whole time be special and unforgettable from day one. More than colleagues, friends that I hope I can keep for life, teaching and learning each day a little further not just on work, but on life itself. A countless amount of names would be written here if I had to thank one by one. I hope each and every single one knows how much I should thank them, for all the time we’ve been together.

To my parents, that are my professional role models, who work hard every day, more than they have to, to make my life easier and better. My parents who despite all my mistakes, all my flaws, all my bad mood, keep supporting me, showing me I can always achieve better goals and aim for the stars. Without them, I would never be able to be where I am today, to be who I am now.

Thanks!

# Abstract

This dissertation intends to create a system capable of cell charging, cell balancing or both at the same time for batteries with multiple cells connected in series. It also tries to understand why there is only few literature connected with cell balancing and cell charging at the same time. For that purpose, this dissertation presents a review on the state of the art of many concepts related both to balancing and charging in order to pick the right methods and equipment to achieve the objectives of this work. This dissertation includes literature review on batteries, cell balancing methods and topologies, cell charging methods and a small review on state of charge estimation methods. Later on, this document studies and explains hardware and software requirements and choices in order to understand the final developed circuit. Lastly, development difficulties, results and conclusions are presented.

# Keywords

Cell balancing, BMS, Battery Management System, battery charging.

# Resumo

Esta dissertação pretende criar um sistema capaz de carregar, balancear ou ambos em simultâneo num pack com diversas células ligadas em série. Tenta ainda perceber a razão de haver tão pouca bibliografia que junte em simultâneo carregamento e balanceamento de baterias.

Para alcançar estes objetivos, esta dissertação conta com uma revisão do estado da arte de vários temas relacionados tanto com balanceamento como com carregamento de forma a perceber os métodos e equipamentos mais adequados para implementar. A dissertação inclui revisão bibliográfica em baterias, métodos de balanceamento e suas topologias, métodos de carregamento de baterias e ainda alguma revisão sobre métodos de estimação de estado de carga.

Posteriormente, este documento estuda e explica os requisitos de *software* e *hardware* e as escolhas feitas para o desenvolvimento do circuito. Finalmente apresentam-se as dificuldades de desenvolvimento encontradas, os resultados e ainda algumas conclusões.

# Palavras-chave

Sistema de Balanceamento de Baterias, BMS, Carregamento de Baterias.

# Table of Contents

## List of Figures

# Nomenclature

| | |
|---|---|
| AC | Alternate Current |
| BMS | Battery Management System |
| CCCV | Constant Current Constant Voltage |
| CCS | Code Composer Studio |
| DC | Direct Current |
| DoD | Depth of Discharge |
| GUI | Graphical User Interface |
| MOSFET | Metal-Oxide Semiconductor Field Effect Transistor |
| OCV | Open Circuit Voltage |
| PHEV | Plug-in Hybrid Electric Vehicle |
| PRE | Produção em Regime Especial |
| REN | Redes Energéticas de Portugal |
| SCI | Serial Communication Interface |
| SCPI | Standard Commands for Programmable Instruments |
| SIMO | Slave in Master Out |
| SOC | State of Charge |
| SOH | State of Health |
| SOMI | Slave Out Master In |
| SPI | Serial Peripheral |
| TTL | Time to Live |
| UBI | Universidade da Beira Interior |
| USB | Universal Serial Bus |
| V2G | Vehicle to Grid |

# Chapter 1 - Introduction

“God wants, man dreams, the work is born” (Fernando Pessoa)

## 1.1 - Topic Introduction

One of society's current concerns is the high consumption of fossil fuels, the possibility of its depletion and the emission of toxic gases released to the atmosphere because of their use. All this reasons make it really important to find an alternative to this kind of energy sources in order to avoid problems that come from the kind of fuel. Vehicles take a big role in the consumption of fossil fuels and all the pollution that they create so it is mandatory to research and develop vehicles that use less and less this kind of fuels and use other resources as a source for its movement [1].

With the development of Plug-in Hybrid Electric Vehicles (PHEV) and the Vehicle-to-Grid (V2G) concept there is the possibility to, with the first one, reduce the pollution that vehicles currently emit to the atmosphere and with the second one reduce the pollution that thermal power stations produce in order to generate energy [2].

Even reducing the usage of fossil fuels in vehicles, it is not a great step if the energy to charge the vehicles batteries comes from power plants that generate electric energy by burning fossil fuels. With the introduction of renewable energies like solar and wind energy there is the need to compensate the fact that these are unpredictable energies and so it is impossible to rely uniquely on them. That compensation could be achieved with thermal power plants working in full-time at low levels, in order to compensate the grids needs at any time, being pretty dispendious for a country. In order to avoid this energy availability problem, there is the possibility to store energy in batteries when there is too much solar, wind or hydraulic energy available and compensate the fluctuation by discharging these batteries [3].

A big step on energy storage systems comes from researching optimization algorithms for balancing batteries in order to store more energy without having a great impact on the systems size [4].

## 1.2 - Work Motivation

Looking at Redes Energéticas Nacionais (REN) website it is possible to understand that the special regime production sometimes overpasses the energy consumption of the country, making it easy to realize that energy is being wasted. This makes it really important to store this excessive energy in order to use it later on high consumption peaks reducing the usage of fossil fuels power plants. Coupled to this subject there is also the fact that household solar panels usually have energy production peaks that are discrepant in time with the house energy consumption peak, which is one more motive for storing energy in batteries.

Out of this motives comes the fact that, the better the storage system, the more energy it is possible to store allowing the correction of the time discrepancy between energy production peak and energy consumption peak, taking a positive impact on household economy.

All that was previously mentioned originated the idea of optimizing batteries charging in order to increase the amount of energy that can be stored in each charging cycle. As there are lots of battery dependent applications the idea grew to the management of any kind of li-ion batteries with the possibility of managing low voltage batteries like those of the laptops, which operate around 19 V and high voltage ones like those used in electric vehicle , which operate at around 400 V.

## 1.3 - Dissertation Structure

This dissertation has the following structure:
Chapter 1 – Introduction – Summarizes the developing project;
Chapter 2 – Literature Review – Studies the previously existent knowledge related to this topic;
Chapter 3 – Circuit Development – Explains the plans for developing the system after achieving the required knowledge;
Chapter 4 – Results – Presents the results acquired in each stage of developing this system;
Chapter 5 – Conclusions – Interprets the results and compares the results to initially defined objectives and suggests future researches related to this topic.

Each Chapter is divided in Sections and Sub-Sections that may or may not have Introduction or Conclusion depending on whether it was considered important to include.

## 1.4 - System Sketch

As charging depends on the battery and its characteristics in each moment, meaning, depends on State-of-Charge, battery capacity, current limits, and battery topology, it is necessary to include a microprocessor to manage all the data and control the whole system during the whole charging procedure. Also, to acquire the parameters previously mentioned, it will also be necessary the use of a Battery Management System (BMS) connected to the microprocessor so that this can decide how to behave during its operating time.

To charge the batteries, it is also necessary a variable power supply that can be controlled by the microprocessor in order to adapt to the charging depending on the decisions taken by the microprocessor.

Finally, as to test the charging procedure there is the need of having the batteries discharged, it is also necessary to include a discharging system capable of removing some energy from the battery pack.

With the previously presented sketch it is possible to create a schematic of how the whole system will be. In Figure 1 there is an overview of the whole system, where there is the microprocessor, the BMS, the power supply, the battery and a discharge system.

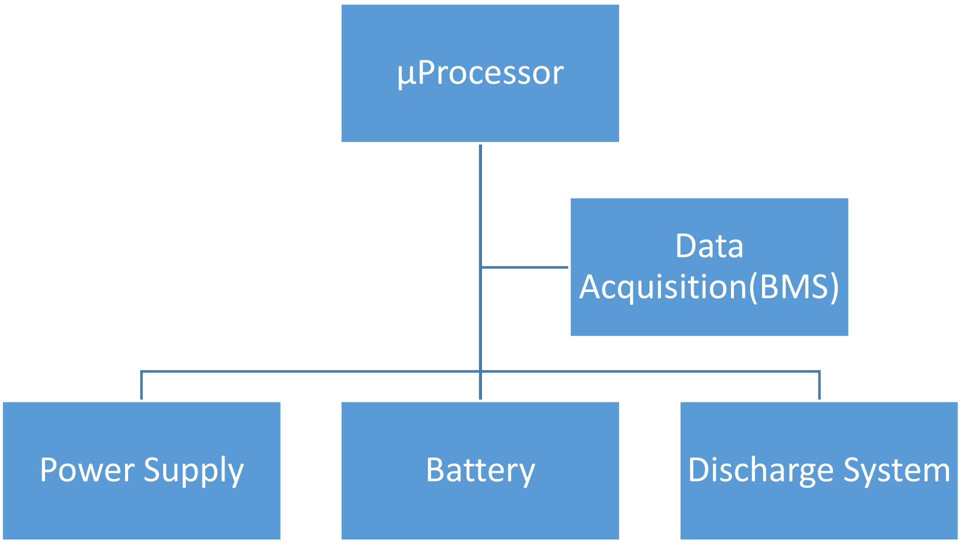


*Figure 1 - System Block Diagram*

## 1.5 - Research Methodology

It is common sense that, as time goes by, evolution is an unstoppable process in everything that is a research topic. Though sometimes it happens by accident, most of the times it happens as the result of many research processes. This makes it important to understand how a research project should be elaborated.

Research comes from the need to answer a problem that has to be accurately formulated in order to understand specifically the kind of information and resources necessary in order to obtain satisfactory results [5].

Applied Research is usually a kind of research that is made in the interest of an area, meaning that it tries to answer people or companies current needs originated by a problem they may have stumbled upon. Joint to Applied Research there might also be an Experimental Research in which after choosing the study object, some parameters are tested in order to answer the problems needs when it was formulated [6].

Although there are many ways to categorize a research, by looking at the literature attached to this document, this research project can be classified as an Applied, Experimental Research as it tries to answer the need for optimizing batteries capacity usage by creating a circuit capable of doing it.

*“Modern planning conception, based on Systems General Theory, consists of four elements: process, efficiency, scheduling and goals.”*

# Chapter 2 – Literature Review

“A mind needs a book as a sword needs a whetstone if it is to keep its edge” – George R. R. Martin

## 2.1 - Batteries

### 2.1.1 - Introduction

Nowadays, an energy storage system is a multiple studies topic as it is possible to understand by the amount of articles emerging everyday related to it [7]-[9]. Each type of battery has its advantages and disadvantages depending on the system they are being implemented on, and it is up to the project designer to decide on the type.

This section studies the various types of batteries in order to select what type is going to be used in this project. Furthermore, an analysis of the chosen battery type mathematic model is made in order to understand its behaviour in the various phases of charging and discharging process. For that analysis the simulation environment was Simulink due to the previous experience the student had with it.

The software provides a help topic where it was easy to find the article that explains the mathematical model used for the program to simulate the battery component. This article shows the complexity of the battery and explains the need of making some approximations in order to obtain the charging and discharging curves [2].

In this section it is possible to understand the type of batteries usually used in this kind of applications where the batteries are frequently charging and discharging. It is also possible to understand the behaviour and the advantages and disadvantages of each type.

### 2.1.2 – Study on Battery Types

In order to pick which one is the most suitable type of battery for this project, it is important to check on which ones can be used, its characteristics and its behaviour.

There are four major types of battery, Lithium-Ion (Li-Ion), Nickel-Cadmium (NiCd), Nickel-Metal-Hybrid (NiMH) and Lead-Acid which have their curves presented in Figure 2 and Figure 3.

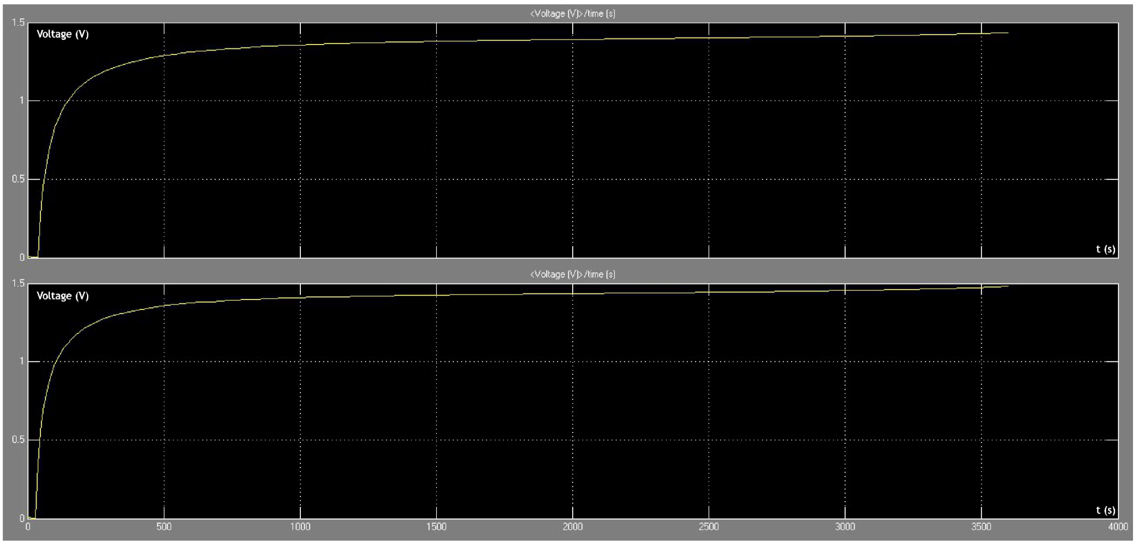


*Figure 2 - Nickel-Cadmium (top) Nickel-Metal-Hybrid (bottom) Charge Curves*

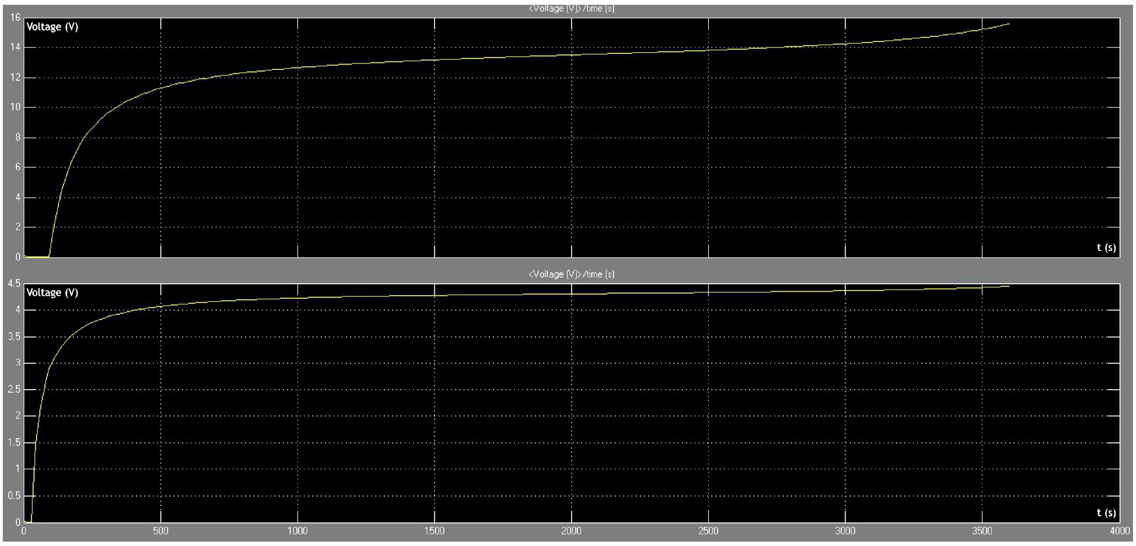


*Figure 3 - Lead Acid (top) Li-Ion (bottom) Charge Curves*

Based on other battery values, each of these batteries was charged at a current equal to the value of its capacity charging in one hour as seen in all of the pictures.

It is easy to conclude that except for the Lead-Acid battery, the other three types have similar start and end of charge behaviours, with different voltage values. For the conditions previously explained, as the curves do not show much of a difference for a decision to be taken, the chosen chemistry for the batteries is Li-ion due to being the ones easier to acquire for the University laboratory and the most used ones for this kind of applications.

### 2.1.2.1 – Li-ion Battery Model

For the used mathematical model a few assumptions were made **[2] [10]**:

- Constant internal resistance;
- Charging and discharging curves are similar;
- Battery's capacity does not change;
- There is no temperature influence.

There are also a few limitations in what comes to its characteristics:

- Voltage goes from 0 to infinite;
- It's capacity is not limited, making it possible to surpass 100% SOC [10].

Concerning Figure 4, it is possible to distinguish three different phases. Considering the discharging curve, when the battery is fully charged and gets to be discharged at constant current, it starts by going down the exponential area, having the phase end at the start of the nominal area where even though state of charge (SOC) keeps decreasing at a constant rate, voltage barely gets to change. After the nominal phase the battery's voltage starts falling pretty fast until the end of the curve that, in this case, as it is a mathematical model reaches 0 V. In a real one it should stop much earlier in order to keep the cell undamaged.

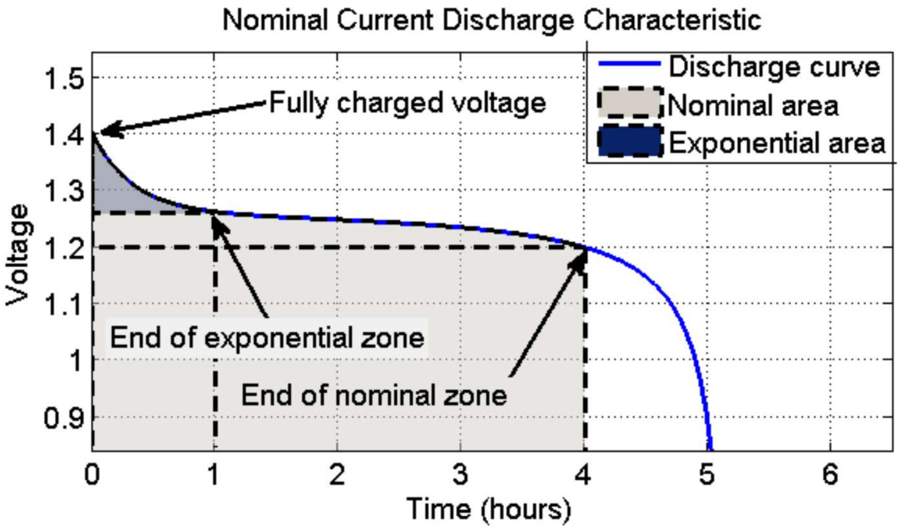


*Figure 4 - Li-Ion Discharge Curve (adapted from [2])*

Despite all these details, the mathematical model gets a pretty good approximation on charging and discharging curves when compared to the curves that are available in the literature.

Charging and Discharging Li-ion batteries models used in Simulink are based on the equations (a) and (b):

Charge:

$$V_{batt} = E_0 - R.i - K\,\frac{Q}{Q-it}\,.i^{*} - K\,\frac{Q}{Q-it}\,.it + A_{exp}\,(-B.it) \tag{2.1}$$

Discharge:

$$V_{batt} = E_0 - R.i - K\,\frac{Q}{Q-\,it}\,.(it +\ i^{*}) + A_{exp}\,(-B.it) \tag{2.2}$$

In which:

$V_{batt}$ is the battery Voltage (V),

$E_0$ the battery Constant Voltage (V),

$K$ the Polarization Constant (V/(Ah));

$Q$ is the Battery Capacity (Ah);

$it$ the= Battery Real Charge (Ah);

$A$ the= Exponential Zone Amplitude (V);

$B$ the exponential zone time constant inverse (Ah$^{1}$);

$R$ is the Internal Resistance (Ω);

$i$ the Battery Current (A);

$i^{*}$ represents the= Filtered Current (A);

Since it is an injective function it is possible to associate to each voltage value a SOC one so an estimation of SOC's value can be achieved by measuring battery's voltage.

## 2.2 – Cell Balancing and Existing Methods

### 2.2.1 – Introduction

Nowadays society lives more and more dependent on mobile devices and their portability is an important factor when picking an equipment. Equipment's charge duration depends not only on its energy consumption but also of the quality of the battery that provides the energy for the device. In what comes to the battery, there are many parameters that can influence its potential like the capacity, the size, etc… [11]. Even more important is the management of the battery since when it is not made in the right way the battery can store much less energy than the one it actually is capable of. A battery without the right management might use only between 20% and 80% of its total capacity, meaning, only 60% of its real capacity. By using a battery management system it is possible to improve this capacity usage by working between 5% and 95% increasing 30% of the real battery capacity usage [12].

The fact that a battery pack is usually made of serially connected cells makes it really important to understand what implications come out of that kind of connections. It's practically certain that a battery with multiple cells can't have them all exactly equal, making them charge and discharge differently due to its differences [13], [14].

Picturing a battery with 5 serially connected cells in which each cell is fully charged when it reaches 4.2 V, the battery will have a 21 V level. As cells get to age, they start developing differences between them originating discrepancies when charging. Even if charging with the same current flowing through all of them they will be changing their SOC at different rates. Because of these differences, when achieving a 21 V reading without a balancing system, it is possible that some of the cells get to be damaged as some of them might be at levels lower than 4.2 V making some other cells to have achieved voltage levels higher than 4.2 V in order to have a 21 V reading. This might cause overcharging and overheating, damaging the cells. That's why systems without cell balancing methods have to tighten up the charging limits, sacrificing some of the energy storage capacity in order to avoid this kind of problems [15], [16].

There are multiple balancing methods. Each one has their pros and cons that might influence the decision on which method to apply. The Figure 5 presents the methods that were found in literature for cell balancing.

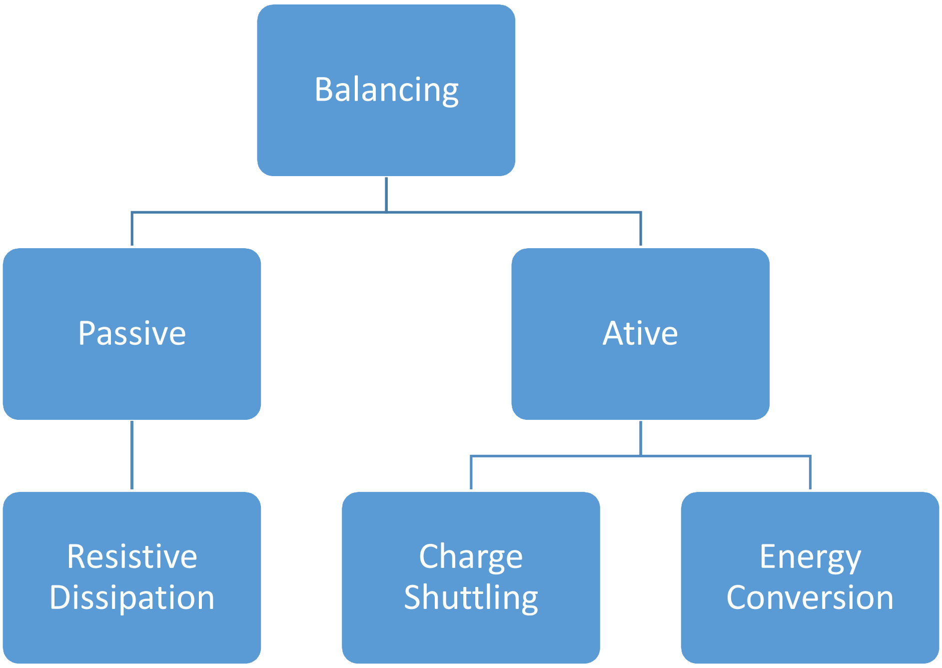


*Figure 5 - Balancing Methods*

In this section, an analysis on the state of the art on battery balancing method is made. The various methods and topologies that exist and the advantages and disadvantages of each one of them are presented in order to understand what the most suitable choice for this project should be [17], [18].

Figure 6 shows how laptop and vehicle batteries are on the inside. It is possible to understand they are created by stacking lots of small cells connected in series and parallel depending on the objective.

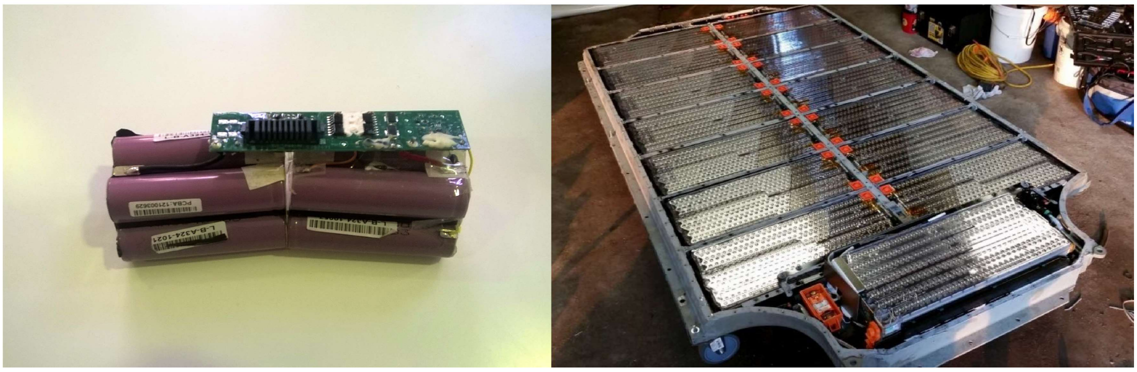


*Figure 6 - Laptop Battery (left) and Electric Vehicle Battery (right)*

In this section topologies figures include a radar plot on the right side presenting a study on eight parameters, Cost, Modular Implementation, Volume, Implementation Complexity, Efficiency, Speed, Switching and Control Complexity rating in three levels where level 1 is “bad” for the system and 3 is “good” [19], [20]

### 2.2.2 – Active Balancing Methods

#### 2.2.2.1 – Charge Shuttling

This balancing method consists of connecting a parallel capacitor with the most charged cell, transferring energy from the cell to the capacitor. On the following step, that connection is undone and a new one is made between the charged capacitor and a less charged cell transferring the “extra” energy from the capacitor to that other cell. Doing this enough times will get the whole cells to get to the similar levels, in what comes to storing energy.

Due to the amount of studies that were already made on this topic, lots of topologies exist where the charge can be transferred from cell to cell, pack to cell and cell pack to cell making the balancing the most efficient and versatile as possible with the implemented topology [20], [21].

Figure 7 presents a Switched Capacitor topology where balancing is achieved from transferring energy from one cell to an adjacent one by using a parallel capacitor to achieve the balancing goal [17], [20], [22], [23].

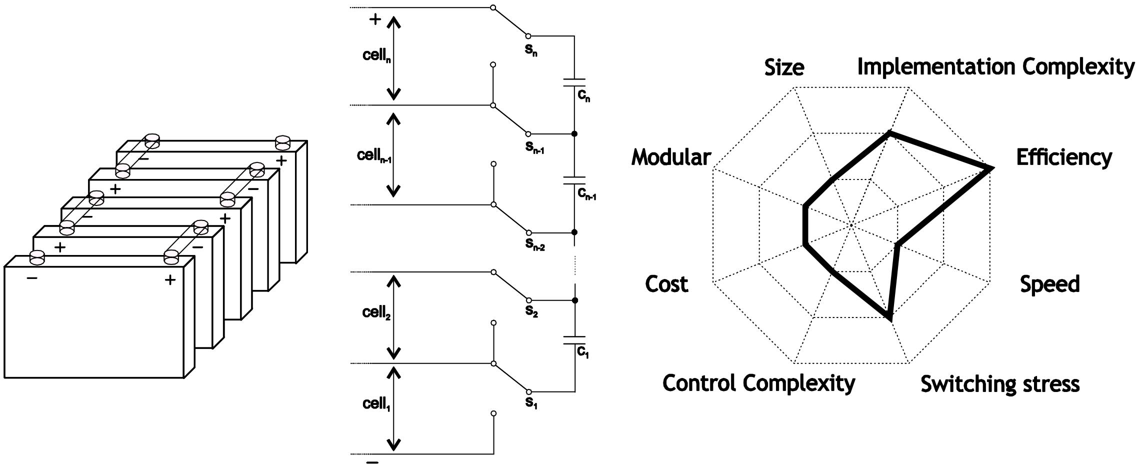


*Figure 7 - Switched Capacitor*

Figure 8 illustrates a Double Tiered Switched Capacitor where the charge shuttling can be made from one cell to another adjacent cell or from a pair of cells to an adjacent pair of cells accelerating the balancing process when compared to the previously presented topology [17], [20], [22], [23].

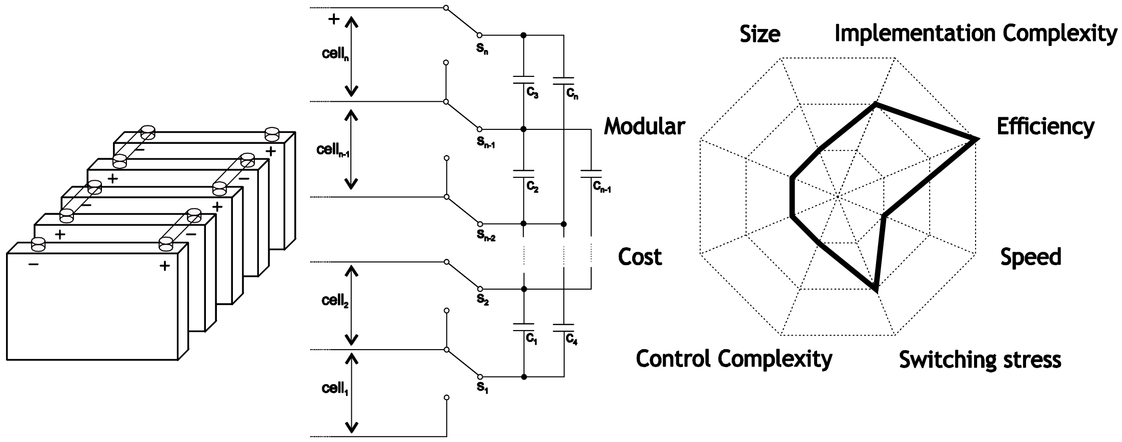


*Figure 8 - Double Tiered Switched Capacitor*

The Single Switched Capacitor topology is presented in Figure 9 and it allows for the charge shuttling to me made from one cell to any other as long as the switches are properly managed by a controlling algorithm [17], [20], [22], [23]. The radar plot evaluating the eight characteristics previously mentioned is included in the figure.

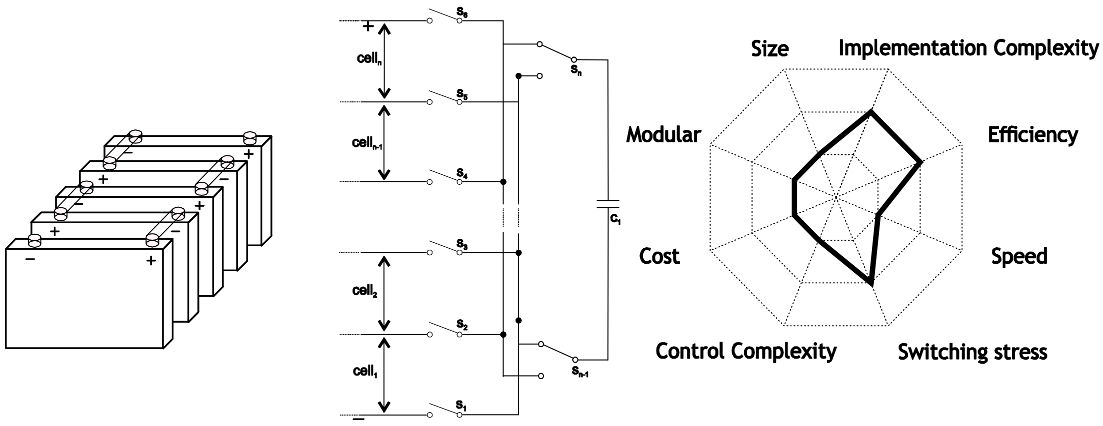


*Figure 9 - Single Switched Capacitor*

Modularized Switched Capacitor topology is presented in Figure 10 and it allows to transfer energy between adjacent cells and, as there are other battery packs present, it allows for them to shuttle energy between them too, approaching their voltage levels.

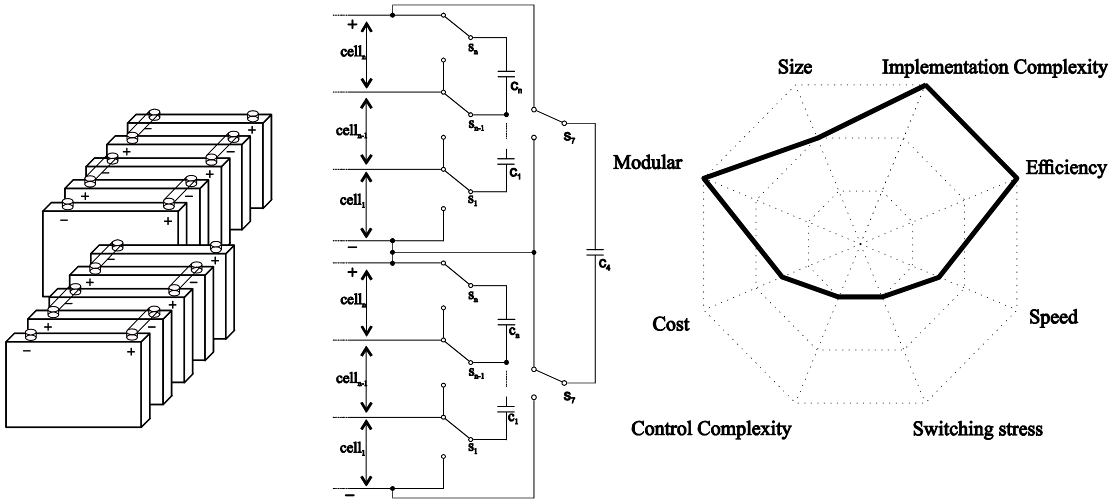


*Figure 10 - Modularized Switched Capacitor*

### 2.2.2.2 – Energy Conversion (DC-DC)

This kind of balancing can be split into two distinct groups, the isolated converters group and the non-isolated converters one.

When balancing, this method uses the battery to supply the converter that will then transfer energy to one or multiple cells in order to balance them. There are numerous topologies having some differences in implementation, complexity and balancing procedure.

Starting on the isolated converters group, the multiple winding transformer topology presented in Figure 11 uses a transformer with one primary coil and as many secondary coils as the number of cells in the battery pack in order to balance them.

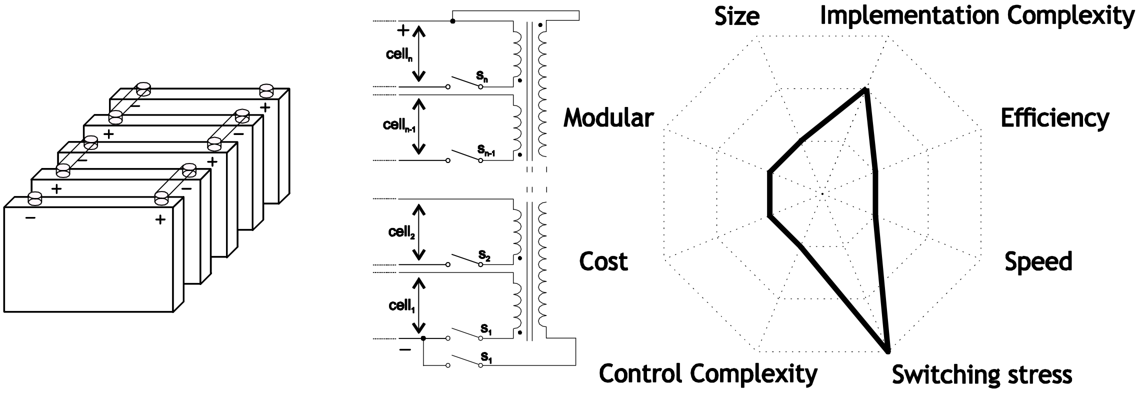


*Figure 11 - Multiple Secondary Winding Transformer*

A variation of the previous topology is the multiple transformers one presented in Figure 12 where there is a primary and a secondary coil for each cell, controlling the balancing with the help of a switch on each primary coil circuit.

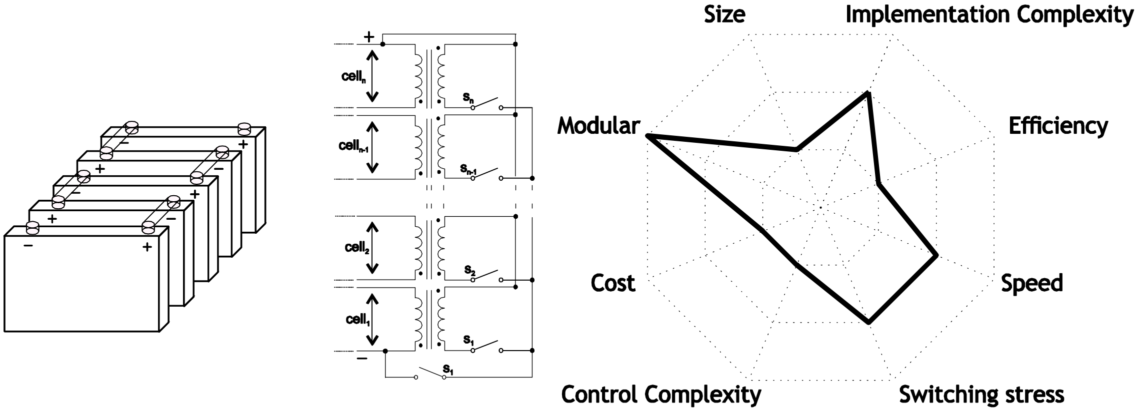


*Figure 12 - Multiple Transformers*

A topology that has much less coils but much more switches is the switch matrix transformer one, which has two switches per cell allowing for the balancing of any cell as desired. Figure 13 presents the schematic of that topology.

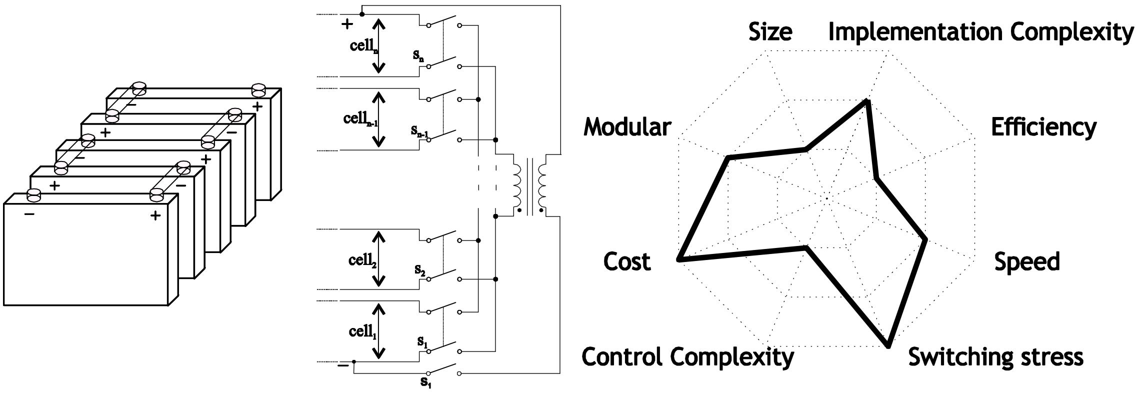


*Figure 13 - Switch Matrix Transformer*

Finally, the topology based on dual active bridge converters completes the isolated converters group study which has a converter for each pack cell. It will have 8n switches, where n represents the number of cells in a pack. Presented in Figure 14, this is a bidirectional topology, allowing the energy to flow both ways [24], [25].

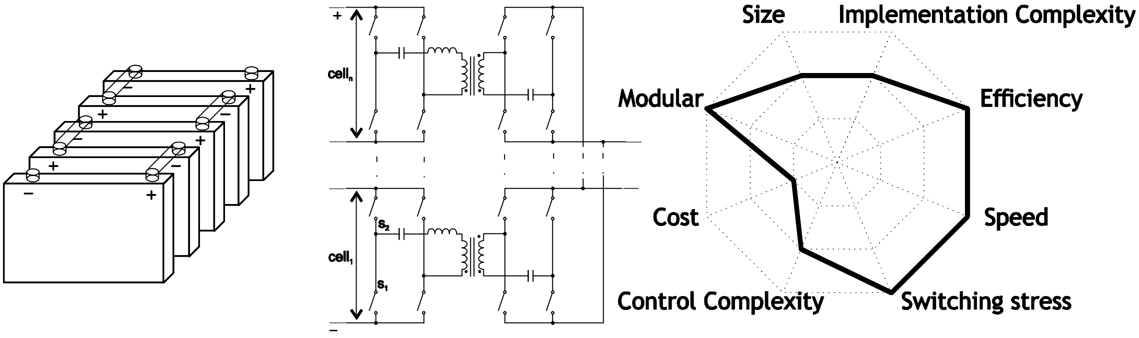


*Figure 14 - Dual Active Bridge Converter*

For the non-isolated group, the topology based on a DC-DC non-isolated buck boost uses that converter in order to balance the cells. The topology is presented in Figure 15 and it's a bidirectional topology allowing for energy transfer from pack to cell and cell to pack [26], [27].

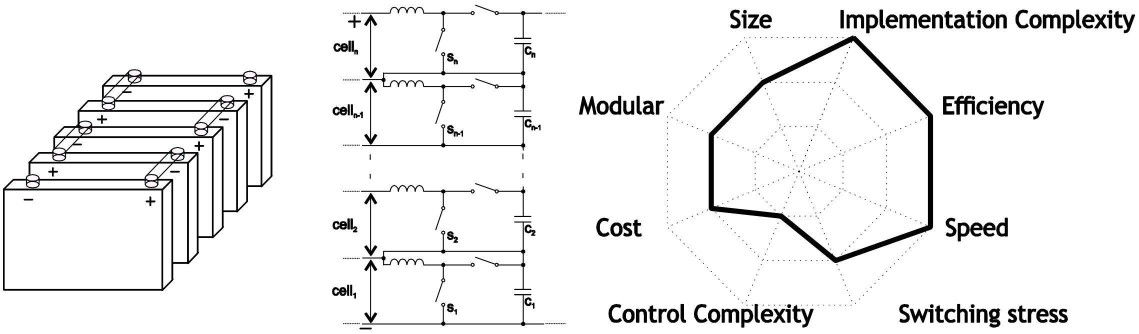


*Figure 15 - Buck-Boost Converter*

One more topology is presented in Figure 16 and is based on the Cuk converter. This topology uses n capacitors and 2n switches to balance the cells, where n represents the number of cells in a pack. The disadvantage of the presented topology is that it can only transfer energy to adjacent cells[20], [28].

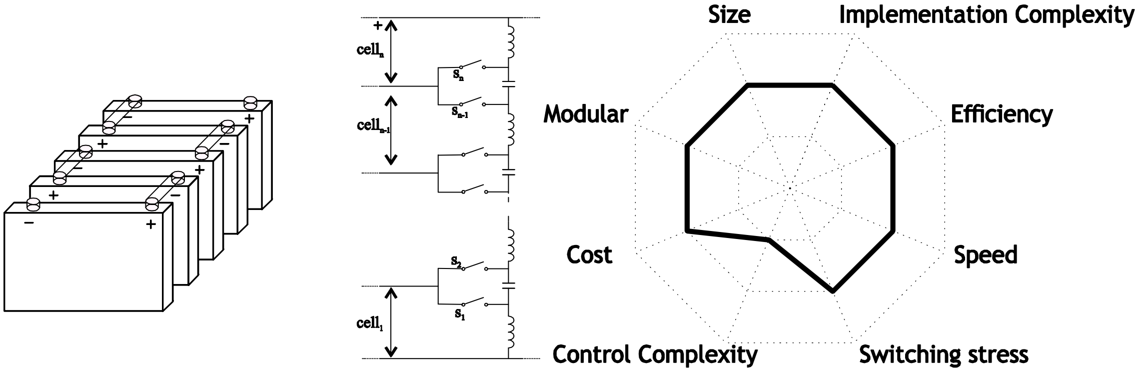


*Figure 16 - Cuk Converter*

## 2.2.3 – Passive Balancing Methods

### 2.2.3.1 – Dissipative Resistors/Cell Bypass

This topology places a high resistor in a parallel connection with each cell, where the current passage on this resistor is controlled by a switch that receives its commanding signals from the Battery Management System. By controlling the switches the system can discharge or force a slower charging on one or more cells in order to balance them with lower voltage cells. It is extremely important to pay attention on which resistors are used for this method since they will be dissipating energy as heat. The high value on the resistors allows for the discharging current to be small in order to have a low impact on the cells.

Figure 17 presents the switched shunt resistor which is the most used topology for the dissipative resistors method. It is a widely used method since its characteristics are pretty appealing [20].

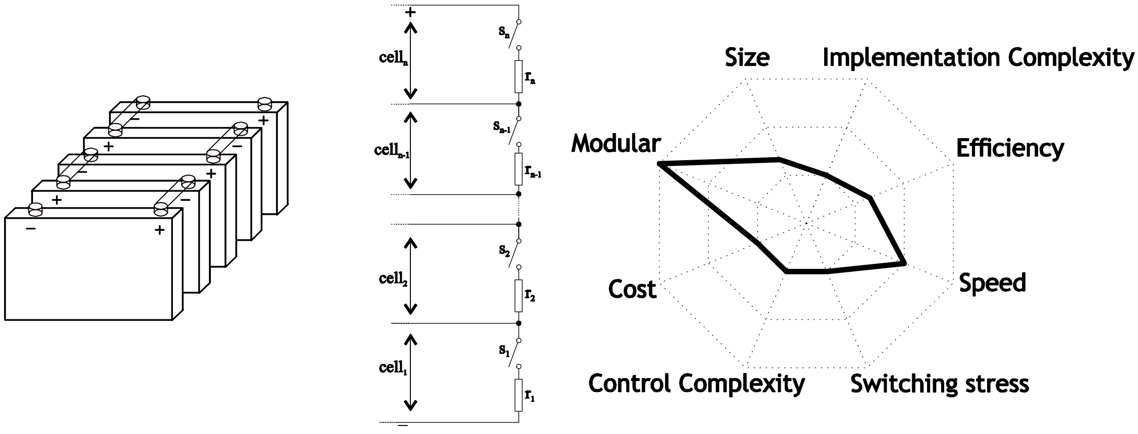


*Figure 17 - Dissipative Resistors*

This is a pretty simple method with a low implementation cost even though its purpose is to dissipate energy, which is a disadvantage when compared to previously presented topologies. As it consists of dissipating energy through heat, it makes it really important to control the temperature in order to avoid damaging any component. There are some other less used methods like fixed shunt resistor, shunt transistor and complete shunting… [20].

## 2.2.4 – Section Conclusion

This section allows for the understanding of many of the balancing methods and topologies existent in the literature and explains the advantages and disadvantages on the implementation of every single one of them. It also helps understanding the method and topology to use in this project considering the resources available and the application goal.

After analysing the advantages and disadvantages of each of the methods and topologies presented and considering the goals and equipment limitations for this project the chosen method was the dissipative resistors method that is a passive balancing method and the chosen topology was the switched shunt resistor one in order to balance the cells on a battery pack.

A comparison on the four mentioned methods was achieved with the help of the literature, making it possible to create the radar plot presented in Figure 18.

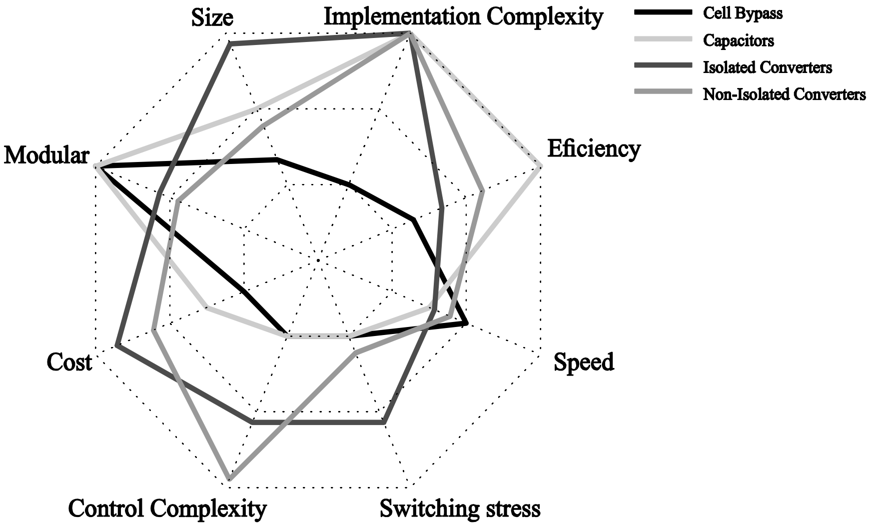


*Figure 18 - Balancing Methods Comparison*

## 2.3 - State of Charge (SOC)

### 2.3.1 - Definitions

State of Charge, from now on mentioned as SOC, can be defined as the relation between currently available capacity and available capacity at full charge state. A 0% SOC means a fully discharged battery and a 100% SOC means a fully charged one [7], [8].
Though it won't be considered in this project, literature also refers on the term State of Health (SOH), which is intimately related to SOC. SOH can be defined as the relation between current maximum capacity and maximum capacity by the time the battery was produced. From this new concept it is possible to think that, a 100 Ah battery with 100% SOH and 50% SOC has 50 Ah but the same 50% SOC on an 80% SOH battery will only have 40 Ah [7].

SOH can be calculated with the following equation:

$$SOH = \frac{C(t)}{C(t_0)} \ , \ t \geq 0 \tag{2.3}$$

One SOH estimation method is acquired by measuring pressure differences in cells [7].
One more possible method to determine SOH would be fully discharging and charging the battery, measuring the impedance with the help of dedicated hardware or software measuring battery's AC or DC resistance. SOH can be related to the impedance of the battery as they both increase as it grows old [29].
Alternatively it is possible to consider the Depth of Discharge (DoD) which is the opposite of SOC. This measures the depth of discharge and usually gets referred when batteries start to grow old. The equation (2.2) may be used:

$$DoD(t) = \ 1 - SOC(\%) \ , \ t \geq 0 \tag{2.4}$$

There are various methods to estimate SOC: Coulomb Counting method, electrolyte physical properties measurement, OCV test, etc… [29].

### 2.3.2 – State of the Art

There are a lot of recent studies and articles on SOC estimation in order to find new and better methods for it. Open Circuit Voltage (OCV) connected to SOC [1], *Coulomb Counting Method* [30], *Kalman Filtering* [31] and *Neural Network* [31], *etc…*

In order to choose a method for this project, it is necessary to understand which ones can be used in the system, and which ones can be implemented for the batteries that this work pretends to study. For that it is necessary to understand what each method's requirements are making it possible to choose the most suitable one. This section resumes a few methodologies studied for this project.

#### 2.3.2.1 – SOC and OCV Relation

Open circuit voltage is intimately related with the state of charge of a battery. It is possible to estimate SOC through OCV as they form an injective function.

The major problem in this method is that it is necessary to use a complex and pretty well developed model in order to keep the estimation value close to the real one. Though in end of charge and end of discharge this method is pretty good, in middle charge it requires a very precise voltage measurement in order to estimate the SOC value correctly. After a while, due to the changes in SOH, the model might get to be less accurate [9].

Figure 19 presents a Simulink test that was made by connecting a fully discharged battery element and a current controlled power supply in order to force a constant current into the battery. This allows for the understanding on how SOC and Voltage behave while charging at constant current. As previously explained the initial phase and ending phase have an acceptable voltage variation but middle phase can get pretty tricky for estimating SOC as it varies really slowly.

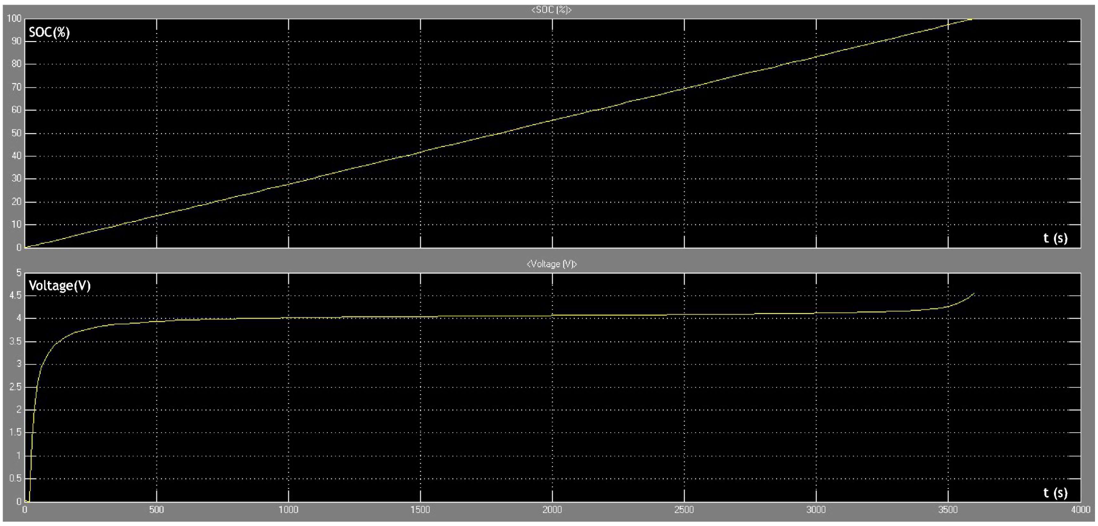


*Figure 19 - Li-Ion SOC vs Voltage*

### 2.3.2.2 - Coulomb Counting Method

The Coulomb Counting Method [30] consists of measuring the current that goes in and out of the battery. To determine the SOC, this method uses an algorithm that considers current provided or received by the battery. Equation 2.3 explains the calculation method:

$$SOC(t) = SOC_{init} - \int_0^t \frac{i(t)}{C_n} \ , \ t \geq 0, \tag{2.3}$$

in which $SOC(t)$ is the current SOC that the algorithm is trying to calculate through this method, $SOC_{init}$ is SOC's value by the time the measuring system is connected to the battery, $C_n$ is the battery's full capacity and $i(t)$ represents the current going in or out of the battery in order to time.

$SOC_{init}$, due to being hard to calculate when connecting to the system, is usually 100% or 0% adjusted when the system is recalibrated by fully charging or discharging the battery.

A great setback on this method is that it suffers from great discrepancy as times goes by so it needs to be recalibrated frequently in order to acquire a reference value.

Due to the need of a reference value this method is not very suitable in cases where the battery is disconnected from the measuring system frequently as it loses reference when disconnecting and so, real SOC value is lost [30].

### 2.3.2.3 - Kalman Filtering

The objective of the Kalman Filter method is to clear out the errors on data acquisition in order to obtain correct information related to the cell. These inputs usually are the internal resistance, the temperature and the current flowing to the cell and the output is usually the voltage. Although this is claimed to be a pretty accurate method on predicting and calculating a battery's state-of-charge, this method has a lot of requirements such as the need for temperature measurement, the need for a pretty accurate battery model and processing capacity.

### 2.3.2.4 - Neural Networks

Neural Networks have self-learning abilities which allow them to improve the system's capacity to estimate battery's state-of-charge [31]. It needs no knowledge on the battery characteristics nor what type of battery it is using as it adapts with the help of training data meaning it doesn't need many attached equipment in order to calculate SOC. The great disadvantage on this method is that it needs really good training data in order to make it a good prediction method with calculated values close to the real ones.

### 2.3.3 – Section Conclusion

As the main purpose of calculating state-of-charge in this system is to understand when to stop charging or discharging procedures, the first method presented is accurate and simple enough to be implemented on this system. By understanding which the cell voltage limits are and considering the end of charge and end of discharge behaviour on the cell battery type used in this project it is possible to detect with an acceptable accuracy if the cell is fully charge or fully depleted.

This means that the SOC and OCV relation method is the chosen one for implementing in this system.

# 2.4 - Battery Charging

## 2.4.1 - Introduction

As this project has a goal of implementing a continuous balancing algorithm during charging, it becomes extremely important to understand what literature's options for charging the batteries are. The importance of this topic only came up when the testing phase started as the initially implemented charging method was inadequate for what was pretended for the system.

Voltage readings when a cell is in open circuit is different from the voltage readings when it has current flowing through the cell which may lead the system to take wrong decisions due to those reading discrepancies. Reading discrepancies between the cell's real voltage and the cell's read voltage increases with the growth of the current flowing through the cell. This makes it really important to find a system that is not influenced by this phenomenon.

This section intends to explain the various methods found in the literature and justify the chosen method for charging this system.

## 2.4.2 - Charging Methods

In the literature a few charging methodologies can be found and they can be split into three distinct groups where one of them can be split into two subgroups as they differ in their implementation and working procedure.

Figure 20 presents a diagram that resumes the methodologies found.

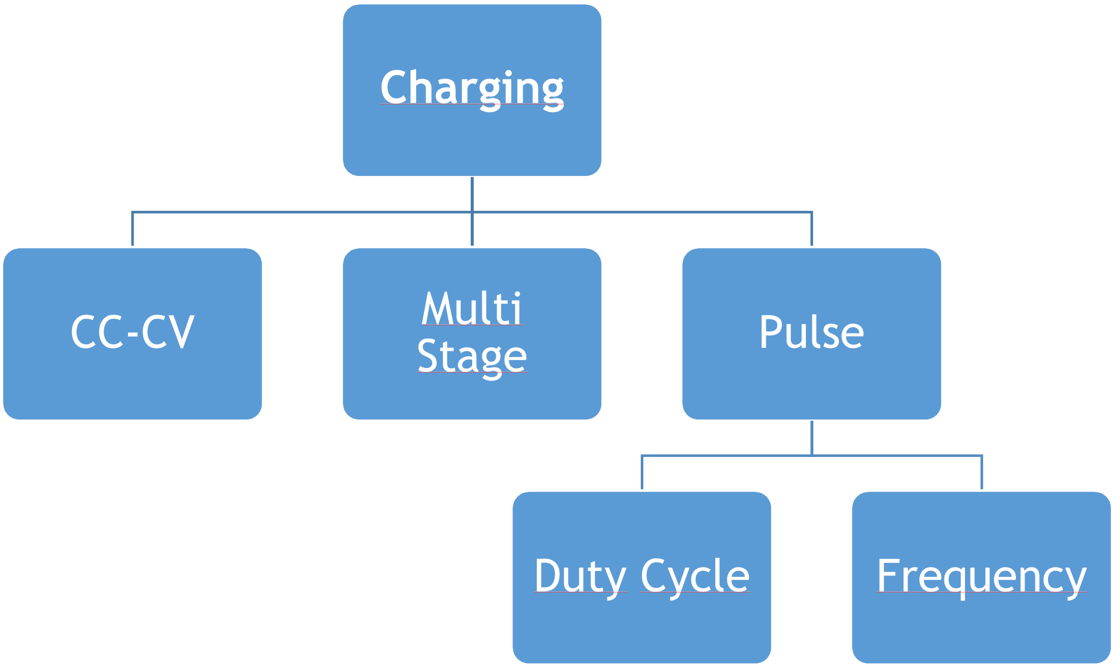


*Figure 20 - Charging Methods*

In this section, a radar plot was added to the figure that explains each method, evaluating the characteristics of each one in. The radars present a three level rating, in which 1 represents “bad” and 3 represents “good”.

### 2.4.2.1 – *Constant Current / Constant Voltage Charging Method*

Constant Current/Constant Voltage (CC-CV) charging method is a widely used method that can be split into three charging stages. The initial stage is only implemented if the battery got to discharge too deeply and is called Trickle Charge Stage and its goal is to charge the battery to voltage levels that are more appropriate for a normal charging. This Stage uses a constant trickle current ($I_{trickle}$) which is a current with a lower level than the one used in constant current stage. The following stage is the Constant Current Stage and it gets to charge the battery with a constant current ($I_{charge}$) until it reaches a predefined voltage level ($V_{charge}$). When it gets to this level it switches stage to the Constant Voltage Stage making the supply voltage constant and letting the current decrease until a minimum current value defined as the one that represents the end of charging ($I_{min}$) [32], [33].

Through implementation and literature reading it was possible to create a radar plot resuming the system’s Control Complexity, Charging Time, Charging Efficiency, and Implementation Complexity and Life Cycle Protection [33], [34]. The whole process and its characterization is visible in Figure 21.

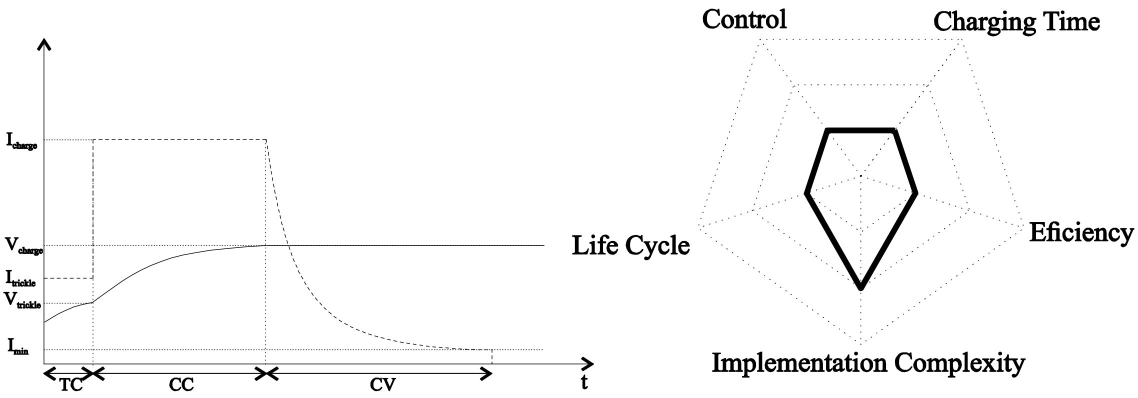


*Figure 21 - CC-CV Charging*

### 2.4.2.2 – *Multi-Stage Charging Method*

Multi-Stage Charging Method consists on charging in multiple distinct stages, where the switching condition between stages is the voltage reading achieving the charge ending voltage ($V_{charge}$), decreasing the charging current from stage to stage ($I_1$,$I_2$,$I_3$…). This method has into account the fact explained at the beginning of this section, where the voltage readings of a cell are more discrepant the higher the current that is flowing through it. Switching from one stage to the following one and reducing the current, makes voltage readings get closer to the battery's real voltage value [34], [35].

In some cases presented in the literature, the final charging stage is based on constant voltage charging just like the Constant Voltage Stage in the Constant Current / Constant Voltage Method presented in 2.4.2.1.

Just like the previous method, by experimentation and literature reading it was possible to create a radar plot resuming the system's Control Complexity, Charging Time, Charging Efficiency, and Implementation Complexity and Life Cycle Protection presented in Figure 22. By looking at the radar plots it is easy to conclude that this method is more complex than the previous one but also more efficient.

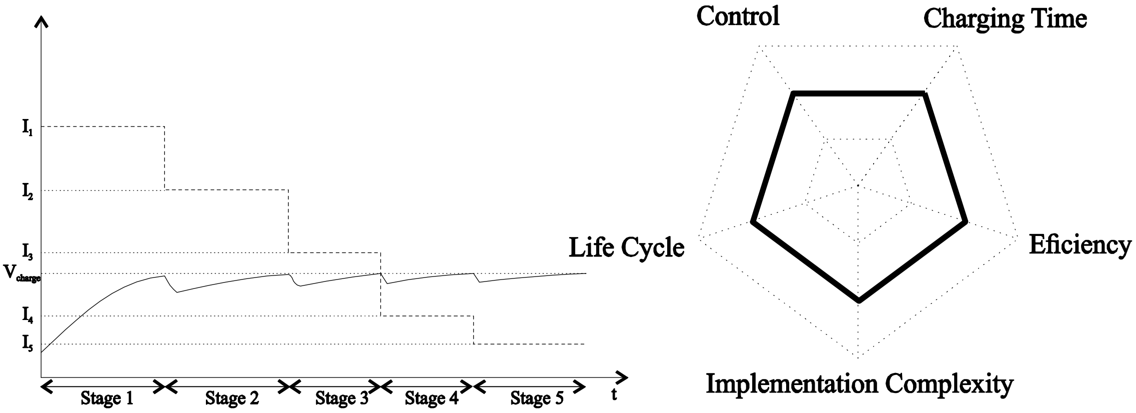


*Figure 22 - Multi-Stage Charging*

## 2.4.2.3 - *Pulse Charging*

Pulse Charging Method consists of charging the battery through pulses. This can be divided into three distinct stages in which the first one, called Detection Mode, is the stage in which battery's charge level is detected in order to understand if the charging is over or not. After the Detection Stage, the Searching Mode Stage, depending on the charging implementation method, tries to find the frequency or the duty cycle that minimizes impedance and maximizes charging current. Finally, the third stage, the Charging Stage, implements the frequency or duty cycle found in the searching stage [36].

By reading the literature it is possible to understand that this is the most efficient method but also the most complex for implementation.

Figure 23 shows the three stages of this method and a radar plot analysing the method's characteristics.

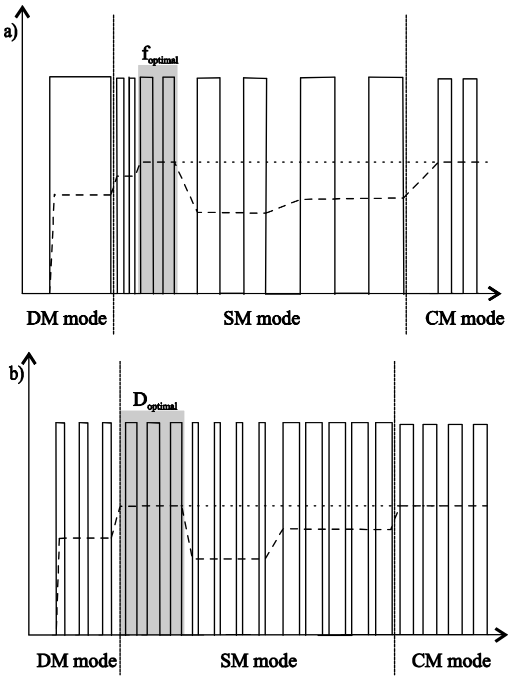
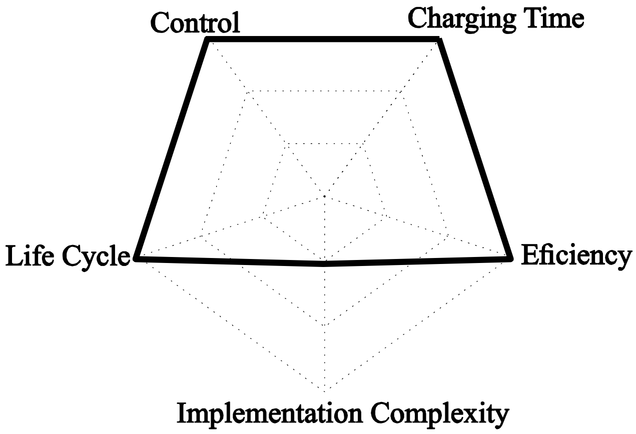


*Figure 23 - Pulse Charging*



## 2.4.3 – Section Conclusion

This section provided an analysis of the numerous charging methods that can be found in literature, showing the characteristics of each of the methods allowing the designer to choose the most suitable charging method for the system. By analysing the characteristics it was possible to get to the conclusion that the method to be implemented in this system would be the Multi Stage one, not only for its complexity/advantages relation but also because using it it's possible to have into account the voltage reading problems [32].

Figure 24 shows the three radar plots presented in this section all in one radar plot resuming this section.

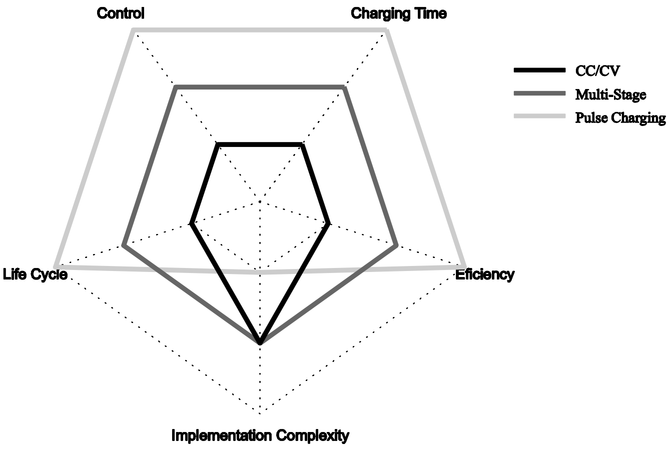


*Figure 24 - Charging Methods Radar Plot Comparison*

# Chapter 3 – Circuit Development

“Intelligence is the ability to adapt to change” – Stephen Hawking

# 3.1 – Battery Management System (BMS)

## 3.1.1 – Introduction

Battery Management Systems is a topic that comes usually coupled to the Cell Balancing topic that was mentioned in “Chapter 2 – Literature Review” as it depends on the balancing topology chosen which in this case was through dissipative resistors.

For the implementation of this method, it is important to choose a system capable of managing the dissipation resistors. As there are many components from numerous companies it is important to run a study on some of those components in order to understand which the most suitable one for this project is.

After picking the component that will be used in this project it is important to study the component even further in order to understand how it works, the connections it needs and the best way to implement it.

## 3.1.2 – Battery Management System Components

As mentioned earlier, there are a lot of components that are suitable for battery management systems so it is mandatory to understand what the project requirements are in order to make it possible to choose between one of them.

For this project, although the tests get to be run at low voltage levels, the battery management component has to be able to allow for the balancing of a 400V battery, as this is the normal voltage for Electric Vehicle Batteries.

Also in order to pick the right components it is vital to understand the parameters to take into account when choosing one so in this triage it was considered the number of cells, the stacking capability in order to manage batteries with more cells, the volume, the accessibility and the communication protocol.

Among all the analysed components there were three that outstood all the others due to their characteristics.

Linear Technology provides the LTC6803 with the ability to monitor 12 cells in each component, a 12 bits analog-to-digital converter and the ability to stack up to 10 components in series connected through a Daisy Chain allowing the system to monitor up to 120 cells in a maximum voltage level close to 500V. This component also has a feature to choose whether internal or external MOSFETS should be used [37].

The ISL94212 that Intersil provides can monitor up to 12 cells in each component and the ability to stack up to 14 components in series communicating through a Daisy Chain allowing for the system to analyse up to 168 cells achieving a maximum voltage level around 700V. This component has a 14 bits analog-to-digital converter having a higher resolution than the previously presented [38].

The third option that hit the spotlight is produced by Texas Instruments named BQ76PL536A-Q1 and has the ability to monitor up to 6 cells in each component. Although it has a lower number of cells per component it can stack up to 32 components communicating through an SPI protocol allowing the system to monitor up to 192 cells in a maximum voltage level around 800V [39].

Figure 25 presents the three components and the packages each one has in order to understand what kind of circuitry is necessary.

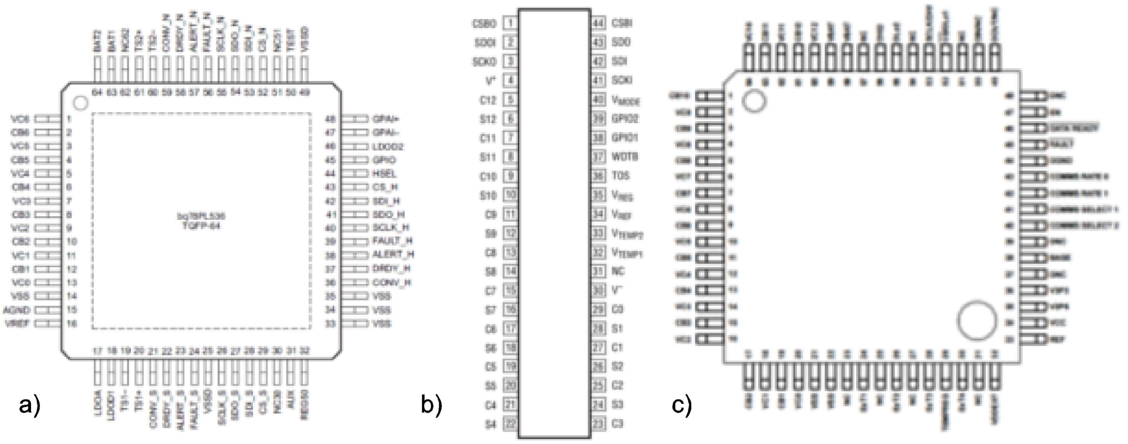


*Figure 25 - Battery Management Components a)BQ76PL536A-Q1 b)LTC6803 c) ISL94212*

Any of the three components meets the initially defined requirements for this project meaning that any of the three would be a good choice to implement in the system. Due to the lower number of components necessary for a bigger battery stack (information acquired by looking at the example given in the components datasheet provided by the company) and being the easiest implementing package of the three the Linear Technology's component was the first choice. Unfortunately the company did not provide samples for this academic project so this option had to be put aside.

Even though the most logic option was to pick Intersil's component, after what happened with Linear Technology and in order to save time, the chosen component was the Texas Instruments one as they have a pretty good image in providing samples for academic environments. This made Texas Instruments BQ76PL536A-Q1 the chosen component for this project.

### 3.1.3 – Chosen BMS

The chosen component has a few features that are important to mention and study for this project like a communication module to its host, which role in this project is played by the microprocessor which has a Serial Peripheral Interface (SPI) capable of communicating with the BQ76PL536A-Q1. The component has a 14 bits analog-to-digital converter integrated whose channel input is controlled by a multiplexer allowing the different battery cells to be monitored. The component also has some temperature inputs and a general purpose input - output. It also has outputs that can control MOSFETS in order to implement the balancing algorithm provided by the microprocessor.

Figure 26 presents a schematic of the component but does not provide a real representation of the ports as they are not in their real place. It’s possible to see that on the right side there are the connections to the cells, the MOSFETS and the temperature sensors, on the left side, there are the communication ports and on the bottom and top of the schematic there are some reference and supply connections.

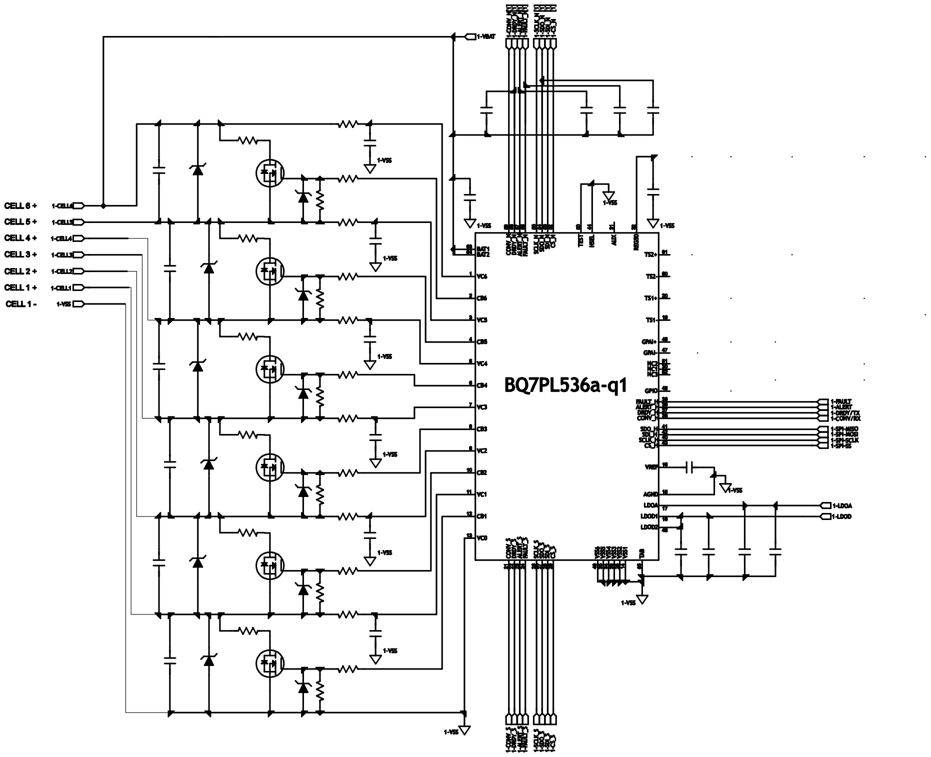


*Figure 26 - BQ76PL536A-Q1 Schematic*

During research, a picture of the Tesla Motors Model S battery management system was found which showed that it uses the same component as the one chosen for this project making the development of this project even more exciting. After some colour editing it was possible to achieve Figure 27 revealing the bq76pl536a-q1 Texas Instruments component that is used for dissipative resistors method.

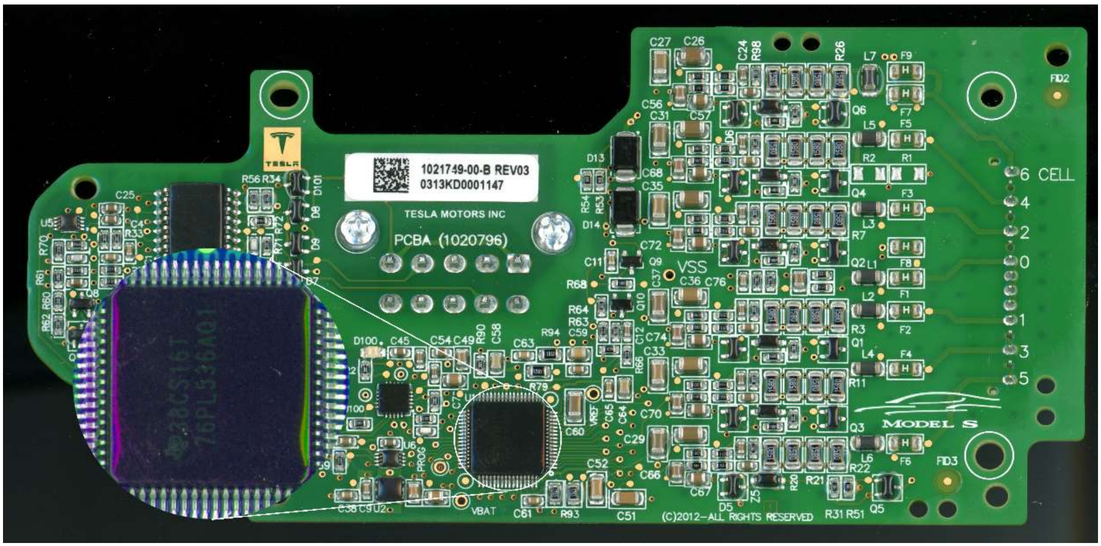


*Figure 27 - Tesla Balancing Circuit*

### 3.1.4 – Section Conclusion

With this section it is possible to understand the type of system that is going to be implemented and the numerous components that market can provide in order to create the project. In this case, the choice was limited not only economically but also by the impasse that came up when acquiring samples for implementation. In any case, all three mentioned components would be a good option for this project as all of them meet the requirements like implementing a dissipative resistors method and managing voltages of up to 400 V.

## 3.2 – Microprocessor

### 3.2.1 – Introduction

When picking the microprocessor for this project it is really important to understand what kind of features the system will need as some of them can be found integrated in the microprocessor. Going further in detail, features like analogic-to-digital converters and pulse width modulation can be integrated in the microprocessor. Although initially these features where taken into account, later on it was possible to understand that they wouldn't be used as with projects evolution, better options came up. The microprocessor should also have communication ports able to communicate with the chosen Battery Management System which in this case is through an SPI protocol.

Taking all that into account and the multiple options available in the academy, the chosen microprocessor was C2000 Picollo TSM3200F28027F provided by Texas Instruments. This microprocessor's features satisfy the project needs as it has SPI and SCI communications, pulse width modulation, 12 bits analogic-to-digital converter and obviously mathematical processing.

Besides all these characteristics, the microprocessor works at a 60 MHz clock rate providing a period of 16.67 ns per cycle and it provides much more freedom on memory management when compared to some other microprocessors evaluated.

Figure 28 is a picture of the microprocessor used in this system.

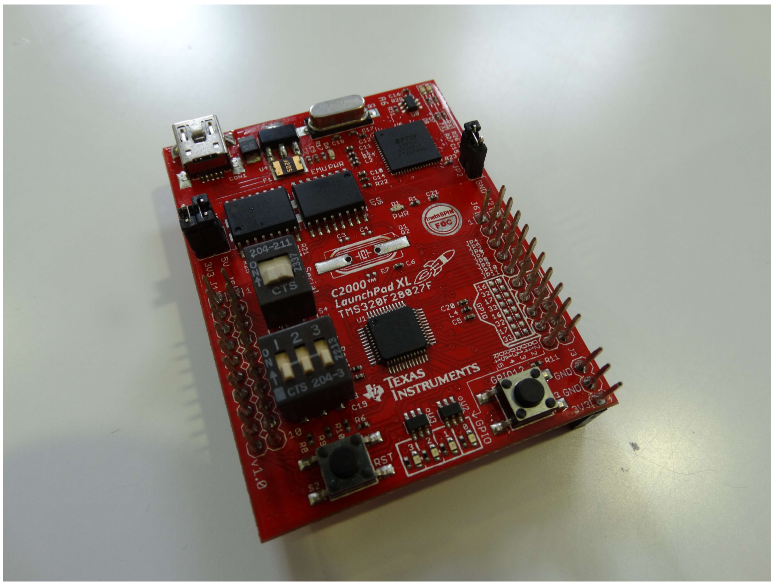


*Figure 28 - C2000 Picollo TSM3200F28027F*

## 3.2.2 – Peripherals

### 3.2.2.1 – *Pulse Width Modulation and Analog to Digital Converter*

As previously mentioned some of this microprocessor features where used while working on this project. Starting by the peripheral ADC, it has 10 bits resolution in a scale between 0 and 3.3 V. It starts conversion through a pulse and can take from 6 to 63 samples, allowing the microprocessor to keep working on other tasks until sampling and conversion end. Although this ADC doesn't have the required resolution for this project and, for that reason, not being the one used for the system, it was used during early learning stages in order to understand how it works and for possible future usage. For the initial application, to get the best reading value possible, maximum number of samples was used.

For the initial application, the pulse width modulation module was also used and this feature has its clock associated with the microprocessor's internal clock. This feature was used in order to generate a pulse to force the ADC start of conversion, making it easy to get the reading period randomly decided of 2 seconds by dividing the microprocessor's clock counter.

$$Comparator = \frac{f}{Clockdiv*T} \tag{3.1}$$

$$Comparator = \frac{60*10^6}{1792*2} = 3348 \tag{3.2}$$

Where $Comparator$ is the variable that will be compared with the value of the counter, $T$ represents the period intended (s), $f$ is the clock frequency (Hz) and $Clockdiv$ the number of times that the frequency is to be divided. Both $Clockdiv$ and $Comparator$ have no units.

By defining the UP_DOWN counting mode and defining the control register of this module to divide the clock's frequency by 1792 times, which is the maximum allowed, and placing the comparator value at 3348 it was possible to achieve the desired reading frequency.

In later stages of the project, it was decided not to use these peripherals as the battery management system has an ADC with a better resolution (14 bits) and the signal for conversion start could be generated by MatLab connected to the microprocessor.

### 3.2.2.2 – *Serial Communications Interface*

Serial Communications Interface (SCI) protocol allows the microprocessor to synchronously communicate with other peripherals that use this kind of serial communication. With the help of this protocol it is possible to transfer information between devices allowing the interaction between multiple components in a system. This communication is very important for this project as having a real time display of the cell parameters is a goal for the system.

In order to provide this display, a graphical environment was created in a personal computer so that a two dimension graphic presents the cell voltage readings as time goes by. SCI protocol implemented between the computer and the microprocessor is achieved with the help of YP-02 USB to TTL by connecting a cable that provides the reference for both devices and two cables, one for the component to send information to the computer, and the other one for the opposite task.

In this kind of protocol it is extremely important to properly configure both communicating devices, which in this case are the computer and the microprocessor. This importance comes from the fact that there is no clock line so they rely on a baud rate to swap data between one another. Besides that baud rate, it is also mandatory to configure the size of the data package, the existence or not of a parity bit and the number of stop bits for the communication.

Figure 29 presents a package transmission where data has the size of a byte. It starts with the Start bit, then the 8 data bits are sent and finally one Parity bit ending with a Stop bit.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Start** | **LSB** | **Bit 2** | **Bit 3** | **Bit 4** | **Bit 5** | **Bit 6** | **Bit 7** | **MSB** | **Parity** | **Stop** |

*Figure 29 - Serial Communications Interface Data Package*

### 3.2.2.3 – *Serial Peripheral Interface*

Serial Peripheral Interface (SPI), just like SCI allows devices to communicate with other devices that use the same communication protocol. This protocol is synchronous and that synchronism is achieved with the presence of a clock line connected between both devices. The clock signal is generated and managed by one of the devices that behaves as a master in order to communicate with other devices configured as slaves.

This protocol relies on the presence of a communication line that the master uses to send data to the slave, “Slave-In Master-Out” (SIMO), one other line for the master to receive information from the slave, “Slave-Out Master-In” (SOMI), a third communication cable for the clock generation (SPICLK) allowing for the system to be synchronous and an enabling line in order to activate de devices configured as slaves when communicating (SPISTE).

Figure 30 resumes this communication protocol showing the behaviour of each line during the data transference.

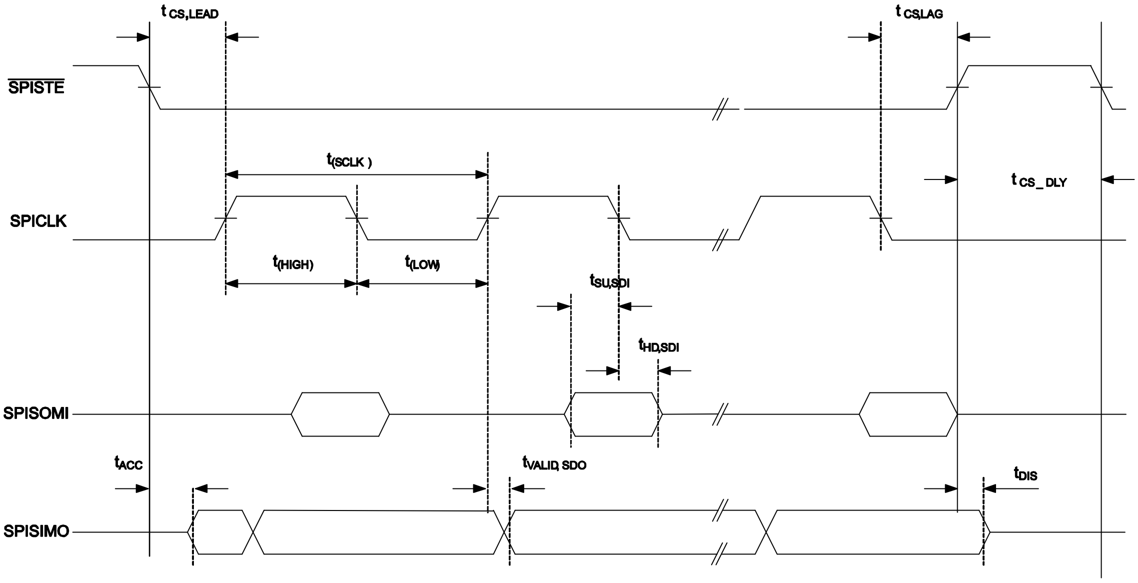


*Figure 30 - Serial Peripheral Interface Communication Schematic*

### 3.2.3 – Section Conclusion

By reading this section it is possible to understand the multiple features of the microprocessor used in this project and the reason for it to be the chosen one. Despite some limitations on choosing the microprocessor, after its implementation, it showed being a great option for the project due to all its features and peripherals, making it easy to interact with the various devices included in this project.

## 3.3 - Power Supply

### 3.3.1 – Introduction

For the multiple tests that have to be run to charge and balance the batteries it is significant to pay attention to the power supply used so that it can be controlled in a symbiosis with the system. In order to achieve this control, the power supply needs to have the ability to communicate with a part of the system, receiving commands to turn channels on or off and change voltage or current values in each channel so that the changing process is made as the implemented algorithm requires.

### 3.3.2 – Power supply

The used power supply, Rigol - DP832A, has  a pretty intuitive display and the capacity to supply up to 195 W adding up the three channels where two of them can provide 30 V each and the third one can go up to 5 V achieving a combined voltage of 65 V. This power supply has also the ability to communicate with the computer through a Universal Serial Bus port using a Standard Commands for Programmable Instruments (SCPI) receiving commands sent by MatLab installed on the computer.

Figure 31 shows the picture of a power supply available at the University projects development laboratory where all previously mentioned characteristics can be seen. This source features the ability to store in memory data with voltage, current and power values for a certain amount of time. Unfortunately the company keeps the files in a proprietary format which can only be decoded and analysed with specific computer software, forcing this display option to be put aside.

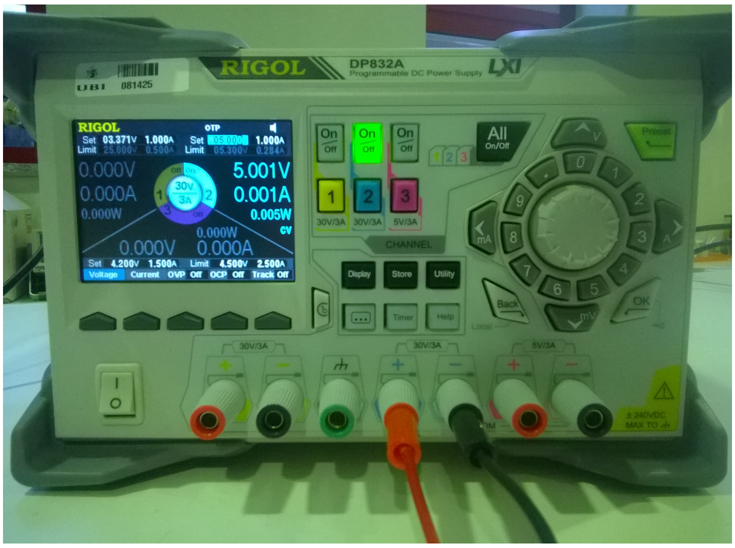


*Figure 31 - Rigol DP832A Power Supply*

The following piece of code presents a command that MatLab send to the power supply. First code line creates the Virtual Instrument Standard Architecture (VISA) object which will receive the command, the second code line opens the communication port and the two following lines configure the power supply to set a certain voltage and current and turn on the channel, finally the last line closes the communication port in order to let it free for other purposes.

```
dp800 = visa( 'ni','USB0::0x1AB1::0x0E11::DP8B171800395::INSTR' );
fopen(dp800);
fprintf(dp800, ':APPL CH1 ,25.2,%s',chargecurr)
fprintf(dp800, ':OUTP CH1,ON' );
fclose(dp800);
```

### 3.3.3 - Section Conclusion

As this is the best power supply available at the laboratory for this project, no further investigation was made on power supplies since it met all the requirements for the implementation of this system. The source is able to be controlled by SCPI protocol making it easy to implement any charging algorithm presented in the literature.

## 3.4 – Management Circuit Design

After picking the Texas Instruments component, BQ76PL536a-q1 and requesting samples it is necessary to develop the circuit design so that it can be implemented on the system. Texas Instruments provides information and advices on how to build the circuit facilitating its development. The schematic overview of a 3 device stack system is presented in Figure 32.

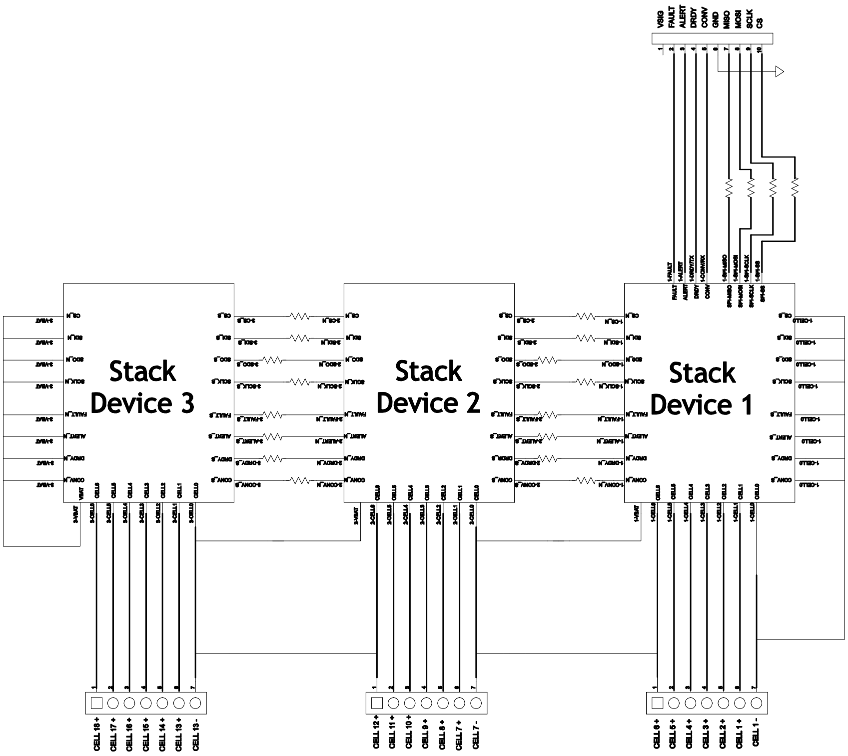


*Figure 32 - BQ76PL536A-Q1 Stack Overview*

Due to previous experience with Design Spark PCB released by RS Components, an electrical and electronic components trading brand, this was the software used for the design of the board. To make the design it is necessary to understand what kind of equipment is available and what can be ordered so that the design has the right footprints.

The software includes features capable of creating schematic design, Printed Circuit Board (PCB) design and with this PCB design it can provide a 3D preview of the whole system. The printed circuit board 2D preview is presented on the top of Figure 33 and the 3D preview is presented on its bottom.

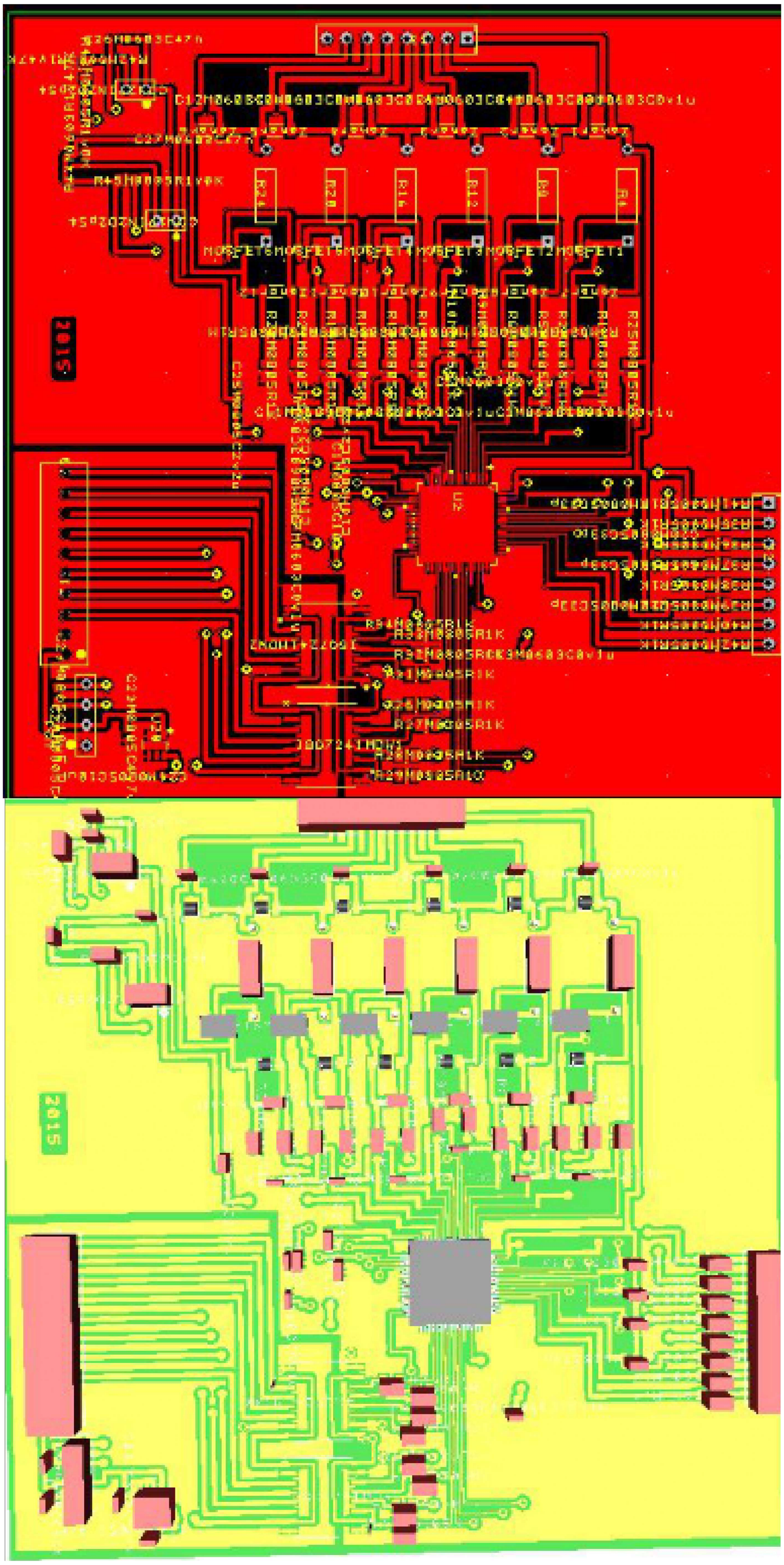


*Figure 33 - Design Spark PCB Previews*

## 3.5 – BQ – Microprocessor Interaction

BQ76PL536A-Q1 needs an initial configuration that can be read in its datasheet. This configuration consists of writing on the various registers related to the components working process.

By starting the microprocessor program, this configures the component that should be connected and running already in order to receive the configuration data, having its address, the number of conversions, the under voltage and overvoltage detection system configured and turning every MOSFET off, blocking currents passage avoiding unwanted discharging.

Besides the communications ports connected between the component and the microprocessor there are four extra connection lines named DRDY_H, indicating the conversion status of the component, ALERT_H and FAULT_H, giving information if there are any errors with the system, and CONV_H that can be used to start a conversion cycle.

Communication is assured by the 4 communication lines that implement the SPI protocol. CS_H that connects to the previously mentioned microprocessor port SPISTEA, SCLK_H which connects both devices clocks, SDI_H to transfer data from the microprocessor to the component and SDO_H to transfer data the opposite way.

A conversion routine begins when a pulse spawned by the microprocessor is received on the component's CONV_H port. This will force the component to lower the DRDY_H line and start the cell voltage values conversion. After finishing the conversion process, DRDY_H is brought to high level again informing the microprocessor that data is ready to be read. As explained in "3.2 – Microprocessor", the microprocessor sends the reading package to the component with the device and register address that it wants to receive data from. The following piece of code presents a reading routine, which will start by reading register 3 and then register 4 in the device with address 1.

```
void leitura(void)
{
	//Reading Structure
	//sdata[0] --> DEVADDR + REGADDR
	//sdata[1] --> CNT+FILLER
	sdata[0]=0x0203;
	sdata[1]=0x0100;

	for(i=0;i<2;i++)
	    {
		 SpiaRegs.SPITXBUF=sdata[i];      // Send data
	    }
}
```

Later on, received data is then processed and converted by the microprocessor in order to present values in a decimal format. To calculate the value of each conversion, the component's datasheet provides the equation (3.3):

$$mV = (REG_{MSB} * 256 + REG_{LSB}) * \frac{6250}{16383} \tag{3.3}$$

Where $mV$is the conversion result (mV), $REG_{MSB}$ is the value read on the most significant byte of the conversion and $REG_{LSB}$ is the value read on the least significant byte of the conversion.

In addition to reading data from the component, the microprocessor needs to configure some registers so it has to be able to write as well. This will allow not only for the component's configuration but also for the MOSFETS control.

Just like the reading procedure, the data package starts with the device and register address, followed by the data byte and finally a cyclic redundancy check (CRC) code in order to test for communication errors. The CRC value is calculated using the whole communication package, meaning, the device address, register address and the sent data. Components datasheet provides the CRC calculation equation (3.4):

$$C(x) = x^8 + x^2 + x^1 + 1 \tag{3.4}$$

Even though the datasheet provides this equation, this features implementation was achieved by using a library available online which can equally calculate the necessary value in order to pass this verification.

There are also some special cases in which some registers are write protected, meaning, due to the importance of the registers, they need a special routine in order to allow them to be written. This makes it necessary to previously write the value 0x35 to the SHDW_CTRL Register (0x3a) of the device, unlocking the writing for this EPROM registers.

## 3.6 – Microprocessor – Computer Interaction

In order to have a visual display of the cell balancing and charging management of the batteries it is necessary to send data stored in the microprocessor to the computer. As mention in “3.2.2 Peripherals”, communication between these two devices is ensured by the YP-02 USB to TTL.

With a graphical environment properly configured, the computer sends a command to the microprocessor and this will react accordingly, running the necessary routines and sending information back to the computer.

In order to facilitate this interaction, a Graphical User Interface (GUI) is to be created where in a data panel, cell values are presented in text boxes after each reading routine, two bi-dimensional graphics are to show live behaviour of cells’ voltage on the top one, and charging current in the bottom one, acquiring this information from the power supply Rigol DP832A. In the data panel there will also be a bar graph where each bar represents a cell, giving a graphical view of cells voltage differences and giving information on what stage are they. If cell are balancing, bars will be printed red, otherwise they will be green. On a second panel called Method Selection there will be buttons for selecting the procedure that the user wants the system to execute, choosing one of four options:

- Charging Only
- Top Balancing;
- Continuous Balancing;
- Balancing Only.

The first one makes the system control the power supply applying a CC-CV, presented in “2.4.2.1 – Constant Current / Constant Voltage Charging Method”, charging with no balancing applied.

The Top Balancing method, found in the second button of the panel, applies a charging algorithm presented in the literature which begins charging the battery, applying no balancing with a multi-stage charging method, explained in “2.4.2.2 – Multi-Stage Charging Method”, until one of the cells reaches the end of charge voltage value. The system then checks if all balancing and charging requirements are met, otherwise it will turn off the power supply, balance the cells and repeat the whole process until ending requirements are achieved.

Third option starts a continuous balancing procedure which is pretty similar to the Top Balancing one except for the fact that during the charging phase there is balancing, minimizing end of charge voltage discrepancy. Just like the previously presented method, by verifying that a cell achieved end of charge voltage, the systems turns the power supply off and balances the cells if necessary. Once again, after balancing, if the battery is not yet fully charged, the whole process is repeated.

The fourth option will apply balancing with no charging, making the cells to get to the same voltage value in order to put them in similar SOC values.

Top Balancing and Continuous Balancing algorithms are presented in Figure 34 which includes flowcharts of both methods.

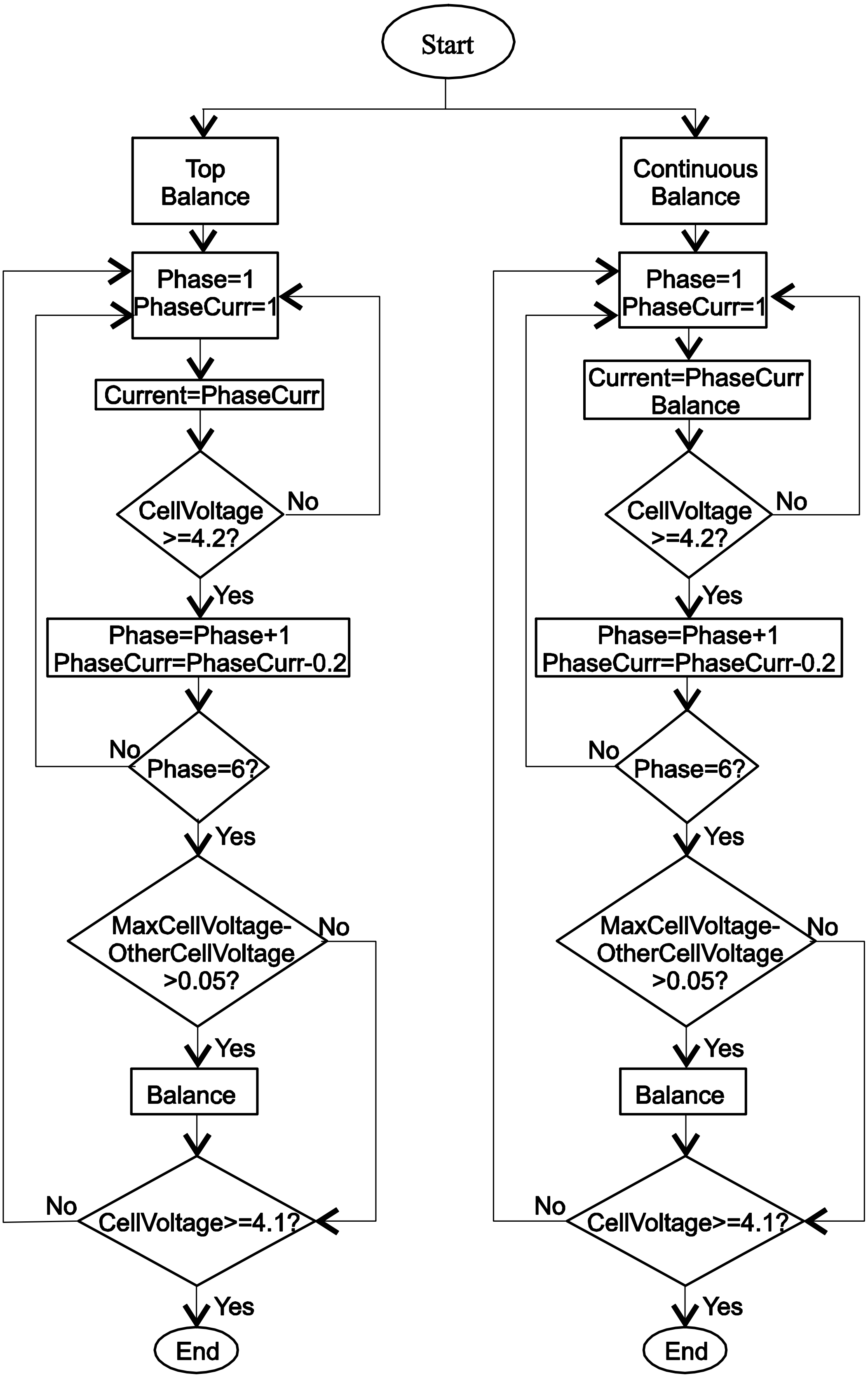


*Figure 34 - Top Balance and Continuous Balance Flowchart*

# Chapter 4 - Results

“Insanity is doing the same thing over and over and expecting different results” – Albert Einstein

## 4.1 – Introduction

During this project's development, some important steps marked the difficulty and quality of this project, which made it seem important to present some information on overcoming some obstacles. This section presents the results obtained on testing the system after each important step and the final result of the whole project.

## 4.2 – Mid-development Tests

### 4.2.1 – Circuit Building

Building the physical part of the project was a hard task as the only method available for creating the printed circuit boards was a really archaic one. On the first step, a circuit was printed on a photosensitive board with the help of ultra violet lamps. Later on the board went on a chemical bath in order to remove some of the copper protection to let it be corroded in the following step. The final step was to put the board on a copper corrosive chemical solution in order to remove the unwanted copper and get the copper printed circuit. The problem of this method is that it is very hard to get the right precision as if it gets to be on the acid for a long time, it takes off too much copper and causes connections to be cut, and if it doesn't stay long enough, it won't remove enough copper and might cause short-circuits. Luckily the first board printed perfectly and had all the connections in good conditions, but unfortunately the following ones did not go so well, making it impossible to add boards on a stack. Figure 35 presents the board produced with this method.

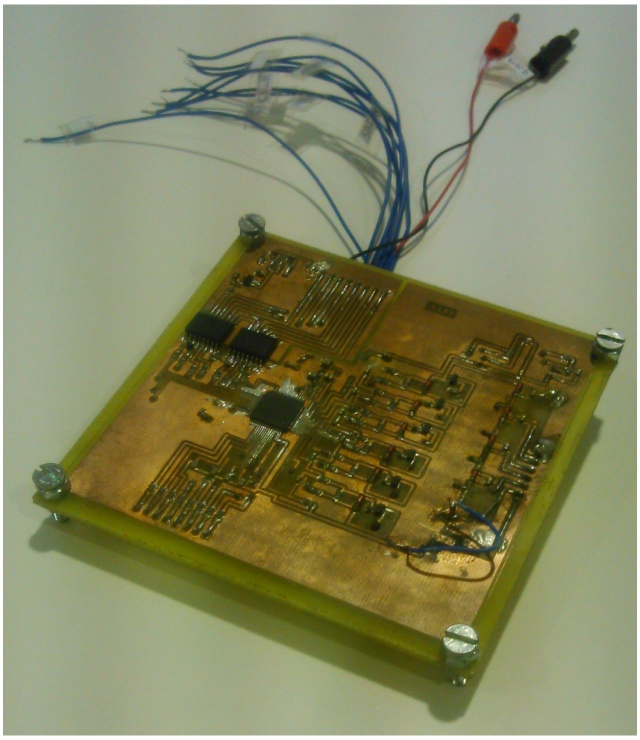
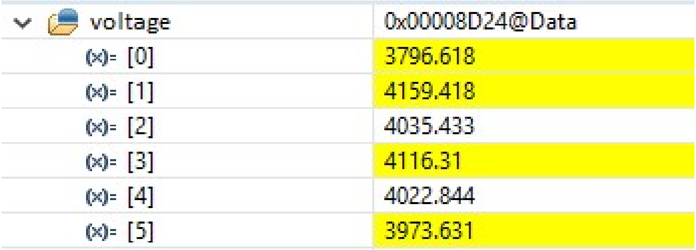


*Figure 35 - Real Life PCB*

### 4.2.2 – Communication Programming

During software programing another hard task was to get the correct communication between the board and the microprocessor as some configurations were hard to find in order to get it to work properly. After days of tests and configurations, SPI finally got to work and to test it, the values read on a voltmeter and the ones presented by the microprocessor were compared. Figure 36 presents the microprocessor data acquired, which matched the voltmeter readings.



```
voltage         0x00008D24@Data
  (x)= [0]      3796.618
  (x)= [1]      4159.418
  (x)= [2]      4035.433
  (x)= [3]      4116.31
  (x)= [4]      4022.844
  (x)= [5]      3973.631
```

*Figure 36 - SPI Communication Test Result*

All values are inside the cells working limits, 3200 mV and 4200 mV. The colour differences in rectangles have no relevance as they only mean that when the print screen was taken, the values were refreshing.

### 4.2.3 – Graphical User Interface Creation

After having the microprocessor and the battery management system working together, it was time to create an interface for a user to be able to control the system as explained in “3.6 – Microprocessor – Computer Interface”. The result of this graphical user interface creation is presented in Figure 37.

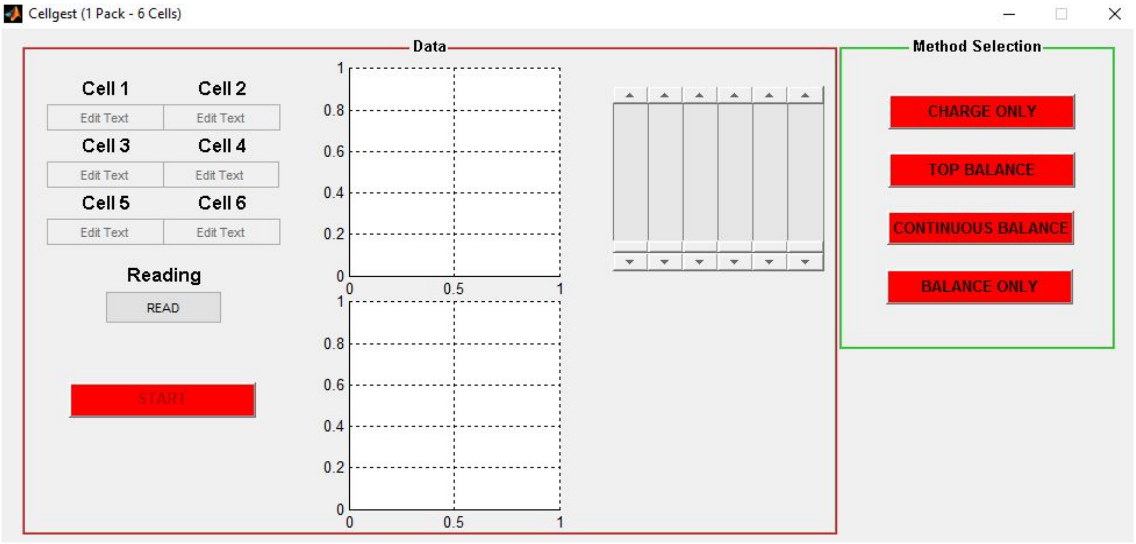


*Figure 37 - Graphical User Interface*

# 4.3 - Final Results

## 4.3.1 - Full System

In Figure 38 it is possible to see the whole system overview, which got to be close to the one initially thought. The Rigol DP832A connects to the battery pack and to MatLab using an SCPI protocol. The BQ76PL536A-Q1 battery management system connected to Texas Instruments TSM3200F28027F microprocessor by an SPI protocol and a computer running MatLab providing a graphical user interface seen on the bottom of the figure.

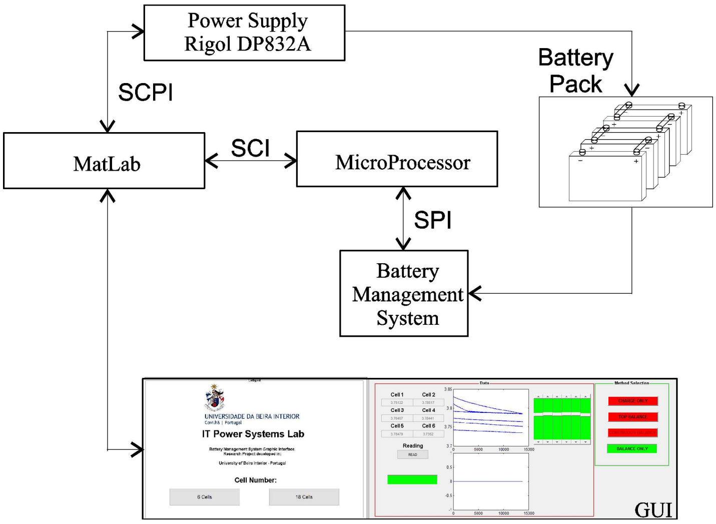
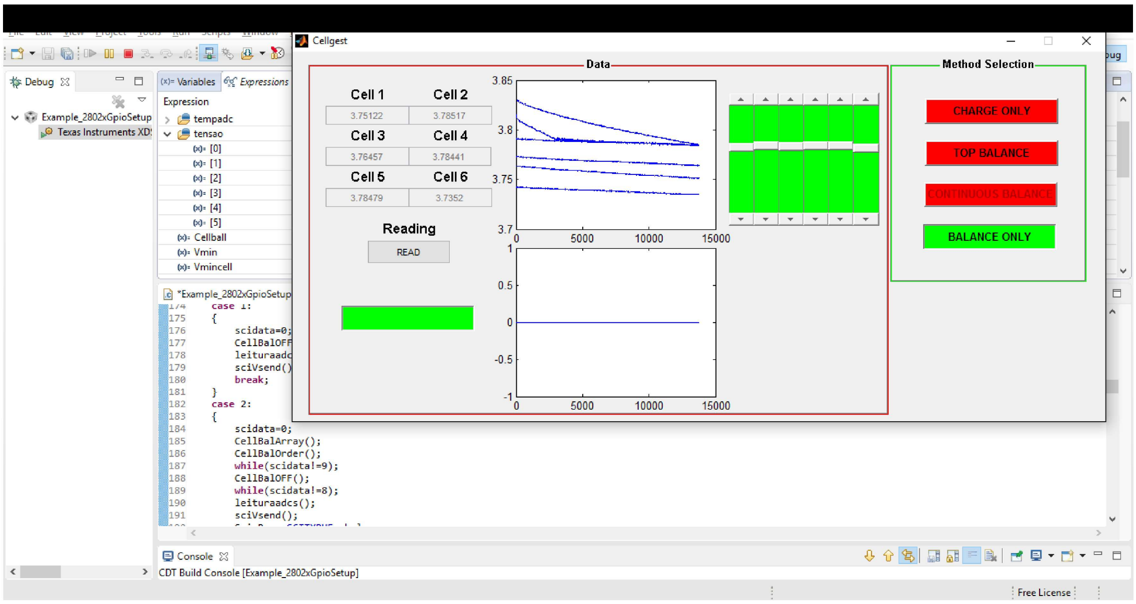


*Figure 38 - Final System Overview*

Figure 39 presents the full system working, as it shows on the background the CCS Composer that was used to program the microprocessor and on the front of the image it is possible to see the graphical user interface showing the whole information designed for the system. Cell values on the left, cell voltage curves on the middle top graph and cell bars on the left of the Data panel. The middle bottom graph presents the current but as it is possible to see in the Method Selection panel, “Balance Only” was the chosen method for this test so no current gets to go in or out of the battery apart from the balancing current spent on dissipative resistors.

[redacted]



Cell 1 3.75122
Cell 2 3.78517
Cell 3 3.76457
Cell 4 3.78441
Cell 5 3.78479
Cell 6 3.7352

```
    case 1:
    {
        scidata=0;
        CellBalOFF
        leituraadc
        sciVsend()
        break;
    }
    case 2:
    {
        scidata=0;
        CellBalArray();
        CellBalOrder();
        while(scidata!=9);
        CellBalOFF();
        while(scidata!=8);
        leituraadcs();
        sciVsend();
```

CDT Build Console [Example_2802xGpioSetup]

*Figure 39 - Fully Working System*

In Figure 40 there is a picture of the whole system. In rectangle number 1 it is possible to see the graphical user interface, in 2, a power supply used to supply the energy for the communications isolating from the power supply number 3 that provides electric power for the battery. In 4 it is possible to see the Battery Management System, which in this case was not the board built on the university as it was a very late stage of this project and multiple battery packs were already being prepared.

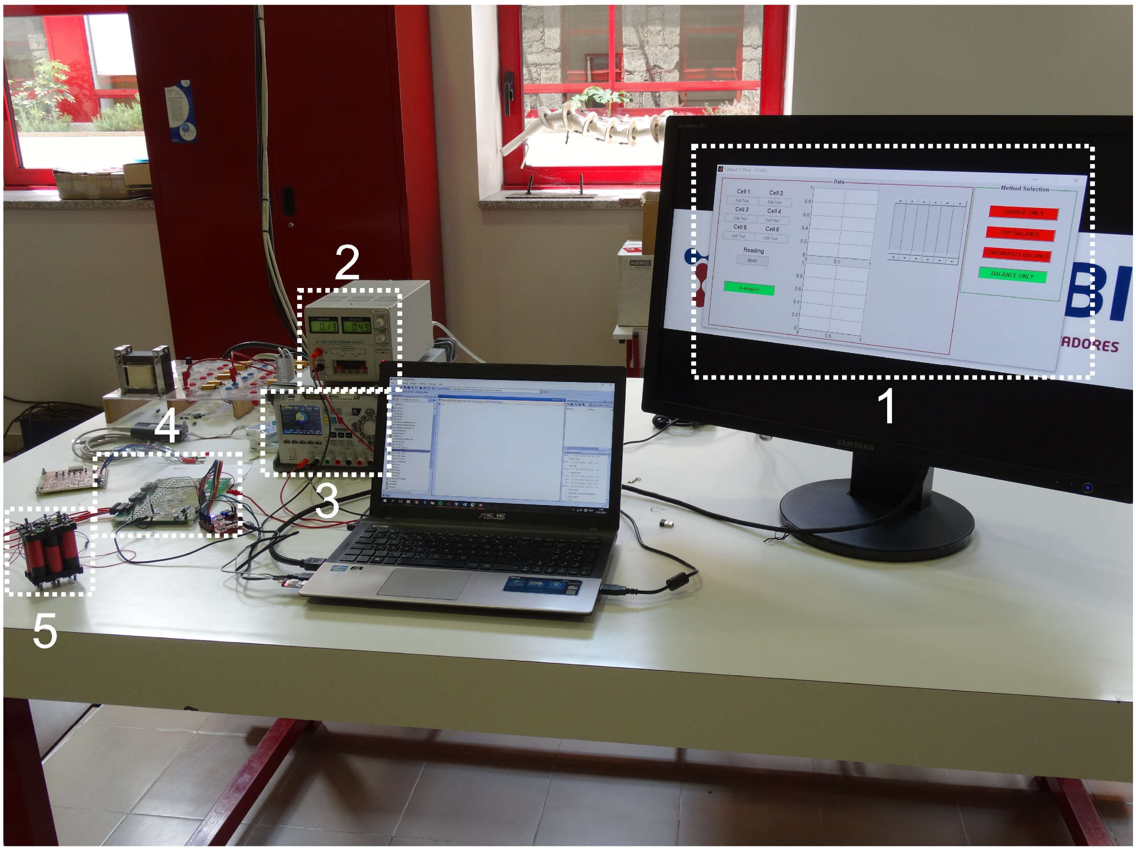


*Figure 40 - Real System Picture*

## 4.3.2 - Multi-Stage Charge Test

In Figure 41 there is a full charge acquisition curve with a part of it zoomed making it possible to check the multi stage implementation as each voltage drop means the stage switch, reducing current transferred to the cells and consequently decreasing the voltage reading, taking it closer to the real value.

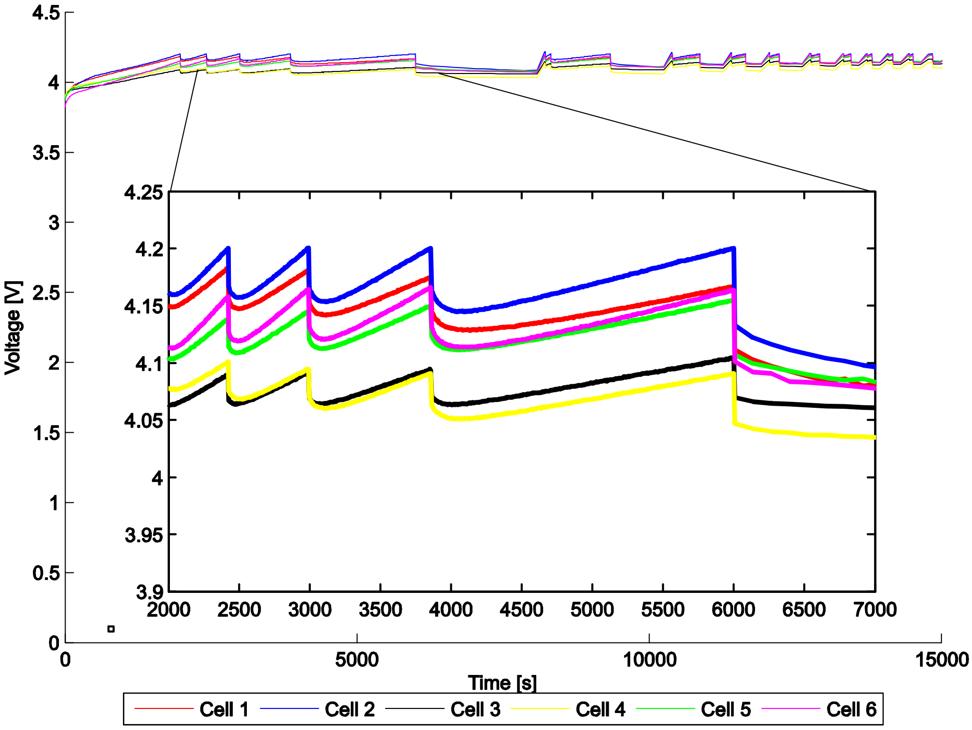


*Figure 41 - Multi-Stage Charging Test*

## 4.3.3 - Balancing Test

In Figure 42 a balancing procedure can be seen, as the system detected that the voltage of 2 of the cells was too high, it balanced them close to other cell's voltage levels. It might seem they are still in too different voltage levels on the zoomed window but the threshold for this test was 0.05 V so the balancing stops when all cells are within 0.05 V. It is also possible to conclude that cells are pretty different one from another as they get to react differently and balance at different rates.

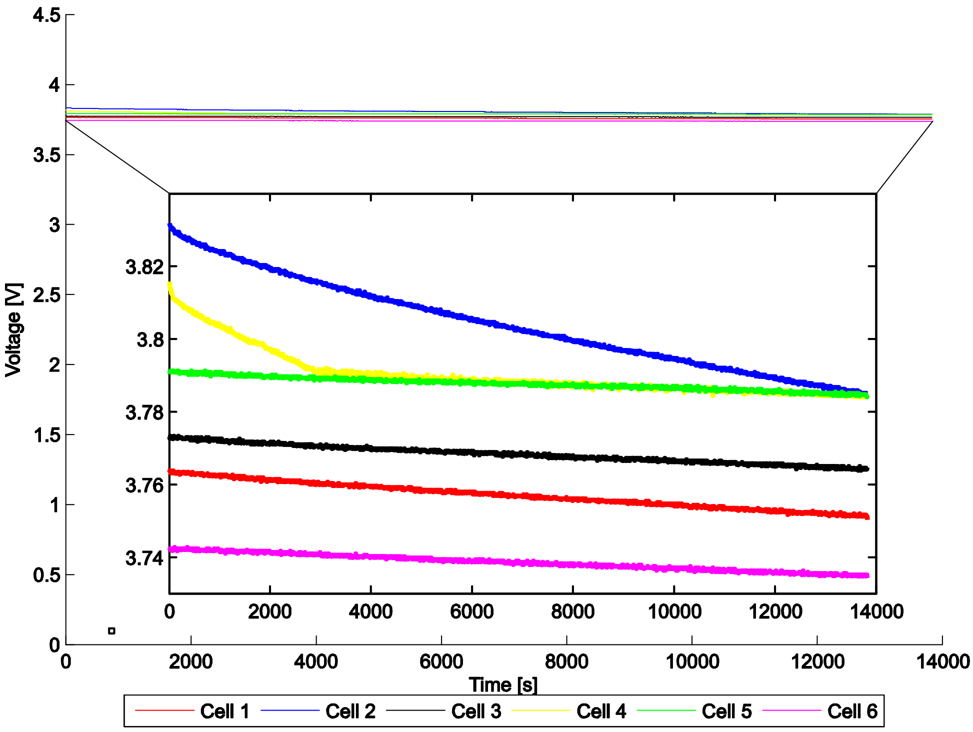


*Figure 42 - Dissipative Resistors Balancing Test*

## 4.3.4 - Charging and Balancing Test

Figure 43 presents an acquisition that was taken while the cells were charging and balancing at the same time. The algorithm was programmed to charge with the multi-stage charging method ending charge when all cells got to be above 4.1 V and dissipative resistor balancing method with a balancing threshold of 0.05 V. This concludes the results section and proves the achievement of this project goals.

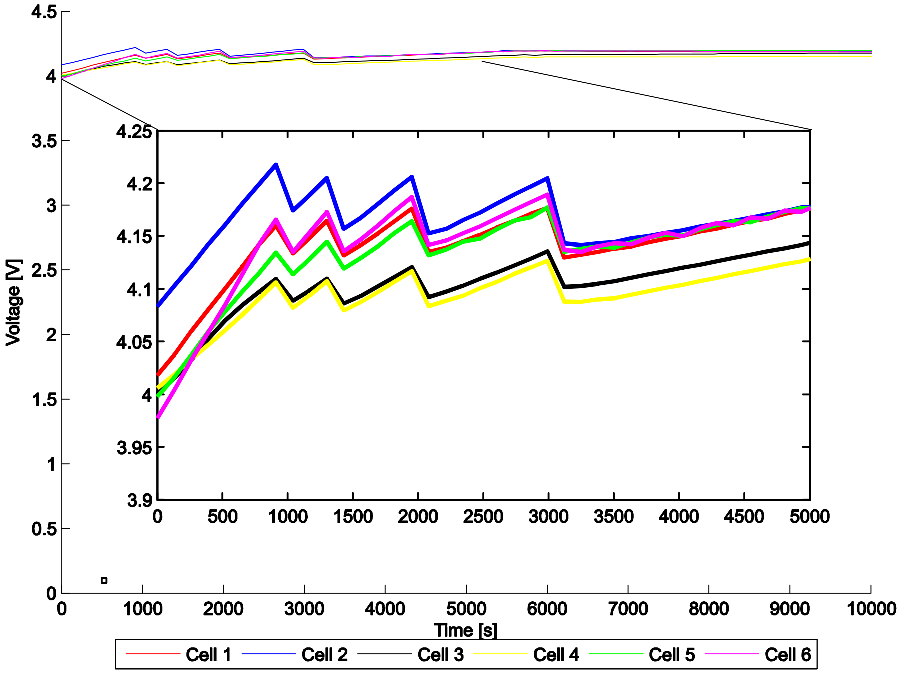


*Figure 43 - Continuous Balancing*

# Chapter 5 - Conclusions

Learn from the past to decide in the present what you want for the future.

## 5.1 – Final Remarks

At the end of this work it is possible to conclude that the system is able to manage cells making them charge as much as possible, improving their energy storage capacity as it avoids stopping charging due to having cells reaching maximum voltage limit too soon.

It was possible to conclude that the fact of having current flowing through a cell originates a discrepancy between measurement equipment voltage readings and cell voltage real values makes it harder to do balancing and charging based on acquired voltage data.

After comparing Top Balancing and Continuous Balancing, as expected, it was also possible to conclude that the Continuous Balance, as long as it is implemented with the correct charging and balancing method can be faster than Top Balancing.

## 5.2 – Future Work and Suggestions

As the main goal of this work was to build a system capable of monitoring and managing battery cells, due to the lack of time, algorithms were not as accurate and optimized as they could so a project on optimizing the whole system would be a good suggestion.

One important detail on battery management is the temperature and this system, though it is ready to apply this feature, doesn’t use it, so, adapting the system to include temperature monitoring and studying temperature’s behaviour during charging and/or balancing would be a good addition.

Given the simplicity of adapting the system, and the procedures, testing other charging algorithms can be a good way on understanding how to improve cell capacity and charging efficiency.

# References

[1] S. Srinivasaraghavan and A. Khaligh, “Time Management,” *IEEE Power Energy Mag.*, vol. 9, no. 4, pp. 46–53, Jul. 2011.

[2] O. Tremblay, L. -a. Dessaint, and a.-I. Dekkiche, “A Generic Battery Model for the Dynamic Simulation of Hybrid Electric Vehicles,” *2007 IEEE Veh. Power Propuls. Conf.*, no. V, pp. 284–289, 2007.

[3] H. Qian, J. S. Lai, J. Zhang, and W. Yu, “High-efficiency bidirectional AC-DC converter for energy storage systems,” *2010 IEEE Energy Convers. Congr. Expo. ECCE 2010 - Proc.*, pp. 3224–3229, 2010.

[4] Y. Drori and C. Martinez, “The benefits of cell balancing,” *Appl. Note, Xicor Inc.*, pp. 1–9, 2005.

[5] A. C. Gil, *Como Elaborar Projetos de Pesquisa*. São Paulo, 2002.

[6] E. L. da Silva and E. M. Menezes, “Metodologia da pesquisa e elaboração de dissertação,” *Laboratório Ensino a Distância da UFSC*, p. 138p, 2001.

[7] J. Cannarella and C. B. Arnold, “State of health and charge measurements in lithium-ion batteries using mechanical stress,” *J. Power Sources*, vol. 269, pp. 7–14, 2014.

[8] M. H. Hung, C. H. Lin, L. C. Lee, and C. M. Wang, “State-of-charge and state-of-health estimation for lithium-ion batteries based on dynamic impedance technique,” *J. Power Sources*, vol. 268, pp. 861–873, 2014.

[9] B. Pattipati, B. Balasingam, G. V. Avvari, K. R. Pattipati, and Y. Bar-Shalom, “Open circuit voltage characterization of lithium-ion batteries,” *J. Power Sources*, vol. 269, pp. 317–333, 2014.

[10] O. Tremblay, “Experimental Validation of a Battery Dynamic Model for EV Applications,” *World Electr. Veh. J.*, vol. 3, no. October, pp. 1–10, 2015.

[11] B. Aksanli and T. Rosing, “Optimal battery configuration in a residential home with time-of-use pricing,” *2013 IEEE Int. Conf. Smart Grid Commun. SmartGridComm 2013*, pp. 157–162, 2013.

[12] D. Cadar, D. PETREUS, T. PATARAU, and N. PALAGHITA, “Active Balancing Method for Battery Cell Equalization,” *Acta Tech. Napocensis*, vol. 51, no. 2, pp. 1–5, 2010.

[13] H. Popp, J. Attia, F. Delcorso, and A. Trifonova, “Lifetime analysis of four different lithium ion batteries for ( plug–in ) electric vehicle,” 2014.

[14] I. Technologies, “Selection of the MOSFET for Faster Cell Balancing of Li-Ion Batteries,” *Appl. Note Infineon*, no. February, 2013.

[15] G. Altemose, “Batteries and Fuel Cells Achieving cell balancing for lithium-ion batteries,” *Hearst Electronics Products*, 2008. [Online]. Available: http://ams.aeroflex.com/. [Accessed: 27-Jan-2016].

[16] M. Einhorn, W. Roessler, and J. Fleig, “Improved performance of serially connected Li-ion batteries with active cell balancing in electric vehicles,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 2448–2457, 2011.

[17] J. Cao, N. Schofield, and A. Emadi, “Battery balancing methods: A comprehensive review,” *2008 IEEE Veh. Power Propuls. Conf. VPPC 2008*, pp. 3–8, 2008.

[18] I. Aizpuru, U. Iraola, J. M. Canales, M. Echeverria, and I. Gil, “Passive balancing design for Li-ion battery packs based on single cell experimental tests for a CCCV charging mode,” *4th Int. Conf. Clean Electr. Power Renew. Energy*

*Resour. Impact, ICCEP 2013*, pp. 93–98, 2013.

[19] M. Daowd, N. Omar, P. Van Den Bossche, and J. Van Mierlo, "Passive and active battery balancing comparison based on MATLAB simulation," *2011 IEEE Veh. Power Propuls. Conf. VPPC 2011*, 2011.

[20] J. Gallardo-Lozano, E. Romero-Cadaval, M. I. Milanes-Montero, and M. A. Guerrero-Martinez, "Battery equalization active methods," *J. Power Sources*, vol. 246, pp. 934–949, 2014.

[21] F. Baronti, R. Roncella, and R. Saletti, "Performance comparison of active balancing techniques for lithium-ion batteries," *J. Power Sources*, vol. 267, pp. 603–609, 2014.

[22] W. C. Lee, D. Drury, and P. Mellor, "Comparison of passive cell balancing and active cell balancing for automotive batteries," *2011 IEEE Veh. Power Propuls. Conf. VPPC 2011*, 2011.

[23] J. Qi and D. Dah-Chuan Lu, "Review of battery cell balancing techniques," *2014 Australas. Univ. Power Eng. Conf. AUPEC 2014 - Proc.*, no. October, pp. 1–6, 2014.

[24] M. M. U. Rehman, F. Zhang, M. Evzelman, R. Zane, and D. Maksimovic, "Control of a series-input, parallel-output cell balancing system for electric vehicle battery packs," *2015 IEEE 16th Work. Control Model. Power Electron.*, pp. 1–7, 2015.

[25] M. M. Ur. Rehman, M. Evzelman, K. Hathaway, R. Zane, G. L. Plett, K. Smith, E. Wood, and D. Maksimovic, "Modular approach for continuous cell-level balancing to improve performance of large battery packs," *2014 IEEE Energy Convers. Congr. Expo. ECCE 2014*, pp. 4327–4334, 2014.

[26] W. Hong, K. S. Ng, J. H. Hu, and C. S. Moo, "Charge equalization of battery power modules in series," *2010 Int. Power Electron. Conf. - ECCE Asia -, IPEC 2010*, pp. 1568–1572, 2010.

[27] T. H. Phung, A. Collet, and J. C. Crebier, "An optimized topology for next-to-next balancing of series-connected lithium-ion cells," *IEEE Trans. Power Electron.*, vol. 29, no. 9, pp. 4603–4613, 2014.

[28] C. M. Young, N. Y. Chu, L. R. Chen, Y. C. Hsiao, and C. Z. Li, "A Single-phase multilevel inverter with battery balancing," *IEEE Trans. Ind. Electron.*, vol. 60, no. 5, pp. 1972–1978, 2013.

[29] J. Xu, B. Cao, Z. Chen, and Z. Zou, "An online state of charge estimation method with reduced prior battery testing information," *Int. J. Electr. Power Energy Syst.*, vol. 63, pp. 178–184, 2014.

[30] A. Ausswamaykin and B. Plangklang, "Design of Real Time Battery Management Unit for PV-Hybrid System by Application of Coulomb Counting Method," *Energy Power Eng.*, vol. 6, no. July, pp. 186–193, 2014.

[31] Z. Chen, S. Qiu, M. A. Masrur, and Y. Lu, "Battery State of Charge Estimation Based on a Combined Model of Extended Kalman Filter and Neural Networks," in *International Joint Conference on Neural Networks*, 2011, pp. 2156–2163.

[32] W. Shen, T. T. Vo, and A. Kapoor, "Charging algorithms of lithium-ion batteries: An overview," *Proc. 2012 7th IEEE Conf. Ind. Electron. Appl. ICIEA 2012*, pp. 1567–1572, 2012.

[33] A. Al-Haj Hussein and I. Batarseh, "A review of charging algorithms for nickel and lithium battery chargers," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 830–838, 2011.

[34] A. Kapoor, "Charging algorithms of lithium-ion batteries: An overview," in *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2012,

pp. 1567–1572.

[35] D. Anseán, M. González, J. C. Viera, V. M. García, C. Blanco, and M. Valledor, “Fast charging technique for high power lithium iron phosphate batteries: A cycle life analysis,” *J. Power Sources*, vol. 239, pp. 9–15, 2013.

[36] L.-R. Chen, “A Design of an Optimal Battery Pulse Charge System by Frequency-Varied Technique,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 398–405, Feb. 2007.

[37] “LTC6803-1 and -3 - Multicell Battery Stack Monitor - Linear Technology.” [Online]. Available: http://www.linear.com/product/LTC6803-1. [Accessed: 09-May-2016].

[38] “ISL94212.” [Online]. Available: https://www.intersil.com/en/products/power-management/battery-management/cell-balancing-and-safety/ISL94212.html#overview. [Accessed: 11-May-2016].

[39] “BQ76PL536A-Q1 | Battery Monitor, Protection & Authentication Solutions | Battery Management Products | Description & parametrics.” [Online]. Available: http://www.ti.com/product/BQ76PL536A-Q1?keyMatch=bq76pl536a-q1&tisearch=Search-EN-Products. [Accessed: 11-May-2016].

# Annex 1 – CCS Code

```
#include "DSP28x_Project.h"
//Interruption Routines
interrupt void xint1_isr(void);
interrupt void xint2_isr(void);
interrupt void spiTxFifoIsr(void);
interrupt void spiRxFifoIsr(void);
interrupt void sciaTxFifoIsr(void);
interrupt void sciaRxFifoIsr(void);

//Configuration Routines
void GPIOstart(void);
void spi_fifo_init(void);
void scia_fifo_init(void);
void error(void);
void comprep(void);
void bqconfig(void);

void conversao(void);
void leitura(void);             //Reading routine
void leituraadcs(void);
void sciVsend(void);

void escrita(void);             //Writing Routine
void escritaeprom(void);

void CRCcalc(void);
//void Vcalc(void);

void CellBalArray(void);
void CellBalOrder(void);
void CellBalOFF(void);

Uint16 Teste;
Uint16 inicio;
Uint16 j;
Uint16 i;
Uint16 x;
Uint16 z;
Uint16 t;
Uint16 sdata[4];     // Send data buffer
Uint16 rdata[2];     // Receive data buffer
volatile Uint16 scidata;
Uint16 spidata;
Uint16 data2;
Uint16 DEVADDR;
Uint16 REGADDR;
Uint16 CNT;
Uint16 FILLER;
Uint16 CRCFILLER;
Uint16 CrcTable[256] = {0x00, 0x07, 0x0E, 0x09, 0x1C, 0x1B, 0x12, 0x15,      0x38, 0x3F, 0x36,
0x31, 0x24, 0x23, 0x2A, 0x2D,    0x70, 0x77, 0x7E, 0x79, 0x6C, 0x6B, 0x62, 0x65,     0x48, 0x4F,
0x46, 0x41, 0x54, 0x53, 0x5A, 0x5D,    0xE0, 0xE7, 0xEE, 0xE9, 0xFC, 0xFB, 0xF2, 0xF5,
      0xD8, 0xDF, 0xD6, 0xD1, 0xC4, 0xC3, 0xCA, 0xCD, 0x90, 0x97, 0x9E, 0x99, 0x8C, 0x8B,
0x82, 0x85,  0xA8, 0xAF, 0xA6, 0xA1, 0xB4, 0xB3, 0xBA, 0xBD,     0xC7, 0xC0, 0xC9, 0xCE, 0xDB,
```

```
0xDC, 0xD5, 0xD2,  0xFF, 0xF8, 0xF1, 0xF6, 0xE3, 0xE4, 0xED, 0xEA,     0xB7, 0xB0, 0xB9, 0xBE,
0xAB, 0xAC, 0xA5, 0xA2,   0x8F, 0x88, 0x81, 0x86, 0x93, 0x94, 0x9D, 0x9A, 0x27, 0x20, 0x29,
0x2E, 0x3B, 0x3C, 0x35, 0x32,   0x1F, 0x18, 0x11, 0x16, 0x03, 0x04, 0x0D, 0x0A,     0x57, 0x50,
0x59, 0x5E, 0x4B, 0x4C, 0x45, 0x42,    0x6F, 0x68, 0x61, 0x66, 0x73, 0x74, 0x7D, 0x7A,
        0x89, 0x8E, 0x87, 0x80, 0x95, 0x92, 0x9B, 0x9C,     0xB1, 0xB6, 0xBF, 0xB8, 0xAD, 0xAA,
0xA3, 0xA4, 0xF9, 0xFE, 0xF7, 0xF0, 0xE5, 0xE2, 0xEB, 0xEC,      0xC1, 0xC6, 0xCF, 0xC8, 0xDD,
0xDA, 0xD3, 0xD4,  0x69, 0x6E, 0x67, 0x60, 0x75, 0x72, 0x7B, 0x7C,     0x51, 0x56, 0x5F, 0x58,
0x4D, 0x4A, 0x43, 0x44,   0x19, 0x1E, 0x17, 0x10, 0x05, 0x02, 0x0B, 0x0C,     0x21, 0x26, 0x2F,
0x28, 0x3D, 0x3A, 0x33, 0x34, 0x4E, 0x49, 0x40, 0x47, 0x52, 0x55, 0x5C, 0x5B,0x76, 0x71, 0x78,
0x7F, 0x6A, 0x6D, 0x64, 0x63,   0x3E, 0x39, 0x30, 0x37, 0x22, 0x25, 0x2C, 0x2B,     0x06, 0x01,
0x08, 0x0F, 0x1A, 0x1D, 0x14, 0x13,    0xAE, 0xA9, 0xA0, 0xA7, 0xB2, 0xB5, 0xBC, 0xBB,
        0x96, 0x91, 0x98, 0x9F, 0x8A, 0x8D, 0x84, 0x83, 0xDE, 0xD9, 0xD0, 0xD7, 0xC2, 0xC5,
0xCC, 0xCB,  0xE6, 0xE1, 0xE8, 0xEF, 0xFA, 0xFD, 0xF4, 0xF3};
Uint16 temp;
Uint16 crc;
Uint16 buffer[3];
Uint16 CELL[6];
Uint16 tempCELL;
Uint16 tempadc[12];
Uint16 HCELL;
Uint16 LCELL;
Uint16 registo;
Uint16 a;
Uint16 Cellball;
Uint16 Vmincell;
Uint16 Sum;
Uint16 balance;
Uint16 status;
Uint16 endereco;
int commini;
int firstcomm;

float Vmin;
float tensao[6];
float tensaotemp[6];

int main(void) {
        for(i=0;i<3;i++)
        {
                buffer[i]=0x0000;
        }
        for(i=0;i<6;i++)
        {
        tensao[i]=0;
        }
    crc=0;
    commini=0;
        registo=0;
        firstcomm=2;
        inicio=0;
        Teste=3;
        t=0;
        Cellball=0;
        balance=0;
        InitSysCtrl();
        InitSpiaGpio();
```

```
      //InitSciaGpio();
      GPIOstart();
      //Interruption Configuration

      DINT;                     //Turn CPU Interruptions OFF
      // Initialize PIE control registers to their default state.

         IER = 0x0000;
         IFR = 0x0000;          // Clear Flags
         InitPieCtrl();
         InitPieVectTable();
         //----------------------->External<-----------------------

         EALLOW;
         PieVectTable.XINT1 = &xint1_isr;
         PieVectTable.XINT2 = &xint2_isr;          // Habilitar o Bloco de interrupções
         PieCtrlRegs.PIEIER1.bit.INTx4 = 1;              // Habilitar a INT4 (EXTINT1) no
Grupo 1
         PieCtrlRegs.PIEIER1.bit.INTx5 = 1;              // Habilitar a INT5 (EXTINT2) no
Grupo 1
         IER |= M_INT1;                                              //Ativar interrupções
externas do Grupo 1
         //Interruption GPIO Definition
         GpioIntRegs.GPIOXINT1SEL.bit.GPIOSEL = 3;       // XINT1 is GPIO3   FAULT
         GpioIntRegs.GPIOXINT2SEL.bit.GPIOSEL = 4;       // XINT2 is GPIO4   ALERT
         //Interruption Polarity Definition
         XIntruptRegs.XINT1CR.bit.POLARITY = 0;      // Falling edge interrupt
         XIntruptRegs.XINT2CR.bit.POLARITY = 0;      // Falling edge interrupt
         //Enable Interruptions
         XIntruptRegs.XINT1CR.bit.ENABLE = 1;        // Enable XINT1
         XIntruptRegs.XINT2CR.bit.ENABLE = 1;        // Enable XINT2

         //----------------------->SPI Communication<-----------------------
         EALLOW;
         IER |= M_INT6;
         PieCtrlRegs.PIECTRL.bit.ENPIE = 1;
         PieCtrlRegs.PIEIER6.bit.INTx1=1;     // Enable PIE Group 6, INT 1 (RX)
         PieCtrlRegs.PIEIER6.bit.INTx2=1;     // Enable PIE Group 6, INT 2 (TX)
         PieVectTable.SPIRXINTA = &spiRxFifoIsr;
         PieVectTable.SPITXINTA = &spiTxFifoIsr;
         spi_fifo_init();
         ERTM;
         PieCtrlRegs.PIEIFR6.bit.INTx1 = 0 ;
         PieCtrlRegs.PIEIFR6.bit.INTx2 = 0 ;

         EDIS;

         //---------------------->SCI Communication<-----------------------
         EALLOW;
         IER |= M_INT9;
         PieCtrlRegs.PIEIER9.bit.INTx1=1;     // PIE Group 9, INT1
         PieCtrlRegs.PIEIER9.bit.INTx2=1;     // PIE Group 9, INT2
         PieVectTable.SCIRXINTA = &sciaRxFifoIsr;
         PieVectTable.SCITXINTA = &sciaTxFifoIsr;
         scia_fifo_init();
         PieCtrlRegs.PIEIFR9.bit.INTx1 = 0 ;
```

```
        PieCtrlRegs.PIEIFR9.bit.INTx2 = 0 ;
        EDIS;

        PieCtrlRegs.PIEACK.all = PIEACK_GROUP6;
        PieCtrlRegs.PIEACK.all = PIEACK_GROUP9;

        comprep();
        EINT;
        while(commini<firstcomm);
        SciaRegs.SCITXBUF=scidata;
        bqconfig();
        a=0;
        scidata=0;
while(1)
      {
      DELAY_US(300);
      leitura();
      registo=spidata;
      switch(scidata)
      {
      case 1:
      {
            scidata=0;
            CellBalOFF();
            leituraadcs();
            sciVsend();
            break;
      }
      case 2:
      {
            scidata=0;
            CellBalArray();
            CellBalOrder();
            while(scidata!=9);
            CellBalOFF();
            while(scidata!=8);
            leituraadcs();
            sciVsend();
            SciaRegs.SCITXBUF= balance;
            scidata=0;
            break;
      }
      case 3:
      {
            scidata=0;
            CellBalArray();
            CellBalOrder();
            while(scidata!=9);
            CellBalOFF();
            while(scidata!=8);
            leituraadcs();
            sciVsend();
            SciaRegs.SCITXBUF= balance;
            scidata=0;
            break;
      }
```

```
        case 4:
        {
                scidata=0;
                leituraadcs();
                CellBalArray();
                CellBalOrder();
                while(scidata!=9);
                CellBalOFF();
                while(scidata!=8);
                sciVsend();
                SciaRegs.SCITXBUF= balance;
                scidata=0;
                break;
        }
        case 5:
        {
                leituraadcs();
                sciVsend();
                scidata=0;
                break;
        }
        }//end switch
        DELAY_US(300);
        }//end while
}//endmain

//----------------------->Interruption Routines<-----------------------

interrupt void xint1_isr(void)        //FAULT
{
    EALLOW;
      PieCtrlRegs.PIEIFR1.bit.INTx4 = 0 ;
      EDIS;
      PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

interrupt void xint2_isr(void)        //ALERT
{
    EALLOW;
      PieCtrlRegs.PIEIFR1.bit.INTx5 = 0 ;
      EDIS;
      PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

interrupt void spiTxFifoIsr(void)
{
            SpiaRegs.SPIFFTX.bit.TXFFINTCLR=1;  // Clear Interrupt flag
            PieCtrlRegs.PIEACK.all = PIEACK_GROUP6;
            DELAY_US(100);
}

interrupt void spiRxFifoIsr(void)
{
        for(i=0;i<2;i++)
    {
          rdata[i]=SpiaRegs.SPIRXBUF;        // Read SPI data
```

```
    }
      if (sdata[0]==0x0200)
      {
             status=rdata[1]&0x00FF;
      };
    spidata=rdata[1]&0x00FF;
      SpiaRegs.SPIFFRX.bit.RXFFOVFCLR=1;  // Clear Overflow flag
      SpiaRegs.SPIFFRX.bit.RXFFINTCLR=1;     // Clear Interrupt flag
      PieCtrlRegs.PIEACK.all = PIEACK_GROUP6;       // Issue PIE ack
}

interrupt void sciaTxFifoIsr(void)
{
      SciaRegs.SCIFFTX.bit.TXFFINTCLR=1;
      PieCtrlRegs.PIEACK.bit.ACK9 = 1;
}

interrupt void sciaRxFifoIsr(void)
{
      if (commini<firstcomm)
      {
             commini++;
      }
      scidata=SciaRegs.SCIRXBUF.all;
      SciaRegs.SCIFFRX.bit.RXFFOVRCLR=1;   // Clear Overflow flag
      SciaRegs.SCIFFRX.bit.RXFFINTCLR=1;   // Clear Interrupt flag
      PieCtrlRegs.PIEACK.bit.ACK9 = 1;     // Issue PIE ack
}


void error(void)
{
    asm("     ESTOP0");    //Test failed!! Stop!
    for (;;);
}

void comprep(void)
{
rdata[0]=0x00;
rdata[1]=0x00;
}

void bqconfig()
{
      sdata[0]=0x013b;          //First device adress definition
      sdata[1]=0x0100;          //Adress 01
      escrita();

      sdata[0]=0x0330;          // ADC_CONTROL REGISTER
      sdata[1]=0x0D00;          //     -      ADC_ON TS2    TS1    GPAI   CELL_SEL[2]  CELL_SEL[1]
      CELL_SEL[0]        5 = convert all cells
      escrita();

      //--------------------------------------------------

      escritaeprom();                  //EEPROM Writing Preparation
```

```
        sdata[0]=0x0340;                //Overvoltage Configuration
        sdata[1]=0x1000;                //
        escrita();

        escritaeprom();                         //EEPROM Writing Preparation
        sdata[0]=0x0342;                //Overvoltage Configuration
        sdata[1]=0x3200;                //
        escrita();

        escritaeprom();                         //EEPROM Writing Preparation
        sdata[0]=0x0343;                //COV Time Delay Configuration
        sdata[1]=0x0000;                //
        escrita();

        escritaeprom();                         //EEPROM Writing Preparation
        sdata[0]=0x0344;                //Undervoltage Configuration
        sdata[1]=0x1700;                //
        escrita();

        escritaeprom();                         //EEPROM Writing Preparation
        sdata[0]=0x0345;                //CUV Time Delay Configuration
        sdata[1]=0x0000;                //
        escrita();

        sdata[0]=0x0320;                //Clear ALERT
        sdata[1]=0x8000;
        escrita();

        sdata[0]=0x0321;                //Clear FAULT
        sdata[1]=0x0400;
        escrita();

        sdata[0]=0x0321;                //Clear FAULT
        sdata[1]=0x0000;
        escrita();

        sdata[0]=0x0333;
        sdata[1]=0x8200;
        escrita();

        sdata[0]=0x0332;                //Clear MOSFETS
        sdata[1]=0x0000;
        escrita();
}

void conversao(void)
{
        Teste=0x00;
        GpioDataRegs.GPASET.bit.GPIO1 = 1;                      //Conversion Start
        while(Teste==0x00)
        {
                Teste = GpioDataRegs.GPADAT.all;
                Teste = Teste & 0x04;
        }
        GpioDataRegs.GPACLEAR.bit.GPIO1 = 1;
        DELAY_US(300);
```

```
}

void escrita(void)
{
      //Reading Structure
      //sdata[0] --> DEVADDR    +     REGADDR
      //sdata[1] --> DATA       +     CRC
      CRCcalc();
      for(i=0;i<2;i++)
                 {
                      SpiaRegs.SPITXBUF=sdata[i];      // Send data
                 }
      DELAY_US(700);
}

void escritaeprom(void)
{
      sdata[0]=0x033a;          //EEPROM Writing Preparation
      sdata[1]=0x3500;          //Mandatory Pre-writing
      escrita();
}

void CRCcalc(void)
{
      buffer[0]=sdata[0]/0x0100;
      buffer[1]=sdata[0]&0x00FF;
      buffer[2]=sdata[1]/0x0100;
      for (i = 0; i<3; i++)
      {
      temp = crc ^ buffer[i];
      crc = CrcTable[temp];
      }
      sdata[1]=sdata[1]+crc;
      crc=0;
}

void leitura(void)
{
      //Reading Structure
      //sdata[0] --> DEVADDR + REGADDR
      //sdata[1] --> CNT+FILLER
      sdata[0]=0x023b;
      sdata[1]=0x0100;

      for(i=0;i<2;i++)
          {
               SpiaRegs.SPITXBUF=sdata[i];      // Send data
          }
      endereco=spidata;
      DELAY_US(1000);
}

void leituraadcs(void)
{
      tempCELL=0;
      sdata[0]=0x0203;
```

```
	sdata[1]=0x0100;
	for(i=0;i<6;i++)
			{
				tensao[i]=0;
			}
	for(z=0;z<10;z++)
	{
		conversao();
		sdata[0]=0x0203;
		sdata[1]=0x0100;
		for(j=0;j<12;j++)
		{
			for(i=0;i<2;i++)
			{
			   SpiaRegs.SPITXBUF=sdata[i];      // Send data
			}
			DELAY_US(2000);
			tempadc[j]=spidata;
			sdata[0]++;
		}
		for(i=0;i<6;i++)
		{
			tensaotemp[i]=tensaotemp[i]+((tempadc[i*2]*256 +
tempadc[i*2+1])*0.381493011)*0.1;
		}
	}
	for(i=0;i<6;i++)
			{
				tensao[i]=tensaotemp[i];
				tensaotemp[i]=0;
			}
}
void sciVsend(void)
{
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[0];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[1];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[2];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[3];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[4];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[5];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[6];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[7];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[8];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[9];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[10];
```

```
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
	SciaRegs.SCITXBUF= tempadc[11];
	while(SciaRegs.SCICTL2.bit.TXRDY==0){}
}
void CellBalArray(void)
{
Cellball=0;
Vmin=tensao[0];
Vmincell=0;
	for(i=0;i<6;i++)
	{
		if(Vmin>tensao[i])
		{
			Vmin=tensao[i];
			Vmincell=i;
		}
	}
	for(i=0;i<6;i++)
	{
		Cellball=(Cellball<<1);
		z=5-i;
		if(tensao[z]-tensao[Vmincell]>50)
		{
			Cellball=Cellball+1;
		}
	}
}

void CellBalOrder(void)
{
	sdata[0]=0x0332;		//MOSFETS Configuration
	sdata[1]=0x0000+Cellball*256;
	escrita();
	balance=Cellball;
	Cellball=0;
}

void CellBalOFF(void)
{
	sdata[0]=0x0332;		//Clear MOSFETS
	sdata[1]=0x0000;
	escrita();
}
```

# Annex 2 – MatLab Code

The graphical user interface code is not displayed as it gets to be pretty long and not very important, the code presented is the one that implements the routines for balancing and charging.

```
% This program takes care of the comunication between the BMS, the
% Microprocessor, the computer and the Power Supply

delete(instrfindall);
global buffer
global method
global cellball
global timer
global current
global CV
global Cell1;
global Cell2;
global Cell3;
global Cell4;
global Cell5;
global Cell6;
global A;
global on;
global fase
s = serial('COM5');
set(s,'BaudRate',9600,'InputBuffer',buffer,'OutputBuffer',1,'Timeout',13);
fopen(s);
clear out
ffase=0;
if(method==1)
    pause(timer)
    fwrite(s,method)
    out = fread(s)
end
if(method==2)
    if(fase==6 && ffase==0)

        timer=130;
        fwrite(s,method)
        display('Waiting for balancing')
        pause(timer)                %Wait balancing time
        fwrite(s,9)
        display('Turn Balancing Off')
        pause(30)
        fwrite(s,8)
        display('Balancing Routine over')
        out = fread(s)
        Cell1=(out(1)*256+out(2))*6.250/16383;
        Cell2=(out(3)*256+out(4))*6.250/16383;
        Cell3=(out(5)*256+out(6))*6.250/16383;
        Cell4=(out(7)*256+out(8))*6.250/16383;
        Cell5=(out(9)*256+out(10))*6.250/16383;
        Cell6=(out(11)*256+out(12))*6.250/16383;
        cellbal=out(13)
        if(cellbal==0 && (Cell1<4.1 || Cell2<4.1 || Cell3<4.1 || Cell4<4.1 ||
Cell5<4.1 || Cell6<4.1))
            display('Balancing over but the desired charge level was not achieved
yet')
```

```
            fase=1;
        elseif(cellbal==0 && Cell1>4.1 && Cell2>4.1 && Cell3>4.1 && Cell4>4.1 &&
Cell5>4.1 && Cell6>4.1)
            dp800 = visa( 'ni','USB0::0x1AB1::0x0E11::DP8B171800395::INSTR' ); %Create
VISA object
            fopen(dp800);                                                %Open the
VISA object created
            fprintf(dp800, ':OUTP CH1,OFF' );
            fclose(dp800);                                                %Close the
VISA object

save('topbal','x','graphCell1','graphCell2','graphCell3','graphCell4','graphCell5','gr
aphCell6','current')
            while(1)
                pause(30)
            end
        end
        ffase=1;
    end
    if(fase<6 && ffase==0)
        timer=5;
        pause(timer)
        fwrite(s,5)
        out = fread(s)
        Cell1=(out(1)*256+out(2))*6.250/16383;
        Cell2=(out(3)*256+out(4))*6.250/16383;
        Cell3=(out(5)*256+out(6))*6.250/16383;
        Cell4=(out(7)*256+out(8))*6.250/16383;
        Cell5=(out(9)*256+out(10))*6.250/16383;
        Cell6=(out(11)*256+out(12))*6.250/16383;
        if(fase<6 && (Cell1>4.2 || Cell2>4.2 || Cell3>4.2 || Cell4>4.2 || Cell5>4.2 ||
Cell6>4.2))
            fase=fase+1;
        end
        ffase=1;
    end
end

if(method==3)
    if(fase==6 && ffase==0)
        timer=130;
        fwrite(s,method)
        display('Waiting for balancing')
        pause(timer)                %Waiting for Balancing
        fwrite(s,9)
        display('Turning Balancing off')
        pause(30)
        fwrite(s,8)
        display('Balancing Routine Over')
        out = fread(s)
        Cell1=(out(1)*256+out(2))*6.250/16383;
        Cell2=(out(3)*256+out(4))*6.250/16383;
        Cell3=(out(5)*256+out(6))*6.250/16383;
        Cell4=(out(7)*256+out(8))*6.250/16383;
        Cell5=(out(9)*256+out(10))*6.250/16383;
        Cell6=(out(11)*256+out(12))*6.250/16383;
        cellbal=out(13)
```

```
        if(cellbal==0 && (Cell1<4.1 || Cell2<4.1 || Cell3<4.1 || Cell4<4.1 ||
Cell5<4.1 || Cell6<4.1))
            display('Balancing over but the desired charge level was not achieved
yet')
            fase=1;
        elseif(cellbal==0 && Cell1>4.1 && Cell2>4.1 && Cell3>4.1 && Cell4>4.1 &&
Cell5>4.1 && Cell6>4.1)
            dp800 = visa( 'ni','USB0::0x1AB1::0x0E11::DP8B171800395::INSTR' ); %Create
VISA object
            fopen(dp800);                                                 %Open the
VISA object created
            fprintf(dp800, ':OUTP CH1,OFF' );
            fclose(dp800);                                                  %Close the
VISA object

save('topbal','x','graphCell1','graphCell2','graphCell3','graphCell4','graphCell5','gr
aphCell6','current')
            while(1)
            end
        end
        ffase=1;
    end
    if(fase<6 && ffase==0)
        timer=130;
        fwrite(s,method)
        display('Balancing while Charging')
        pause(timer)                %Waiting for Balancing
        fwrite(s,9)
        display('Turn Balancing Off')
        pause(30)
        fwrite(s,8)
        display('Balancing Routine Over')
        out = fread(s)
        Cell1=(out(1)*256+out(2))*6.250/16383;
        Cell2=(out(3)*256+out(4))*6.250/16383;
        Cell3=(out(5)*256+out(6))*6.250/16383;
        Cell4=(out(7)*256+out(8))*6.250/16383;
        Cell5=(out(9)*256+out(10))*6.250/16383;
        Cell6=(out(11)*256+out(12))*6.250/16383;
        if(fase<6 && (Cell1>4.2 || Cell2>4.2 || Cell3>4.2 || Cell4>4.2 || Cell5>4.2 ||
Cell6>4.2))
            fase=fase+1;
        end
        ffase=1;
    end
end

if(method==4)
    display('Waiting for Balancing to run')
    pause(timer)
    fwrite(s,9)
    display('Sent 9')
    pause(10)
    fwrite(s,8)
    display('Sent 8')
    display('Balancing Routine Over')
    out = fread(s)
end
```

```
if (buffer==13)
    cellbal=out(13)
end

Cell1=(out(1)*256+out(2))*6.250/16383;
Cell2=(out(3)*256+out(4))*6.250/16383;
Cell3=(out(5)*256+out(6))*6.250/16383;
Cell4=(out(7)*256+out(8))*6.250/16383;
Cell5=(out(9)*256+out(10))*6.250/16383;
Cell6=(out(11)*256+out(12))*6.250/16383;

    display('Measuring Current')
    dp800 = visa( 'ni','USB0::0x1AB1::0x0E11::DP8B171800395::INSTR' ); %Create VISA 
object
    fopen(dp800);                                                       %Open the VISA 
object created
    fprintf(dp800, ':MEAS:ALL? CH1' );
    meas_CH1 = fscanf(dp800);
    A=str2num(meas_CH1);
    A=A(2);
    if(x==0)
        current=A;
    else
        current=[current A];
    end
    fclose(dp800);
    x;
    current;
    axes(handles.currentgraph)
    plot(x,current)

fclose(s)
delete(s)
clear s
```

# Annex 3 – Article

# Balancing management system for improving Li-ion batteries capacity usage and lifespan

Miguel D. Beirão, Maria do Rosário A. Calado, José A.N. Pombo, Sílvio J.P.S. Mariano
mdbeirao@gmail.com; rc@ubi.pt; Jose_p@portugalmail.com; sm@ubi.pt
IT – Instituto de Telecomunicações, Universidade da Beira Interior
Covilhã, Portugal

***Abstract*— Society's' growth of use on mobile devices makes the study of batteries a really important subject. Mobile phones and electric vehicles are, among others, battery dependent devices. Li-ion cells are the main choice for battery dependent applications. Different charging of cells as they age put the balancing management systems on the spotlight as a study subject. This paper proposes a balancing management system (BMS) for improving the capacity usage and lifespan of Li-ion batteries. Algorithms to implement the proposed BMS are proposed and experimental results are obtained for multiple battery packs with up to 192 series cells.**

***Keywords—BMS; Battery Charge Equalization; Cell Balancing; Battery Management System.***

## I. INTRODUCTION

Society's mobile devices usage grows on a regular basis making the study of batteries a really important subject. The longevity of battery charges is nowadays a big issue for people who are usually outdoors. Electric Vehicles (EV), a worldwide growing market, are pretty dependent on batteries capacity and usage as EVs can only go as far as the battery allows [1], [2]. Home battery devices that can store energy from either local renewable sources or from the grid are also a promising technology highly dependent on battery usage patterns [3].

In a battery pack, all cells are different and as time goes by, each cell in a battery pack gets to age distinctly originating differences between one another when charging them. Although parallel cells are self-balanced, that does not happen when putting them in series [4], [5].

Some battery balancing methods were proposed in literature, in order to create a system capable of applying a charging algorithm [6]. The method presented in this paper is based on the most frequently used balancing system [7].

The Li-ion cells used in this project were designed to have a voltage drop between 3 V and 4.3 V and if that voltage goes below 3 V it may enter into a deep discharge state that takes a really long time to recover from. Also, if the voltage reaches values above 4.3 V the cell can be destroyed due to overheating. This makes it really important to be aware of each cell's voltage and to take some precautions in order to prevent severe damages [8], [9].

To have more capacity we can easily think on adding some parallel cells to the battery resulting in a significant increase of the battery size. What if it was possible to "increase" the batteries capacity by managing its charge in order to allow it to store some more energy?

In order to prevent crossing battery safe voltage limits, some batteries are used between 20% and 80% of their capacity using only 60% of it. With the right charging management system it is possible to use batteries between 5% and 95% of the battery capacity increasing its capacity usage to 90%, which represents an increase of 50% of the previously available energy [10].

This paper is organized as follows: Section II presents and evaluates cell balancing methods available in literature and then approaches cell charging methods also proposed in literature. Section III describes the whole proposed system architecture, the Battery Management System and explains the algorithms designed and implemented. Section IV shows the results obtained by the developed system. Section V concludes the paper and discusses the system implementation.

## II. BACKGROUND

### A. Balancing Methods

To balance batteries there are two kinds of methods as shown in Fig. 1. , the active ones and the passive ones. Active cell balancing methods have a lot of different topologies and can be split into two groups.

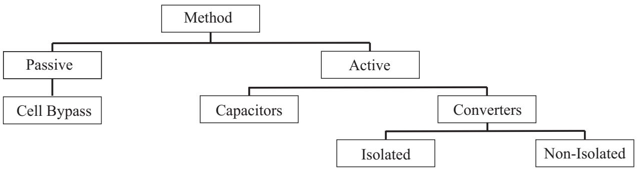


Fig. 1. Balancing Methods

The charge shuttle group, where the energy transference is achieved by using a capacitor in paralel with a chosen cell or pack in order to transfer the energy from higher voltage cells or packs to lower voltage ones, and the converter group where the battery pack is the source to the converter that will then charge the chosen cells. Most topologies in both groups are based on switches in order to control cell balancing. Depending on the topology it is possible to have balancing cell-to-cell, cell-to-pack, pack-to-cell and cell-to-pack-to-cell [11], [12].

978-1-5090-2320-2/16/$31.00 ©2016 IEEE

In the charge shuttle group there are lots of topologies from the most simple ones like the switched capacitor to more complex ones like the modularized switched capacitor [6], [12], [16–18]. In literature, besides this two methods it is possible to find the double tiered switched capacitor, automatic switched capacitor [13], single switched capacitor [14], [15], chain structure of switched capacitor [16], and series-parallel switched capacitor [17]. Fig. 2 presents a modularized switched capacitor where it is possible to transfer energy from cell to cell on the first tier capacitors and from pack to pack on the second tier capacitors.

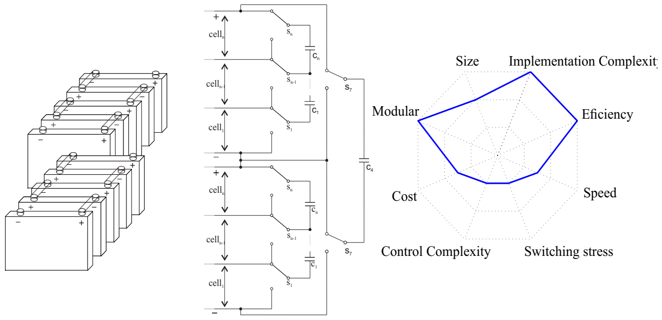


Fig. 2. Modularized Switched Capacitor

The converter group can be split into two other groups that may be nominated as isolated converters and non-isolated converters. This kind of balancing is based on using the pack or cell as the source for the converter which will then transmit energy to another cell or pack. The first group is based on isolated DC-DC converters and has various topologies like multi-winding transformer, multiple transformers, and single switched transformer. Fig. 3 shows an example of an isolated converter [19], [20].

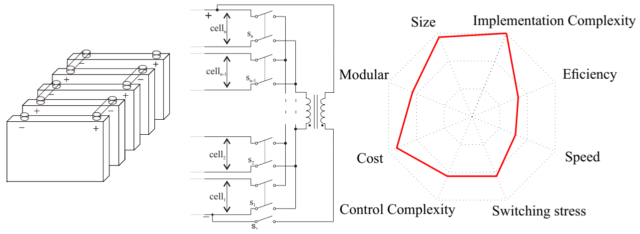


Fig. 3. Switch Matrix Transformer

The second group is based on non-isolated DC-DC converters like buck, boost, buck-boost or Cuk that can be unidirectional or bidirectional depending on the topology. Fig. 4 presents an example of a non-isolated converter [21].

In passive cell balancing methods, the cells are balanced by discharging, when the battery is not being used, or by providing an alternative path for the current to flow while it is being charged. There are a few topologies for passive methods like fixed shunt resistor, shunt resistor, complete shunting [12] but the most popular is the one presented in Fig. 5 called switched shunt resistor where, to balance the cells, a semiconductor is used to allow or not the current from flowing through the resistor [22].

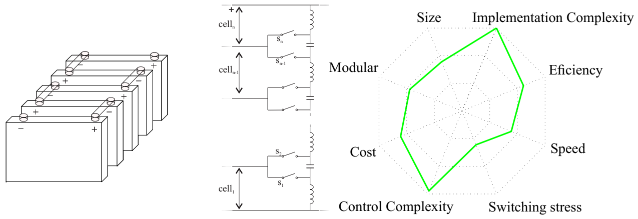


Fig. 4. Cuk converter

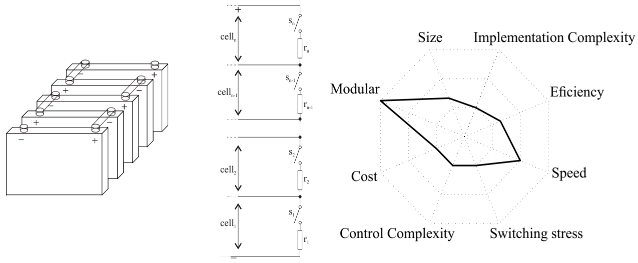


Fig. 5. Switch Shunt Resistor

The Fig. 6 shows a scheme to compare the different balancing methods. Due to its reliability and simplicity and for the fact that it is widely used in vehicle industry the dissipative resistors method was the chosen in this paper. This method is effective during the whole charging process and if the resistor value is properly selected, it can be even more efficient. Also, this method has the advantage of not requiring complex control like the active ones [23].

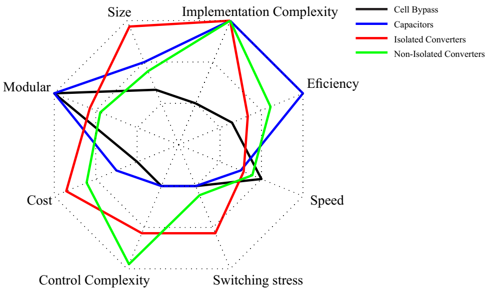


Fig. 6. Balancing methods comparison

## B. *Charging Methods*

When balancing a battery pack, during charging, a lot of issues might come if it is made without taking some considerations into account. When charging a cell, the fact that it has current flowing to it makes voltages rise. The higher the current, the bigger the difference between the real voltage and the one read by the measurement system, making it really important to study charging methods in order to avoid misleading voltage readings. This issue gets to be amplified with the increasing number of series cells. In literature it was possible to find different charging methods, schematized in Fig. 7.

The CC-CV method is pretty good for charging battery packs but, when using a voltage based balancing method, due to high currents flowing through the cells it can mislead the system to interpret that the cells are fully charged when in fact they are pretty far from that making the whole process much slower because of switching to constant voltage phase too early. Fig. 8 shows that this method has three main phases where the first one, trickle phase, is only implemented when the cell gets to discharge more than $V_{trickle}$ (recommended value). Phase two consists of a constant current power supply until it reaches a $V_{charge}$ where it changes to constant voltage phase decreasing the current as it charges finishing charge when it reaches $I_{min}$ [24], [25].

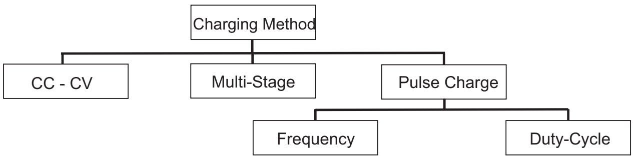


Fig. 7. Charging Methods

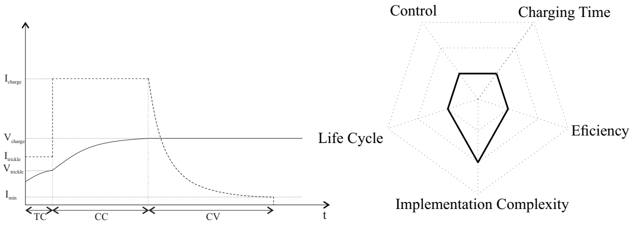


Fig. 8. CC/CV Charging Method

Multi Stage charging consists of having multiple stages where in each stage there is a different current, Fig. 9. This method can vary on the currents and the conditions for changing the stage. A good criterion to switch between stages is when the cell voltage reaches $V_{[charge]}$ making the current reduce and the voltage readings approach a more realistic value. There are also a few works that suggest using constant voltage charging on the last charging stage [24], [26].

Pulse charging method can be implemented by controlling its frequency or its duty cycle [27]. In both methods, it's possible to see that they consist of finding the best duty cycle or frequency during a searching process based on minimizing impedance and maximizing charging current. This method consists of sending voltage pulses to the battery, Fig. 10.

The charging method adopted in the proposed system was the Multi Stage due to the tradeoff between its efficiency, control simplicity and charging time as can be seen in the radar plots on each charging method figures[28].

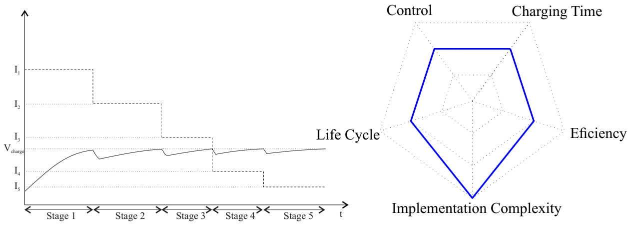


Fig. 9. Multi-Stage Charging Method

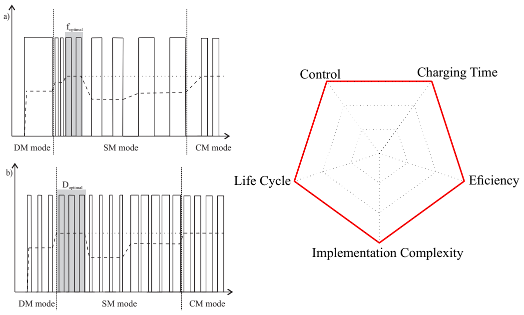


Fig. 10. Pulse Charging Methods: a) Variable Frequency Pulse Charge; b) Variable Duty Pulse Current.

## III. HARDWARE DEVELOPMENT

### A. *System Architecture*

To implement the proposed balancing algorithms it was adopted a BMS controlled by a Texas Instruments Picollo C2000 connected to MatLab through a Serial Communications Interface (SCI) protocol. In order to control the BMS the microprocessor and the BMS are connected using a Serial Peripheral Interface (SPI). The charging and balancing are controlled during the whole process by the connection of MatLab with a Rigol DP832A power supply based on a Standard Commands for Programmable Instruments (SCPI) protocol. The whole system is represented in Fig. 11.

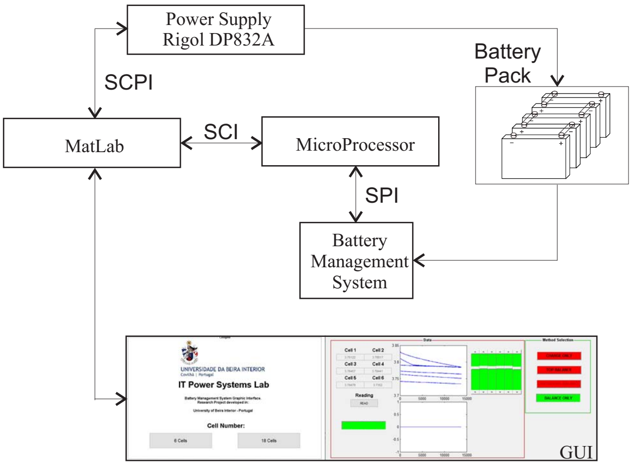


Fig. 11. System Overview

By means of a Guided User Interface (GUI), the user can choose the kind of algorithm he wants the system to run between Top Balance, balancing only when the charging phase ends and Continuous Balance, balancing through the whole charging process. Afterwards MatLab takes care of the rest of the process by controlling the power supply and the BMS system through the whole process.

### B. *BMS Development*

The proposed BMS system was based on the integrated circuit (IC) bq76pl536a-q1 from Texas Instruments for balancing the batteries, allowing multiple IC stacking in order to manage up to 192 cells. This circuit can measure the voltage on each cell sending that information to the microprocessor which can then apply the balancing algorithm by controlling the IC MOSFETs. The whole BMS system is controlled by the microprocessor that communicates with the IC through an SPI connection, Fig.12.

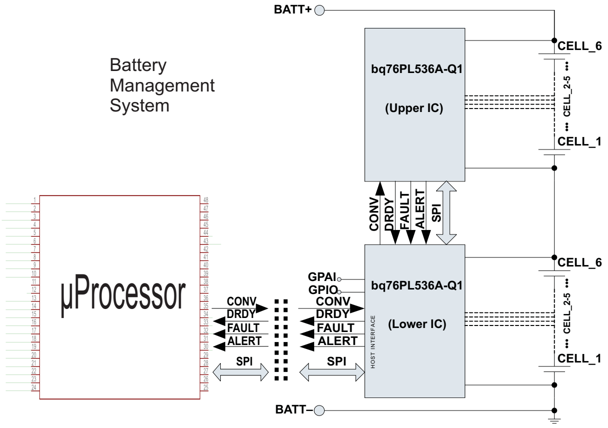


Fig. 12. BMS Overview

To balance the cells, the microprocessor sends a conversion start signal to the IC that will activate the Analog to Digital Converters (ADCs), reading and replying the voltage in each cell. Then it is up to the microprocessor, depending on user initial choices, to decide what to do in order to run the chosen algorithm. This IC has 14 bit ADCs performing conversions with great precision and accuracy. In order to balance the cells, the microprocessor can send commands to the IC to control the MOSFETS allowing the higher voltage cells to discharge or to charge a little slower in order to achieve more desirable levels permitting the system to achieve better capacity usage from the whole pack.

### *C. Algorithm Comparison*

In this paper, two charging algorithms where compared, one found in the literature and the other proposed by the authors in this paper. In order to understand the algorithms of Fig. 13 shows how the whole procedure occurs starting by picking what kind of balancing is desired, Top Balance or Continuous Balance.

If Top Balance is chosen, the system starts by charging the pack through a Multi Stage Charging Algorithm switching between 6 stages where the first five have different charging currents and the switching condition is achieved when a cell voltage reading is over 4.2 V. The last stage is the balancing one where charging is stopped and balancing is made. After that balancing stage, if the cells are all above a chosen value considered the one for charge ending, the algorithm ends, otherwise it restarts the whole procedure until this condition is guaranteed. On the other hand, if Continuous Balance is chosen, it will still use a Multi Stage Charging Algorithm with 6 stages but it will be balancing and charging all at the same time by charging some cells more than others through the control of the IC MOSFETS. Once a cell reaches 4.2 V it stops charging, balances and just like Top Balancing if all cells aren’t charged yet, it restarts the whole procedure.

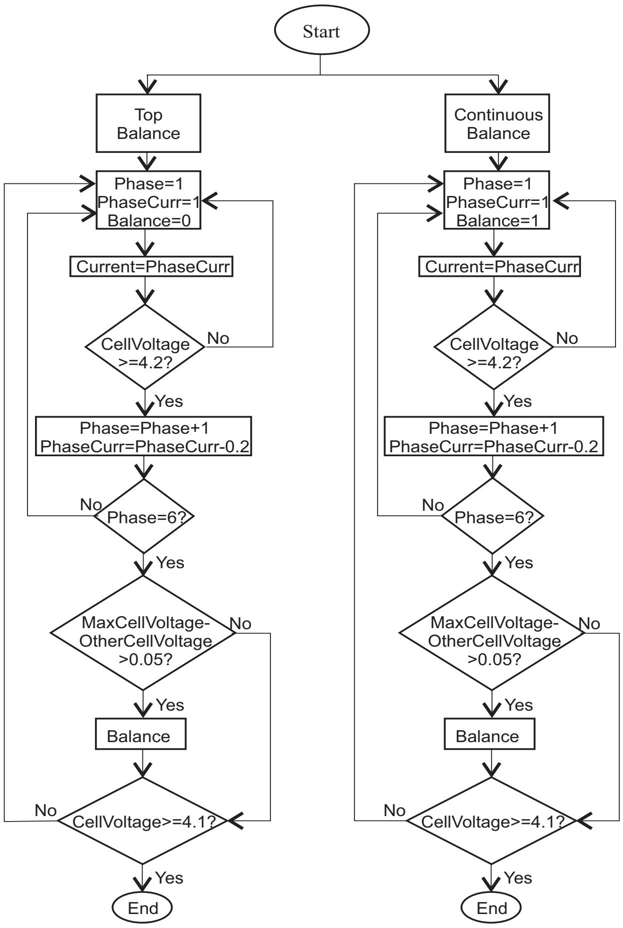


Fig. 13. Cell Balancing Routine

## IV. RESULTS

The experimental stand was designed to be possible to work with as many cells as desired up to 192, the IC’s limit. Each existing board can control up to 6 cells and has a board lower surface connection to the previous board and a board upper surface connection to the next board. The main board has a host connection that is used to communicate with the microprocessor, passing the information from and to all the stacked boards, Fig. 14.

Fig. 14. Real Life Overview

The implementation of the previously presented system allowed obtaining the characteristics shown in Fig. 15, for Top

Balance, and in Fig. 16, for Continuous Balance. Each characteristic in the figures represents one cell in a six cell battery pack. It is important to notice that time axis in Fig. 15 goes up to 15000 seconds and in Fig. 16 it reaches 10000 seconds. The acquisition was stopped once the system considered that all stopping conditions were met, in this case, all cells voltages above 4.1 V and difference between them lower than 0.05 V. Another fact that is pretty important to consider for this system is that no tests were made on the cells that were obtained from previously working battery packs so their aging conditions might have been really discrepant. Equipment limitations made it impossible to be even more rigorous on this project but further studies will be made for the improvement of this system.

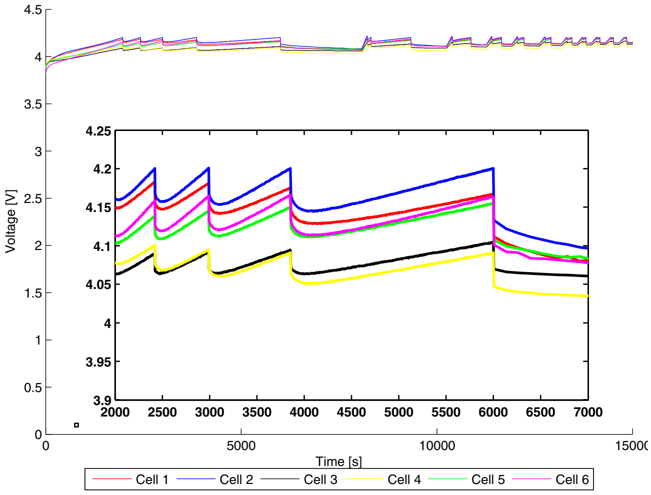


Fig. 15. Top Balance Acquisition

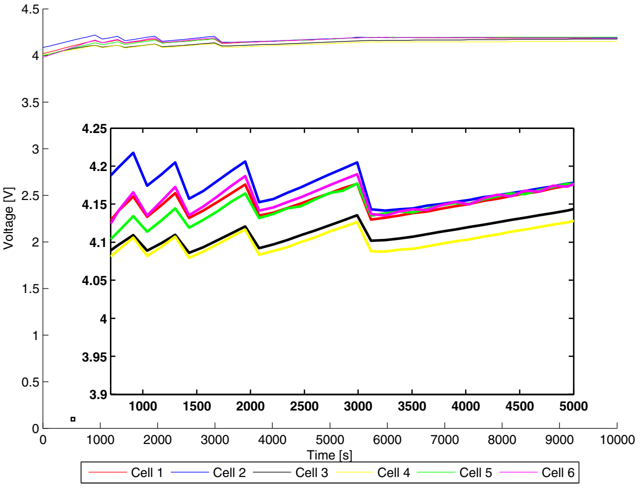


Fig. 16. Continuous Balance Acquisition

## V. CONCLUSIONS

This paper proposed a combined charging and balancing management system for improving the capacity usage and lifespan of Li-ion batteries. The system is able to manage multiple battery packs with up to 192 series cells.

A proposed algorithm was implemented showing better performances when compared with a Top Balance algorithm.

Obtained data shown that improvements can be made to the system since the thresholds for balancing can be smaller allowing the system to be even more efficient and Multi Stage charging method can be optimized.

## REFERENCES

[1] C. Piao, Z. Wang, J. Cao, W. Zhang, and S. Lu, “Lithium-Ion Battery Cell-Balancing Algorithm for Battery Management System Based on Real-Time Outlier Detection,” Hindawi Publ. Corp., vol. 2015, no. Mathematical Problems in Engineering, 2015.

[2] Y. Drori and C. Martinez, “The benefits of cell balancing,” Appl. Note, Xicor Inc., pp. 1–9, 2005.

[3] B. Aksanli and T. Rosing, “Optimal battery configuration in a residential home with time-of-use pricing,” 2013 IEEE Int. Conf. Smart Grid Commun. SmartGridComm 2013, pp. 157–162, 2013.

[4] H. Popp, J. Attia, F. Delcorso, and A. Trifonova, “Lifetime analysis of four different lithium ion batteries for ( plug–in ) electric vehicle,” 2014.

[5] I. Technologies, Selection of the MOSFET for Faster Cell Balancing of Li-Ion Batteries, no. February. 2013.

[6] J. Cao, N. Schofield, and A. Emadi, “Battery balancing methods: A comprehensive review,” 2008 IEEE Veh. Power Propuls. Conf. VPPC 2008, pp. 3–8, 2008.

[7] I. Aizpuru, U. Iraola, J. M. Canales, M. Echeverria, and I. Gil, “Passive balancing design for Li-ion battery packs based on single cell experimental tests for a CCCV charging mode,” 4th Int. Conf. Clean Electr. Power Renew. Energy Resour. Impact, ICCEP 2013, pp. 93–98, 2013.

[8] G. Altemose, “Batteries and Fuel Cells Achieving cell balancing for lithium-ion batteries,” Hearst Electronics Products, 2008. [Online]. Available: http://ams.aeroflex.com/. [Accessed: 27-Jan-2016].

[9] M. Einhorn, W. Roessler, and J. Fleig, “Improved performance of serially connected Li-ion batteries with active cell balancing in electric vehicles,” IEEE Trans. Veh. Technol., vol. 60, no. 6, pp. 2448–2457, 2011.

[10] D. Cadar, D. PETREUS, T. PATARAU, and N. PALAGHITA, “Active Balancing Method for Battery Cell Equalization,” Acta Tech. Napocensis, vol. 51, no. 2, pp. 1–5, 2010.

[11] F. Baronti, R. Roncella, and R. Saletti, “Performance comparison of active balancing techniques for lithium-ion batteries,” J. Power Sources, vol. 267, pp. 603–609, 2014.

[12] J. Gallardo-Lozano, E. Romero-Cadaval, M. I. Milanes-Montero, and M. A. Guerrero-Martinez, “Battery equalization active methods,” J. Power Sources, vol. 246, pp. 934–949, 2014.

[13] Y. Ye and K. Cheng, “An Automatic Switched-Capacitor Cell Balancing Circuit for Series-Connected Battery Strings,” Energies, vol. 9, no. 3, p. 138, 2016.

[14] S. R. Raman, X. D. Xue, and K. W. E. Cheng, “Review of Charge Equalization Schemes for Li-ion Battery and Super-Capacitor Energy Storage Systems,” in IEEE Conference Publications ICAECC, 2014, pp. 1–6.

[15] J. Qi and D. Dah-Chuan Lu, “Review of battery cell balancing techniques,” 2014 Australas. Univ. Power Eng. Conf. AUPEC 2014 - Proc., no. October, pp. 1–6, 2014.

[16] M. Y. Kim, C. H. Kim, J. H. Kim, and G. W. Moon, “A chain structure of switched capacitor for improved cell balancing speed of lithium-ion batteries,” IEEE Trans. Ind. Electron., vol. 61, no. 8, pp. 3989–3999, 2014.

[17] Y. Ye and K. W. E. Cheng, “Modeling and Analysis of Series-Parallel

Switched-Capacitor Voltage Equalizer for Battery/Supercapacitor Strings," IEEE J. Emerg. Sel. Top. Power Electron., vol. 3, no. 4, pp. 977–983, 2015.

[18] W. C. Lee, D. Drury, and P. Mellor, "Comparison of passive cell balancing and active cell balancing for automotive batteries," 2011 IEEE Veh. Power Propuls. Conf. VPPC 2011, 2011.

[19] M. M. U. Rehman, F. Zhang, M. Evzelman, R. Zane, and D. Maksimovic, "Control of a series-input, parallel-output cell balancing system for electric vehicle battery packs," 2015 IEEE 16th Work. Control Model. Power Electron., pp. 1–7, 2015.

[20] M. M. Ur. Rehman, M. Evzelman, K. Hathaway, R. Zane, G. L. Plett, K. Smith, E. Wood, and D. Maksimovic, "Modular approach for continuous cell-level balancing to improve performance of large battery packs," 2014 IEEE Energy Convers. Congr. Expo. ECCE 2014, pp. 4327–4334, 2014.

[21] F. Mestrallet, L. Kerachev, J. C. Crebier, and A. Collet, "Multiphase interleaved converter for lithium battery active balancing," IEEE Trans. Power Electron., vol. 29, no. 6, pp. 2874–2881, 2014.

[22] J. D. Welsh, "A Comparison of Active and Passive Cell Balancing Techniques for Series / Parallel Battery Packs," The Ohio State University, 2009.

[23] K. Vitols, "Design of an embedded battery management system with passive balancing," in 2014 6th European Embedded Design in Education and Research Conference (EDERC), 2014, pp. 142–146.

[24] W. Shen, T. T. Vo, and A. Kapoor, "Charging algorithms of lithium-ion batteries: An overview," Proc. 2012 7th IEEE Conf. Ind. Electron. Appl. ICIEA 2012, pp. 1567–1572, 2012.

[25] A. Al-Haj Hussein and I. Batarseh, "A review of charging algorithms for nickel and lithium battery chargers," IEEE Trans. Veh. Technol., vol. 60, no. 3, pp. 830–838, 2011.

[26] D. Anseán, M. González, J. C. Viera, V. M. García, C. Blanco, and M. Valledor, "Fast charging technique for high power lithium iron phosphate batteries: A cycle life analysis," J. Power Sources, vol. 239, pp. 9–15, 2013.

[27] L.-R. Chen, "A Design of an Optimal Battery Pulse Charge System by Frequency-Varied Technique," IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 398–405, Feb. 2007.

[28] A. Kapoor, "Charging algorithms of lithium-ion batteries: An overview," in 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2012, pp. 1567–1572.