Martin Gärtner, BSc

Design and Implementation of a NFC-based Solution for Secure Battery Management Systems

Master's Thesis

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Information and Computer Engineering

submitted to

Graz University of Technology

Supervisor

Ass.Prof. Dipl.-Ing. Dr.techn. Christian Steger

Advisors

bak. elektr. Dipl.-Ing. Fikret Basic
Dipl.-Ing. Robert Kofler (NXP Semiconductors Austria GmbH und Co KG)

Institute of Technical Informatics

Graz, Mai 2021

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used.
The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date	Signature

Acknowledgement

This work was carried out at the Institute of Technical Informatics at the Technical University of Graz.[1]

I would like to thank everyone who made this master thesis possible and supported me. My special thank goes to Fikret Basic for the countless hours of review and revision, the mentorship during project as well as the numerous valuable tips.
In particular, I would like to thank Professor Steger for the supervision of the overall project as well as the intensive support especially during the writing process of the thesis. I would also like to express my sincere gratitude to Claudia Laube and Christian Seifert for their support and good cooperation during the development of the prototypes.

Furthermore I would like to thank all members of the workforce of NXP Semiconductors Austria GmbH und Co KG who supported this thesis. My special thank goes to Robert Kofler, for the supervision and intensive guidance during the prototyping process. Also I want to thank Kyriakos Neophytou for his assistance and extensive support regarding all NFC related problems. I would like to thank Radoje Ivantic too, in particular for his support in organizational matters.

[1]This master thesis was carried out within the scope of the funding project „EFREtop: Securely Applied Machine Learning - Battery Management Systems“ (Acronym „SEAMAL BMS“, FFG Nr. 880564).

Kurzfassung

Das Ziel dieser Masterarbeit ist die Evaluierung von Batteriemanagementsystemen bezüglich ihrer Security. Angriffsvektoren werden ausgemacht und passende Sicherheitsvorkehrungen diskutiert und ergriffen. Desweiteren soll der Einsatz und die Sinnhaftigkeit des Einsatzes von NFC Technologie in automotive Batteriemanagementsystemen erforscht werden.

Dafür wurde im Zuge der Arbeit ein Prototyp für ein verteiltes Batteriemanagementsystem entworfen. Verschiedene Möglichkeiten die gewünschte Funktionalität zu implementieren werden im Zuge der Arbeit vorgestellt und deren Vor- und Nachteile gegeneinander abgewogen.

Im Speziellen wird der Anschluss der Temperatursensoren drahtlos über eine NFC- Schnittstelle ausgeführt, was in weiterer Folge Produktionskosten einspart. In Verbindung mit dieser Anbindung wird auch die Möglichkeit einer Maßnahme gegen Produktpiraterie demonstriert, die den Einsatz von Nachbauzellen verhindert und somit die Sicherheit des gesamten Systems steigert.

Der Einsatz von NFC Technologie wurde auch für das Auslesen des Fehlerspeichers im Zuge dieser Masterarbeit getestet. Hier wird mittels Prototyp evaluiert wie das ganze System in einen Energiesparmodus versetzt und mittels NFC über einen externen Reader aufgeweckt werden kann.

Abstract

The goal of this master thesis is the evaluation of Battery Management Systems regarding their security. Attack vectors are identified, and appropriate security measures are considered and implemented. Furthermore, the use and feasibility of NFC technology in automotive Battery Management Systems will be investigated.

For this purpose, a prototype for a distributed Battery Management System was developed in the course of the thesis. Different ways to accomplish the desired functionality will be introduced in the course of the thesis and their benefits and drawbacks will be compared against each other.

In particular, the connection of the thermal sensors is realized wirelessly via a NFC interface, which subsequently saves production costs. In combination with this connection, the possibility of an anti-counterfeiting measure is also being demonstrated, which inhibits the use of replica cells and thus enhances the safety of the overall system.

The use of NFC technology was also investigated for the readout of the fault memory in the course of this master thesis. Here, the prototype is assessed regarding the possibility to put the whole system into a power saving state and to wake it up via an external reader by utilizing NFC.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The Battery Management System (BMS) is an element of the safety system for battery packs. Thus, it is the one of the tasks of the BMS to keep the operating conditions of the battery cells in a certain range and to take precautions if necessary.

The most devastating incident, thermal runaway, should be detected as early as possible for appropriate countermeasures to be taken. A thermal runaway precedes a lead time of several hours, but once the tipping point is passed, it is often too late [Dou12]. In conventional implementations of BMS, the transition to a safety critical state is detected by temperature sensors. The temperature inside the cell is difficult to measure, but it is the most reliable indication that a thermal runaway is imminent [Dou12]. By the time a critical cell state is recognized by a temperature sensor placed in the vicinity of the cell, it is often already too late [Law].

Therefore, it would be beneficial to use additional pressure sensors to monitor the cells. The use of pressure sensors by BMS has been difficult or not even feasible so far [Law]. However conventional implementations of BMS are using the measured temperature values to approximate the pressure inside the battery cell. A potential solution would be to incorporate pressure or temperature sensors inside the battery cell via a NFC interface.

1.2 Goals

The aims of this thesis are to highlight the security aspects of Battery Management Systems used in electric vehicles and the use of NFC interfaces to connect pressure and temperature sensors, as well as the wireless transmission of the cell history to determine the State Of Health (SOH).

The following goals are defined:

- Define the security requirements of BMS and find out vulnerabilities
- Developing an application prototype
  - showing the contactless readout of sensors used in BMS
  - demonstrating the contactless readout of the cells fault memory
- Evaluate different key parameters of the designed prototype, including communication time and communication latency, among others
- Define the template for further development and set a guideline for future work

1.3 Structure

The thesis is roughly structured into the following chapters:

Chapter 2 deals with the available literature of BMS. Here, the functions of a BMS are summarized, as well as the latest developments in BMS, such as the use of wireless interfaces in BMS design. Then, an overview of the literature dealing with threats and risk analysis of BMS is given.

In chapter 3 an overview of the system design is given. Here the requirements to be fulfilled by the design are specified. Then, different high-level design options are evaluated and their advantages and disadvantages are weighed against each other. Also, a threat analysis of the different designs is accomplished here, as well as necessary and meaningful countermeasures are discussed.

In chapter 4 specific details of the implementation are discussed. In this chapter, the toolchain used is described and the details of the software implementation are discussed in more detail. The data transmitted on the various buses of the prototype are also discussed here. Hardware dependent implementation details and implemented security measures are also discussed in more detail in this chapter.

The implemented prototype is evaluated in chapter 5. Execution times and the achieved transmission speeds are measured and discussed.

A conclusion is given in chapter 6. Also advancements of the prototype developed and possibilities for further research are outlined in this chapter.

Chapter 2

Related work

2.1 Battery Management Systems

The main purpose of a Battery Management System is to keep the battery in a so called Safe Operating Area (SOA). In doing so, the functions and some parameters must be constantly monitored, because if they are not sufficiently monitored, the battery cells go into a state that cannot be controlled. The battery cells then heat up themselves and the system reaches a state of singularity. A thermal runaway is the result [Zim17].

The following sections are describing parameters determining the SOA. If one of those parameters is out of bounds it could lead to a thermal runaway. Therefore those parameters should be monitored by a Battery Management System while the batteries are in operation.

2.1.1 Functions of a Battery Management System

In this chapter, the tasks of a Battery Management System will be examined in more detail. The key figures used by the Battery Management System to fulfil its task, such as SOC or SOH, are also dealt with in more detail in this chapter.

2.1.1.1 Temperature measurement

Temperature measurement is necessary to keep the battery in its SOA. Usually NTC sensors are used to measure the temperature. Since the temperature change is relatively slow, a measurement every second is sufficient. The temperature measurement is important to keep the battery inside the SOA. If the battery exceeds a certain temperature, it is the task of the BMS to take certain measures, such as switching off consumers or switching on the cooling, to bring the battery back to a safe operating state [Law].

2.1.1.2 Cell Voltage measurement

The voltage of the individual cells must be measured by the Battery Management System at least once per hour. The required measuring range here is between 0 and 5 volts [Zim17].

2.1.1.3 Current Measurement

The cells of a battery pack are usually connected in series. Therefore, the current flowing through the battery pack is usually the same for each cell. When the current is measured by BMS, this usually happens once for the entire battery pack. Usually the current is passed through a shunt resistor and the voltage drop across this resistor is measured. The current flowing through the battery can then be calculated back from the voltage drop across the shunt resistor [And10].

2.1.1.4 Balancing

The cells that make up a battery pack always have different internal resistances and different capacities and are therefore charged at different rates. Since the cells in a battery module are usually connected in series, the charging current always flows through all the cells of the battery and therefore all the cells are charged simultaneously. Due to the different physical constellation of the individual cells, however, some cells always finish charging earlier than others. Because all cells are charged as soon as a charging current flows, the charging current must be switched off as soon as the first cell has finished charging. Otherwise, this particular cell would be overcharged, which would inevitably put it outside the Safe Operating Area [And10].

When discharging, one has the same problem. After the cells are connected in series, as soon as a current is drawn from the battery, all cells are discharged equally. When the first cell is completely discharged, the discharge current must be stopped, otherwise this cell would be deeply discharged and thus be outside its Safe Operating Area (SOA). The charges remaining in the other cells remain unused [And10].

When the battery is balanced, it is the cell with the lowest capacity that determines when to stop charging and when to stop discharging. However, if the battery is used for a longer period of time, the states of charge of the respective cells shift and before the cell with the lowest capacity is fully discharged, another cell is fully discharged and stops the discharge process. The discharge process has now been stopped earlier than would have been necessary with a balanced battery [And10].

To stop this, the battery must be balanced. The battery is charged and energy is removed from the cell that stops charging because it is full. Then the whole battery continues to charge until another cell is full, from which energy is then taken. This continues until all cells are fully charged. The battery is now balanced. There are different methods of extracting energy from the full cell, which are illustrated in figure 2.1 [And10].

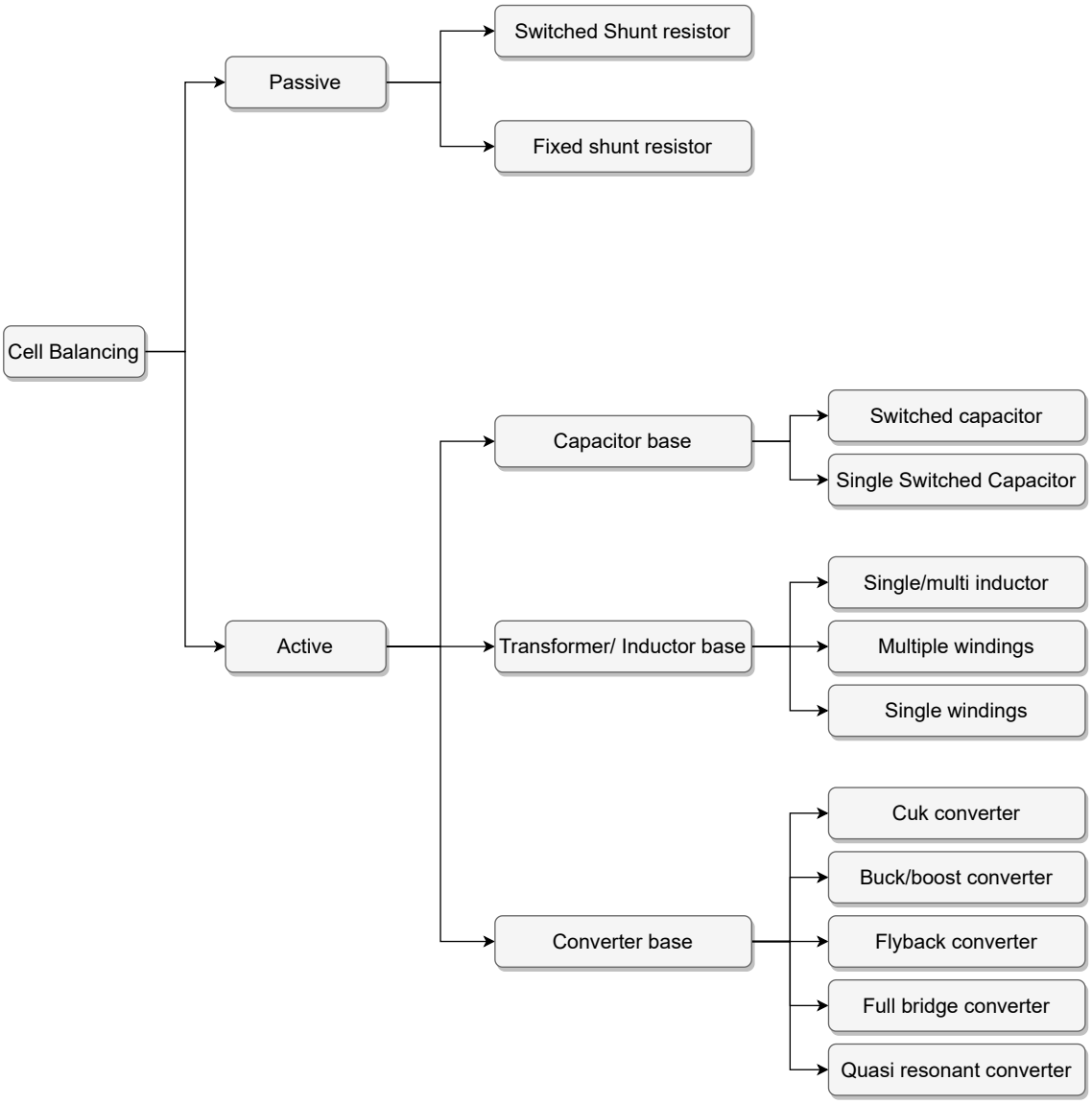


Figure 2.1: Overview of cell balancing techniques [Mar14; Bra+; CSE08]

Depending on how the energy is taken from the full cell, one decides between active and passive cell balancing. In passive cell balancing, the excess energy is converted into heat via a resistor. This balancing method is inexpensive to produce, but takes a very long time because the energy stored in the cell can only be converted into heat very slowly, otherwise the battery would overheat. In active balancing, the excess charge is actively taken from the respective cell by transformers, capacitors or converters and used to charge the other cells. The advantage with this method is that the balancing process is faster and less energy is lost during balancing. However, the hardware required for this balancing method is more expensive to manufacture [And10].

2.1.1.5 Battery cell behaviour

Battery models are used to understand the behavior of the battery in the BMS. A distinction is made between online and offline methods. In the case of offline methods, the model and its parameters are determined statically in the laboratory. In online methods, the individual parameters of the model are dynamically adjusted during the battery runtime [Ber01; AMA12; Ng+09; PPJ01; Col+07; Hue98; SRB03].

2.1.2 Battery Management System Architectures

Depending on the size and the field of application of the BMS, different types of Battery Management Systems are used. One decides here among centralized topologies which are mainly used for the management of small battery systems. For large batteries, modular and distributed topologies are predominantly used. To build these architectures, the task domain of the BMS is divided into different components 2.1.2.1, which are responsible for different individual tasks [And10].

2.1.2.1 Components

Battery Management Systems are usually composed of several layers with different modules that perform specific tasks. In the scientific literature the individual components are named differently by different researchers. Since the project is developed jointly with NXP, the prevailing nomenclature at NXP will be briefly summarized below.

Battery Pack Controller: The Battery Pack Controller (BPC) is the top level of the battery management stack. Usually it consists of a powerful Micro Controller Unit (MCU) and the required communication periphery. One function of the Battery Pack Controller is controlling the safety peripherals. This encompasses the breaker as well as the heating and cooling system of the battery. The acquisition of the measured values, i.e. the communication with the Battery Cell Controller is a further task of the Battery Pack Controller. The communication with other ECU's in the vehicle via the in-vehicle network is also one of the BPC's responsibilities. This is typically done via a CAN interface, whereby it must be ensured that the CAN voltages are isolated from the high voltage levels within the battery [And10].

Battery Cell Controller: The purpose of the Battery Cell Controller (BCC) is the measurement of the cell voltages and temperatures for the individual cells with sufficient precision and speed. Additionally the BCC carries out the cell balancing. The cell balancing can be performed individually for each cell or across the cell stack. Although inter cell balancing is more complex, it can save on component costs and increase efficiency. If each cell has its own BCC, one speaks of a Cell Monitoring Unit (CMU). If this unit is installed on a separate PCB, the board is called a cell board. A BCC which is responsible for monitoring several cells is referred to as a Module Management Unit (MMU) [And10].

2.1.2.2 BMS Topologies

The topology describes how the different components are physically arranged. Depending on where the individual components of the BMS are distributed, there are various advantages and disadvantages.

According to the technical literature, Battery Management Systems can be classified by topology into 3 basic types: Centralized, Modular and Distributed. These systems differentiate in the way the individual components of the Battery Management System are split up and spatially arranged [And10; Bra+].

- **Centralized** Battery Management System architectures are gathering all components of the Battery Management System on a single PCB. This kind of BMS is primarily used for the management of small batteries [And10; Vem19].

  The NXP reference design for the centralized BMS system is outlined in chapter 2.1.2.3.

- **Modular** Battery Management Systems are spreading out the different modules throughout the battery. Each of this modules is responsible for both, the measurement and the evaluation. The master-slave BMS is a particular structure of a modular BMS. In the modular BMS, each individual module could be the master, whereas in the master-slave BMS, the masters only responsibility is the evaluation, while the slave modules are solely responsible for the acquisition of measurement data [And10; Vem19].

  The modular NXP reference design, as described in chapter 2.1.2.3, represents a master slave Battery Management System.

- **Distributed** Battery Management Systems are fundamentally different to the other architectures. In a distributed BMS, each cell has its dedicated cell management board [And10; Vem19].

2.1.2.3 BMS Demonstartors of NXP

Since the research is carried out in cooperation with NXP, the prototype developed in the course of the thesis is to be based on NXP reference designs. Currently there are two NXP reference designs. One based on a centralized BMS architecture, presented in paragraph 2.1.2.3 and one based on a Distributed BMS architecture, which is described in more detail in paragraph 2.1.2.3.

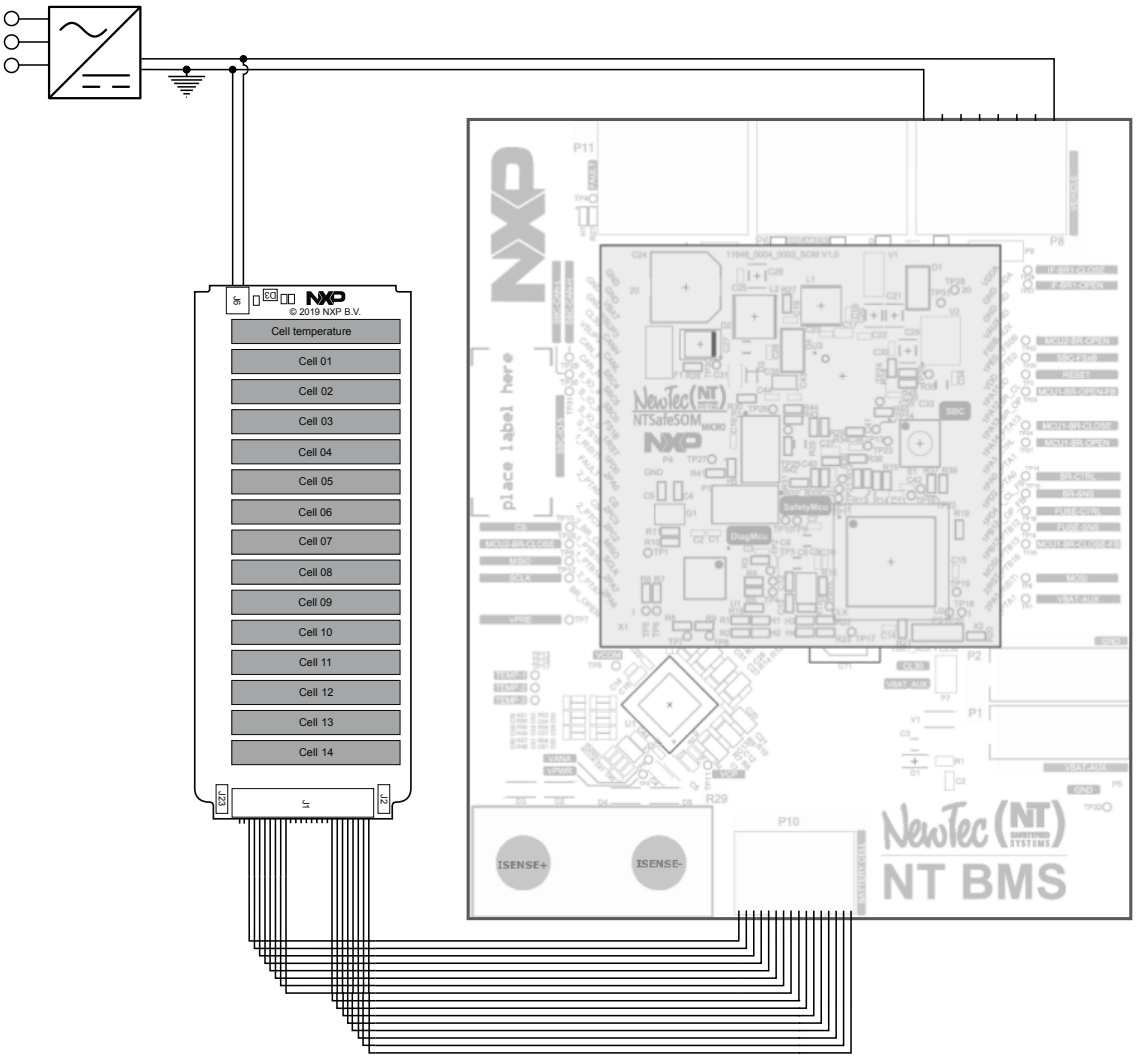


Figure 2.2: The Setup for the NXP reference design for the centralized BMS prototype

Centralized: The setup includes the NewTec board, a battery emulator and a power supply. The battery emulator imitates a battery stack of 14 batteries, while the NewTec Board supports 6 monitored batteries. Therefore the connector from the battery emulator has 26 pins, while the connector on the board has 16 pins. The wiring of the individual pins of these two connectors is depicted in figure 2.2.

Modular (Master Slave): For a more detailed evaluation and a feasibility study, a prototype was built in the course of the thesis. In the proposed prototype, only the temperature data should be transmitted via NFC, since the example implementation of the BCC is expecting only temperature values from the sensors too. Pressure sensors and other measurement components can be integrated into this setup via extensions in subsequent setups.

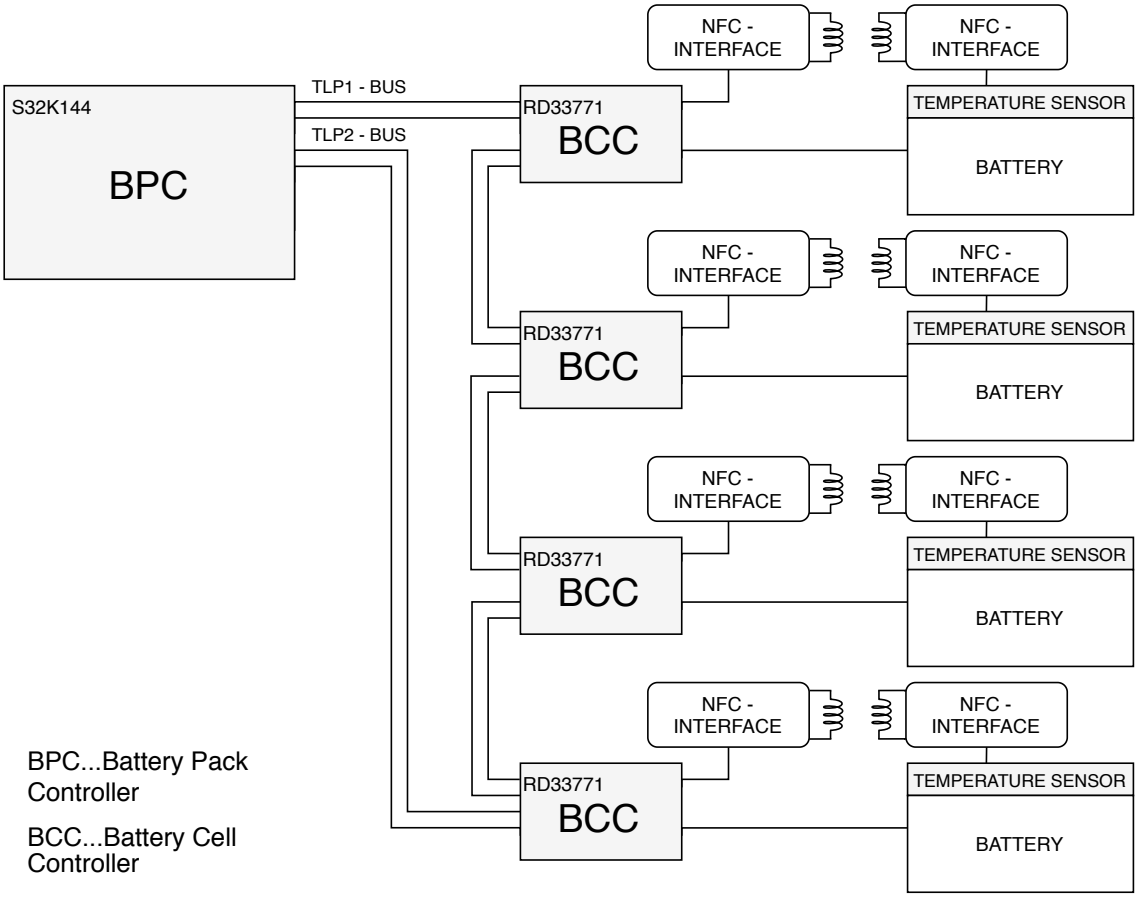


Figure 2.3: The Setup for the NXP reference design for the distributed BMS prototype

In the given reference implementation of the BMS, the interface between the Battery Management Unit and the battery emulator contains four connections on which temperature values are transmitted via analogue values. As shown in figure 3.11, these can be substituted by an NFC interface for the prototype developed in the course of this thesis. On the active side of the NFC connection a S32 microcontroller with NFC front end is proposed, on the passive side a NFC temperature sensor.

2.1.2.4 Wireless vs. wired battery management systems

With the increasing spread of electric vehicles and the associated large accumulators, the modular and distributed architectures of the Battery Management Systems are also being produced in ever greater numbers. The additional expensive cabling in these architectures, which is also prone to errors, is becoming increasingly important. Wireless Battery Management Systems try to replace these cabled connections with wireless connections. Various connections such as Bluetooth [De +15; She+15] or Zigbee [RRR17] have already been tested for the communication of the individual Battery Cell Controllers and their usability evaluated. The basic advantages and disadvantages of such systems are illustrated in table 2.1.

Considerations	Wired BMS	Wireless BMS
Weight	Increases weight	Saves wire weight
Measurement	Time synchronization must be propagated manually	Measurements are time synced automatically by wireless protocol
Reliability	Communication error under EMI Physical connection failure under vibratory environment	No wires to maintain Reduced number of failure points
Security	Enclosed and therefore secure	Faulty implementation can cause security flaws
Costs	Difficult in automated manufactoring	Reduced costs due to reduced number of components

Table 2.1: Considerations wired vs. wireless BMS [Lee+13; Vog20; She+15]

2.1.3 Cell characteristics

There are various indicators which describe the condition of a cell and provide information about the further usability and operational safety of this cell. These codes will be explained in more detail in the following paragraphs.

2.1.3.1 State of Charge

The State Of Charge (SOC) describes the currently available capacity of the battery in relation to the total capacity of the battery. It is usually expressed in percent [Pat+08].

$$SOC = \frac{Capacity - DOD}{DOD} \cdot 100 \tag{2.1}$$

SOC ..... State Of Charge [%]
$Capacity$ .. Estimated capacity $[AH]$
DOD ..... Removed capacity $[AH]$

2.1.3.2 Depth of Discharge

The Depth Of Discharge (DOD) is an indicator that describes how much charge has already been removed from the battery. It is usually expressed in ampere-hours. The

DOD is especially useful when special states of charge of the battery are to be described. For example, if the state of charge of the battery exceeds 100% the SOC would stop at 100%, but the DOD can overwrite this particular state of charge of the battery. Also, in the case of a deep discharge of the battery, the SOC would get stuck at 0% state of charge, the DOD can provide a more accurate description about the current state of the cells. While the DOD has the fully charged battery as the only reference, the SOC refers to two references, the fully charged battery, as well as the fully drained battery [And10].

2.1.3.3 State of Health

The State Of Health (SOH) gives an indication of the general indication of the battery condition in relation to a fresh battery with new cells. [Law] The SOH describes the ability of the cells to store energy, to deliver or absorb high currents and to maintain the state of charge over longer periods of time [Bha+05]. It is a very unspecific indicator, the meaning of which is specified by each BMS manufacturer in another way [And10]. It is an exact designation for witty synonyms such as: fresh, aged, old or worn out [MR05]. SOH is a mixture of the energy the battery can still deliver and the remaining capacity [Pat+08]. The SOH can be determined in two ways:

- Log Book Function [Law]
- Measurement of power fade and capacity fade [Pat+08]

The SOH metric has no meaning inside a BMS. It is useful to provide the external user with information about whether the battery is still usable for the purpose, or whether it needs to be replaced [And10].
Some metrics that often go into the SOH variable are:

- increase in cell resistance [And10]
- decrease in cell capacity [And10]
- number of charging cycles [And10]
- ability to accept charge [And10]
- inner resistance of the cells [MR05]

2.1.3.4 Remaining Useful Life

The Remaining Useful Life (RUL) provides information about how many charge cycles the battery can still be used for. It can be determined from the current internal resistance of the cells and their storage capacity [Pat+08].

2.1.3.5 Thermal model

One of the responsibilities of a Battery Management System is to observe and control the temperature of the battery cells and maintain it within a range that would correspond to the Safe Operating Area of the cells. The electrical behavior of the cells is often described by a battery model, which is addressed in more detail in paragraph 2.1.1.5. There are also ways to model the transient thermal behavior of the cells. This can be used to improve

the cooling strategies of the batteries, because a strong limiting factor of the cell lifetime is the temperature that prevails during its operation.

In [PJ03] the temperatures next to the cell are approximated based on the measured current using a thermal model. A discrepancy of $3°C$ between the estimated values and the values physically measured in the cell vicinity is reported by the authors of this paper. [KYK13] introduces a method for a plausibility check in which the temperature at the sensors is approximated via a neural network. In paper [TMT11] a thermal electric battery model for battery simulations is described. The works [VLA01] and [V02] are comparing the common battery models used in thermal and electrical simulation tools.

There are ways to describe the thermodynamic model of a battery with knowledge from the electrical engineering domain. Thermodynamic quantities are mapped to quantities in the electrotechnical domain, and can then be inserted into the corresponding electrotechnical equations and calculated. Table 5.1, taken from [NXP08], shows this mapping of variables between the two different domains.

electrical domain			**thermal domain**		
variable	symbol	unit	variable	symbol	unit
Current	I	$[A]$	Heat Flux	P_D	Joules/s
Voltage	V	$[V]$	Temperature	T	$°C$
Electrical Resistance	R	$[\Omega]$	Thermal Resistance	$R_{\Theta AB}$	$°C/W$
Electrical Capacitance	C	$[F]$	Thermal Capacitance	C_Θ	$Joules/°C$

Table 2.2: Relationships between thermal and electrical domain [NXP08]

Thermal transport mechanisms: There are 3 different mechanisms how heat can be transported. These are described in more detail in the following chapters. The temperature model presented in appendix A, to which the background knowledge compiled in this chapter refers, was designed to be as simple as possible. Therefore, only the conduction was considered as a temperature transport mechanism in this temperature model. In the reference model generated with the help of Matlab, with which is compared in this chapter, all 3 known temperature transport mechanisms, the conduction, the convection and the radiation are considered [NXP08].

- **Conduction** is the simplest of the 3 possible heat transport mechanisms. It is described by the thermal resistance, and can be modeled in the simple thermal equivalent circuit. The relation for the conversion is shown in table 5.1 [NXP08].
- **Convection** plays a role if the temperature transport happens actively through moving air (fan) or through a moving liquid. For passively cooled modules, convection as a heat transport mechanism can be neglected. The convection cannot be modeled via a thermal equivalent circuit [NXP08].
- **Radiation** is a very complex heat transport mechanism and cannot be described by the thermal equivalent circuit. However, radiation only plays a role at large temperature differences and large surface areas [NXP08].

2.2 Security and Safety Aspects in BMS

Security and safety often strongly interact with each other in the automotive sector. A vulnerability in the security can quickly have an impact on the safety of the overall system. In this chapter, the smart batteries 2.2.1 section deals with the communication buses at the interface of a BMS to the outside world. This interface is available from the environment and therefore security considerations must be considered. The safety chapter 2.2.2 discusses safety precautions commonly used in battery packs. The Security chapter summarizes several papers that address possible attack vectors and countermeasures in modern BMS.

2.2.1 Smart Batteries

If the BMS contains an interface via which external communication with the cells can take place, it is a smart battery.

Smart batteries contain an interface that can be used to communicate with the cells. The cells of a smart battery log the environmental conditions, the charge cycles, the temperature profile of the cell, the internal cell impedance, errors that occurred during operation, and circumstances in which the limits of the Safe Operating Area were exceeded [Law; Ber01].

One area in which smart batteries are already established and widely used is in notebook computers. A Smart Battery System (SBS) is used in mobile devices and portable computers unifies this interface and makes the use of battery packs with different chargers possible. Since the use of smart batteries in the automotive field is equally desirable, it would be conceivable to introduce a similar system for smart batteries and their interfaces in the automotive field. The individual elements of the specification already in use in the notebook sector are explained in the thesis [Ber01] and are summarized in the following enumeration:

- System Management Bus (SMB)
- Smart Battery Data (SBD)
- Smart Battery Charger (SBC)
- Specification between SMB and Basic Input Output System (BIOS)

2.2.2 Safety

Lithium ion batteries can become very dangerous if they get outside the Safe Operating Area (SOA) during operation [And10]. Misuse can result in battery failure, explosion, or toxic gas emission. Therefore, it is important to closely monitor the operating conditions of the battery. The most dangerous condition when operating a battery is the thermal runaway. This occurs when the temperature exceeds the maximum allowable temperature range for operation, resulting in chemical reactions that cause the temperature range of the battery to increase even further. The result is a positive feedback loop that is difficult to break from the outside and therefore difficult to control. In charged lithium ion batteries, a lot of energy can be stored, very densely packed. To quickly remove this energy from

the battery in such an exceptional state is difficult if not impossible. Since the battery system in an electric vehicle is an essential component, its failure would severely limit the functionality of the vehicle. A BMS operated in the automotive context should ideally support Automotive Safety Integrity Level D according to ISO 26262 [Mar14; Law].

The following are some safety measures used in battery cells to prevent dangerous cell operating conditions according to [Law]:

- **Cell chemistry** is more reactive in energy dense cell designs.
- **Electrolytes** can contain chemical inhibitors with a self-extinguishing function.
- **Cell Construction** which takes into account the thermal model in the cell design to quickly dissipate temperature away from the cell during operation.
- **Durable Separators** because standard plastic separators can melt.
  - **Rigid separators** which do not deform even under extreme temperatures.
  - **Ceramic coated separators** which have a special coating that stays more rigid at high temperatures.
  - **Shut down separators** whose resistance increases as the temperature rises, cutting off the current flowing through the battery.
- **Robust Packaging** and resistant cell housings are preventing the cells from deforming and thus the separators from getting damaged or loose.
- **Circuit Interrupt Device (CID)** is a built-in fuse that interrupts the current flow when the pressure inside the cell becomes too high.
- **Safety vents** are opening automatically at high pressure allowing the pressure that has built up inside the cell to escape into the environment.
- **Keyed and shrouded terminals** which are preventing the battery from being incorrectly connected to the system or the user from getting into the electrical circuit during connection.

2.2.2.1 Safe Operating Area

The Safe Operating Area (SOA) specifies that operating range of the cells in which a safe operation of the battery cells is guaranteed. The SOA specifies the maximum current, the allowed temperature and the allowed cell voltage [And10].

2.2.2.2 Operating Temperature

If the temperature of the cells is too high, it can lead to a thermal runaway, an undesirable operating state of the cells. If the battery cells are operated at a too low a temperature, they can also be damaged permanently. The temperature range in which battery cells may be charged is even narrower than that in which they may be discharged [And10].

2.2.2.3 Operating Current

Excessive charging currents can lead to overheating of the cells. Also too high discharge currents can bring the cells into a dangerous temperature range and damage them permanently. Also high pulse currents, which flow through the battery for more than a few seconds, should be avoided if possible to prevent damaging of cells [And10].

2.2.2.4 Operating Voltage

If battery cells are overcharged, they are permanently damaged or even start to burn. If they are discharged to a cell voltage that is too low, they can be damaged too [And10].

2.2.2.5 Authentication and identification

All the safety measures discussed in chapter 2.2.2 might be useless, if the battery is not manufactured carefully. A small metal particle inside the electrolyte can cause a sudden short circuit due to vibration during operation. Inadequately secured separators can slip and cause short circuits. These are just a few examples of a safety critical condition that can be caused by operating a carelessly produced cell. Therefore it is important to control where the cells, which are for example used to repair the battery pack are coming from. This safety precaution is considered in the prototype developed in this thesis too [Law].

Here, the individual cells are to be authenticated in order to prevent cheap replicas from being used for repairs that have not been checked and approved by the manufacturer. If unknown battery cells are used as spare parts, this can have devastating effects on the behavior in the vehicle and thus on the resulting safety, especially in the automotive sector. If batteries from certain manufacturers break down or even catch fire, this can have devastating consequences for the brand's reputation. Therefore, it is important to make sure that only authenticated battery cells can be taken as spare parts [Law].

For authentication, some feature like a unique code is embedded somewhere in the battery. This could be a digital signature, ensuring that the cell comes from an authorized manufacturer. If the verification of this unique code fails, the battery can be electronically prevented from being used. This can prevent the battery from being operated in an undefined state through the use of unauthorized spare parts [Law].

In the following, a few common ways to prevent counterfeiting will be listed and their applicability in the automotive domain will be evaluated.

- **Radio Frequency Identification (RFID)** can be used in the automotive field for tracking and anti counterfeiting of battery cells in the future [Law].
- **Holograms** can detect counterfeiting, but not automatically prevent it and are therefore unsuitable for use in the automotive sector, where the battery cells are usually not visibly installed in the vehicle [Law].
- **Barcodes** are cheap and can be used for tracking, but cannot be automated [Law].
- **Labels** are cheap to implement and can be used for tracking battery cells, but not for anti-counterfeiting [Law].

2.2.3 Security

In Table 2.3 an overview of the current research status regarding automotive security and security in the context of BMSs is given. It lists the existing papers that are dealing with the topics of security in BMS, automotive security and wireless solutions in BMS.

Paper	**Similarities**	**Advancements**
Smart Cells for Embedded Battery Management [FW09]	Similar architecture of used CMU	Proposing a completely decentralized structure
Cybersecurity for BMS in Cyber-Physical Environments[Stu+18]	Security evaluation for common threats regarding BMS	Focusing on the interface between BPC and cloud
Cybersecurity Issues in electric Vehicle[Kha+19a] BMS	Summarizing threats and attack trees	Paper is focusing on a standard BMS architecture
Wireless BMS for Electric Vehicles [MPJ12]		Compares different wireless protocols for usage in BMS
Wireless Battery Management System in Electric Transport [RRR17]	Measuring the temperature via a digital sensor	ZigBee Sensor Network between BCC and BPC
Wireless Battery Management System [Lee+13]	Attempt to avoid the sensor wire-harness	Using a wireless connection between BCC and BPC
Security Challenges in Automotive Hardware Software Architecture Design [Sag+13]	Evaluating attacks that are imposing a safety threat to the battery system	Outlining the security threats of a broad range of wireless connections in the automotive domain
Volpe Technical Meeting on Electric Vehicle and Charging Station Cybersecurity Report [Har+18]	Identifying security gaps in modern electric vehicles	Identifying security hazards in a broader domain in a more generic design
An RFID Authentication and Anti-counterfeit Transaction Protocol [Che+12]	Proposing an anti counterfeiting solution for RFID transactions	Signed number is randomly generated and renewed on every transaction
Automotive Battery Monitoring by Wireless Cell Sensors [Sch+12]	The usage of wireless cell sensors	Utilizing the wireless connection for galvanic isolation
Towards a smarter BMS: A critical review on battery T state of health monitoring methods [Ali+19]	Determining the current SOH of the battery	Using a battery model for SOH determination instead or a cell history readout
Cyber-Physical-Security Framework for Building Energy Management System [Par+16]	Determining security threats in Energy Management Systems (EMS)	Security threat in the domain of EMS and BMS in buildings is determined

Potential Cybersecurity Issues of Fast Charging Stations with Quantitative Severity Analysis [POO19]	Evaluating Security threats in BMS	Charging port offering an additional attack vector is evaluated in the analysed system
Cybersecurity of Battery Management System [Mad19]	Analyzing threats and attack vectors in BMS	Analyzed space includes the wider in vehicular system
Design of CAN to Bluetooth gateway for a BMS [De +15]	Evaluating the interfaces commonly exposed by a BMS	Readout of the BMS via a can- Bluetooth gateway
FACTS Approach to Address Cybersecurity Issues in Electric Vehicle Battery Systems [Kha+19b]	Determining threats regarding the sensor data acquisition and transmission over BMSdata lines	Battery swapping functionality adding an additional attack vector when evaluating security threats
Hardware-Based Anti-Counterfeiting Techniques for Safeguarding Supply Chain Integrity [ASS17]	Anti counterfeiting via signing an unique UID	Additional methods other than tagging with a UID [1]
Implementation of a Wireless Battery Management System (WBMS) [She+15]	Utilization of wireless transmission technologies in BMS to reduce wire harnesses and failure points	Proposing a Bluetooth connection between BCC and BPC
Lightweight Authentication Protocol for NFC Based Anti-Counterfeiting System in IoT Infrastructure [AMK20]	Anti counterfeiting based on unique codes transmitted via NFC	Using RFID mutual authentication
Automotive Security: From Standards to Implementation [Soj14]	Utilisation of AES-128 crypto- algorithms in the automotive context	Security measures in a broad automotive domain
Wireless sensor/actuator device configuration by NFC with secure key exchange [Kla+17]	Connection between an embedded system and an external readout via NFC	Tracking of the embedded system via UID and security measures for the RFID interface
Secure and Authentic Communication on Existing In-Vehicle Networks [GR09]	Evaluating security measures for the interface of a single ECU to the in- vehicle network	Focusing on the broader in-vehicle network
Smart Cells for Embedded Battery Management [Ste+14]	Utilizing cell boards for the management of individual cells	Proposing a distributed BMS architecture

Table 2.3: Similarities and differentiation of this thesis in relation to already existing literature

[1] Watermarking, Fingerprinting, Circuit Obfuscation, Parametric and Functional Tests, Hardware Metering, Physically Unclonable Functions (PUF), Aging Models and Sensors

Chapter 3

System Design

In this chapter, the different demands on the design are defined, and different possibilities of realising this design are evaluated. Each of these possibilities has advantages and disadvantages, and it is important to weigh these benefits and drawbacks against each other in order to achieve the best possible solution that takes into consideration all the demands placed on the design.

3.1 Requirements

The setup has as its requirements, among other things, the standard demands that are placed on every BMS. These requirements include cell balancing, monitoring the cell voltage, and safety measures such as monitoring cell temperature that a BMS has to provide. To further advance the design, some additional requirements, which are exceptional for a BMS, have been included in the following list:

- **Anti counterfeiting** for the battery modules.
- **Storage capability** of the cell module without draining the cells (Low power State).
- **Contactless readout** of the error logs occurred during the operation.
- **Wireless interface** between the thermal sensors and the cell boards to lower production costs during assembly.

3.2 Overall design

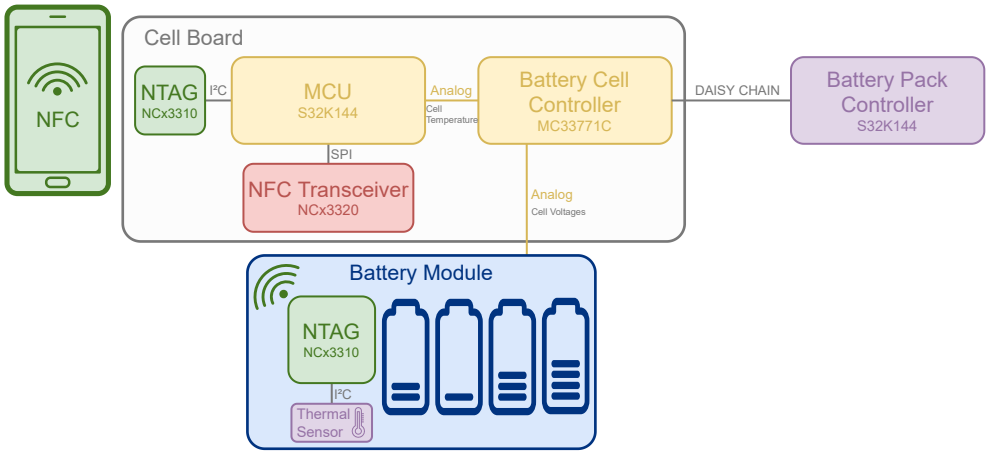


Figure 3.1: Block diagram of the overall design

Since lithium cells are quite expensive, and batteries are in many cases still usable after the life cycle of the vehicle, two cycles of use are generally envisioned for the batteries. The first utilization cycle takes place after manufacturing and represents the use for powering the electric vehicle. In the following, this usage cycle of the battery is referred to as on vehicle use.

The batteries are sometimes the most expensive components of electric cars and therefore cannot simply be disposed at the end of the car's life cycle. After this usage cycle, the individual battery modules can be recycled for energy storage, for example as house batteries. To determine which battery cells are still usable after the use in the vehicle, a contactless readout of the fault memory in the battery module should be implemented. If the battery was already error-prone during the usage in the vehicle and already has too many errors in the error memory, it has to be disposed. Otherwise, it can be used for its second cycle of use away from the vehicle, which will be referenced as off vehicle use in the following. This metric, which is an indication of the battery's remaining usability, is known in the popular literature as SOH. It can be estimated by measuring cell specific values such as capacitance and internal cell resistance, or by using a logbook function as discussed in more detail in paragraph 2.1.3.3. In this design, the logbook function is used to derive the SOH. Faults are logged as the battery is used in the vehicle and can be read out afterwards using the NFC interface presented in the design.

The requirements placed on the design can be fundamentally divided into 3 sub-areas:

- **The battery management functionality** is the battery management functionality with its balancing and monitoring functions required for safe operation.
- **On Vehicle Features** which are anti-counterfeit and contactless sensor reading.
- **Off Vehicle Features** which contain the contactless error log readout and the low power functionality.

3.3 Battery management (On-vehicle)

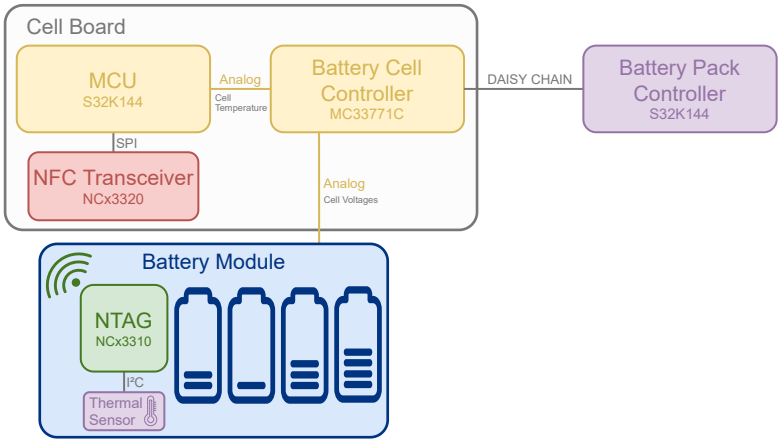


Figure 3.2: Block diagram of the Battery Management System design

The architecture of the module, which has to fulfil all the requirements listed in paragraph 3.1, is shown in the block diagram in figure 3.2.

The required battery management features are the standard ones found in similar setups. In order to comply to these requirements, the components of the current NXP prototype for a distributed BMS are used. Those components are used to built upon in the subsequent thesis.

The components the prototype is composed of are described in the following. One component of the setup is the Battery Cell Controller (MC33771C). It accomplishes the task of cell balancing and the determination of the voltage level, i.e. the state of charge of the battery cells. The chip also takes over the transmission of the evaluated values via the daisy chain. The TLP high-speed network, which forms the daisy chain, is electrically isolated and connected to the Battery Cell Controller chip. This potential separation has been omitted in the block diagram 3.2 for the sake of simplicity. The Battery Cell Controller is also responsible for transmitting the measured cell temperature to monitor the operating state of the battery. The chip used offers an analogue interface for this task, which is intended for the connection of an analogue temperature sensor.

This analogue interface is used by the prototype to feed the temperature values into the NXP reference design. One of the design requirements was that the temperature sensors should be wirelessly connected to the circuit board that houses the Battery Cell Controller. For this purpose, the NFC transceiver NCx3320 was connected to the microcontroller via SPI, which is connected to an NFC smart sensor from which the temperature values can be obtained.

The smart sensor consists of the NCx3310 chip, which offers a special mode that allows sensors to be used via the I^2C bus without the use of a microcontroller. This setup is powered by the NFC field of the NFC transceiver via energy harvesting.

The requirement of anti-counterfeiting can be solved via the NTAG too. The NCx3310 offers a freely configurable originality signature. Each NCx3310 chip is delivered from the factory with a fixed 8 byte long UID. With the originality signature, this UID is signed using the secp128r1 protocol. This signature can then be stored in a designated memory location on the chip and verified during use. If the verification is successful, it is certain, that the chip whose signature was verified comes from the corresponding manufacturer who created the signature. This ensures that the NFC temperature sensor read by the microcontroller comes from the specified manufacturer. If the signature verification fails, the microcontroller can switch off the battery, for example by reporting a temperature in an unacceptable range to the Battery Pack Controller.

3.3.1 Attack vectors and security analysis

The threat analysis was started by presenting the system design with the assets to be protected and possible threats in a Data Flow Diagram (DFD). There is no single and correct way to protect a system against all possible attacks. One method, and part of any security concept, is threat modeling. In this system, threat modeling, based on the STRIDE model presented by Microsoft [Cor], will be used to analyze the possible attack surfaces of this system. In the DFD, the data flow is graphically represented with the assets to be protected, the possible threat actors and the security controls. The red dashed boxes are representing trust boundaries, gateways that must first be overcome by a possible attacker before components inside this subsystem can be accessed [Her+07].

Regarding the security of the NFC sensor interface, it would be important to find out whether it can be interacted with the NFC interface at all from outside the vehicle. The following security analysis is made supposing that the NFC connection of the setup is exposed to the environment an provides an attack vector exposing values and providing access to the inside of the BMS

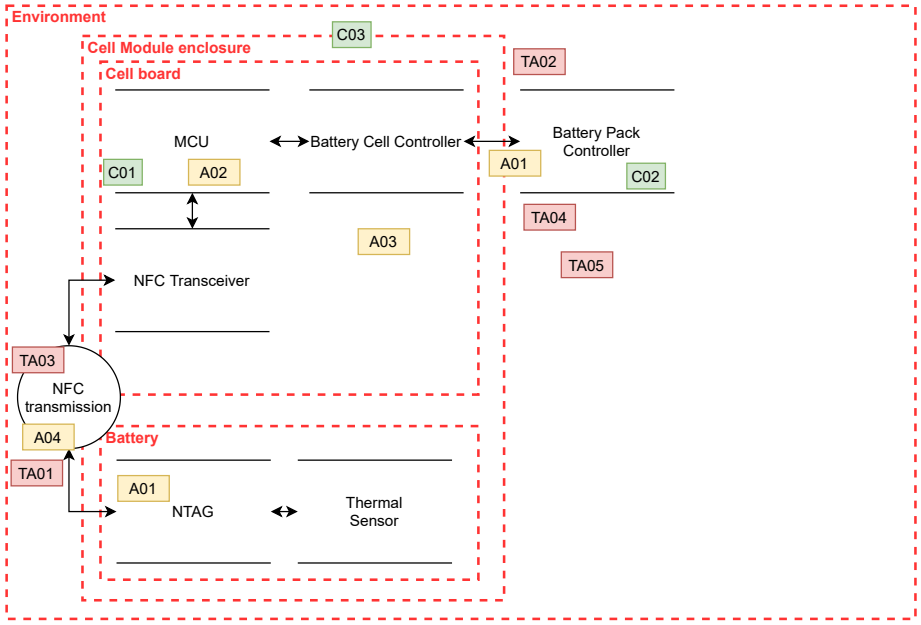
Using this assumed setup, a security analysis as described in the EVITA paper [Rud+09] was started. In the following, security analyses from the software domain that are not focused on the automotive sector, such as the STRIDE or the DREAD [Mic02] analysis, could be continued. This would help to assess the individual risks and relevance of the different attack vectors and to be able to take the necessary countermeasures.

The target of the attack can obviously be bringing the system into a safety critical state (e.g. by reporting a too low cell temperature) or to prevent the safety measures taken by the BMS. Other objectives would be to inhibit the start-up, which would damage the reputation of the manufacturer. The driver's privacy could also be compromised by a wireless BMS, as tracking of the vehicle could be facilitated [A+]. Also the attack scenario described in [A+] whereby the driver is made to stop by a fake error message of the tire pressure sensor and is then robbed is relevant for a wireless implementation of the BMS.

Proceeding into this area would include the assessment of the possible security concepts for the communication channel. Furthermore, possibilities would have to be found for adapting the firmware of the BMS to changes or advances in the security sector during its service-life. This might include updating the software on the wireless transmitter and the wireless receiver. If this is infeasible, it should be investigated how much of the battery

monitoring would be required with a wired connection in order to maintain the safety requirements even if the wireless connection fails or is manipulated.

Another option would be to detect the manipulation of the wireless communication and switch to a fail safe mode. The detection could be done by redundant wired transmission of some sensors. The last possibility, which has been considered so far, is to have all safety relevant functions of the BMS, such as opening the breaker directly at the battery management component. The information transmitted via the wireless channel would then only be used for diagnostic functions, which would make the consequences of an attack, possible by an outdated security protocol less severe. However, security-critical manipulations would still be feasible with this solution. For example, the demand management of the battery could be influenced such that the battery is overcharged by regenerative braking.



Threat Actors	
ID	Description
TA01	Compromise drivers privacy
TA02	Tamper with dashboard warning message
TA03	Boost vehicle battery
TA04	Prevent vehicle from starting
TA05	Pretend safety critical state while driving

Security Controls	
ID	Description
C01	signature validation
C02	plausibility check of sensor values
C03	Sealed cell module enclosure

Assets	
ID	Description
A01	Sensor Values
A02	Software integrity
A03	Hardware integrity
A04	Vehicle identifiability

Figure 3.3: Data Flow Diagram of the Battery Management System setup with the exposed NFC Interface

In order to use STRIDE, the data, subsystems, and attacks transmitted in the system are presented in a Data Flow Diagram representation (3.3). The Data Flow Diagram is used to break down the system into subsystems. Then each of these subsystems is checked for

potential threats and protected against them. Security controls are then defined to protect the assets from the threat actors. This process is repeated until all assets are sufficiently protected from threats [Her+07].

In the Data Flow Diagram in Figure 3.3, the cell module enclosure is drawn as a trust boundary. This cell module enclosure is a mechanically sealed component. For further threat analysis it is assumed, that the only ways to get in and out of this cell module enclosure are the data flows drawn in DFD. Thus, the cell module enclosure is the first trust boundary, which is propagated in 3.3 as security control C03.

For securing the NFC connection, a signature validation is introduced as a security control. This allows the Cell Board to ensure that the NFC connection has been made to the correct NTAG and thus to the correct battery. This can ensure that the sensor data transmitted over the NFC interface is from a manufacturer verified sensor and cannot be easily tampered with. Another danger with the NFC interface is Vehicle Identifiability. Each NTAG has a UID which is also used for authentication. Via this UID the vehicle could be uniquely identified and thus tracked, if it can be read from a distance that is large enough. If this UID can be received from the roadside, the vehicle could be tracked via stations permanently installed at the roadside. Such a scenario is discussed in more detail in [A+]. However, the NFC interface has a very limited range. Especially in an environment inside a vehicle where metal is predominant as a building material, one has great problems receiving anything via the NFC interface at all from outside the vehicle.

Another interface shown in the Data Flow Diagram 3.3 is the data interface between Battery Cell Controller and Battery Pack Controller. This represents another interface through which threat actors can interact with the system. The assets to be protected, which are transmitted through this interface, are the sensor data (A01). This sensor data is the measured cell temperatures and the voltages of the individual cells. To protect this interface, a plausibility check (C02) is proposed. The plausibility check would pass through a second data transmission channel, the current flowing through all the battery cells connected in series is measured directly at the Battery Pack Controller by a shunt resistor or a hall sensor. The course of the individual cell voltages can be verified by a battery model in the Battery Pack Controller. For example, coulomb counting can be used to compare the cell voltage transmitted to the Battery Pack Controller with the current measured at the Battery Pack Controller [Vel+17]. If the values differ too much, the plausibility check fails, the Battery Pack Controller, which controls the Battery Management System algorithms, can react accordingly. A similar plausibility check can also be performed for the temperature values measured and transmitted to the Battery Pack Controller. The current flowing through the battery cells correlates with the cell temperatures. The approximate cell temperature can be inferred from the current measured at the Battery Pack Controller via a temperature model. The cell temperature transmitted to the Battery Pack Controller can thus be checked for plausibility. A possible implementation of this plausibility check is explained in more detail in paragraph 3.3.1.2. A simple temperature model in which the differential equations used for transient thermal description has been extrapolated down to simple relationships is presented in appendix A. The equations in this model are simple enough that it can be computed on a MCU such as the S32K144 microcontroller. Thus, it can be used for a possible plausibility check.

3.3.1.1 Threat Actors

For the operation of the Battery Management System, several threat actors were identified, which are visualized via error trees and will be discussed in more detail below. In order to obtain a picture as comprehensive as possible of the possible sources of attack with the threat analysis, the 6 main threats of the STRIDE model were assumed. For each of these main risk factors, fault trees were created which are breaking down the attack surface into sub-attack threats and can be applied to the specific model shown in Figure 3.3. The 6 threat categories identified by STRIDE and elaborated more closely in the subsequent paragraphs are spoofing authenticity, tampering with the integrity, repudiation of actions, denial of service and elevation of privilege [Cor].

Spoofing identify: A potential threat when operating the Battery Management System would be to forge the identity. In doing so, the whole BMS could be replaced with an BMS from another manufacturer. This would be very costly, but a possible option to tune the vehicle and extract more power from the battery cells. Such a tuning attempt can be prevented by the BMS authenticating itself to the gateway installed in the vehicle.

The other option to forge the authenticity would be to use cells from cheap third party suppliers for a repair, which may not have the quality and production grade required by the manufacturer.

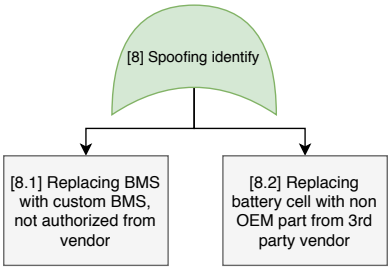


Figure 3.4: Fault tree - Spoofing identify

Tampering with data: As discussed in Paper [A+], it is a serious potential security risk if the dashboard alerts can be manipulated by an intruder. Such a notification might cause the driver to stop and subsequently be robbed. It would also be possible to misuse such a security vulnerability to undermine the trust in the manufacturer by continuously reporting false error messages to the driver.

On the contrary, necessary alerts could also be suppressed. This might lead to the driver being deprived of crucial information about the condition of the battery. As a result, the driver would react too late to the changed vehicle condition, which could result in a hazardous event.

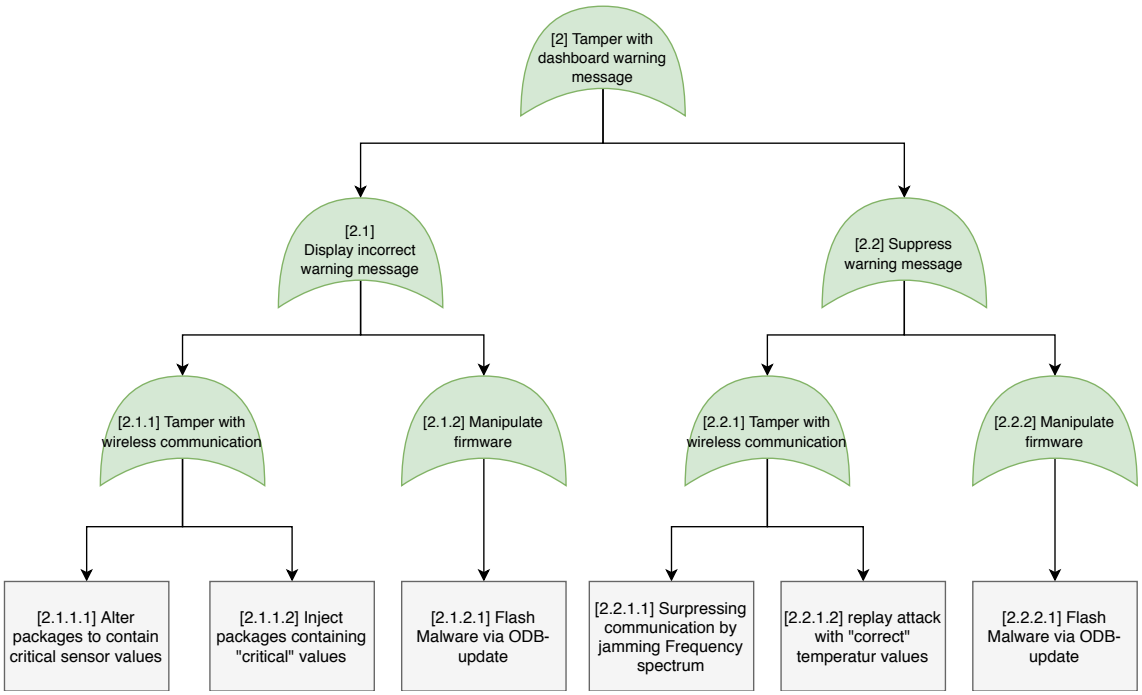


Figure 3.5: Fault tree - Tamper with dashboard warning message

A possible target of an attack would be to simulate a security-critical state while driving. Since in the case of an electric vehicle, the traction battery is an essential part of the drive train, its removal would have a devastating effect on the vehicle. Since it is not possible to shut down the battery entirely in certain driving situations, the battery would have to be set to an emergency program. In any case, attack vectors leading to the mentioned target should be detected and closed.

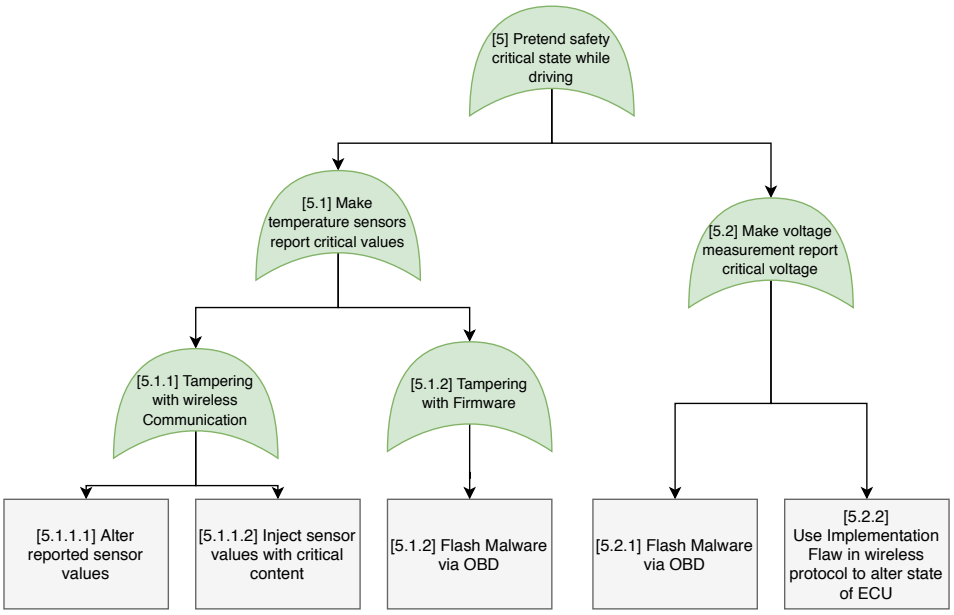


Figure 3.6: Fault tree - Pretend safety critical state while driving

Repudiation: A general problem in securing systems against security threats, especially in the automotive domain, is deniability. If the Battery Management System is manipulated in any way, it should be obvious afterwards and not easily deniable.

Ways to eliminate the traces after manipulation of the BMS would be to delete the non-volatile memory.

Another danger would be to open the case of the BMS without being able to detect it. In the BMS presented here, the sealed enclosure is an essential part of the security concept. If the housing can be opened without it being traceable, individual hardware modules can be replaced at any time, and the BMS can be manipulated in a variety of ways.

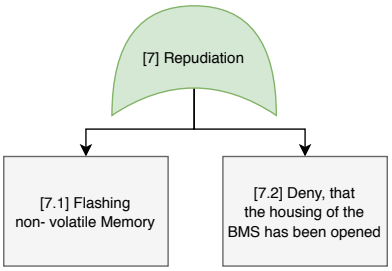


Figure 3.7: Fault tree - Repudiation

Information disclosure: Emissions from the wireless communication of a wireless BMS can result in the vehicle being identifiable. This identifiability can be exploited for tracking purposes, as discussed in paper [A+]. In order to prevent vehicle tracking, it is important to keep the radio communication observable from the surroundings as generic as possible.

In addition to the location information, the BMScan also leak data about the respective driving style and driving habits.

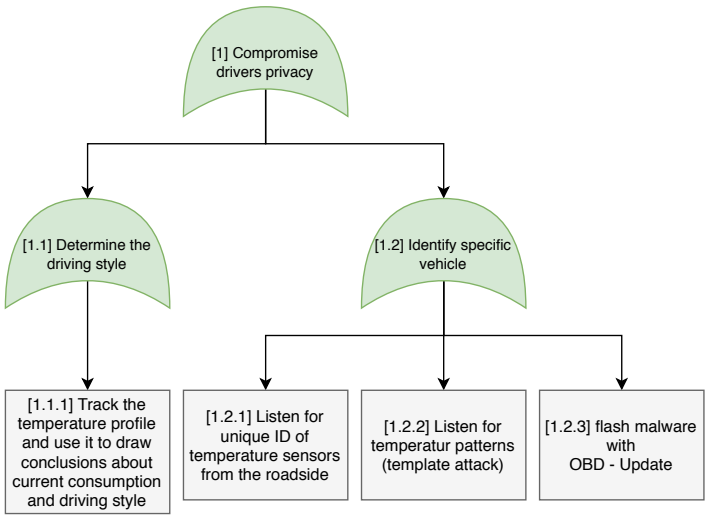


Figure 3.8: Fault tree - Compromise drivers privacy

Denial of service: Security vulnerabilities in the BMS can also be exploited for a denial of service attack. The goal of such an attack is to prevent the vehicle from being started. This decreases the availability of the vehicle, which has a devastating effect on its reliability. The possibility to start at any time, for example to remove the vehicle from a danger zone, is also a part of the safety requirements a modern vehicle has to fulfill.

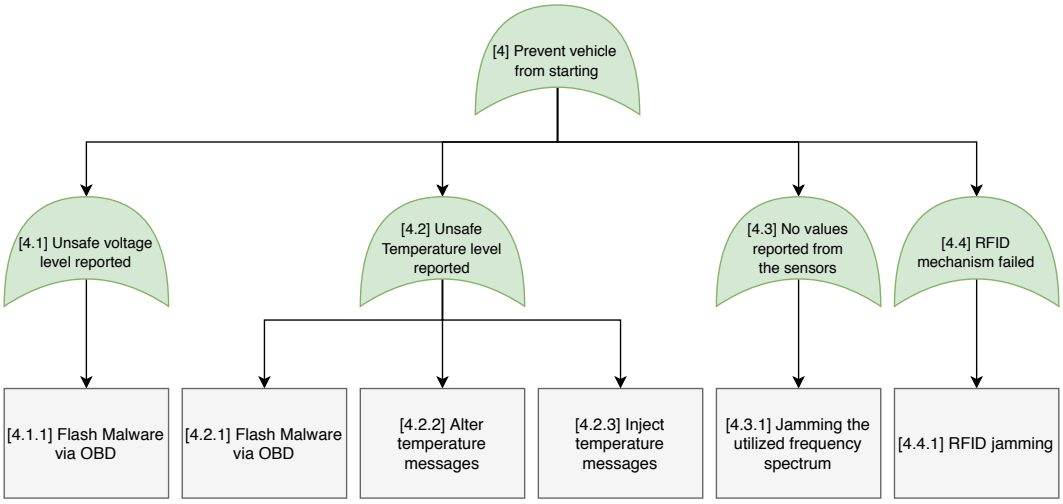


Figure 3.9: Fault tree - Prevent vehicle from starting

A possible target of attack is to deactivate the safety functions that the BMS offers. Although there is usually still a mechanical fallback to prevent a thermal runaway, such as

pressure valves, those safety precautions will most likely destroy the battery. In addition, the reliability decreases due to the reduced safety functionality.

An attack as depicted in figure 3.10 can be carried out either by a malicious attacker, or by a garage trying to fix the car with uncertified spare parts.

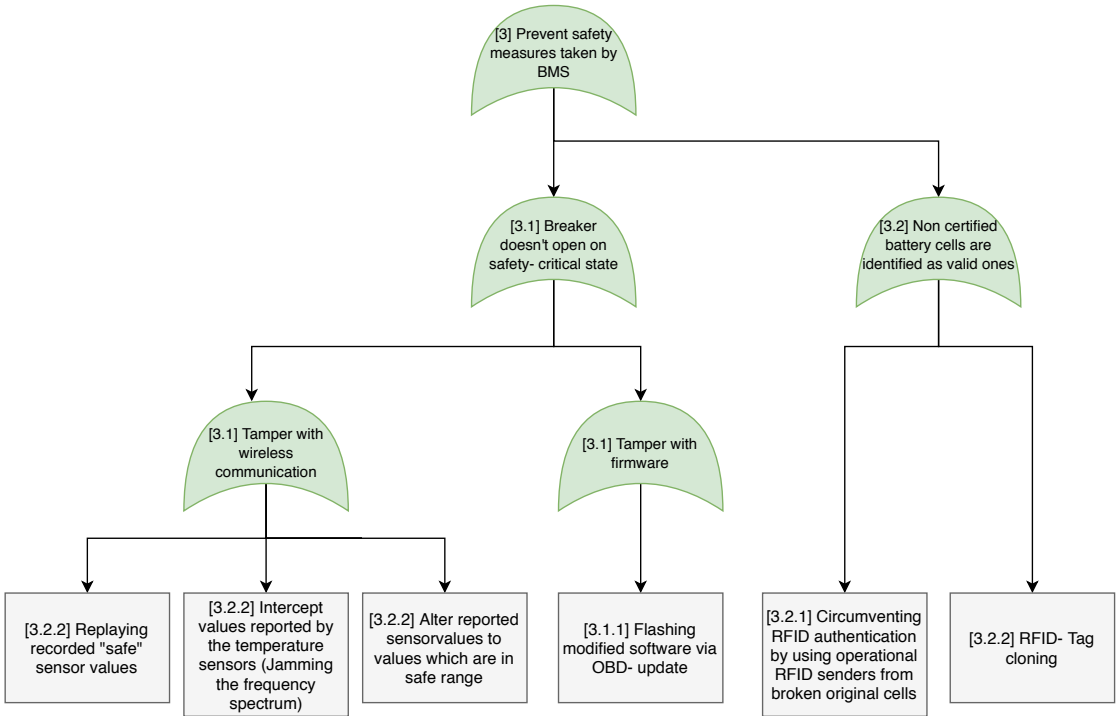


Figure 3.10: Fault tree - Prevent safety measures taken by the BMS

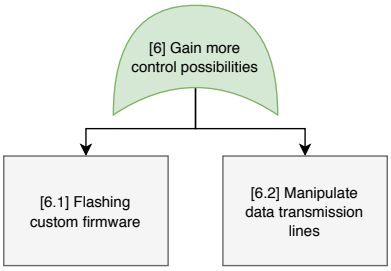


Figure 3.11: Fault tree - Compromise drivers privacy

Elevation of privilege: Security is a race between evolving attacks and protection methods. Any security barrier can be circumvented or broken, as long as enough resources are available. The security measures must be sufficient to convince a potential attacker that an attack will not be worthwhile. Over time, security methods evolve and attacks on existing systems are getting cheaper. To combat this, existing systems that are subject to security requirements must be updated. In the automotive sector, an update function

is particularly important because those products have a much longer service life than conventional consumer electronics [Sim19].

3.3.1.2 Security measures

To counteract the threat actors discussed in chapter 3.3.1.1, some security measures have to be incorporated into the design, which are discussed in more detail below. This security measures, namely the signature validation, described in chapter 3.3.1.2 and the plausibility check, described in chapter 3.3.1.2 are also named as security measures in the Data Flow Diagram depicted in figure 3.3.

Signature validation: To ensure the safe operation of a battery, it is essential that the cells have been produced carefully and under controlled circumstances. For example, if there are metal splinters in the electrolyte, all the safety measures taken by the BMS are useless, because a short circuit can still occur inside the battery, which in the case of a fully charged battery would inevitably lead to a thermal runaway. Therefore, it is essential to authorize the cells monitored by the BMS. If cells are detected which have not been authorized by the manufacturer, the battery should be switched off as a safety precaution. The 2.2.2.5 paragraph discusses this security feature used by BMS in more detail.

Therefore, this design will incorporate a way to authenticate the individual battery blocks. This authentication is used for anti counterfeiting purposes. Each battery block consists of a cell module, in which the battery cells and the temperature sensors are located, and the cell board, which is responsible for evaluating and transmitting the information collected by the battery module. These two components are located in a sealed housing. When individual cells are reaching the end of their life cycle, an entire battery block is replaced at once. The authentication option must therefore be integrated within this battery block. After replacement, the new battery block authenticates itself to the Battery Pack Controller, which resumes operation after successful authentication.

The NCx3310 board used for wireless temperature transmission offers the possibility of creating a digital signature for anti counterfeiting purposes. The advantage here would be that the temperature sensors are closer to the cells. If a battery block is actually opened to exchange only the single cells the authentication would possibly fail. This is especially true if the wireless interface is used to measure not only the temperature but also the pressure inside the cells. In this case, the NFC smart sensor must be implemented inside the battery cell. Otherwise, it would not be the pressure inside the cell that would be measured, but the ambient pressure. An authenticated battery inside the cell cannot be easily exchanged or manipulated by opening the battery block.

Another possibility would be to implement the authentication with a battery on the cell board. Here, even the MCU itself could possibly be used for authentication, if it offers the corresponding functionality. Otherwise, a secure element could be installed on the cell board to provide the possibility of an authentication.

Another option, which results from the special design of this setup, would be to use the NCx3310, which is installed on the cell board for the off-vehicle use case for authentication.

Plausibility check: Over and over again, it happens that users try to manipulate their BMS to boost their battery packs. In this way, the Battery Pack Controller is made to believe that the battery is in the Safe Operating Area while the cells themselves are operated far outside the safe range. Such methods can be used to get more current out of the battery pack for example to accelerate faster. The resulting safety risk is neglected by the user, who is often unaware of the devastating effects that overheated lithium cells can have. If the Battery Pack Controller, on which the battery management algorithms are executed, receives incorrect information about the cell temperature, no suitable measures are taken and a thermal runaway can be the consequence.

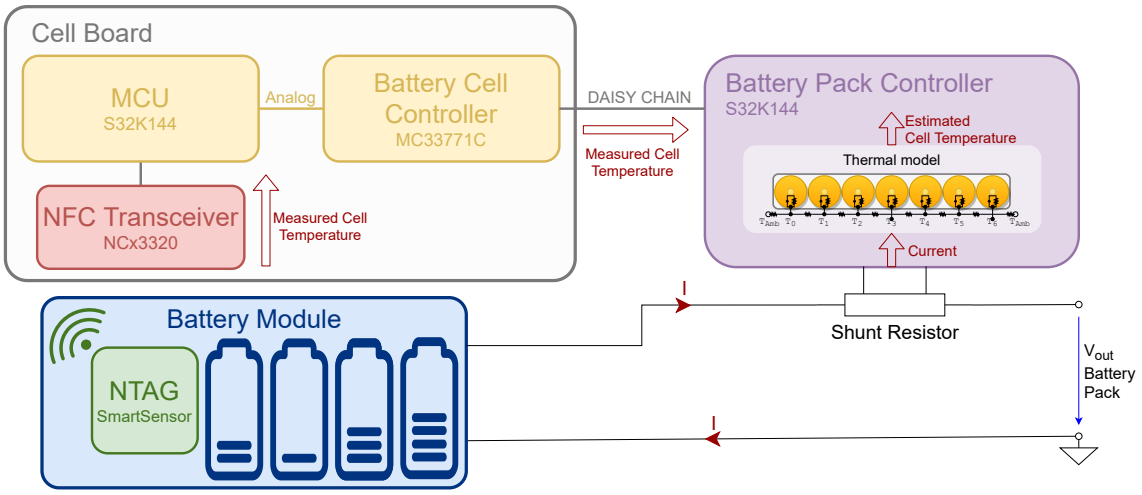


Figure 3.12: Redundant transmission channel of the temperature

Therefore, it is important to detect incorrectly transmitted data from the BMS in order to be able to react appropriately. Figure 3.12 shows an option for redundant transmission of the cell temperature in the prototype presented. Such a redundant transmission enables a plausibility check which is widely used in the automotive field. The temperature is transmitted once via the cell board and the daisy chain on the designated path to the Battery Pack Controller. The second transmission path would be the current flowing through the Battery Pack. The current flowing through the battery pack shows a clear correlation with the cell temperature. Using a thermal model, which takes into account the thermodynamic conditions inside the battery pack, the temperature prevailing in the cells can be inferred. The current consumed by the battery pack, which can be measured at the Battery Pack Controller via a shunt resistor or a Hall effect sensor [And10], can be utilized for approximating the cell temperature with this method.

3.4 NFC wireless readout (Off-vehicle)

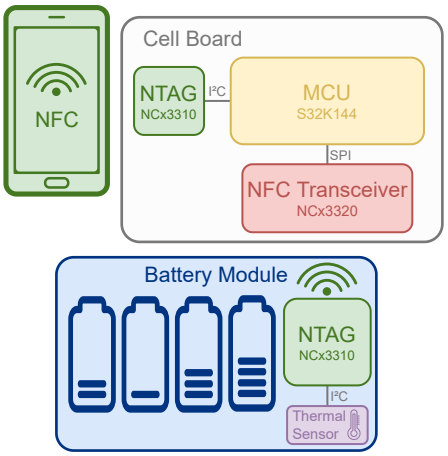


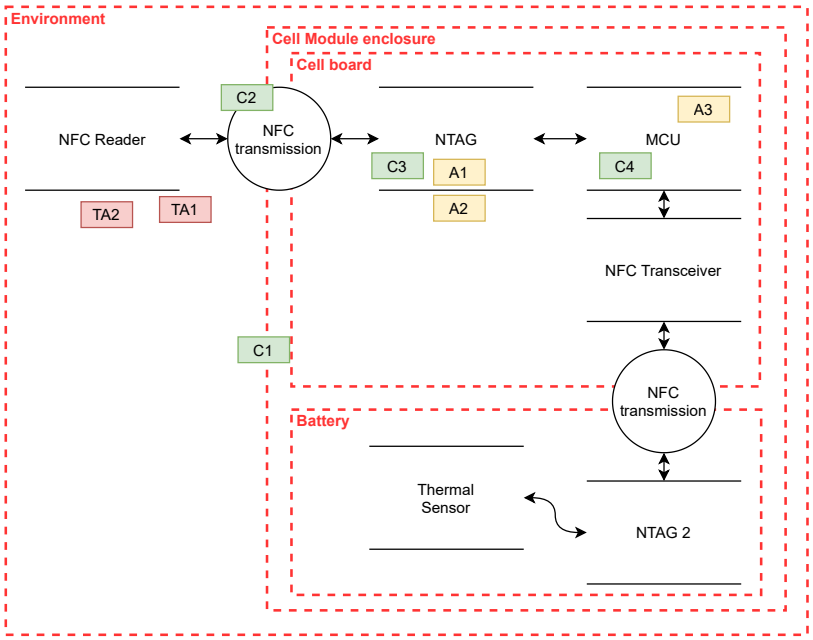
Figure 3.13: Block diagram for the NFC wireless readout

3.4.1 Security considerations

Different types of security are to be considered. The assets that have to be protected in the 3.14 setup as well as the security measures and the threat actuators are discussed in chapter 3.4.1.1. Parts of this setup are not only used in the off vehicle use case, but also during the operation in the vehicle. Security breaches in this part have an influence on the safety of the overall setup, which must be considered in raised security measures.

3.4.1.1 Threat model

For the detailed security evaluation a DFD of the setup was created which is depicted in figure 3.14. The assets to protect, as well as the threat actors are listed and the potential countermeasures to mitigate those threats are discussed in the following.



Threat Actors	
ID	Description
TA1	Conceal damage
TA2	Boost battery pack

Security Controls	
ID	Description
C1	Sealed cell module enclosure
C2	AES mutual authentication
C3	Read only interface
C4	Signature validation

Assets	
ID	Description
A1	Temperature data integrigy
A2	Temperature data authenticity
A3	Software integrity

Figure 3.14: Data Flow Diagram of the proposed setup for the NFC wireless readout

The DFD in Figure 3.14 shows the threat model for the off vehicle use case. In this use case, it should be possible to read the fault log after use inside the vehicle via an NFC interface to make an approximate estimate of the State Of Health of the battery cells.

The NFC interface between battery and cell board is again secured via signature validation. This security measure has already been discussed in the On Vehicle Use case and is explained in more detail in chapter 3.3.1.2. The Cell Module enclosure is again assumed to be physically sealed in this threat model, severely limiting the interfaces through which a potential attack can take place.

The Second Interface in the Data Flow Diagram 3.14 which is externally accessible and therefore vulnerable to threat actors is the exposed NFC interface between the external NFC reader and NTAG. Here, two ways are considered to secure it. The first possibility would be to make the NFC interface a read only interface. This possibility is evaluated in paragraph 3.4.1.2. The second option would be to use the AES 128 mutual authentication feature of the NTAG. This possibility is discussed in paragraph 3.4.1.3.

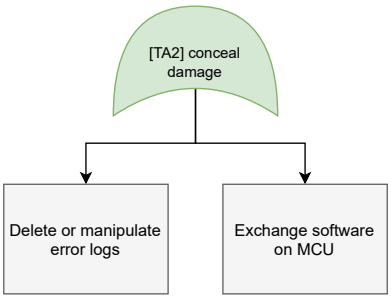


Figure 3.15: Fault tree - Pretending a damaged cell to be usable

In the fault tree in Figure 3.15, a possible threat of the off vehicle use case is evaluated. The structure could be manipulated to allow broken battery modules to be reused or resold. This could be achieved by changing or deleting the error logs, or even changing the complete software on the MCU so that the diagnostic data is no longer transmitted correctly.

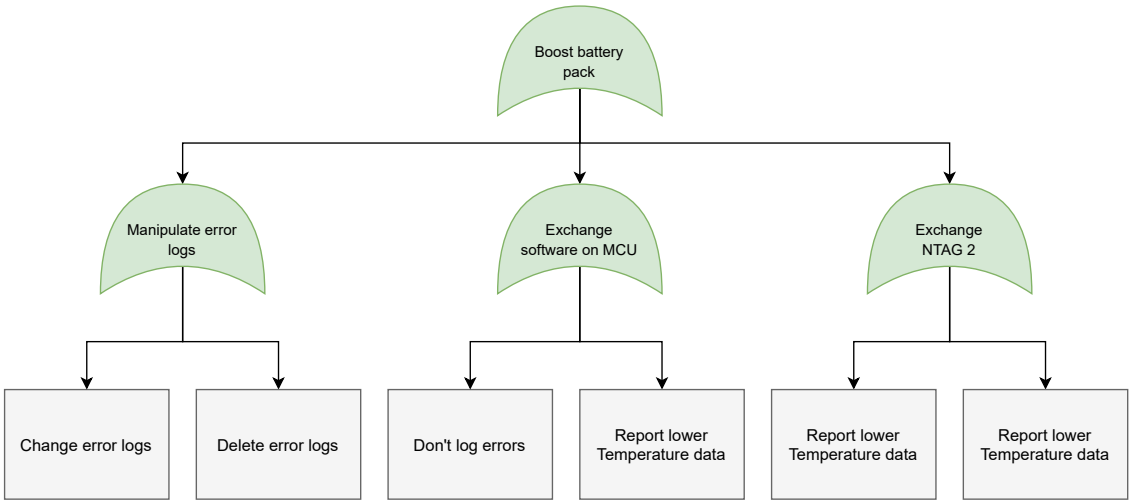


Figure 3.16: Fault tree - Boosting the battery pack

In the fault tree in figure 3.16 the potential attack is evaluated whose goal is to boost the battery pack. Boosting can take place both in the sense that the battery pack is overcharged, shuts down later when discharging, or delivers larger currents to accelerate faster. Boosting takes place by transmitting corrupted sensor data to the Battery Pack Controller and not storing or suppressing the errors that occur during boosting.

3.4.1.2 Design variant 1 - Read only interface

Now the risk factors identified in the threat model in Figure 3.14 must be counteracted by appropriate security measures in the design of the model. If the first security measure is to seal the cell module housing, a secure environment can be assumed for the communication

buses inside the setup. The only remaining attack surfaces are the both NFC interfaces which are at least partially exposed to the environment.

The first threat, a concealed damage, can be avoided by a read only interface configured at the NTAG 1. This read only interface is configurable on the used chip and makes sure that no malicious data can enter via the NFC interface exposed by NTAG 2 into the setup. For the second attack surface exposed, a valuable countermeasure would be a signature validation conducted by the MCU. This would ensure that the MCU is communicating with the intended NTAG and not a malicious one. Thus the MCU would receive and log only correctly measured temperature data. Therefore, the two attack vectors exposed would be closed for the evaluated threat and thus no damage which occurred during the use can be concealed.

The threat actor 2 would be the boosting of the battery pack. This boosting could occur in different ways including the possibility to overcharge the battery gaining more capacity or boost the available current and thus the available power. The current which can be drawn out of the battery cells is limited through a fuse. The extractable steady state current is mainly limited by the BMS. The current flowing through the battery cells has the largest influence on the cell temperature. To prevent the most serious safety risk of a battery module, the thermal runaway, the BMS monitors the temperature and limits the current if the temperature rises too high. One way to increase the current available from the cells and thus increase the power output of the battery module would be to pretend to the BMS that the temperature is less than the actual temperature. The BMS would allow the power drawn from the battery to remain in an unacceptable level for a longer period of time, resulting in excessive heat up of the cells and a severe safety risk.

3.4.1.3 Design variant 2 - AES mutual authentication

The second way to achieve the necessary security precautions for the after vehicle use case would be to use the AES mutual authentication feature for the NFC interface of NTAG 1. The remaining security measures would be identical to those described in the design variant described in chapter 3.4.1.2. The benefit of using the AES mutual authentication would be that the NFC transmission of NTAG 1 would no longer have to be restricted to read only, but could be used bidirectionally. This would enable authenticated reader devices to perform software updates via the NFC interface. The security measures for the second exposed NFC interface would remain the same as in design 1, as described in section 3.4.1.2 All other attacks would be fended off by the sealed cell model enclosure.

3.4.1.4 Conclusion

The sealed housing is a mandatory requirement for both of these possible design variants. Without it, hardware can always be replaced, which makes a design that is protected against all possible attacks extremely complex. The only difference lies in the protection of the vulnerability created by the NFC interface of NTAG 1. If the external reader is to be used for software updates, the interface should be secured via AES mutual authentication to enable bidirectional data transfer. If this update feature is not requested, the more secure variant would be to design the NFC connection as a read only interface. This

prevents data from entering the system via this interface and closes this attack vector entirely.

3.4.2 Energy awareness

At the intended use case, the battery modules are laying unused in the shelf after they were used in the vehicle. Then they are waiting for an NFC reader to read out the error codes. There are powered batteries in the system, so the power provision is granted. Nevertheless it must be ensured that the cells are not drained during the storage of the battery modules in the shelf, so the power consumption should be kept to a minimum.

To achieve the lowest possible standby energy consumption for the entire setup, the standby consumption of the individual components of the setup must be minimized. The energy-saving modes of the setup components will be evaluated more closely in the following.

3.4.2.1 S32K144 microcontroller

The S32k144 microcontroller supports several low power modes. These differ primarily in the time required for wake-up and in energy consumption. Table 3.1 shows the different power modes and the power dissipated in each mode.

power mode	wakeup time	current drawn
VLPS	$17\mu s$	$29.8\mu A$
STOP1	$0.08\mu s$	$7mA$
STOP2	$0.08\mu s$	$7.7mA$

Table 3.1: Low power modes and wakeup times of the S32K144 [20c][1]

Since the time needed by the NTAG to wake up from the low power state is much longer than the wake up time of the microcontroller, the VLPS power mode is chosen for the implementation. Furthermore, the sleep times are extremely long, which makes the energy consumption during sleep a more important criteria than the wake-up time.

3.4.2.2 NTAG

Mode	Standby energy consumption		wakeup time
Supply voltage:	$5.5V$	$3.3V$	
Hard power down	$0.31\mu A$	$0.25\mu A$	$\sim 1ms$
Stand by	$6.9\mu A$	$5.9\mu A$	
Energy harvesting	not possible	$0\mu A$	$3.1ms$ [20b]

Table 3.2: Differences of the energy conservation modes of the NTAG [20a][1]

[1]The values shown in the table are taken from product data sheets

While the battery module is in storage, the NTAG should be set to a state in which as little energy as possible is consumed. Only when a NFC field of a reader is recognized, the NTAG should be woken up. There are several possibilities to realize the requirement of the energy saving mode with subsequent wake up via the NTAG. These possibilities together with the advantages and disadvantages are outlined in table 3.2 and will be discussed in the following paragraphs.

Hard power down: The NTAG 5 provides a dedicated pin for enabling the hard power down mode. If this mode is enabled, the NTAG is switched off completely. That means, that the event detection pin is disabled either and cannot be used to wake up the MCU. The NTAG is booted from the hard power down mode, if the hard power down pin is released. Therefore, the hard power down mode cannot be used for this application where the automatic wake up via NFC is required. If the wake up of the whole setup with some external utility like a button is allowed, the hard power down mode for sleeping would become an option again. Even though the power consumption of the hard power down mode is lower than of the standby mode, it is more important to provide a good user experience where the setup is woken up automatically if an NFC reader approaches [20a].

Stand by: NTAG 5 can be set to stand-by mode by writing some value to a specified session register. It can also be configured to be entered automatically by the NTAG if no communication from the I^2C interface nor an NFC field is detected. The NTAG automatically wakes up from the stand by mode if an NFC field is detected. The event detection pin is fully functional during the standby since the NTAG is woken up automatically if anything happens which could require a notifying of the event detection pin.
During the standby mode, all session registers are staying initialized. The values in the SRAM are kept too. Therefore when using this mode the MCU can write values into the SRAM of the NTAG and go to sleep afterwards. The event detection pin is configured to trigger an event if the last byte of the SRAM is read by the NFC reader. So if a reader approaches, the NFC field will wake up the NTAG from the sleep mode. The NTAG will transfer the data stored in the SRAM via the NFC interface. If this transmission is finished, the Event Detection Pin will wake up the MCU which can write more data into the SRAM and go to sleep afterwards [20a].

Powered by NFC field: The NTAG 5 has an energy harvesting capability, where it is powered completely via the NFC field. Therefore, if no NFC field is available, no power is available too which means, that there is no power consumed at all by the NTAG if no reader is nearby. The drawback of this solution is, that there is only a limited voltage range covered by the energy harvesting function. The rest of the setup would have to be matched to a voltage level offered by the energy harvesting function. Otherwise a level shifter would have to be introduced between the components adding additional complexity and thus lowering the reliability of the overall system [20a].

Chapter 4

Implementation

The implementation section is broadly divided into the two use cases evaluated in this thesis. The on vehicle use case during the operation of the battery in the car is described in more detail in paragraph 4.2. For the off vehicle use case, the cell board is modified. This configuration is described in further depth in chapter 4.3.

4.1 Toolchain

In the following chapter the toolchain used for compiling and building is presented. The used IDE is presented in paragraph 4.1.1 and the compiler used is specified in paragraph 4.1.2. The sources from which the software stack running on the microcontroller is assembled are described in paragraph 4.1.3.

4.1.1 Integrated Development Environment

For the development of the firmware for the microcontroller used in the prototype, the S32K144, the S32 Design Studio from NXP was used. The S32 Design Studio is based on Eclipse and is, according to NXP, a Integrated Development Environment (IDE) for the development of automotive applications.

4.1.2 Compiling, flashing and debugging

The firmware developed for this setup is executed on the S32K144 microcontroller from NXP. Therefore, the software must be compiled for the corresponding architecture, cortex-m4 ARM target architecture. The gcc-6.3-arm32-eabi was used as compiler.
For flashing and debugging purposes, the PEmicro debugger via the OpenSDA Embedded Debug Interface was used.

4.1.3 Software

On the driver side, the **S32 Software Development Kit** was used to implement the Hardware Abstraction Layer (HAL). This includes drivers for all hardware modules and bus systems available on the microcontrollers of the S32K series. From this library the SPI driver was used to interact with the NFC reader. The I^2C driver of the S32 SDK was used to communicate with the NCx3310 in the off vehicle use case. Also to implement the sleep mode in the off vehicle use case an API of the S32 SDK was used. To perform the hardware configurations, the Processor Expert software was used. This is a software module that is available in the S32 design Studio, the SDK used here and presented in chapter 4.1.1 presented in chapter IDE is integrated. In the processor expert, hardware-dependent configurations for the S32 SDK software can be made using a GUI. The Processor Expert software then generates configuration files which can be used to initialize the S32 SDK at runtime.

The **NFC reader library** is provided by NXP too. It was used to implement the application logic for the NFC reader used in the setup, the NCx3320.

The **BMP 180 driver** is provided by Bosch®, which also produces the I^2C temperature sensor used in this setup, the BMP 180.

For the authentication and signature validation of the NTAG, a library is needed that supports the secp128r1 ECC curve. Here the **ecc-nano** library was chosen, which is provided by user kmackay on GitHub.

The logic for the **NTAG 5 I^2C driver** was taken from a demo software for the NTAG 5. This is written for a microcontroller of the MCUXpresso series, and can be built with the help of the MCUXpresso IDE and flashed to the corresponding microcontroller. The interface to the HAL of this driver, was adapted to the API of the S32 SDK to work with the API of the I^2C driver implemented there.

The **FTM pulse width driver** was taken from a software sample for the S32k144 microcontroller. This driver is described and provided in the NXP document AN5413 called S32K1xx Series Cookbook.

4.2 Battery management Prototype

Since this BMS is a modular BMS, it is separated into different modules whose functionality and implementation will be described in more detail below. The individual modules are the Battery Pack Controller, the Battery Cell Controller and the Battery Module. The Battery Pack Controller is presented in more detail in paragraph 4.2.1, the Battery Cell Controller in paragraph 4.2.2 and the Battery Module in paragraph 4.2.3.

4.2.1 Battery Pack Controller

The Battery Pack Controller consists of the FRDMDUAL33665EVB evaluation board and a S32K144EVB evaluation board.

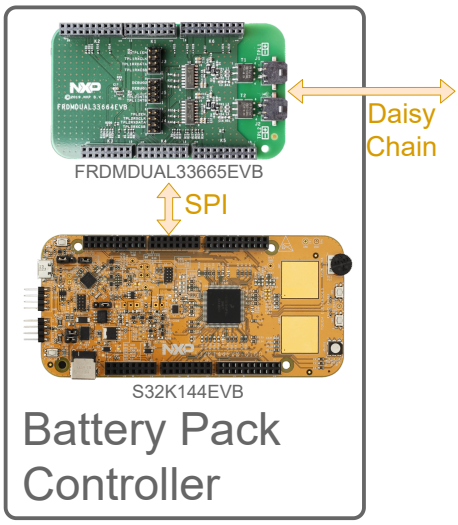


Figure 4.1: Battery Pack Controller

The FRDMDUAL33665EVB houses two MC33664, which are converting SPI signals from the microcontroller into signals of a high speed TPL network. A bus isolator transformer provides isolation between the SPI interface and the high speed network. The SPI signals of the microcontroller are directly converted to pulse bits for the high speed network.

The second component of the Battery Pack Controller is the S32K144EVB, which carries the S32K144 automotive microcontroller. This microcontroller implements in this design the function of the Battery Pack Controller via its firmware. The software summarizes the data collected by the individual Battery Cell Controllers. The various battery management algorithms are also executed on the Battery Pack Controller, i.e. on this microcontroller.

4.2.2 Battery Cell Controller (Cell Board)

The Cell Board is composed of the RD33771CDSTEVB, a S32K144EVB and the NCx3320. The RD33771CDSTEVB houses the MC33771C Battery Cell Controller, which is responsible for measuring the cell voltage, the cell temperature and the cell balancing.

The microcontroller on the S32K144EVB is used to implement the additional functionality desired by the cell board. The temperature is detected by a wireless interface from the sensor, and the signature from the battery module is verified for authenticity to prevent the use of cheap aftermarket products. The MC33771C Battery Cell Controller is intended for connecting analog temperature sensors to it. These analog temperature values are emulated by the microcontroller and thereby fed into the rest of the setup for further processing.

The NCx3320 is used to establish the required wireless connection to the temperature sensor. This NFC connection is performed by the NCx3320 board on the Cell board as active transceiver as shown in figure 4.2 and the NCx3310 on the battery module as the passive NFC TAG as shown in figure 4.5.

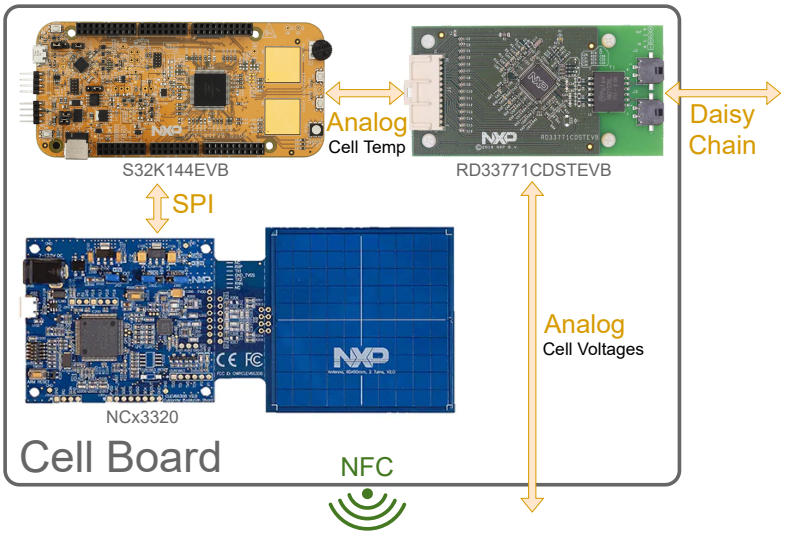


Figure 4.2: Cell board components for in vehicle use

To achieve the desired functionality the S32 Software Development Kit is used on the S32K144 microcontroller. This contains drivers for the different bus interfaces provided by the S32K144 microcontroller as well as drivers for the other functionalities such as the low power modes or security functions. From this Software Development Kit the SPI driver for the NCx3320 NFC transceiver is used. To realize the analog interface to the RD33771CDSTEVB board the flexible timer module is used. This module generates a pulse width modulation with a duty cycle depending on the actual measured temperature. A subsequent low-pass filter converts the pulse-width modulated signal into an analog signal, which is transmitted from the RD33771CDSTEVB to the Battery Pack Controller via the TPL bus.

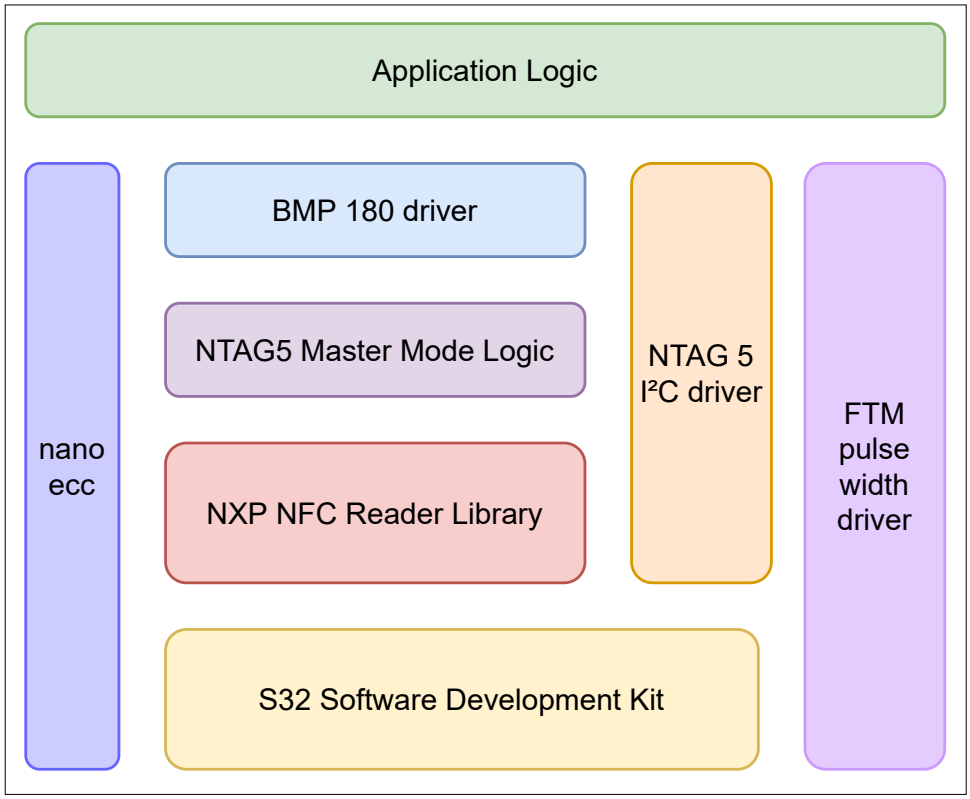


Figure 4.3: Software stack running on S32K144 MCU

Figure 4.3 depicts the software stack used to operate the S32K144 microcontroller.

The foundation of the firmware is the S32 Software Development Kit, which combines drivers for the bus systems and other features of the microcontroller.

The NXP NFC reader library is build upon this driver. It uses the SPI driver from the S32 Software Development Kit to interact with the NCx3310 NFC transceiver connected to the SPI bus. Also a timer is used by the NFC reader library to be able to measure certain time intervals. Since the implementation of this timer is hardware specific, this timer driver is not implemented in the Reader Library, but utilizes an API from the S32 Software Development Kit.

The temperature sensor BMP180 is not directly connected to the microcontroller as shown in 4.5. In between there is the NCx3310, which is run in I^2C master mode. This implies that the NCx3310 is the bus master when the I^2C bus is connected to the BMP180. The commands received via the NFC interface are passed to the BMP180 sensor. The NCx3310 has an adapter role between NFC communication and I^2C communication. Since the BMP180 driver expects a I^2C bus, the NTAG5 master mode logic has been implemented, which makes the NFC connection to the NTAG treated as a I^2C connection for the BMP 180 driver.

The BMP 180 driver is designed to operate the BMP180. At the interface to the NTAG5 master mode logic it sends the I^2C commands for the BMP180 driver. At the interface to the application logic it provides functions to initialize the BMP180 temperature sensor, measure temperature and air pressure.

The NTAG 5 I^2C driver builds upon the I^2C driver of the S32 Software Development Kit. It is not used in this setup. It is needed for the functionality in the off vehicle use case where an Nx3310 board is connected directly to the microcontroller via an I^2C bus.

The FTM pulse width driver is used as a driver for the analog interface over which the temperature values are emulated for the RD33771CDSTEVB, which expects an analog temperature sensor as input. The RD33771CDSTEVB expects an NTC resistor as a sensor, which is connected in such a way that at the upper end of the automotive temperature range, i.e. at $125°C$ 0V is applied to the interface. If a voltage level of $5V$ is applied to the analog input of the RD33771CDSTEVB board, this is interpreted as the lower end of the automotive temperature range, i.e. as $-40°C$. Since an NTC resistor is expected for measurement, the mapping between analog values and interpreted temperature is severely non-linear. This non-linearity is taken into account by the FTM pulse width driver. The FTM pulse width driver receives a temperature from the application logic and adjusts the duty cycle of the pulse width modulation at the flexible timer module accordingly, so that after a low pass filtering at the analog input of the RD33771CDSTEVB the correct temperature is interpreted.

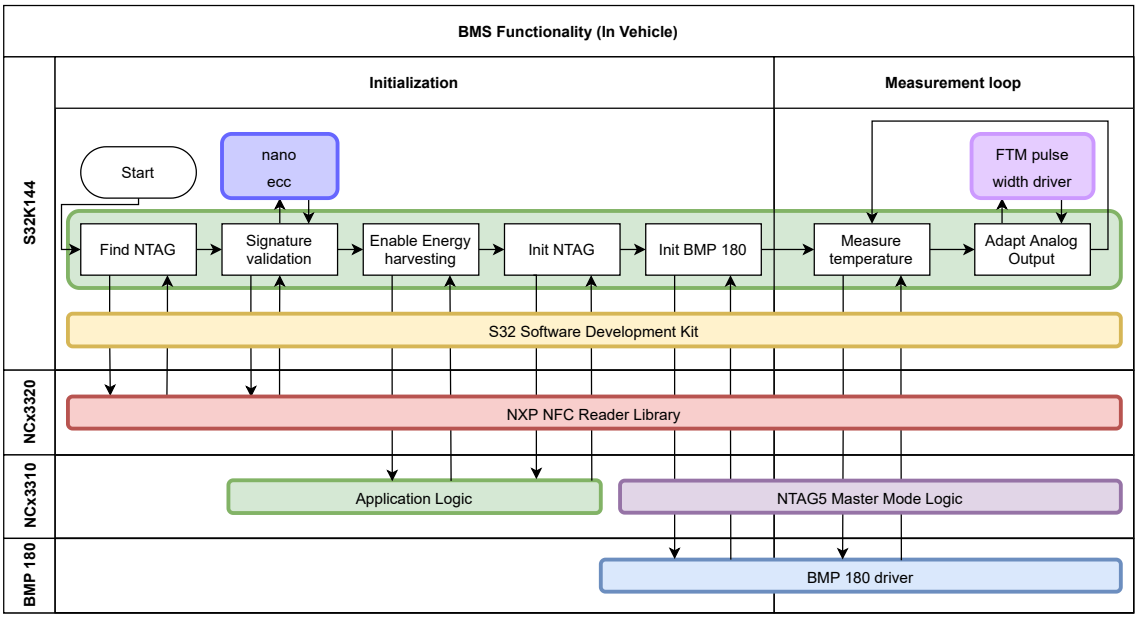


Figure 4.4: Sequence diagram of the software for the on vehicle use case

4.2.3 Battery Module

The purpose of the battery module in the emulator is to provide information about cell temperature and cell voltages. The cell temperatures are accumulated digitally via an NFC interface, while the cell voltages are measured via an analog interface on the BATT-14CEMULATOR.

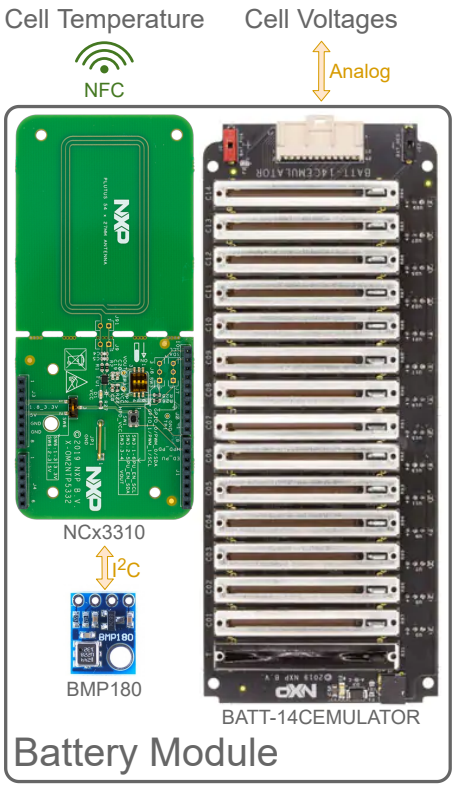


Figure 4.5: Components of the battery module

Although the components for determining the cell temperature and the BATT-14CEMULATOR for emulating the cell temperature are located on the same module, they can be treated as separate entities since they do not interact with each other in any way.

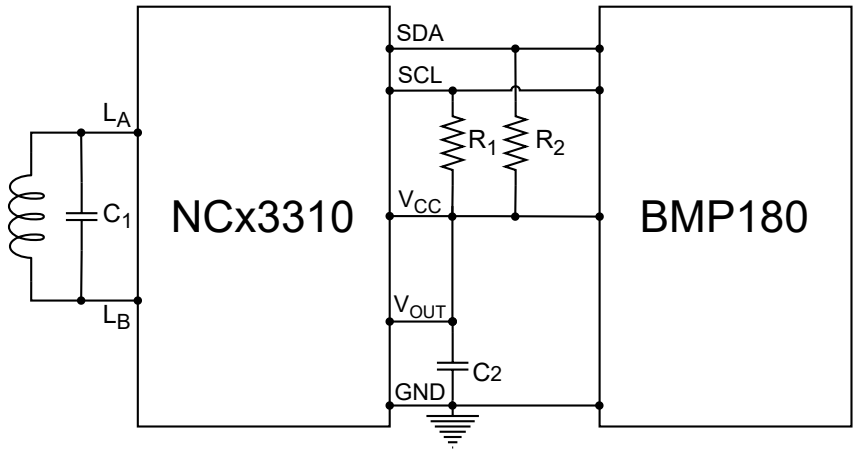


Figure 4.6: Interconnection of NCx3310 and the temperature sensor[20a]

To provide the temperature values the NCx3310 is used, which is attached to the BMP180 sensor via a I^2C connection. By using energy harvesting (section: 4.2.3.2) and the master mode (section: 4.2.3.1) the setup does not need many additional components. Figure 4.6 shows the connection between NCx3310 and BMP180. Resistors R1 and R2 are pull up resistors required for the I^2C bus to function. Capacitor C2 is necessary for proper operation of the energy harvesting functionality. According to the data sheet, it prevents the voltage from dropping too far during modulation breaks.

4.2.3.1 NCx3310 master mode

The NCx3310 is in the I^2C master mode. In this mode it operates as a bus master on the I^2C bus. This means that no MCU is needed in this setup, keeping the overall setup extremely simple. Figure 4.6 shows all the components needed to run the NCx3310 and the BMP180. In the I^2C master mode, the NCx3310 takes over the function of an adapter. It forms an interface between the NFC and the I^2C bus, translating or passing on the commands directly, as shown in the sequence diagram 4.7.

Custom NFC Commands: For the use of the master mode implemented on the NCx3310, some new commands with special command codes are implemented on the NCx3310, which are not yet included in the NFC standard. Therefore they are not supported by the NXP NFC reader library and have to be implemented in a way that they interact with a lower layer of the library. These commands are listed below along with their command codes.

- **I^2C write (D4):** writes a command to the I^2C line of the NCx3310.
- **I^2C read (D5):** receives a command from the I^2C line and stores it in the SRAM.
- **Read SRAM (D2):** reads the addressed content from the SRAM.

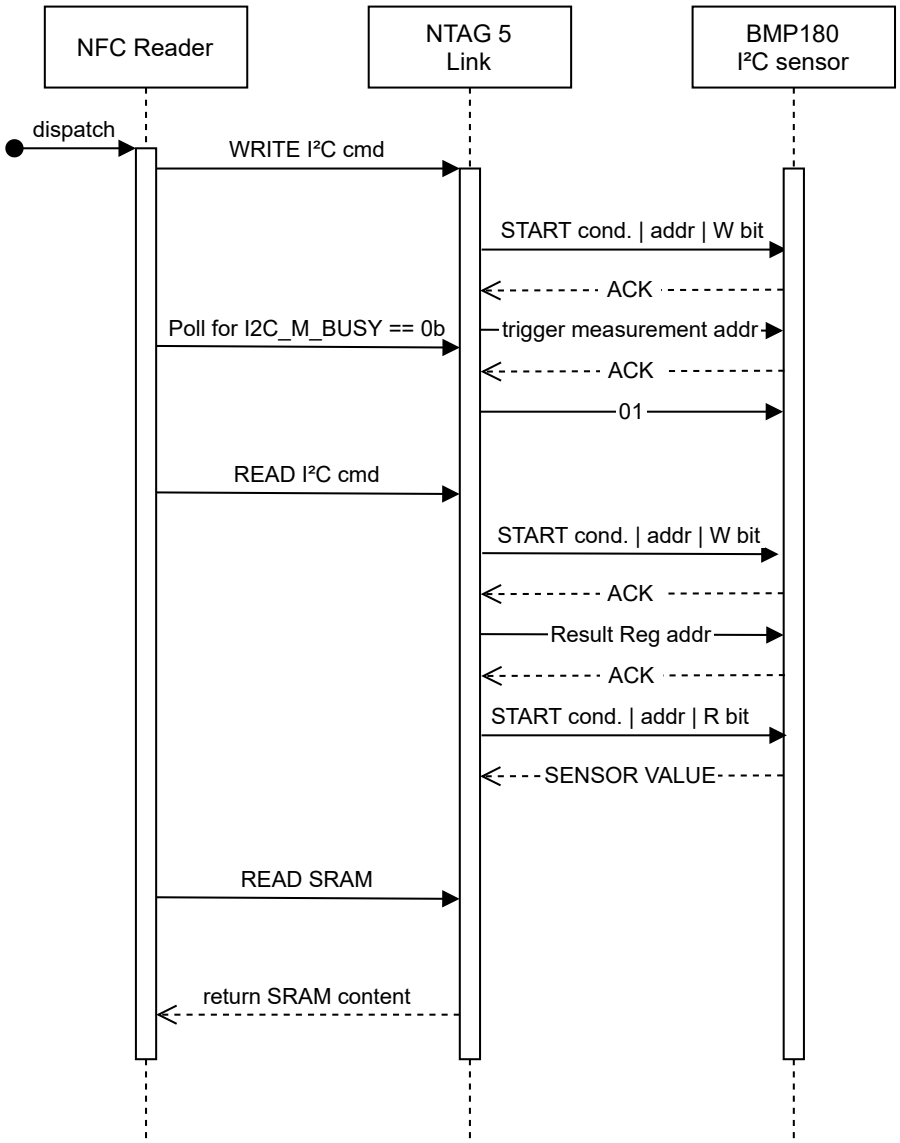


Figure 4.7: Sequence diagram of the I^2C master mode [20a]

4.2.3.2 Energy harvesting

Energy harvesting is used to power the BMP180 sensor. The energy required for operation is drawn entirely from the NFC field of the reader. This obviates the need for an additional energy supply.

Figure 4.6 shows the wiring diagram between the NCx3310 and the BMP180. The energy retrieved via energy harvesting is provided by the NCx3310 at V_{OUT}. Since the NCx3310 itself also requires a power supply to operate the SRAM, the power is fed back into the NCx3310 itself at V_{OUT}.

Energy harvesting is not enabled immediately by the NCx3310 after boot up. First it checks if enough energy can be harnessed from the existing NFC field. If energy harvesting is enabled although not enough energy can be harvested from the NFC field, a reset cycle would occur. Via session registers the NTAG is configured to match the required voltage level.

One can choose between $1.6V$, $2.4V$, and $3V$. In this implementation a voltage of $3V$ was chosen. The used sensor needs a voltage of $3.3V$ for operation. If $3V$ is selected as voltage for energy harvesting, the NCx3310 delivers a voltage between $2.9V$ and $3.1V$ which is sufficient for the operation of the temperature sensor.

A desired minimum current can be selected. If a current is selected, the energy harvesting will only be switched on if the current that can be obtained from the currently present NFC field is at least as high as the selected current. For the required minimum current several setting variants can be chosen, where $> 0.4mA$ is the lowest possible setting and $> 12.5mA$ is the highest adjustable minimum current.

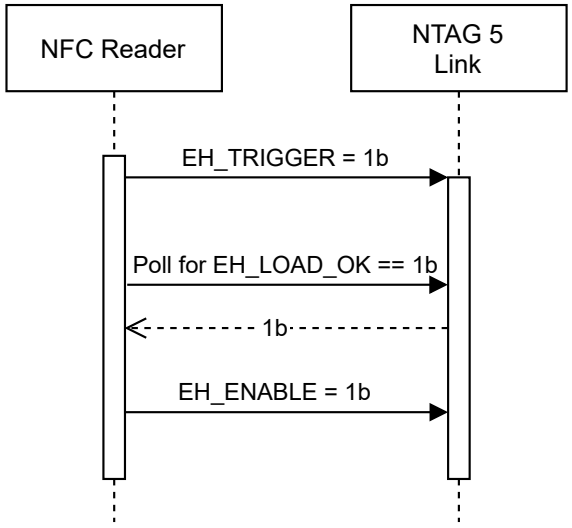


Figure 4.8: Sequence diagram of the transmission to enable energy harvesting [20a; 20b]

The procedure for enabling the energy harvesting function is shown in Figure 4.8. First, the power check needs to be enabled by writing 1b into the EH_TRIGGER register.

Then the NTAG checks if enough energy can be provided. The result of this check is supplied in the EH_LOAD_OK register. The MCU reads this register in a loop until 1b. is received from the register. This means, that the requested energy can now be supplied and energy harvesting can be enabled.

Subsequently, the energy harvesting is activated by the MCU by writing the value 1b into the register EH_ENABLE. This can also be done without prior checking, but there is a risk that the NTAG will be overloaded by the connected load and therefore reset.

4.2.3.3 Signature validation

One requirement of the prototype is the anti counterfeiting functionality. The NCx3310 offers a reprogrammable originality signature.

Each board is shipped with an 8 byte Unique Identifier (UID). This UID is signed with a Elliptic Curve Digital Signature Algorithm (ECDSA), resulting in a 32 byte signature that can later be verified using the public key. The 32 byte long signature can be stored on a certain designated location on the NCx3310, which can then be locked against modification.

```
1                phalICode_ReadSignature(psalI15693, &signature, &pSignLen);
2
3                if (ecdsa_verify(&PubKey, UID, signature, &signature[16])) {
4                        DEBUG_PRINTF("Signature is valid");
5                } else {
6                        DEBUG_PRINTF("Signature is not valid");
7                }
```

Listing 4.1: Verification of the NTAGs originality signature

For reading the signature, a special NFC command READ SIGNATURE is available, which sends the whole signature at once as response. One can also read the 32 bytes with normal READ REGISTER commands, but it is recommended to use the dedicated command for this purpose. For the READ SIGNATURE command there is an own function in the NXP NFC Reader Library as shown in the code example in listing 4.1.

The second function call in the code sample 4.1 edsa_verify is a call to the nano-ecc library. This is used to verify the signature obtained from the phalCode_ReadSignature function call by using the public key.

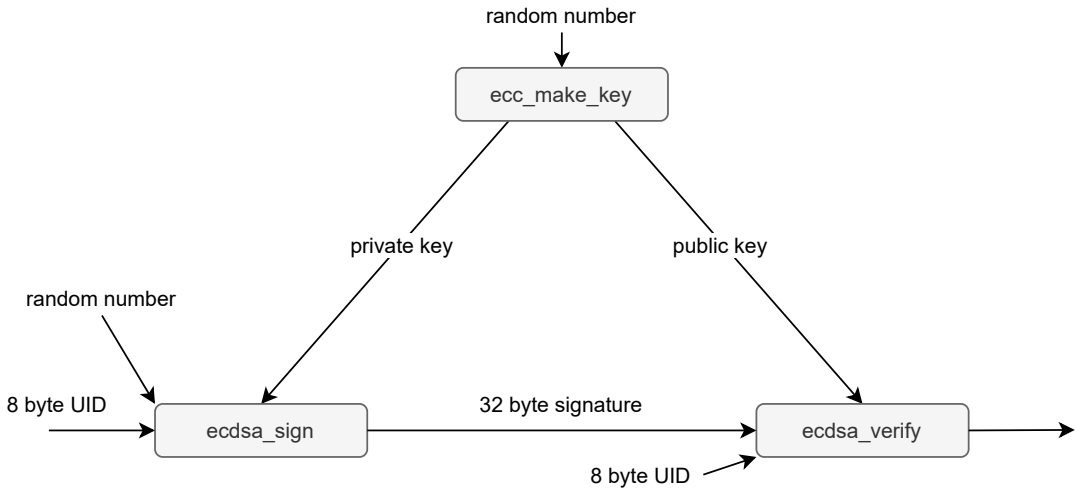


Figure 4.9: Generating and verifying originality signature using nano-ecc libary

Reprogramming the originality signature: Figure 4.9 shows how to create a new key-pair, sign the UID with it and verify the created signature. The whole procedure can

be done with the ecc-nano library. The gray boxes represent functions of the API of the ecc-nano library. The arrows represent variables. The UID is assigned to each NCx3310 during manufacturing and can be read out. The 32 byte signature can be stored on the NCx3310. The blocks provided for this purpose are blocks 00h to 07h. After writing, these blocks can then be locked so that they cannot be modified any longer.

During the verification this originality signature is read from the NCx3310. Also the UID is requested from the NCx3310. Together with the public key the function edsa_verify can be used to verify that the person who created the signature is the owner of the private key. The code for this verification with the function call in the ecc-nano library is shown in listing 4.1.

4.3 Off vehicle use (Error log readout)

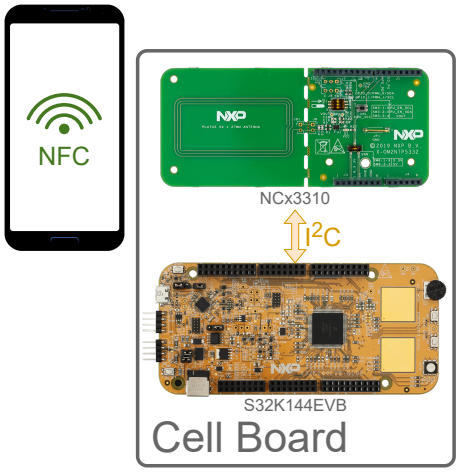


Figure 4.10: Cell board components for the off vehicle use case

In this setup, it should be possible to read data from the microcontroller via an external NFC reader. For this purpose, the NCx3310 has a pass-through mode implemented in hardware, which is intended specifically for this use case. The data should be transferred from the microcontroller to the NFC reader. The microcontroller writes the data to the SRAM of the NCx3310 chip via the I^2C bus. Then it goes into a low power state and waits for the NFC reader to read the contents from the SRAM. When the last byte is read from the SRAM, the MCU gets an interrupt from the event detection pin through which it wakes up and writes new data to the SRAM. Since the SRAM cannot be read from the NFC interface and the I^2C interface at the same time, an arbiter is implemented in the NCx3310. Depending on which of the two interfaces a request comes from, this arbiter switches to the respective interface and blocks access from the other. If no access happens for 25 milliseconds, the arbiter switches to a neutral mode and gives access again to the first interface which tries to read or write the SRAM. The arbiter can also be switched manually via a configuration bit. This bit can also be polled to find out if the MCU has finished writing to the SRAM and the NFC reader can start reading.

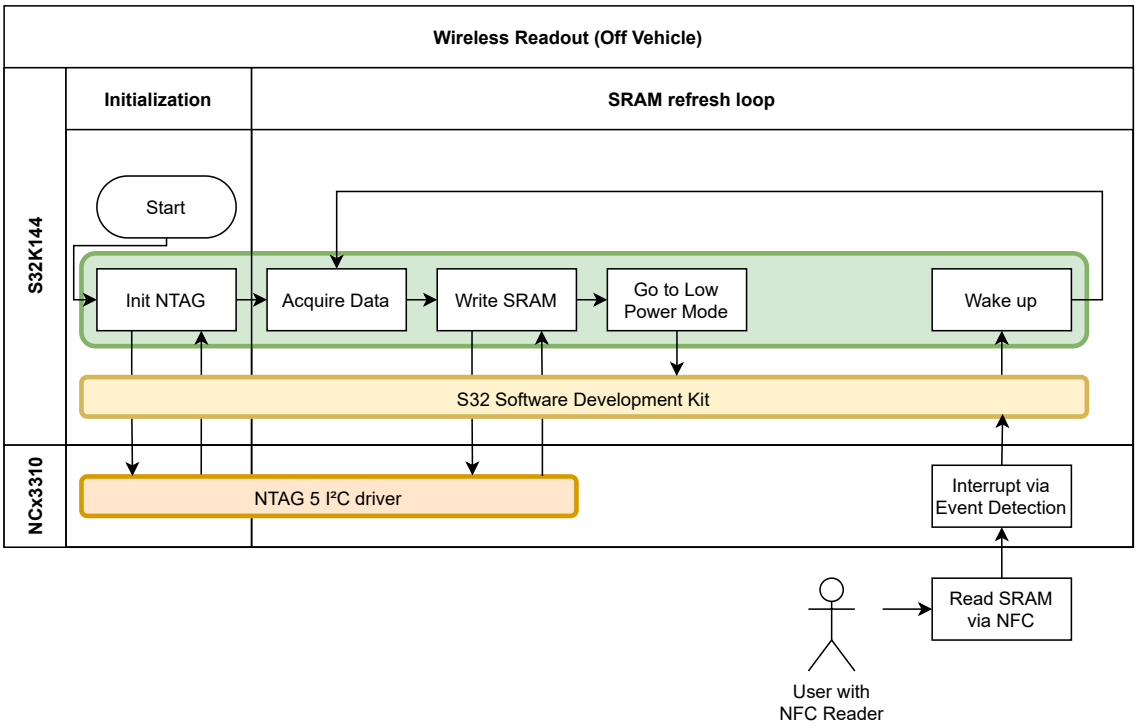


Figure 4.11: Sequence diagram of the software for the wireless NFC readout

4.3.1 Cellboard wakeup via NFC

After the battery unit is in storage, there are 2 ways to wake it up which both have their advantages and disadvantages. The wake-up from standby mode, which is described in chapter 4.3.1.1 works very fast and also the content of the SRAM is kept while in standby mode, but the chip needs some power in this mode.

The wake-up from power down mode takes longer and also the SRAM is not initialized because of the missing power supply. Before the SRAM can be read out by the reader, it has to be initialized.

4.3.1.1 Wakeup from stand by

The NCx3310 chip can be configured to automatically enter a standby mode when the NFC field disappears and there is no communication on the I^2C bus. The NCx3310 automatically wakes up from this standby mode when it receives a communication from either the NFC interface or the I^2C interface. The SRAM is also powered during standby mode and therefore its contents are preserved. Thus, the NCx3310 can be put into the low power mode in order to preserve energy. The MCU writes to the SRAM via the I^2C interface and then puts itself into a low power state. The NCx3310 chip enters standby mode as well, as there is no communication on the I^2C bus and no NFC field is present. The setup is now in the low power state.

If a NFC field from a reader is sensed, the NCx3310 will automatically wake up from stand by mode. The SRAM that still contains the written data can be read by the reader. If the reader reads the last byte of the SRAM the event detection pin triggers. This triggers an interrupt on the microcontroller, which wakes up from its low power mode and writes new data to the SRAM. After that it goes back to a low power state.

If the reader is still in the vicinity, it can now read the SRAM again and will trigger a notification at the event detection pin when it has finished reading. If the reader, and thus the NFC field, is removed from the NCx3310, it returns to standby mode as well and the setup is in low power mode again.

4.3.1.2 Powered by NFC field

As shown in table 3.2, the NCx3310 still requires energy in standby mode. To avoid that the battery cells are completely drained or to extend its depletion time, the NCx3310 should be switched off completely. Then, the MCU would be the only component that still draws current.

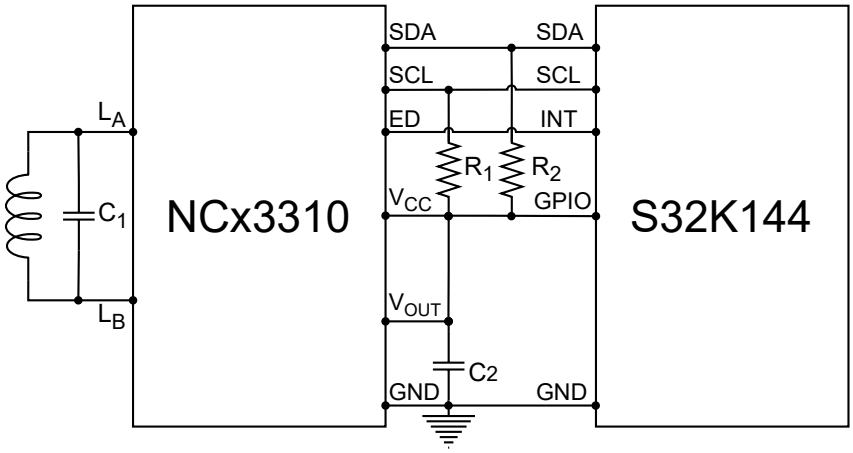


Figure 4.12: Interconnection between the NCx3310 and the S32K144 microcontroller [20a]

The NCx3310 can be supplied with energy via energy harvesting. If no NFC field is present anymore, the NCx3310 has no energy supply any longer and is consequently switched off.

Figure 4.12 illustrates how the NCx3310 has to be connected to the microcontroller in order to achieve this functionality. The V_{OUT} output of the NCx3310 is connected to the V_{CC} input of the NCx3310 and to a GPIO pin of the S32K144. The microcontroller can use this input to determine if the NCx3310 is currently in energy harvesting mode, or if it is currently powered off. The ED pin remains connected to an interrupt input of the microcontroller. However, this interrupt input must now be configured so that an interrupt is triggered on both a rising and a falling edge. Thus an interrupt is triggered every time the NFC field is turned on or off, as well as when the last block of the SRAM has been read. In the ISR, the microcontroller can then determine via the GPIO pin whether the interrupt was triggered because the last block of the SRAM was read, or whether a reader device was detected or removed.

4.3.2 Security - one directional data transfer

As a security precaution, data transfer was restricted to a one direction transmission only. Data can only be read by the external NFC reader, but not written. It is essential that the data and error logs stored on the microcontroller cannot be altered. A threat actor determined in the threat model in figure 3.14 for this prototype is to conceal some damage happened during the usage of the battery. This can happen by deleting or rewriting error logs. As the NFC interface is an interface to the outside world, the possibility of a bidirectional data transfer via the interface would offer the option of changing the stored data. Therefore, the interface for reading out the error memory was implemented in one direction only.

This is assured by the fact that the MCU in the currently implemented version only sends write commands, and no read commands to the NCx3310 via the I^2C bus. Furthermore, SRAM is set to read only towards the NFC interface. This is a configurable security feature provided by the NCx3310 chip.

4.3.3 Implementation details

The NCx3310 chip expects a very unique format of its I^2C messages.

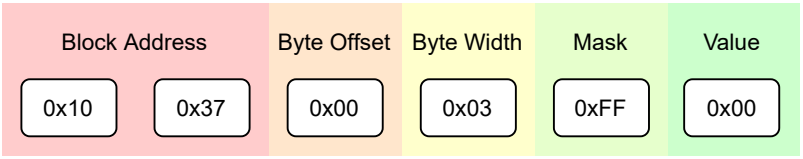


Figure 4.13: Message format for writing a value in a specific register via I^2C

Each I^2C address of the NCx3310 is 2 bytes long. The first byte is always the value 0x10, and the second byte has the same address as the NFC interface. Since the memory on the NCx3310 is divided into blocks and each block is 4 bytes long, the next byte to be sent is a 1-byte long byte offset, which has a value between 0 and 3, depending on which byte is to be written. This is followed by a bit mask which is exactly one byte long and defines which bits of the sent byte should be taken over in the memory of the NCx3310. If it is 0xFF, as shown in figure 5.7, the entire value is taken over into the respective memory location of the NCx3310. The last byte, shown as value in figure 5.7, is the value to be saved in the addressed register.

For reading a value from a register via I^2C, the NCx3310 specifies a nomenclature similar to that for writing shown in figure 5.7. The first 2 bytes constitute the block address. The next byte is again the byte offset. After these 3 bytes sent via the I^2C bus, the NCx3310 transmits the value of the addressed register via the I^2C bus as a reply.

Chapter 5

Results

For the evaluation of the built prototype, the timing and the transmission rate should be evaluated. This evaluation will again distinguish between the two setups, the on vehicle use case and the off vehicle use case.

In chapter 5.1 the setup is described, which was built up, in order to accomplish the time measurements and to determine the result. In chapter 5.2 the measured oscilloscope images are discussed. In 5.3 the measured times are elaborated and illustrated.

5.1 Measurement Setup

For the evaluation of the results the time, which is needed for the execution of different code sections has to be determined. To measure the execution time of the corresponding code sections, a command was inserted in the code at the beginning of the measurement, which sets an output pin on the microcontroller to high. At the end of the code section, a command was inserted in the code that sets the corresponding output pin back to the logic level low. Then, using an oscilloscope, the time that the output pin was high can be determined. This corresponds to the time that the microcontroller needed to execute the corresponding code section.

The oscilloscope used for the measurements is the Analog Discovery 2 from the company DILIGENT®. According to the manufacturer, it has a sample rate of $100M^{samples}/_{second}$. The oscilloscope was set to the maximum resolution during the measurement. Only $8000^{samples}/_{second}$ were produced at this resolution. This corresponds to one sample every $0.125ms$. The measurement range at which the values were taken was 2 seconds. The reason for the low resolution could be that no more values could be stored in the cache of the device and when recording shorter periods of time the resolution would increase. Also an error in the settings, i.e. in the usage, cannot be precluded and might explain the poor captured resolution. To detect when the MCU is at a certain code section, output pins were set to high or low. The rise and fall time of the pin was measured by the oscilloscope with a time of 0.25 milliseconds. As a reference, a static code section without bus communication was measured several times. This time should always be the same, since the execution speed is determined exclusively by the clock of the MCU and not by external

bus communication. A standard deviation of $\pm 0.3ms$ was calculated for this reference measurement. It can therefore be expected, that the measurement setup itself contributes this value to the standard deviation of the measurements.

5.2 Time measurements

The microcontroller was run with an operating voltage of 5V. Thus in the oscilloscope pictures one can see the voltage going up to 5V if the corresponding output pin was set to high and going back to 0V if the corresponding output pin was set to low again.

5.2.1 NTAG discover and initialize

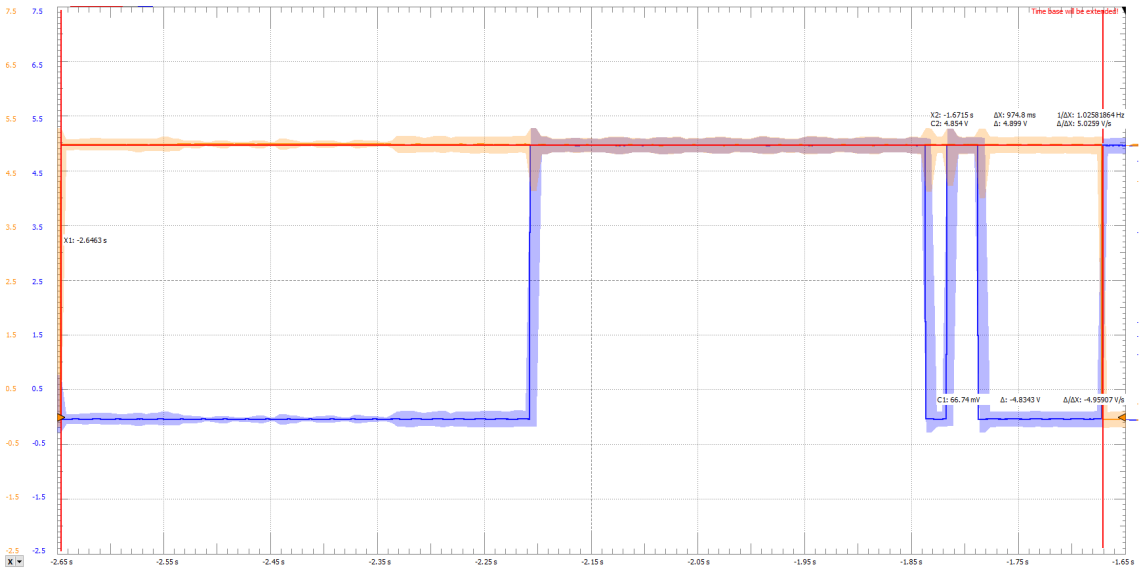


Figure 5.1: Time the NFC reader takes to recognize ntag and enable energy harvesting

The recognition of the NTAG by the reader, the activation of the master mode, and the activation of the energy harvesting take 3.3 seconds according to the measurement in 5.1. How long the validation of the signature alone takes is determined in paragraph 5.2.2. The time of the boot process measured in this paragraph varies widely. It depends on how fast the NFC reader recognizes the NTAG. Times from 600 milliseconds to over 6 seconds have been measured for this boot process.

5.2.2 Signature validation

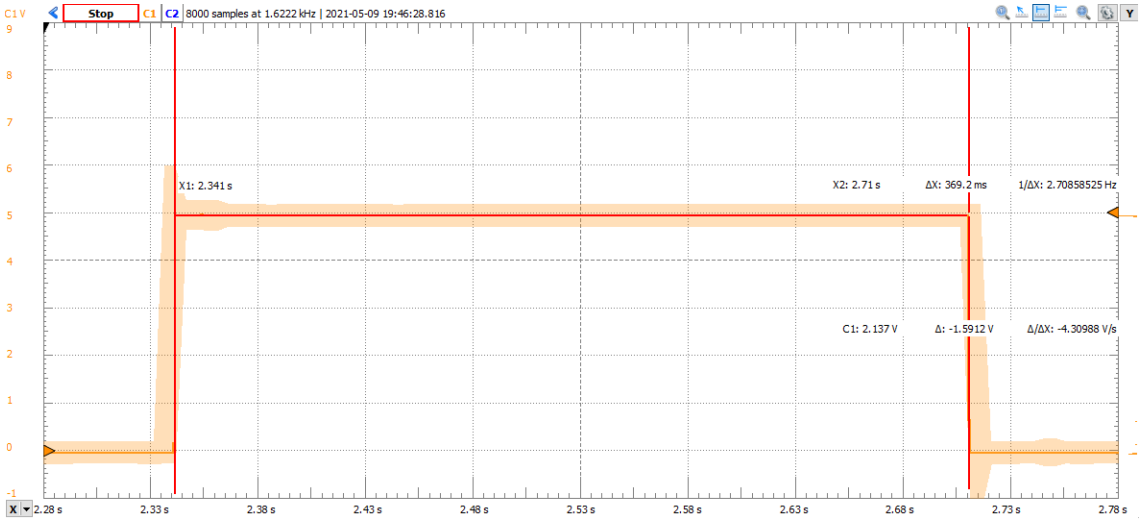


Figure 5.2: Time the validation of the originality signature takes

The validation of the signature takes 369.2 milliseconds according to the measurement as shown in Figure 5.2.

5.2.3 SRAM transaction

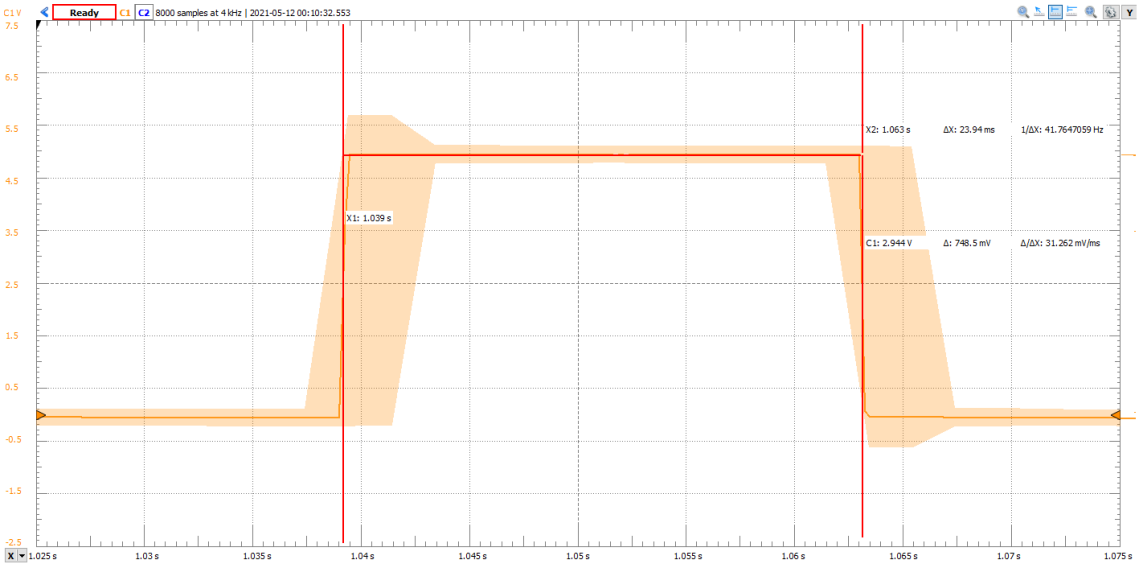


Figure 5.3: Time to write the SRAM via I^2C

When writing the SRAM via I^2C one comes to the limits of the resolution of the oscilloscope used. In image 5.3 the SRAM was written 100 times in succession via I^2C and then

the duration for a single writing of the SRAM write was calculated. The approximate duration for writing the SRAM once over I^2C is $0.2394ms$.

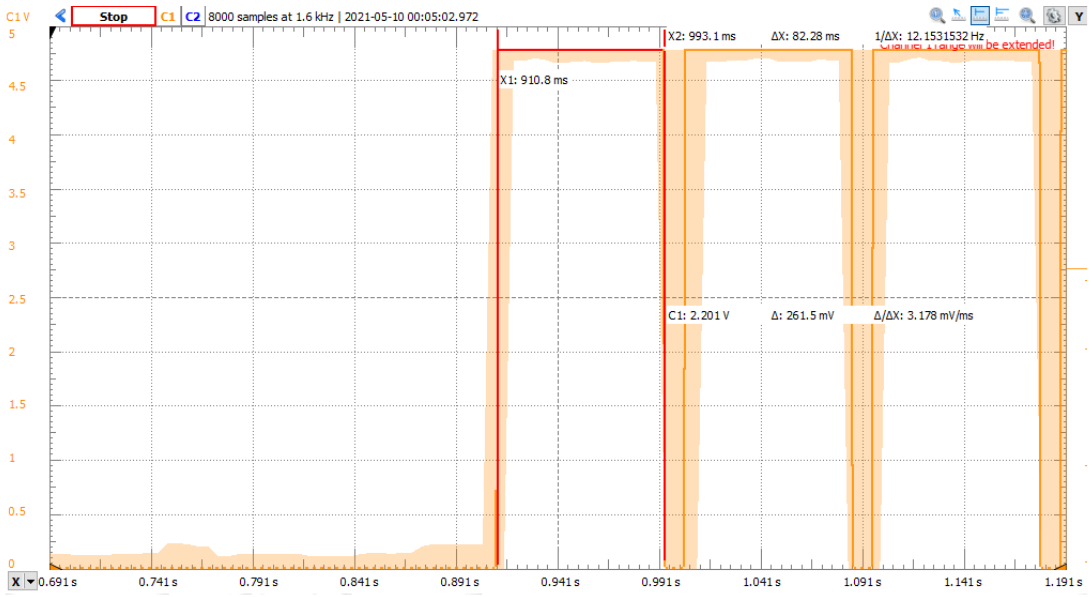


Figure 5.4: Time to read the SRAM via NFC

Reading the SRAM through NFC takes $82.28ms$. In figure 5.4 the SRAM is read three times in a row, each read takes about the same amount of time.

5.2.4 Time for one temperature measurement

```
1 PINS_DRV_WritePin(PTC, 15, 1);                          // set pin to high
2 v_uncomp_temp_u16 = bmp180_get_uncomp_temperature();
3 temperature = bmp180_get_temperature(v_uncomp_temp_u16);
4 PINS_DRV_WritePin(PTC, 15, 0);                          // set pin to low
```

Listing 5.1: Measurement of temperature readout time

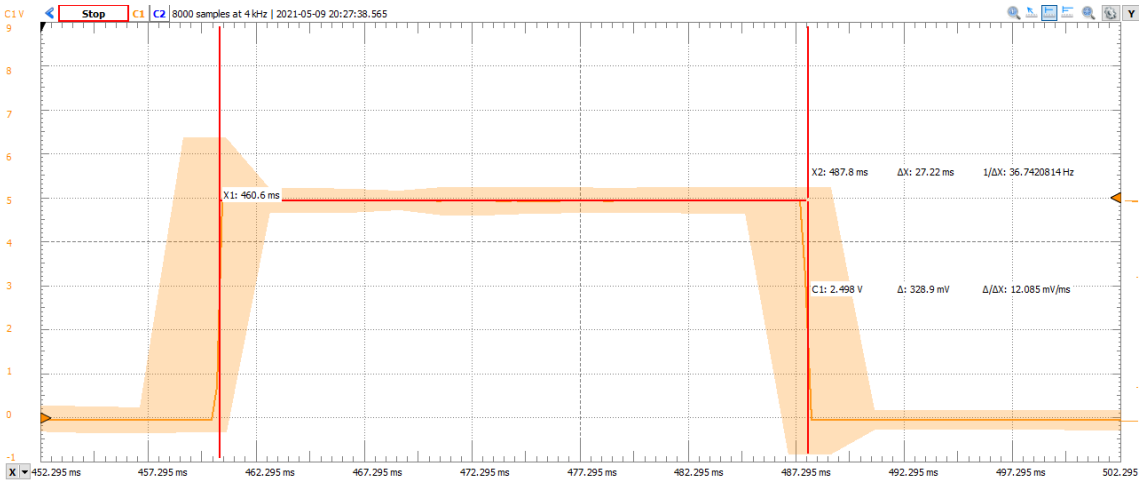


Figure 5.5: Measurement time BMP180

The code in Listing 5.1 shows the code executed to get the measurement in Figure A.3. In line 1, the pin that is measured on the oscilloscope image is turned on, and the measured value goes from 0V to 5V. The function in line 2 sends the appropriate I^2C commands to the BMP180 sensor to trigger the measurement. Afterwards it waits for the BMP180 to take the measurement. Subsequently. the measured values are read out from the BMP180 sensor.

In line 3 the temperature values read out by the sensor are compensated to get a real temperature. This compensation takes place purely on the microcontroller, there is no communication with the sensor. The execution of this function takes less than a millisecond and is barely significant because it hardly contributes to the overall measured time.

In line 4 the pin measured by the oscilloscope is set to low again.

5.3 Result Evaluation

In this chapter, the measured times are summarized and explained. The two prototypes developed in the course of this thesis are discussed and evaluated separately from each other again. First, the time measurements of the in vehicle use case are discussed in chapter 5.3.1. Afterwards those of the off vehicle use case are elaborated in chapter 5.3.2.

5.3.1 In Vehicle use case (Battery management)

In the In vehicle use case, the emphasis of the evaluation is on the NFC readout of the temperature sensor. Hence, the time required to initialize the temperature sensor and the NFC periphery is of interest, as well as the time required to take measurements with the temperature sensor. These times measured on the prototype are explained in more detail in chapter 5.3.1.1 The second interesting result would be how long the range of the NFC transmission actually is. This affects where the temperature sensors can be placed in relation to the cell board when the system is installed. These measurements will be discussed in chapter 5.3.1.2.

5.3.1.1 Timing

NTAG discover and init	BMP 180 measurement
$974.8ms$	$27.22ms$

Table 5.1: Timing of the main elements of the BMS functionality[1]

NTAG discover and init: The NTAG discover and init function includes finding the NTAGS from the NFC reader, the signature validation, which as can be seen in paragraph 5.2.2 takes 369.2ms. Activating the energy harvesting function and setting the corresponding operating mode, in this case the I^2C master mode, is also part of the NTAG boot process. In the end, the used temperature sensor, the BMP 180 is initialized.

NFC discovery	Signature	Energy harvesting	Init NTAG	Init BMP 180
$652.5 \pm 322.9ms$	$369.4 \pm 0.4ms$[1]	$21.4 \pm 1.6ms$	$29.7 \pm 0.6ms$	$118.8 \pm 0.6ms$

Table 5.2: Single timings of the NTAG discover and init function

[1] Measurement in chapter 5.2

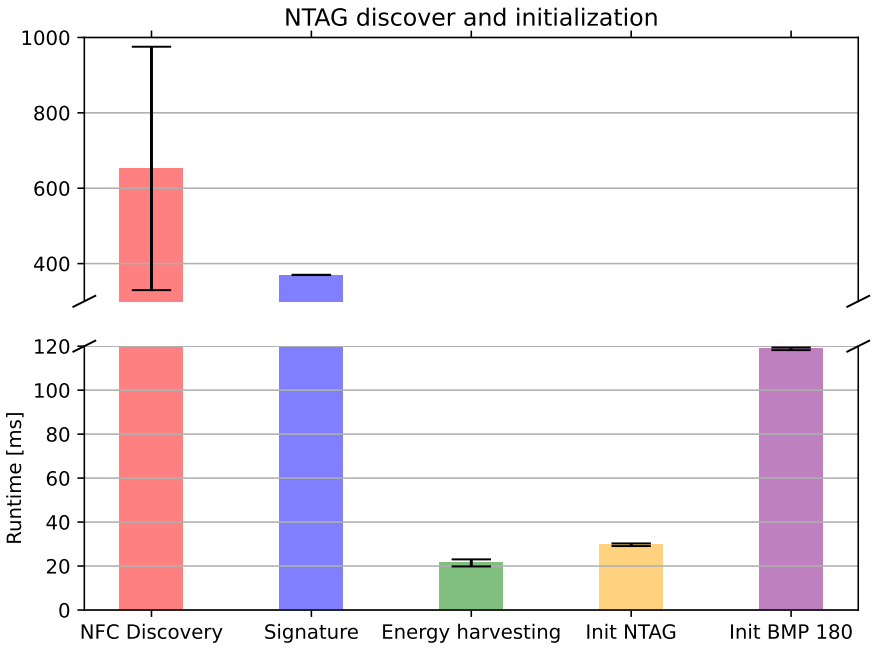


Figure 5.6: Timings of the NTAG discover and init sequence (values in Table 5.2)

BMP 180 measurement: The measurement from the BMP 180 temperature sensor includes triggering the measurement via the I^2C interface. After that it waits for 4.5 milliseconds [15]. This is how long the BMP 180 needs to measure the temperature. After that the measured temperature values are read out via the I^2C bus. Then the final temperature is calculated at the microcontroller from the raw data read out via the I^2C interface. The timing of the temperature measurement function is explained in more detail in paragraph 5.2.4.

5.3.1.2 Positioning (NFC connection)

The limiting factor in the NFC connection to the wireless sensor readout is the energy obtainable from the NFC field. The sensor does not have its own energy supply and is powered by the NFC field via energy harvesting. The energy harvesting feature of NTAG is designed to first check if the existing NFC field is strong enough to harvest the required energy. If the NFC field is not strong enough, or becomes too weak during operation, the NTAG is automatically reset. This check is the limiting factor for the NFC range in this setup. Energy harvesting can be activated up to a distance of 5.4 cm between the reader antenna and the antenna of the NTAG. If the distance between the antennas is greater, an NFC connection can be established, but the energy harvesting feature can no longer be activated. Thus, the function of the setup is no longer guaranteed.

5.3.2 Off Vehicle use case (NFC wakeup and readout)

In the off-vehicle use case, the attention is focused on the wireless readout of the temperature values via an external NFC reader. The NFC temperature sensor is reused in this use case as well. The same initialization and measurement times as explained in chapter 5.3.1.1 are valid here. Of primary interest for this setup is the time required to transfer data via the SRAM to the external NFC reader. These times are visualized and discussed in chapter 5.3.2.1. Subsequently, the transmission rate to achieved between the Battery Cell Controller and the external reader device is examined in Chapter 5.3.2.2.

5.3.2.1 Timing

NTAG wakeup	BMP 180 measurement	NTAG I^2C SRAM	NTAG NFC SRAM
$3.1ms$ [20b]	$27.1 \pm 0.5ms$	$\sim 0.24ms$	$82.5 \pm 0.4ms$

Table 5.3: Timing of the main elements of the NFC wakeup and readout[2]

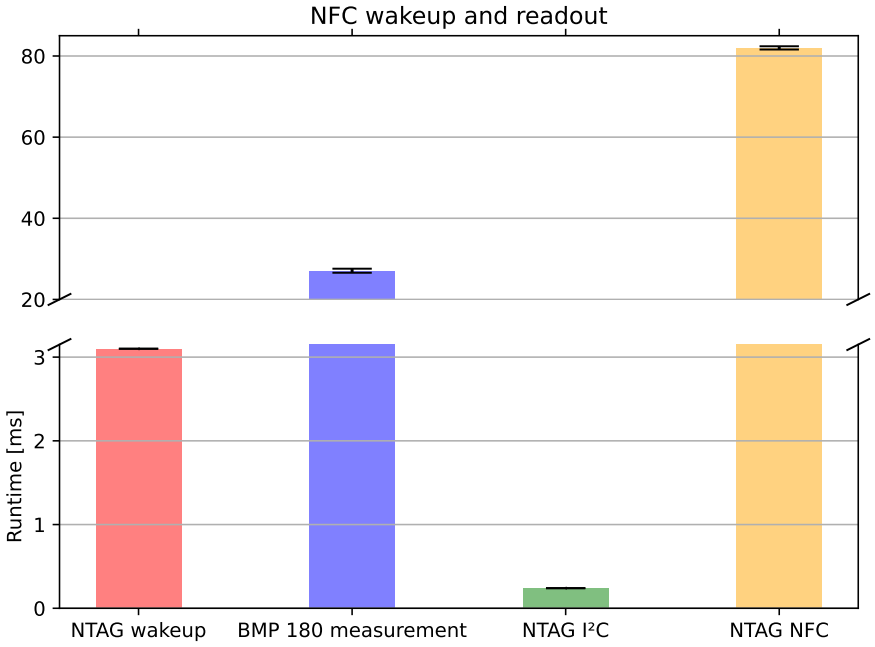


Figure 5.7: Visualisation of the timings[3](values in Table 5.3)

[2]Measurement in chapter 5.2
[3]NTAG wakeup time is taken from the data sheet

NTAG wakeup: There are 2 variants for the wakeup of the NTAG. The first one is the wakeup from standby mode. This is described in more detail in paragraph 4.3.1.1. The exact timing to wakeup the NTAG from the standby could not be obtained or measured.

The second implementation variant would be the wakeup of the NTAG from power down mode via the energy harvested from the NFC field. This is also the variant which is handled in table 5.3. It is described in more detail in paragraph 4.3.1.2. In this mode, the wakeup takes approximately $3.1ms$, according to the datasheet [20b].

NTAG I^2C SRAM: The transfer via I^2C to SRAM takes $0.24ms$. The measurement process is explained in chapter 5.2.3. This time is mainly limited by the I^2C bus itself. The I^2C bus to the NTAG runs in fast mode. This means that the bus is clocked at 400kHz.

NTAG NFC SRAM: The data transfer between NTAG and SRAM via the NFC reader takes $82.28ms$. In paragraph 5.2.3 the measurement is elaborated in more detail. The NFC reader is connected to the microcontroller via an SPI bus. The data is thus transmitted via the NFC interface, and then further to the microcontroller via the SPI bus.

5.3.2.2 Data throughput

Reading the SRAM via NFC takes $82.28ms$. Writing the SRAM via I^2C takes about $0.24ms$. It takes 25 milliseconds for the arbiter to unlock a locked interface automatically[4]. So that specifies the time passing before switching between I^2C and NFC. For switching from NFC to I^2C in this setup, there is no need to wait for the 25 milliseconds needed by the arbiter to switch. In the setup tested here, the pass-through mode intended for data transmission is used. In this mode the arbiter switches automatically when the last byte of the SRAM has been read.

Adding everything up, $107.52ms$ are passing for a complete transmission of the SRAM. Thus the amount of one completely written SRAM can be transferred in this time. This corresponds to 256 bytes, which results in a data throughput of 2 381,38 bytes/second, i.e. an approximate data throughput of 19 kbit/s.

The time required by the microcontroller to provide the data and by the NFC reader to process the data is negligible compared to the transmission time. Therefore, the required transmission time consists of the time required by the I^2C bus and the NFC connection and the switching time required by the arbiter.The transfer times for the I^2C bus and the NFC interface are directly proportional to the amount of transferred data. The arbiter has to switch once every 256 bytes of data, which adds an extra 25 milliseconds to the transmission time.

[4] According to own testing (Not from the datasheet)

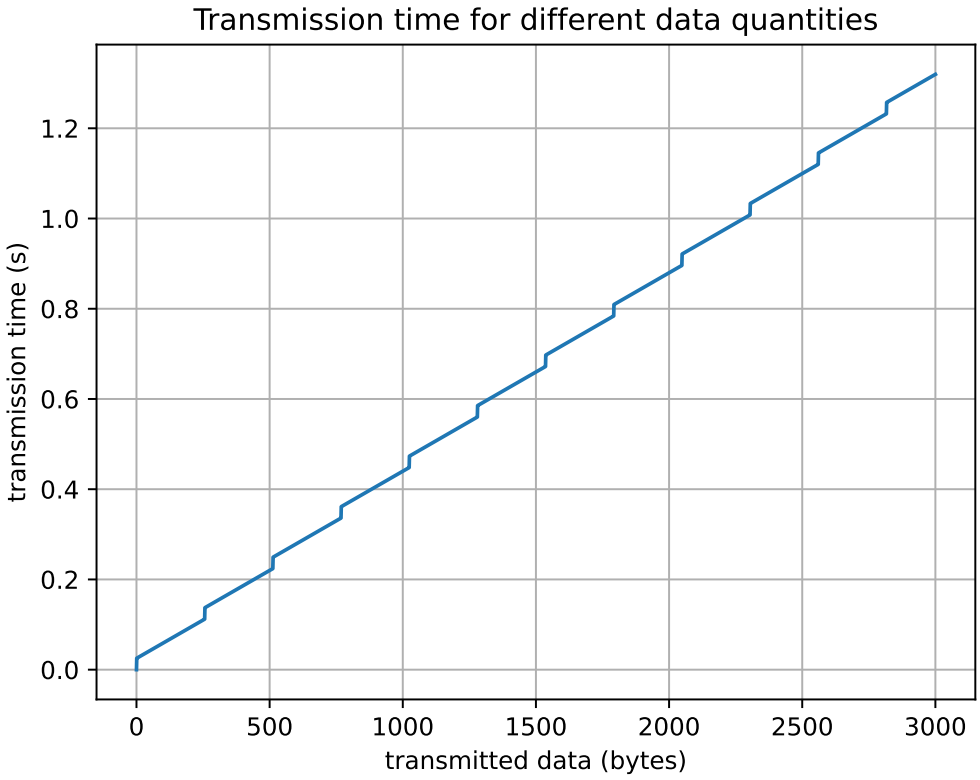


Figure 5.8: Transmission time for the different data quantities[1]

Figure 5.8 shows the time required for the transmission of a certain amount of data. The time required for the transmission is made up of the time required for the NFC transmission, the time required for the I^2C transmission and the switching time required by the arbiter. Since the SRAM has a maximum size of 256 bytes, the arbiter must switch over every 256 bytes. If less bytes than the 256 bytes are transferred, the arbiter must still switch over, since the data written to the SRAM via I^2C must be read out via NFC. This switching, which must happen at least every 256 bytes, causes the non-linearity shown in plot 5.8.

[4]Not all data shown in this diagram is acquired via measurements - Measured values were mathematically interpolated and processed

Chapter 6

Conclusion and further Work

6.1 Conclusion

In this thesis, the security implications and the use of NFC in BMS in the automotive context are investigated. It is shown that the use of NFC in the in-vehicle environment is very difficult. On the one hand, it is very difficult to align the NFC antennas appropriately in the confined environment inside a vehicle. On the other hand, there are a lot of metal components, which makes communication via NFC almost impossible. Nevertheless, the implemented NFC sensor interface could be used. This interface would allow integrating pressure sensors inside the battery cells, which would not be possible without NFC technology. The pressure sensors could be powered via energy harvesting, and the distance that the NFC interface would have to bridge could be kept small enough to allow reliable communication despite the harsh conditions inside the vehicle.

The second aspect evaluated in this thesis are security precautions in automotive BMS. In the course of the security evaluation carried out in this thesis, some security risks were determined. Currently, Battery Management Systems are only protected against external attacks at the interface. This mainly includes the interface with the gateway and thus to the cloud, and the interface between the BMS and the charger, which is part of current security evaluations. However, there is also the risk that BMS implementations are optimized for higher car performance. Squeezing more power out of the cells inevitably leads to an operation outside the Safe Operating Area of the cells. In the worst case, this can cause thermal runaway, which would have devastating effects on safety and would at least leave a nifty mark on reputation of the manufacturer. Current BMS reference designs are not protected against such attacks. This work suggests how such battery optimizations can be prevented and subsequent accidents inhibited.

6.2 Further work

Like the rest of the thesis, the further work can be divided too into further work for the in vehicle use case, and further work for the off vehicle use case. For the in vehicle use case, the NFC range problem for the NFC sensor readout would need to be evaluated, which is explained in more detail in paragraph 6.2.1. The use of pressure sensors, to forecast safety critical operation conditions more reliably is discussed in section 6.2.2. Also, the interface between Battery Pack Controller and Battery Cell Controller is not yet sufficiently secured, which is further explained in paragraph 6.2.3. How the off vehicle use case can be continued and possibly expanded with a functionality for software updates is discussed in section 6.2.4.

6.2.1 NFC antenna orientation

A major problem with the NFC sensor readout of the current prototype is that the NFC antennas are orthogonal to each other in the installed setup. In addition to this, there is a lot of metal inside the vehicle battery, which further limits the NFC range. A possible solution would be to evaluate the use of the active load modulation of the NTAG boost series. Currently, the temperature sensor is powered via energy harvesting. The NTAG boost series does not offer this possibility, but for this particular use case the battery cells themselves could be used for powering the NTAG.

6.2.2 NFC pressure sensor

If the NFC sensor interface cannot be used for the connection of temperature sensors due to the difficult conditions inside the vehicle, the use of this interface for pressure sensors should be evaluated. By means of NFC, pressure sensors can be installed inside the cells, informing the Battery Management System much earlier about a critical condition. By the time the measured temperature rises into a range where the BMS takes safety precautions, it is often too late and the battery gets into a state outside the Safe Operating Area. If there is the possibility to measure the pressure inside the cells in addition to the cell temperature, this would be a huge advantage. The additional measurement not only creates redundancy of the measured values, but also safety precautions of the BMS can be taken much earlier, and thus the safety of the overall system is increased.

6.2.3 Unsecure connection

The demonstrator presented in the paper is based on the reference design of the distributed Battery Management System architecture presented by NXP. In this prototype, no security precautions are provided in the data connection between Battery Cell Controller and Battery Pack Controller. However, the security analysis revealed that there is a risk that the battery pack could be boosted by manipulating the measurement data transmitted on this interface. In the Data Flow Diagram in Figure 3.3, the corresponding data connection with the associated threats is shown graphically. The possibilities to secure this connection would be either to encrypt the transmitted data or at least to sign it. The second possibility would be to perform a plausibility check of the received temperature and voltage data at the Battery Pack Controller. The cell voltage could be checked for plausibility using a battery model, and the cell temperature could be checked for plausibility using a thermal model of the battery pack. For both plausibility checks, a measurement of the charge or discharge current at the Battery Pack Controller would be sufficient.

6.2.4 AES mutual authentication

For the NFC error log readout, a possible extension would be to introduce mutual authentication on the NFC interface between NTAG and the external NFC reader. In this prototype, the NFC interface was implemented as a read-only interface to avoid security vulnerabilities. An AES mutual authentication would allow to open the interface in both directions. This would lead to completely new possibilities like for example to integrate a firmware update functionality via this interface.

Bibliography

[15] *BMP180 Data sheet*. BST-BMP180-DS000-12. Rev. 2.8. Bosch Sensortec. July 2015. URL: https://datasheetspdf.com/pdf-file/770150/Bosch/BMP180/1.

[20a] *Datasheet: NTAG 5 link - NFC Forum-compliant I²C bridge*. 544533. Rev. 3.3. NXP Semiconductors. July 2020. URL: https://www.nxp.com/docs/en/data-sheet/NTP53x2.pdf.

[20b] *NTAG 5 - How to use energy harvesting*. 530412. Rev. 1.2. NXP Semiconductors. May 2020. URL: https://www.nxp.com/docs/en/application-note/AN12365.pdf.

[20c] *S32K1xx Data Sheet*. S32K1XX. Rev. 13. NXP Semiconductors. Apr. 2020. URL: https://www.nxp.com/docs/en/data-sheet/S32K-DS.pdf.

[A+] Ishtiaq Rouf A et al. *Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study*.

[Ali+19] Muhammad Ali et al. "Towards a Smarter Battery Management System for Electric Vehicle Applications: A Critical Review of Lithium-Ion Battery State of Charge Estimation". In: *Energies* 12 (Jan. 2019), p. 446. DOI: 10.3390/en12030446.

[AMA12] Ciprian Antaloae, James Marco, and Francis Assadian. "A Novel Method for the Parameterization of a Li-Ion Cell Model for EV/HEV Control Applications". In: *IEEE Transactions on Vehicular Technology* 61.9 (2012), pp. 3881–3892. DOI: 10.1109/TVT.2012.2212474.

[AMK20] Bander A. Alzahrani, Khalid Mahmood, and Saru Kumari. "Lightweight Authentication Protocol for NFC Based Anti-Counterfeiting System in IoT Infrastructure". In: *IEEE Access* 8 (2020), pp. 76357–76367. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2989305.

[And10] D. Andrea. *Battery Management Systems for Large Lithium-ion Battery Packs*. EBL-Schweitzer. Artech House, 2010. ISBN: 9781608071043. URL: https://books.google.at/books?id=nivOtAEACAAJ.

[ASS17] Md Tanvir Arafin, Andrew Stanley, and Praveen Sharma. "Hardware-based anti-counterfeiting techniques for safeguarding supply chain integrity". In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2017, pp. 1–4. DOI: 10.1109/ISCAS.2017.8050605.

[Ber01] H. Bergveld. "Battery management systems : design by modelling". In: *International Journal of Chemical Reactor Engineering - INT J CHEM REACT ENG* (Jan. 2001).

[Bha+05] B.S. Bhangu et al. "Nonlinear observers for predicting state-of-charge and state-of-health of lead-acid batteries for hybrid-electric vehicles". In: *IEEE Transactions on Vehicular Technology* 54.3 (2005), pp. 783–794. DOI: 10.1109/TVT.2004.842461.

[Bra+] M. Brandl et al. In: *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*.

[Che+12] Chin-Ling Chen et al. "An RFID Authentication and Anti-counterfeit Transaction Protocol". In: *2012 International Symposium on Computer, Consumer and Control*. 2012, pp. 419–422. DOI: 10.1109/IS3C.2012.112.

[Col+07] Martin Coleman et al. "State-of-Charge Determination From EMF Voltage Estimation: Using Impedance, Terminal Voltage, and Current for Lead-Acid and Lithium-Ion Batteries". In: *Industrial Electronics, IEEE Transactions on* 54 (Nov. 2007), pp. 2550–2557. DOI: 10.1109/TIE.2007.899926.

[Cor] 2005 Microsoft Corporation. *The STRIDE Threat Model*. www.msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx. Accessed: 2021-05-19.

[CSE08] Jian Cao, Nigel Schofield, and Ali Emadi. "Battery balancing methods: A comprehensive review". In: *2008 IEEE Vehicle Power and Propulsion Conference*. 2008, pp. 1–6. DOI: 10.1109/VPPC.2008.4677669.

[De +15] Giuseppe De Maso-Gentile et al. "Design of CAN to Bluetooth gateway for a Battery Management System". In: *2015 12th International Workshop on Intelligent Solutions in Embedded Systems (WISES)*. Oct. 2015, pp. 171–175.

[Dou12] Daniel H. Doughty. "Vehicle Battery Safety Roadmap Guidance". In: 2012.

[FW09] Benjamin Weyl Frederic Stumpf Christian Meves and Marko Wolf. "A Security Architecture for Multipurpose ECUs in Vehicles". In: *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*. 2009.

[GR09] Andre Groll and Christoph Ruland. "Secure and Authentic Communication on Existing In-Vehicle Networks". In: July 2009, pp. 1093–1097. DOI: 10.1109/IVS.2009.5164434.

[Har+18] Kevin Harnett et al. "DOE/DHS/DOT Volpe Technical Meeting on Electric Vehicle and Charging Station Cybersecurity Report". In: Mar. 2018, p. 44. DOI: DOT-VNTSC-DOE-18-01.

[Her+07] Shawn Hernan et al. "Uncover Security Design Flaws Using The STRIDE Approach". In: *Microsoft Journal of developers* (Mar. 2007).

[Hue98] F. Huet. "A review of impedance measurements for determination of the state-of-charge or state-of-health of secondary batteries". In: *Journal of Power Sources* 70.1 (1998), pp. 59–69. ISSN: 0378-7753. DOI: https://doi.org/10.1016/S0378-7753(97)02665-7. URL: https://www.sciencedirect.com/science/article/pii/S0378775397026657.

[Kha+19a] Asadullah Khalid et al. "FACTS Approach to Address Cybersecurity Issues in Electric Vehicle Battery Systems". In: *2019 IEEE Technology Engineering Management Conference (TEMSCON)*. 2019, pp. 1–6. DOI: 10.1109/TEMSCON.2019.8813669.

[Kha+19b] Asadullah Khalid et al. "FACTS Approach to Address Cybersecurity Issues in Electric Vehicle Battery Systems". In: *2019 IEEE Technology Engineering Management Conference (TEMSCON)*. June 2019, pp. 1–6. DOI: 10.1109/TEMSCON.2019.8813669.

[Kla+17] Bernd Klauer et al. "Wireless sensor/actuator device configuration by NFC with secure key exchange". In: *2017 IEEE AFRICON*. Sept. 2017, pp. 473–478. DOI: 10.1109/AFRCON.2017.8095528.

[KYK13] Tae Kim, Byeng Dong Youn, and Hyun Kim. "Battery Pack Temperature Estimation Model for EVs and Its Semi-transient Case Study". In: vol. 33. Jan. 2013, pp. 955–960. ISBN: 978-88-95608-24-2. DOI: 10.3303/CET1333160.

[Law] Barrie Lawson. *Battery and Energy Technologies Electropaedia*. www.mpoweruk.com. Accessed: 2021-05-05.

[Lee+13] Minkyu Lee et al. "Wireless battery management system". In: *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*. 2013, pp. 1–5. DOI: 10.1109/EVS.2013.6914889.

[Mad19] Richard Stocker Madeline Cheah. "Cybersecurity of Battery Management Systems". In: *State of the Internet Security Report*. 2019, pp. 82–89.

[Mar14] José Miguel Branco Marques. *Battery Management System (BMS) for Lithium-Ion Batteries*. 2014.

[Mic02] David LeBlanc Michael Howard. *Writing Secure Code*. 2nd edition Microsoft Press. 2002.

[MPJ12] Shema Ann Mathew, R. Prakash, and Philip C. John. "A smart wireless battery monitoring system for Electric Vehicles". In: *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*. 2012, pp. 189–193. DOI: 10.1109/ISDA.2012.6416535.

[MR05] Eberhard Meissner and Gerolf Richter. "The Challenge to the Automotive Battery Industry: The battery Has to Become an Increasingly Integrated Component within the Vehicle Electric Power System". In: *Journal of Power Sources* 144 (June 2005), pp. 438–460. DOI: 10.1016/j.jpowsour.2004.10.031.

[Ng+09] Kong Soon Ng et al. "Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries". In: *Applied Energy* 86.9 (2009), pp. 1506–1511. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2008.11.021. URL: https://www.sciencedirect.com/science/article/pii/S0306261908003061.

[NXP08] NXP. "Thermal Analysis of Semiconductor Systems". In: *freescale semiconductor* (2008), p. 24. DOI: BASICTHERMALWP/REV0.

[Par+16] Kaveh Paridari et al. "Cyber-Physical-Security Framework for Building Energy Management System". In: *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*. 2016, pp. 1–9. DOI: 10.1109/ICCPS.2016.7479072.

[Pat+08] Bharath Pattipati et al. "Automotive battery management systems". In: Oct. 2008, pp. 581–586. DOI: 10.1109/AUTEST.2008.4662684.

[PJ03] Chanwoo Park and Arun Jaura. "Dynamic Thermal Model of Li-Ion Battery for Predictive Behavior in Hybrid and Fuel Cell Vehicles". In: *SAE Transactions* 112 (June 2003), pp. 1835–1842. DOI: 10.4271/2003-01-2286.

[POO19] Yongwan Park, Omer C. Onar, and Burak Ozpineci. "Potential Cybersecurity Issues of Fast Charging Stations with Quantitative Severity Analysis". In: *2019 IEEE CyberPELS (CyberPELS)*. 2019, pp. 1–7. DOI: 10.1109/CyberPELS.2019.8925069.

[PPJ01] Sabine Piller, Marion Perrin, and Andreas Jossen. "Methods for state-of-charge determination and their applications, journal = Journal of Power Sources". In: 96.1 (2001). Proceedings of the 22nd International Power Sources Symposium, pp. 113–120. ISSN: 0378-7753. DOI: https://doi.org/10.1016/S0378-7753(01)00560-2. URL: https://www.sciencedirect.com/science/article/pii/S0378775301005602.

[RRR17] Ataur Rahman, Md Rahman, and Mahbub Rashid. "Wireless Battery Management System of Electric Transport". In: *IOP Conference Series: Materials Science and Engineering* 260 (Nov. 2017), p. 012029. DOI: 10.1088/1757-899X/260/1/012029.

[Rud+09] Alastair Ruddle et al. "Security requirements for automotive on-board networks based on dark-side scenarios. Deliverable D2.3: EVITA. E-safety vehicle intrusion protected applications". In: *Fraunhofer ISI* (Jan. 2009).

[Sag+13] Florian Sagstetter et al. "Security Challenges in Automotive Hardware/Software Architecture Design". In: Jan. 2013, pp. 458–463. ISBN: 978-1-4673-5071-6. DOI: 10.7873/DATE.2013.102.

[Sch+12] Matthias Schneider et al. "Automotive battery monitoring by wireless cell sensors". In: *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. 2012, pp. 816–820. DOI: 10.1109/I2MTC.2012.6229439.

[She+15] Cody Shell et al. "Implementation of a wireless battery management system (WBMS)". In: *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*. 2015, pp. 1954–1959. DOI: 10.1109/I2MTC.2015.7151581.

[Sim19] Balázs Simacsek. "Can we trust our cars?" In: *NXP* (2019).

[Soj14] Richard Soja. "Automotive Security: From Standards to Implementation". In: Jan. 2014, p. 18.

[SRB03] B. Schweighofer, K.M. Raab, and G. Brasseur. "Modeling of high power automotive batteries by the use of an automated test system". In: *IEEE Transactions on Instrumentation and Measurement* 52.4 (2003), pp. 1087–1091. DOI: 10.1109/TIM.2003.814827.

[Ste+14] Sebastian Steinhorst et al. "Smart Cells for Embedded Battery Management". In: *2014 IEEE International Conference on Cyber-Physical Systems, Networks, and Applications*. 2014, pp. 59–64. DOI: 10.1109/CPSNA.2014.22.

[Stu+18] Frederic Stumpf et al. "A Security Architecture for Multipurpose ECUs in Vehicles". In: 2018.

[TMT11] Y. K. Tan, J. C. Mao, and K. J. Tseng. "Modelling of battery temperature effect on electrical characteristics of Li-ion battery in hybrid electric vehicle". In: *2011 IEEE Ninth International Conference on Power Electronics and Drive Systems*. 2011, pp. 637–642.

[V02] Johnson V. "Battery performance models in ADVISOR". In: *Journal of Power Sources - J POWER SOURCES* 110 (Aug. 2002), pp. 321–329. DOI: 10.1016/S0378-7753(02)00194-5.

[Vel+17] Ricardo Velho et al. "Management System for Large Li-Ion Battery Packs with a New Adaptive Multistage Charging Method". In: *Energies* 10 (May 2017), p. 605. DOI: 10.3390/en10050605.

[Vem19] Prashanth Vemireddy. "BMS Application Programming Interface". In: *Everlasting* (Aug. 2019), p. 27.

[VLA01] Johnson V.H., Pesaran A.A. (National Renewable Energy Laboratory), and Sack T. (Saft America). "Temperature-Dependent Battery Models for High-Power Lithium-Ion Batteries". In: Jan. 2001. DOI: 10.3303/CET1333160.

[Vog20] Taylor Vogt. "Wired vs. Wireless Communications in EV Battery Management". In: *TI Power* (Oct. 2020), p. 6.

[Zim17] Zilberman Zimmermann Sturm. "Requirements and architecture concept of a highly modular prototyping hardware platform". In: *Everlasting* (Feb. 2017), p. 20.

Acronyms

AES Advanced Encryption Standard. 26, 41, 43, 73

API Application Programming Interface. 47, 50, 57

ARM Advanced RISC Machines. 46

BCC Battery Cell Controller. 15, 18, 19, 25, 26, 29, 32, 48, 49, 68, 72, 73

BIOS Basic Input Output System. 22

BMS Battery Management System. 3, 10–13, 15–20, 22, 23, 25–27, 29–33, 35, 36, 38, 39, 43, 48, 66, 71–73, 83

BPC Battery Pack Controller. 15, 25, 26, 30, 32, 38, 39, 42, 48, 49, 72, 73

CAN Controller Area Network. 15

CID Circuit Interrupt Device. 23

CMU Cell Monitoring Unit. 15, 25

DFD Data Flow Diagram. 30–32, 38, 40, 41, 73

DOD Depth Of Discharge. 19, 20

DREAD Damage, Reproducibility, Exploitability, Affected users, Discoverability. 30

ECC Elliptic Curve Cryptography. 47

ECDSA Elliptic Curve Digital Signature Algorithm. 56

ECU Electronic Control Unit. 15, 26

ED Event Detection. 45, 59

EMI Electro Magnetic Interference. 19

EMS Energy Management System. 25

EV Electric Vehicles. 25

EVITA Electric Vehicle Infrastructure Transportation Alliance. 30

FTM Flexible Timer Module. 47, 51

GPIO General Purpose Input Output. 59

GUI Graphical User Interface. 47

HAL Hardware Abstraction Layer. 47

I²C Inter Integrated Circuit. 29, 45, 47, 50, 51, 53, 54, 57, 58, 60, 63–70

IDE Integrated Development Environment. 46, 47

ISR Interrupt Service Routine. 59

M Million. 61

MCU Micro Controller Unit. 7, 15, 32, 38, 42, 43, 45, 50, 53, 55, 57–61, 83

MMU Module Management Unit. 15

NFC Near Field Communication. 1–3, 10, 18, 26, 28–32, 38, 40, 41, 43–45, 47, 49, 50, 52–60, 62, 64, 66–73

NTC Negative Temperature Coefficient. 12, 51

PCB Printed Circuit Board. 15, 16

PUF Physically Unclonable Functions. 26

RFID Radio Frequency Identification. 24–26

RUL Remaining Useful Life. 20

SBC Smart Battery Charger. 22

SBD Smart Battery Data. 22

SBS Smart Battery System. 22

SDK Software Development Kit. 47, 49–51

SMB System Management Bus. 22

SOA Safe Operating Area. 12, 13, 20, 22, 23, 39, 71, 72

SOC State Of Charge. 12, 19, 20

SOH State Of Health. 10, 12, 20, 25, 28, 41

SPI Serial Peripheral Interface. 29, 47–50, 69

SRAM Static Random Access Memory. 45, 53, 54, 57–60, 63, 64, 68–70

STRIDE Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. 30, 31, 33

UID Unique Identifier. 26, 30, 32, 56, 57

VLPS Very Low Power Stop. 44

WBMS Wireless Battery Management System. 25, 26

Appendices

Appendix A

Thermal model

In a BMS, a variety of sensors such as temperature sensors are employed. Those temperature sensors are spread in a kind of sensor network among the battery cells and transmit the measured values to the main MCU, which is responsible for the evaluation of the collected values.

Particularly in a wireless version of the BMS, those sensor values might be modified during the transmission. A wrongly reported temperature value might also be caused by a broken sensor. This would severely reduce the safety of the overall system, since the control signals of the MCU are based partly on the reported temperature values.

In many BMSs the current flow is determined via a shunt resistor. The current measured correlates with the power consumed by the battery pack and therefore also with the measured temperature in the vicinity of the cell. Via this channel it is possible to verify the measured temperature via a plausibility check. A thermal model can be employed to estimate the prevalent temperature at the sensor with the overall pack charging/discharging current as an input. This approximated temperature is subsequently compared with the values reported by the sensor. If the values reported by the sensor are deviating too much from the calculated temperatures, the plausibility check fails and the BMS has the option to react appropriately to the recognized error.

A.1 Thermal model

In the following paragraphs a battery model is described, which approximates and mathematically describes the thermodynamic processes inside the battery. There are battery models, which were created from an electrical viewpoint, and models which were developed from a chemical viewpoint. The battery model described in the following deals exclusively with the thermodynamic processes inside the battery. This thermal model is based on the basic thermal principles explained in paper [NXP08].

There are 3 types of heat transfer: convection, conduction and radiation. The thermal model described below considers only the thermal conduction. Convection and radiation are neglected. The battery model discussed in the following should differentiate itself

from the existing battery models in respect to providing sufficient accuracy for a plausibility check while being computable on restricted available computing resources like a microcontroller.

A.2 Assumptions

In the model described, some assumptions are made to simplify the formulas applied. By utilizing those measures the computational load on the microcontroller should be reduced.

- Internal resistance of the cells is assumed to be constant
- The total dissipated power is assumed to be converted into heat at the core of the battery cell
- Thermal convection and thermal radiation are neglectable

The internal resistance is assumed to be constant. In reality this is not the case, the internal resistance strongly depends on external parameters such as the current currently flowing through the cell, the current cell temperature, the state of charge and the number of cycles the cell has already gone through during its life [And10].

The present internal resistance of the cell can be determined from the cell voltage and the current currently flowing through the cell. Alternatively, the inner resistance of the battery cells can be adjusted via a lookup table using empirical values.

A.3 Thermal model

$$Q_{dis} = I^2 \cdot R_x \tag{A.1}$$

Q_{dis} ..... heat flux $[W]$
I ..... charging and discharging current $[A]$
R_x ..... inner resistance of the battery $R_x = f\left(SOC, Temperature, I, V\right)$ $[\Omega]$

During the charge and discharge process of a cell heat is generated, which has to be dissipated via the cell envelope into the environment of the cell. The heat generated inside the cell can be approximated via equation A.1.

Equation A.1 is used under the premise of a constant internal resistance. In paper [TMT11] this formula was used under the same preconditions for the estimation of the thermal losses during the charging and discharging process.

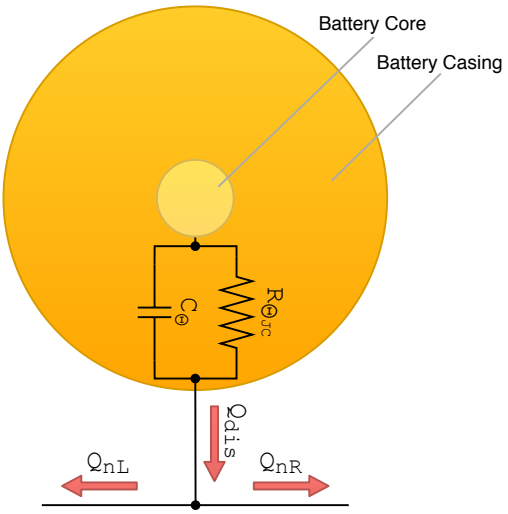


Figure A.1: Thermal model of a single battery Cell

$$Q_n = \left(Q_{dis} - Q_{n_{(start)}}\right) \cdot \left(1 - e^{-(t/\tau)}\right) + Q_{n_{(start)}} \tag{A.2}$$

Q_{dis} ..... temperature difference between sensor and ambient $[W]$
$Q_{dis_{(start)}}$ ..... temperature difference between sensor and ambient at t=0 $[W]$
t ..... time $[s]$
P_{dis} ..... dissipated power during charging and discharging $[W]$
τ ..... the time constant ($\tau = R_{\Theta_{JC}} \cdot C_\Theta$) $[s]$
$R_{\Theta_{JC}}$ ..... thermal conductivity $[K/W]$
C_Θ ..... thermal capacity $[J/K]$

Every substance can store a certain amount of heat, something which is also the case with battery cells. The heat storage capacity of the battery cell, also known as the thermal mass is modeled via C_Θ in Figure A.1.The insulating effect of the cell material is considered in this temperature model via the value $R_{\Theta_{JC}}$.

The capability to store heat inside the cell envelope results in a nonlinear step response to a temperature change within the cell. This nonlinear step response is modeled via formula A.2.

$$\begin{aligned} Q_{nL} &= Q_n \cdot \frac{k-n}{k} \\ Q_{nR} &= Q_n \cdot \frac{n}{k} \end{aligned} \tag{A.3}$$

Q_n ..... heat flux of the n^{th} cell $[W]$
n ..... number of the cell [0 to k-1] $[W]$
k ..... overall number of cells in the cell array $[W]$

As depicted in figure A.1, the heat flux emitted by the battery is disseminated in the environment. The temperature model presented here contains 2 possible temperature

paths, which are shown in figure A.1. With the help of equation A.3 it can be determined in which ratio the emitted heat flux is divided between these 2 possible temperature paths.

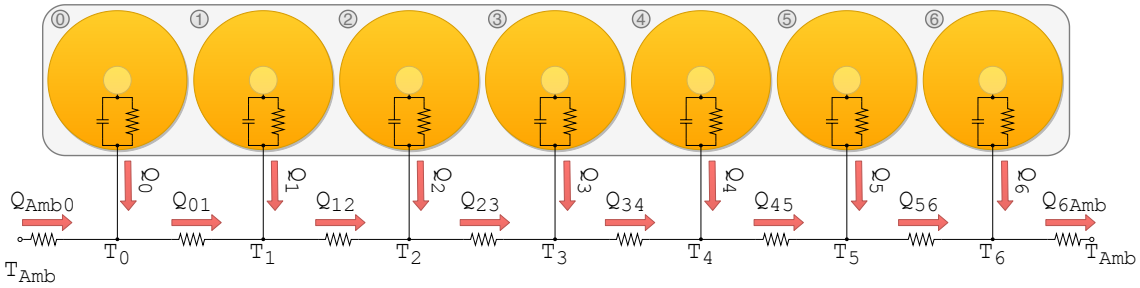


Figure A.2: Thermal model of multiple cells connected to a battery array

In figure A.2 the positioning of the individual cells and the thermodynamic interaction between those individual cells is shown.

$$\begin{aligned} Q_{01} &= Q_{1R} - Q_{2L} - Q_{3L} - Q_{4L} - Q_{5L} - Q_{6L} \\ Q_{12} &= Q_{1R} + Q_{2R} - Q_{3L} - Q_{4L} - Q_{5L} - Q_{6L} \\ Q_{23} &= Q_{1R} + Q_{2R} + Q_{3R} - Q_{4L} - Q_{5L} - Q_{6L} \\ Q_{34} &= Q_{1R} + Q_{2R} + Q_{3R} + Q_{4R} - Q_{5L} - Q_{6L} \\ Q_{45} &= Q_{1R} + Q_{2R} + Q_{3R} + Q_{4R} + Q_{5R} - Q_{6L} \\ Q_{56} &= Q_{1R} + Q_{2R} + Q_{3R} + Q_{4R} + Q_{5R} + Q_{6R} \end{aligned} \tag{A.4}$$

$$\begin{aligned} Q_{Amb0} &= Q_{01} - Q_0 \\ Q_{6Amb} &= Q_{56} + Q_6 \end{aligned} \tag{A.5}$$

Q ..... Heat flux between adjacent cells $[W]$

$$T_y - T_x = Q_{xy} \cdot R \tag{A.6}$$

T ..... Temperature of the cell envelope $[K]$
Q ..... Heat flux between two adjacent cells $[W]$
R ..... Thermal resistance between two adjacent cells $[K/W]$

A.4 Reference model

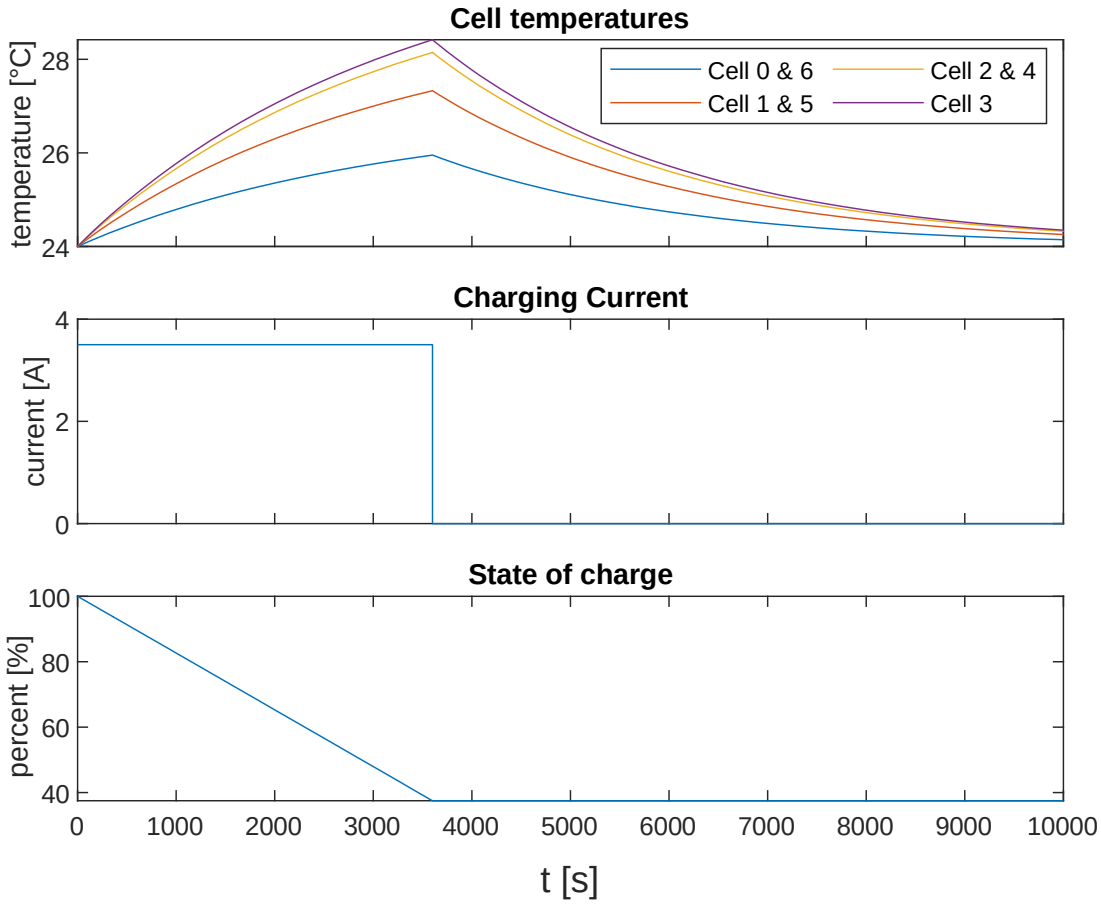


Figure A.3: Transient cell temperatures of the Simulink reference model

Since testing with real battery cells poses a safety risk and is hence difficult, a battery model from the Simulink library was used to estimate the thermal behavior of the battery during the charging and discharging process. The thermal behavior of the battery cells estimated by this model is shown in figure A.3. Figure A.4 compares the temperature behavior simulated in Simulik- with the temperature behavior estimated by the formulas presented in chapter A.3.

A.5 Results

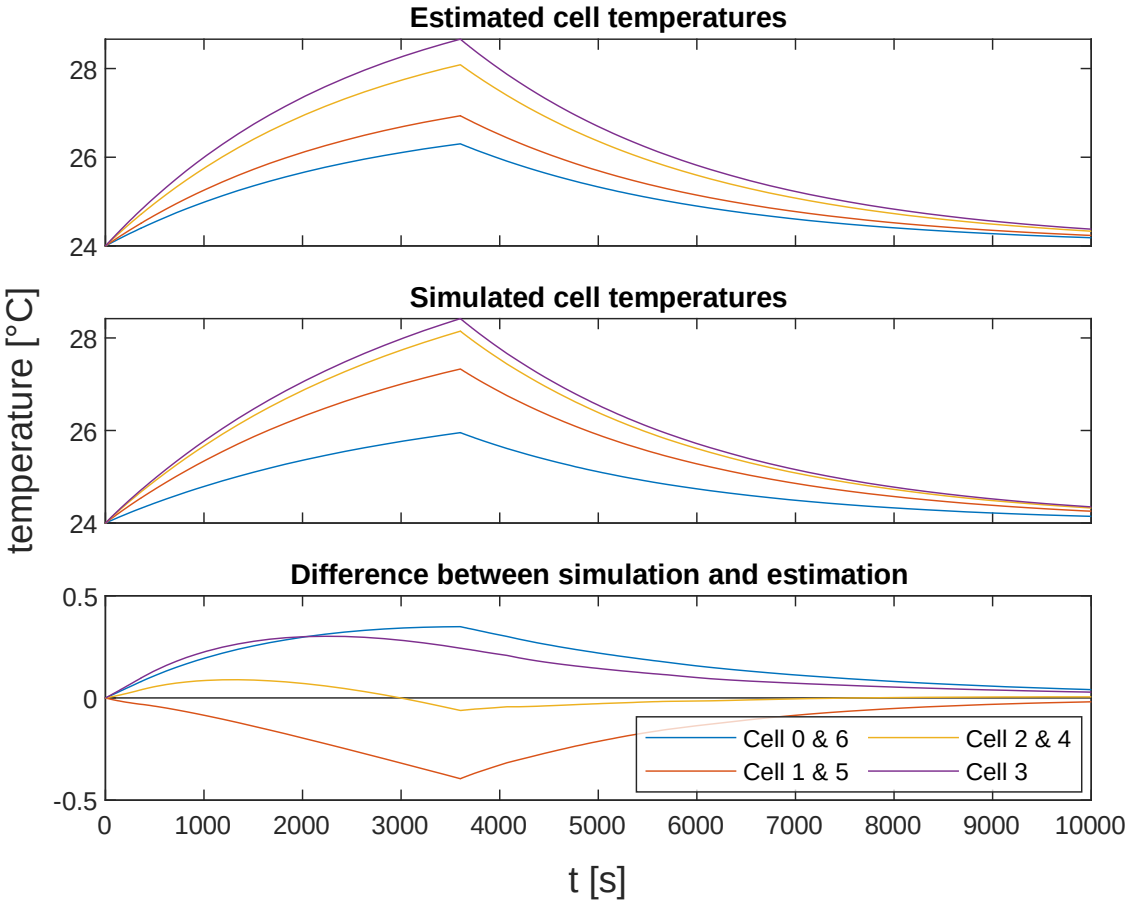
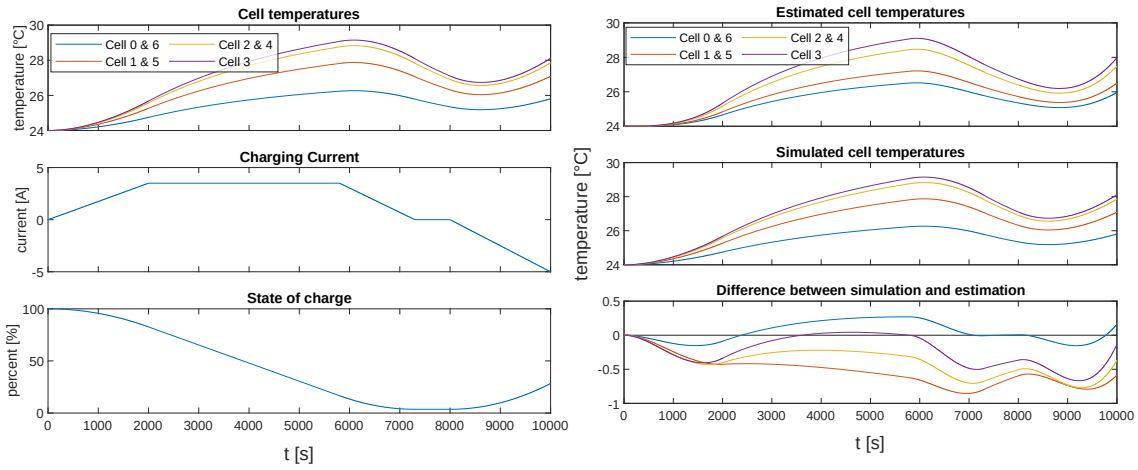


Figure A.4: Comparison between simulated and estimated cell temperatures

Figure A.4 compares the Simulink simulation battery model with the battery model presented in chapter A.3. The inner resistance and the thermal mass of the Matlab model were adjusted to fit the Simulink battery model used in the test. The remaining parameters were left at the same values for the estimated model as for the Simulink model.



(a) Simulated by the Simulink model

(b) Estimated vs Simulated results

Figure A.5: Estimation of the thermal cell behaviour during the Charging Process

In figure A.5 the results were tested with a more complicated charge current curve. All simulation parameters were identical to those of the simulation with the constant discharge current depicted in figure A.3 and A.4.