N°PFE : GE3-Speciality_year

Electrical Engineering Department

FINAL STUDY PROJECT REPORT

Presented with a view to obtaining the

National Engineer Diploma in Electrical Engineering

Speciality: electrical system

By Mr: ABDELLAWI ACHRAF

Bringing Clarity to Electric Mobility: Diagnostics Implementation and problems Rectification in Electric Vehicles

Produced within BAKO MOTORS Company

Defended on, 28/06/2024, before the jury composed of:

President:	**Mrs.** SAIDI IMEN
Rapporteur:	**Mrs.** HADOUK AMIRA
University supervisor:	**Mr.** BACHA FAWZI
Industrial supervisor:	**Mr.** TRIKI HAMZA

Academic Year : 2023 - 2024

Acknowledgments

The success and final outcome of this project required a lot of guidance and assistance from many people, and I am extremely fortunate to have got this all throughout the completion of my project work.

I am deeply grateful to my professional supervisor Mr. **Hamza TRIKI** for his enthusiasm, patience and insightful comments that made me dive deeply into the professional life and enjoy my work. Know that your LOVE for what you are doing was my source of inspiration during my internship.

I owe my profound gratitude to Mr. **FAWZI BACHA**, my Academic supervisor, who took keen interest on my project and guided me all along not only with her knowledge but also with his encouragement and support. Thank you for being always available for help.

I am thankful to all of Bako team members and Collaborators and The Higher Institute of Computer Science staff who gave me extensive personal and professional guidance which helped me in successfully completing my project.

In the end, This project could not have been accomplished without the insights of **ME** who believed in me through ups and downs. You made this journey fruitful enough to be the start of a successful career.

The Learning NEVER Stops

Contents

General Introduction **1**

1 Company Presentation and Project Context **2**

Introduction 3

1.1 General context of the project 3

1.2 Description of the company 3

1.2.1 Host company: Bako Motors SAS 3

1.2.2 Partners 4

1.2.3 Departments 5

1.3 The BAKO Motors vehicles 10

1.3.1 The B1 vehicle 10

1.3.2 The BEE vehicle 12

1.3.3 The EVAN vehicle 14

1.4 Project context 17

1.4.1 Problematic 17

1.4.2 Project objectives 18

1.4.3 Project planning 18

1.5 design methodology 19

1.5.1 Choosing the Agile Methodology 19

1.5.2 SCRUM Methodology 20

1.5.3 Product Backlog 21

Conclusion 21

2 Communication Technologies and Protocols **22**

Introduction 23

2.1 Introduction to the Onboard Charging System in Electric Vehicles 23

2.1.1 Functionality of the Onboard Charging System 23

2.1.2 Components of the Onboard Charging System 23

2.2 Plug Standardizing with IEC 61851-1 24

2.2.1 Overview of IEC 61851-1 24
2.2.2 Plug Types and Their Standards 24
2.2.3 Communication and Safety Protocols 25
2.2.4 Plug Types and Their Standards 25
2.3 CAN bus (Controller area network) 26
2.3.1 Electronic Control Units (ECUs) 26
2.3.2 CAN frames 27
2.3.3 CAN characteristics 28
2.4 Battery management system (BMS) controller 30
2.4.1 Battery management system's architecture 30
2.4.2 Battery management systems functionality 31
2.4.3 The BMS frame 33
2.5 Diagnostic systems 34
2.6 OBD-II (On-Board Diagnostics) 34
2.6.1 The OBD2 connector 35
2.6.2 Connection between CAN bus and OBD2 35
2.6.3 Integrating OBD II 36
Conclusion 36

3 Problem Identification 37

3.1 Introduction 38
3.2 Importance of SOC in Electric Vehicles 38
3.3 Display of SOC in the BEE Model 39
3.3.1 Discharge Characteristics: Lead-Acid vs. Lithium Batteries 39
3.3.2 Problem with SOC Display in the BEE and B1 Model 40
3.4 Display of SOC in the EVAN Model 41
3.4.1 SOC Display via CAN: 41
3.4.2 Communication Breakdown: 41
3.5 the Onboard Charger 41
3.5.1 Overview of Onboard Charger (OBC) in EVs 41
3.5.2 Relevant ISO Standards for OBC 42
3.5.3 Problem Identification 42

3.5.4 Performance Deficiencies . . . 42
3.5.5 Impact Analysis . . . 42
3.6 lack of Diagnostic System Bako Motors Vehicles . . . 43
3.6.1 Importance of a Diagnostic System: . . . 43
3.6.2 Introducing the Diagnostic System as a Proposed Solution . . . 43

4 Solution Design **44**
Introduction . . . 45
4.1 Display Solution Design . . . 45
4.1.1 architecture solution . . . 45
4.1.2 solution organizational chart . . . 46
4.2 Plug Standardizing . . . 47
4.2.1 Type 2 Connector (Mennekes) . . . 47
4.2.2 Specifications of Type 2 Plugs . . . 47
4.2.3 Rationale Behind Selection . . . 48
4.2.4 Pin Configuration and Communication Process . . . 49
4.2.5 Communication Protocols . . . 49
4.2.6 Communication with STM32 Microcontroller . . . 50
4.2.7 Architecture Diagram . . . 52
4.3 diagnostic systeme . . . 52
4.4 System functionality . . . 54
4.4.1 diagnostic architecture . . . 54
4.4.2 System organizational chart . . . 54
4.4.3 Perspective: User Interface Development . . . 55
4.5 project global archirtarchitecture . . . 55
4.5.1 Cloud Communication . . . 57
4.6 Hardware choice . . . 57
4.6.1 Micocontroller . . . 57
4.6.2 Tranceiver . . . 59
4.6.3 voltage regulator . . . 60
4.6.4 RELAY module . . . 61
4.6.5 USB module . . . 61

4.6.6 GSM module . . . 62
4.6.7 Detailed architecture . . . 63
4.7 supply system . . . 64
4.7.1 Power Requirements . . . 64
Conclusion . . . 65

5 Implementation and Test **66**
Introduction . . . 67
5.1 Software and hardware environment . . . 67
5.1.1 Hardware Components . . . 67
5.1.2 Software Tools . . . 68
5.2 Arduino emulation . . . 69
5.3 Embedded application development . . . 70
5.3.1 STM32 connection with BMS system . . . 71
5.3.2 STM32 connection with Serial Monitor . . . 75

General Conclusion **77**

Bibliography **79**

Appendix **80**
Appendix 1. . . . 80

List of Figures

1.1 Bako Motors manufacturing facility 4
1.2 Bako Motors' partners. 5
1.3 Bako Motors' design process 5
1.4 Bako Motors' design draft 6
1.5 Bako Motors poster example 7
1.6 Bako Motors' mold workshop 8
1.7 Bako Motors' welding workshop 8
1.8 Bako Motors' warehouse 9
1.9 Bako Motors' assembly line 9
1.10 Bako Motors' B1 model 10
1.11 Bako Motors' B1 configuration 10
1.12 Bako Motors' BEE model 12
1.13 Bako Motors' BEE configuration 13
1.14 Bako Motors' EVAN model 15
1.15 Bako Motors' BEE configuration 15
1.16 The project's Gantt chart 19

2.1 plug Type 2 25
2.2 CAN communication 26
2.3 CAN frame 27
2.4 CAN node 28
2.5 CAN high speed variation 29
2.6 CAN low speed variation 30
2.7 The BMS architecture 31
2.8 BMS monitoring process 32
2.9 BMS thermal management 32
2.10 BMS frame 33
2.11 Diagnostic system 34
2.12 the obd2-connector-pinout 35

2.13 The OBD port to tranceiver 36

3.1 Comparison of Discharge Curve: Lithium vs Lead-Acid Battery 39
3.2 B1 DISPLAY 40
3.3 Delayed SOC Drops in Electric Vehicles 40
3.4 EVAN DISPLAY 41

4.1 display solution architecture for BEE AND B1 45
4.2 display solution architecture for EVAN 46
4.3 Solution organizational chart for bee and B1 47
4.4 Type 2 plug architecture 52
4.5 The project's global architecture 54
4.6 Application organizational chart 55
4.7 project global archirtarchitecture 56
4.8 The STM32F407G-DISC1 development board 59
4.9 MCP2551 Tranceiver 59
4.10 MCP2515 Tranceiver 60
4.11 mosfet 60
4.12 RELAY module 61
4.13 The CP2102 usb module 62
4.14 GSM module 63
4.15 Detailed architecture of the diagnostic system 63

5.1 Oscilloscope 67
5.2 Multimeter 68
5.3 STM32CubeIDE 68
5.4 VS code 68
5.5 Draw.io 68
5.6 Arduino IDE 69
5.7 FreeRTOS 69
5.8 Arduino and MCP2515 connected to the BMS, and the received frames 70
5.9 The STM32 getting the received frame 71
5.10 The received frame 'rxMessage' from the BMS 72

5.11 STM32 translation of the data received . . . 73
5.12 STM32 transmission to the vehicle . . . 74
5.13 The transmitted frame 'txMessage' . . . 75
5.14 displayed data in the serial monitor . . . 76
5.15 The UART receive function . . . 76

6.1 STM32 pinouts . . . 81
6.2 Arduino uno pinouts . . . 81
6.3 USB module pinouts . . . 82
6.4 MCP2515 pinouts . . . 82
6.5 MCP2551 pinouts . . . 82
6.6 STM32 pinout configuration . . . 83
6.7 STM32 clock configuration . . . 83

List of Tables

1.1 B1 specifications 11
1.2 BEE Specifications 13
1.3 EVAN Specifications 15
1.4 Roles in SCRUM 20
1.5 product backlog of our project 21

4.1 Microcontroller selection and explanation 58
4.2 Tranceiver selection and explanation 59
4.3 USB module selection 62

5.1 Manufacturer translation manual example 72

6.1 Libraries used 80

List of Acronyms

- **OBC** = **O**n **B**orad **C**"harging
- **SPI** = **S**erial **P**eripheral **I**nterface
- **SoC** = **S**tate **O**f **C**harge
- **UART** = **U**niversal **A**synchronous **R**eceiver-**T**ransmitter
- **BMS** = **B**attery **M**anagement **S**ystem
- **CAN** = **C**ontroller **A**rea **N**etwork
- **ECU** = **E**lectronic **C**ontrol **U**nit
- **EV** = **E**lectrical **V**ehicle
- **GUI** = **G**raphical **U**ser **I**nterface
- **OBD** = **O**n **B**oard **D**iagnostics
- **SAS** = **S**ociéte par **A**ctions **S**implifiée

General Introduction

Electric vehicles (EVs) represent a pivotal advancement in modern transportation, offering significant environmental benefits, energy efficiency, and a reduction in greenhouse gas emissions. As the adoption of EVs continues to grow, ensuring their reliability and performance becomes increasingly crucial. One critical component of an EV's performance and user experience is the accurate display of the State of Charge (SoC). The SoC provides drivers with essential information about the remaining battery capacity and the estimated driving range. However, inaccuracies in the SoC display can lead to substantial issues, such as range anxiety, diminished user trust, and potential safety risks.

The primary objective of this project is to address and resolve the problem of incorrect SoC display in electric vehicles. This issue, if left unaddressed, can severely impact the reliability and user satisfaction of EVs. Through a comprehensive analysis, we identified the root causes of the SoC inaccuracies and developed an effective solution to correct them. Additionally, to ensure the long-term accuracy and reliability of SoC readings, we propose the implementation of a diagnostic system. This system is designed to proactively detect and address potential issues, thereby maintaining the integrity of the SoC display over time.

Our approach involved a detailed investigation of the factors contributing to the inaccurate SoC display. By implementing targeted software updates and hardware modifications, we successfully corrected these inaccuracies. Furthermore, the development of a diagnostic system will enable early detection of any anomalies, facilitating proactive maintenance and reducing vehicle downtime.

The expected outcomes of this project include improved accuracy of the SoC display, enhanced user confidence in the vehicle's reliability, and better overall vehicle performance. The diagnostic system will provide additional benefits by allowing for early intervention and maintenance, thus ensuring the longevity and dependability of the SoC readings.

In this report, we will detail the steps taken to identify and resolve the SoC display issue, describe the development and functionality of the diagnostic system, and discuss the anticipated benefits of these improvements. By addressing the SoC display problem and introducing robust diagnostic capabilities, this project aims to significantly enhance the performance and user experience of electric vehicles.

Chapter 1

Company Presentation and Project Context

Plan

Introduction 3
1 General context of the project 3
2 Description of the company 3
3 The BAKO Motors vehicles 10
4 Project context 17
5 design methodology 19
Conclusion 21

Introduction

This chapter begins with an introduction to the organization that facilitated this project. Following that, we will examine its internal operations, customer base, and strategies for maintaining a competitive edge against its rivals. Finally, we will outline the project's overall context and planning.

1.1 General context of the project

Named "Bringing Clarity to Electric Mobility: Diagnostics Implementation and problems Rectification in Electric Vehicles" this project serves as a vital link between academic knowledge and practical application within the startup "Bako Motors SAS." It represents an opportunity to apply embedded systems engineering expertise acquired at ENSIT, the National Higher School of Engineers of Tunis, in a real-world context.

1.2 Description of the company

This section presents the host company.

1.2.1 Host company: Bako Motors SAS

BAKO Motors is a dynamic Tunisian startup specializing in the design and production of electric vehicles. Established by Mr. Boubaker Siala in 2021, the company is dedicated to introducing sustainable and eco-friendly electric vehicles to Tunisia's automotive industry. With a focus on environmental responsibility, BAKO Motors initially concentrates on manufacturing last-mile electric delivery vans equipped with solar panels, aimed at reducing charging costs. The flagship model, the B1, boasts a spacious loading capacity of up to 300 cubic meters and a speed range spanning 100 to 200 kilometers.

Positioned as a leading manufacturer of solar-powered vehicles catering to micromobility and last-mile delivery across Africa, BAKO Motors is committed to democratizing sustainable transportation and promoting eco-mobility accessibility for all. Their primary objective revolves around fostering responsible and sustainable mobility through innovative solar energy solutions. The company's 2024 target milestone emphasizes three strategic priorities: enhancing client engagement, accelerating innovation initiatives, and diversifying product offerings, including data-driven solutions aimed at augmenting the company's overall value. [1].

Figure 1.1: Bako Motors manufacturing facility

Strategically positioned across multiple locations, Bako Motors operates from various key hubs. The company's headquarters are situated in Luxembourg, serving as the central administrative center for Bako Motors' global operations. In France, the company hosts its Research and Development facility, where a dedicated team of engineers and researchers continuously strive to innovate and enhance Bako Motors' electric vehicle technologies. Additionally, the company's manufacturing facility is based in Tunisia, facilitating the production of Bako Motors' electric vehicles.

1.2.2 Partners

Recently, Bako Motors and a significant industry participant entered into a strategic relationship. This partnership advances our goal of building a more sustainable future by significantly increasing our ability to deploy solar-powered cars. Additionally, Bako Motors has partnerships with some esteemed financial and technology establishments. Notable technical collaborators with extensive automotive knowledge and inventiveness are Assad, Amin, and Marquardt. In addition, we have agreements in place with important financial partners like Enda Tamweel and Zitouna Tamkeen, whose resources and assistance enable us to meet our challenging objectives and preserve our financial stability. Together with our partners, Bako Motors is in a strong position to drive innovation and lead the transition to eco-friendly transportation options. [2].

Figure 1.2: Bako Motors' partners.

1.2.3 Departments

This section outlines Bako Motors' departments, their responsibilities, functions, and contributions to the business's overall operations.

- **Design department:** The design section of the BAKO startup is made up of industrial product designers who are in charge of creating the vehicle's visual idea. This includes ergonomic and technical studies in addition to the external and interior design elements. The design team also prioritises user experience design, making sure that the car's layout improves driver and passenger comfort and offers an easy-to-use interface.

  Prototyping is another important duty of the design department, which uses cutting-edge methods like 3D modelling to refine design iterations and bring concepts to life. The Design team at BAKO advances the organization's goal by utilising their knowledge of cutting-edge technologies and design concepts to shape the cars' characteristics that are important to users and visually appealing.

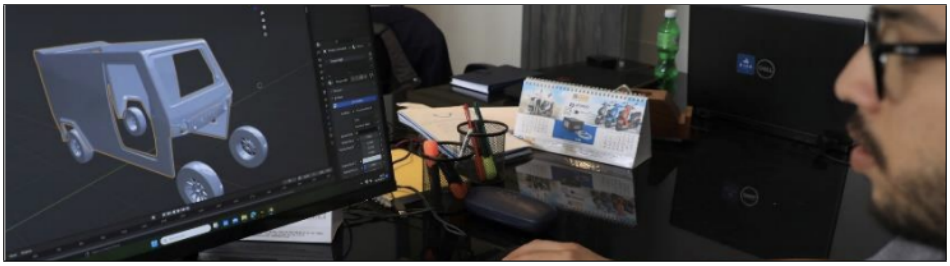

Figure 1.3: Bako Motors' design process

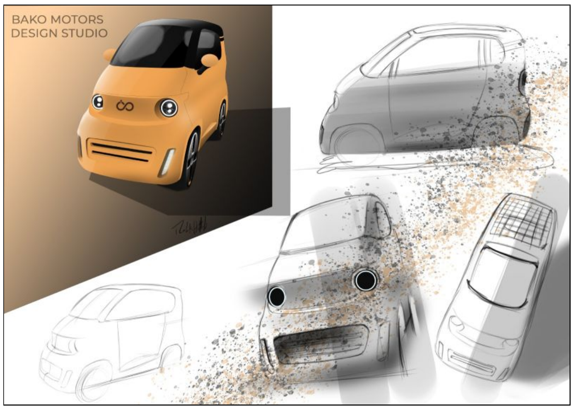

Figure 1.4: Bako Motors' design draft

- **Finance department:** Their primary duties include organising the creation of financial and accounting information and testing management controls, as part of their strategic role to offer privileged counsel to the general management. They also assist with budgeting and specify readability tools for activities.

  They essentially act as crucial advisors to upper management, ensuring that accounting and finance functions run smoothly, putting in place efficient management controls, and optimising budgeting processes to boost overall performance within the company.

  Their support and experience have a significant impact on decision-making procedures and the achievement of strategic objectives.

- **Marketing department:** Establishing and guiding management towards the key tenets of the business plan is the department's responsibility. It gathers information about industry competitors and consumer expectations in order to achieve this.

  To find possibilities and risks, this entails conducting market research, evaluating client input, and monitoring industry trends. The department creates strategic plans and works to position the company competitively in the market based on these findings. Additionally, it collaborates with other departments to guarantee that the strategy is properly applied throughout the company and to synchronise corporate objectives.

  Ultimately, the department plays a pivotal role in propelling the business's expansion and prosperity by converting market intelligence into workable strategies that create value and maintain a competitive edge.

Figure 1.5: Bako Motors poster example

- **Engineering department:**

  Engineers from several fields, including as electrical, mechanical, embedded systems, programming, and others, make up the engineering department. They are responsible for constructing the car's mechanics, creating the necessary wiring, enhancing the engine, and adjusting the speed control. This multidisciplinary team collaborates closely to ensure that every aspect of the vehicle's engineering is carefully planned and integrated, producing a finished product that is both high-quality and efficient. The engineering department is in charge of bringing innovative concepts to life and guaranteeing the vehicle's performance and operation from conception to completion. Their abilities and collaboration are essential to accomplishing the company's objectives and offering clients state-of-the-art solutions.

- **Production line:**

  - Mold workshop: The mould workshop is where the car's interior and cabin components are manufactured. Fibreglass and resin are laminated to produce precise structures that combine strength and beauty. This process brings the vehicle's basic structures to life by combining technology and expertise, ensuring lifespan and the appropriate look.

Figure 1.6: Bako Motors' mold workshop

– Welding workshop: The core of the automobile chassis production process is the welding department, which uses state-of-the-art welding techniques to build structural components. Modern equipment and skilled welders ensure the strength and accuracy of the chassis, which in turn ensures the lifetime and safety of the vehicle. Establishing a strong foundation upon which all automobiles are constructed requires this division.

Figure 1.7: Bako Motors' welding workshop

– Warehouse: The warehouse needs to be well-organized in order to guarantee efficient assembly processes. Every component expedites retrieval and shortens assembly time when it is properly labelled, categorised, and kept in designated areas. Inventory management systems are also used to keep an eye on supply levels, facilitate reordering, and prevent shortage-related delays.

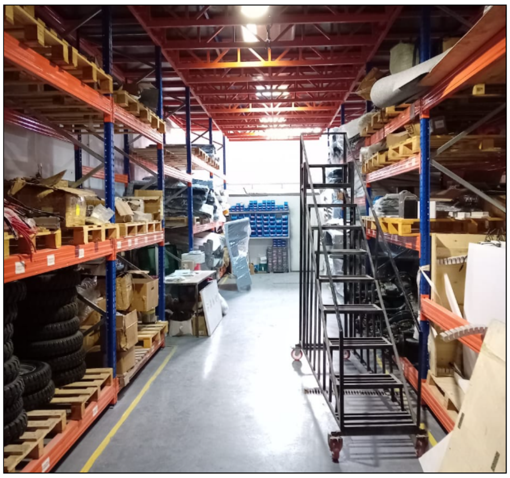

Figure 1.8: Bako Motors' warehouse

– Assembly line: Every car takes shape at the assembly line, which serves as a central hub. It consists of a number of stations, such as :

  * Mechanical assembly
  * Cabin marriage
  * Electrical wiring
  * Installation of safety devices
  * Installation of the solar panels and windshields
  * The quality control station

Figure 1.9: Bako Motors' assembly line

1.3 The BAKO Motors vehicles

1.3.1 The B1 vehicle

The B1 vehicle is a little electric van designed with delivery in mind. Its spacious interior makes the most of the storage space for goods or parcels, presumably since the driver can only access two of the doors. With the help of its solar panel, the B1 may be able to extend the battery's operating range. This battery has an impressive range of 100 to 150 km on a single charge, using only 1.5 TND for every 100 kilometres. The B1 is a preferred choice for providing services due to its affordability, durability, and ease of use.

Figure 1.10: Bako Motors' B1 model

Here the configuration options for the van including potential GPS integration and selectable battery range.

Figure 1.11: Bako Motors' B1 configuration

Specifications

Table 1.1: B1 specifications

Motorization	
Motor	Electric
Power	3000W
Couple	95 N.m
Version/Autonomy	
B10	100 km
B15	150 km
Dimensions	
L/W/H (mm)	3460/1760/1853
Loading space	
Volume	2433L
L/W/H (mm)	1600/1300/1170
Payload	300 kg
Battery	
Type	Lithium
Charging time	3h, 6h
Consumption	
Cost	1.5 DT / 100 km
17500 km/an	via solar energy
On the road	
Max range	150 km
Max speed	45 km/h
Emission CO2	0g / 100km
Exterior	
100 LED lighting	
Interior	
Places	2
Air conditioner	

Entertainment system	
GPS	
Rear camera	

1.3.2 The BEE vehicle

The BEE model from Bako Motors is a small electric car made for eco-conscious city driving. It emphasises mobility and a small footprint with only two doors and two seats, which makes it perfect for navigating city streets and confined parking spaces. With its four wheels, the BEE offers a stable and pleasant ride, and its unique roof-mounted solar panel adds a unique touch. By using the sun's energy to charge the battery, this solar panel maximises driving range while reducing environmental impact and the need for conventional charging stations. Because of this, the BEE seems like a desirable option for singles or couples searching for a green and sustainable way to get around town.

Figure 1.12: Bako Motors' BEE model

The configuration choices for the BEE model are displayed below, along with the associated prices for each colour, range, and additional function. Customers may choose their preferred colour and add other amenities like air conditioning using the selection system, which offers them a choice between three different range versions: Start, Wave, and Horizons.

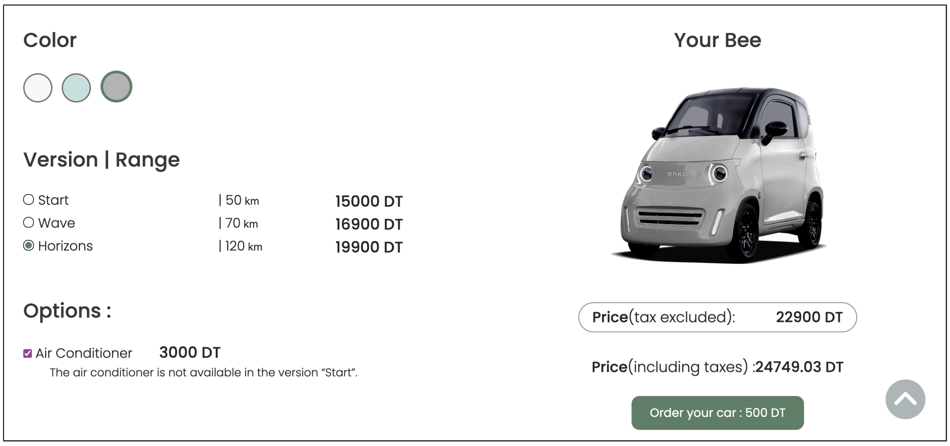

Figure 1.13: Bako Motors' BEE configuration

Specifications

Table 1.2: BEE Specifications

Motorization	
Motor	Electric
Power	2000W
Couple	95 N.m
Version/Autonomy	
Start	50 km
Wave	70 km
Horizon	120 km
Dimensions	
L/L/H (mm)	2350/1100/1530
Battery	
Type	Lithium
Charging time	3h, 6h
Consumption	
Cost	1.5 DT / 100 km
7300 km/year	via solar energy
On the road	
Max range	170 km

Max speed	45 km/h
Emission CO2	0g / 100km
Exterior	
100 LED lighting	
Wheels	12"
Interior	
Places	3
Adjustable driver's seat	
Air conditioner	
Touch screen	10"
USB port	
Radio	
Bluetooth connectivity	
Demisting system	
Reversing camera	
Adjustable rear window	

1.3.3 The EVAN vehicle

The EVAN model from Bako Motors is a compact electric vehicle designed for eco-conscious urban logistics and delivery. It emphasizes functionality and a minimal footprint with its small, utilitarian design, making it perfect for navigating city streets and tight parking spaces. With its four wheels, the EVAN offers a stable and efficient ride, and its unique roof-mounted solar panel adds an innovative touch. By harnessing solar energy to charge the battery, this solar panel maximizes driving range while reducing environmental impact and the need for conventional charging stations. Because of this, the EVAN is an attractive option for businesses seeking a green and sustainable way to manage deliveries and city transportation.

Figure 1.14: Bako Motors' EVAN model

The EVAN model offers various configuration choices, each with associated prices for different colors, ranges, and additional features. Customers can select their preferred color and opt for extra amenities like air conditioning using the selection system. This system provides a choice between three range versions: Start, Wave, and Horizons.

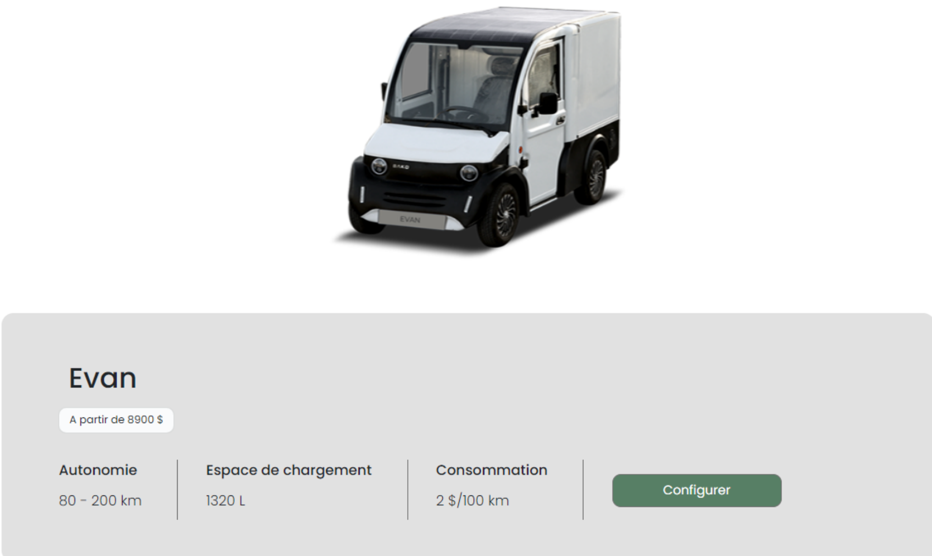

Figure 1.15: Bako Motors' BEE configuration

Table 1.3: EVAN Specifications

Motorization	
Motor	Electric
Power	7500W
Couple	110 N.m

Version/Autonomy	
Start	80 km
Horizon	200 km
Dimensions	
L/L/H (mm)	2815/1240/1680
Battery	
Type	Lithium
Charging time	3h, 6h
Consumption	
Cost	1.5 DT / 100 km
17500 km/year	via solar energy
On the road	
Max range	150 km
Max speed	45 km/h
Emission CO2	0g / 100km
Exterior	
100/100 LED lighting	
Wheels	12"
Interior	
Places	1
Adjustable driver's seat	
Air conditioner	
Touch screen	10"
USB port	
Radio	
Bluetooth connectivity	
Demisting system	
Reversing camera	
Adjustable rear window	

1.4 Project context

We start by discussing the problematic aspects that the project aims to address to provide context for the initiative. Next, we specify the specific objectives that the project seeks to fulfill. Lastly, we offer the project plan that was developed to guide the process of execution.

1.4.1 Problematic

The "QQOQCP approach" will be used to thoroughly understand the project's problematic features, aiding in the development of effective solutions and strategies for successful project implementation.

Question (Quoi?)- What is the problem? Bako Motors is facing significant issues with the inaccurate display of the State of Charge (SoC) in their electric vehicles. This problem not only leads to potential misjudgments in vehicle range and battery health but also undermines customer trust and satisfaction. In addition to this primary issue, Bako Motors lacks an effective diagnostic tool, making it challenging for staff to diagnose and maintain faults quickly and accurately.

Question (Qui?):Who is harmed by this problem? Stakeholders affected include Bako Motors, technicians, and customers. Customers may experience service delays and reduced vehicle reliability, while technicians face increased difficulty in performing accurate diagnostics and maintenance. Regulatory authorities could also be concerned about the reliability and safety of the vehicles.

Question (Où?):- Where does the problem occur? The issue primarily arises during the everyday use and servicing of Bako Motors' electric vehicles, both in the workshop and potentially in the field.

Question (Quand):- When does the problem occur? The problem occurs whenever Bako Motors' electric vehicles are in operation, during routine servicing, troubleshooting customer-reported issues, and pre-delivery inspections.

Question (Comment):- How does the problem manifest? The problem manifests as follows:

- Inaccurate SoC readings lead to misinformed decisions regarding vehicle range and battery status.
- Technicians spend excessive time manually diagnosing issues without suitable tools.

- Delays in detecting and correcting potential issues due to the inability to access real-time diagnostic data.
- Lack of defined communication protocols may cause compatibility concerns with various car models.

Question (Pourquoi?): Why is the problem important? Accurate SoC readings are vital for ensuring reliable vehicle performance and maintaining customer trust. Efficient diagnostics and maintenance are critical for customer satisfaction and retention. Delayed or inaccurate diagnostics can result in increased downtime and higher maintenance costs for Bako Motors.

1.4.2 Project objectives

The purpose of this project is to address the issue of incorrect SoC displays in Bako Motors' electric vehicles and to develop an effective diagnostic system to enhance vehicle reliability and maintenance efficiency. The specific objectives are:

- To implement solutions that ensure the accurate display of the State of Charge (SoC), thereby improving vehicle reliability and user experience.
- To design a diagnostic system that enables quick and accurate fault detection and maintenance.
- To integrate real-time data access and communication protocols to enhance the diagnostic process.
- To reduce vehicle downtime and maintenance costs through efficient diagnostics and accurate SoC readings.
- To enhance overall customer satisfaction and trust in Bako Motors' electric vehicles.

By meeting these goals, the diagnostic tool will provide Bako Motors with a dependable solution for maintaining and servicing their electric vehicles, hence increasing dependability and minimizing downtime.

1.4.3 Project planning

The development timeline for the project is displayed in this table as a Gantt chart, similar to the one in figure 1.14. It lists the tasks to be accomplished during the project, broken down into specific assignments for each week and arranged by month. The project timeline is represented graphically in the table by horizontal bars that span months and have lengths equivalent to the duration of the tasks. This timeline makes the development stages easy to see, facilitating the visualisation of the project flow and its timely conclusion.

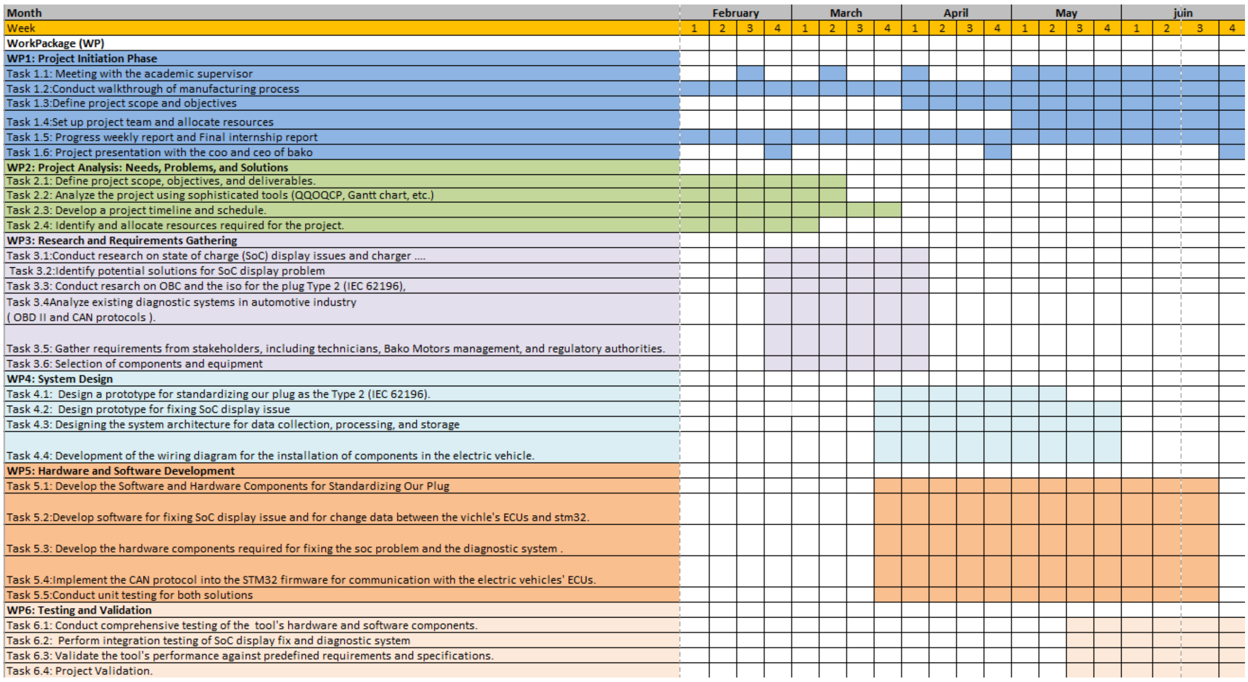

Figure 1.16: The project's Gantt chart

1.5 design methodology

A design methodology outlines the different phases and their sequences in the development process of a software application. The most recurrent methodologies include the analysis of the existing system, specification, design, implementation, and evaluation. To transition from the needs of the company and project requirements to application code, we need to adopt a development methodology that we will adhere to throughout the project cycle. After choosing the methodology, we need a modeling language to model our project. In this section, we will provide a general idea of the methodology followed during the design and development of our application. Then, we will define the modeling language we adopted.

1.5.1 Choosing the Agile Methodology

At the start of the project, the first instinct is to try to anticipate and plan everything, which is not always possible and sometimes leads to failure. We can cite two limitations of this predictive approach:

Risk Management: It is impossible to anticipate all risks as unforeseen events can arise at any moment.

Development: It is impossible to foresee all tasks and how to execute them, as new possibilities and ideas are constantly being discovered throughout the project.

These previously mentioned problems led IT professionals to reinvent project management and design methods by introducing what is known as the agile methodology. It is an incremental and iterative approach, conducted in a collaborative spirit, with minimal formalism. It can generate a high-quality product while considering the evolving needs of clients. By following this approach, the software is designed as a whole and can be built step by step like the SCRUM method.

1.5.2 SCRUM Methodology

This is an iterative agile methodology based on short-duration iterations called Sprints. It is characterized by:

Its ease of implementation and flexibility to changes. Its focus on real project progress rather than theoretical progress. Its organizational style, which focuses on the process of producing a product. SCRUM primarily involves three main roles:

Table 1.4: Roles in SCRUM

Role	**Description**
Product Owner	The project leader from a business perspective. They communicate a clear vision of the product and define its main characteristics. They invest in the process by prioritizing product features. Their support is based on real project progress rather than theoretical progress.
SCRUM Master	Their task is to optimize the team's production capacity and help them work efficiently and improve constantly. They are also responsible for the correct implementation of SCRUM.
Team	The team that handles product development incrementally with a series of short periods called Sprints. This team is responsible for carrying out the assigned tasks within the time frames.

SCRUM proposes three essential principles:

The Product Backlog: Represented as a list of ideas for the final product vision, maintained in a required order.

The Sprint Backlog: A detailed development plan for the next Sprint.

The Product Increment: The result at the end of a Sprint, representing an integrated version of the product with an acceptable quality level, ready to be deployed once validated by the Product Owner. This methodology breaks tasks into subtasks, then handles them in the form of sprints. Sprints last between a few hours and a month. Each sprint begins with an estimation followed by planning. A "daily meeting" is held at the same time and place every day to ensure synchronization among team members and evaluate progress toward the sprint goal. At the end of the sprint, a demonstration of what has been achieved is shown, contributing to the product's business value. Before starting a new sprint, a retrospective meeting is held to analyze what happened during the previous sprint and improve for the next one.

1.5.3 Product Backlog

Before starting each project and using the SCRUM methodology, it is necessary to define the product backlog. The following table summarizes the elements of our project backlog:

Table 1.5: product backlog of our project

Identifier	Backlog Item
1	Preliminary study / Specifications
2	Development of a solution for the problem
3	Testing the prototype functionalities
4	Implementation and testing of the state machine

Conclusion

In this chapter, we introduced Bako Motors, the company that provided the platform for this senior project, and discussed the project's goals and background. We identified the shortcomings in Bako Motors' current diagnostic processes, particularly the inaccurate SoC display and limited data accessibility. The project's objectives were defined to develop an effective diagnostic system with accurate SoC readings and fault detection, ensuring it meets industry standards.

We outlined our project plan, focusing on implementing standardized sensor calibration, software updates, and real-time data protocols. In the next chapter, we will explore the communication technologies and protocols essential for our project.

Chapter 2

Communication Technologies and Protocols

Plan

Introduction 23

1 Introduction to the Onboard Charging System in Electric Vehicles . . 23

2 Plug Standardizing with IEC 61851-1 24

3 CAN bus (Controller area network) 26

4 Battery management system (BMS) controller 30

5 Diagnostic systems 34

6 OBD-II (On-Board Diagnostics) 34

Conclusion 36

Introduction

This chapter will begin with an introduction to the onboard charging system, followed by an exploration of the Controller Area Network (CAN bus). It will then examine the Battery Management System (BMS) and provide an introduction to diagnostic systems. Finally, it will conclude with an overview of the On-Board Diagnostics (OBD-II) system.

2.1 Introduction to the Onboard Charging System in Electric Vehicles

Electric vehicles (EVs) are rapidly becoming a significant component of the global automotive market due to their environmental benefits and advances in technology. A crucial part of the EV infrastructure is the onboard charging system (OBC), which enables the vehicle to charge its battery using external power sources. This introduction outlines the essential aspects of the OBC, including its functionality and components.

2.1.1 Functionality of the Onboard Charging System

The primary function of the OBC is to convert alternating current (AC) from the power grid into direct current (DC) suitable for charging the vehicle's battery. This conversion is necessary because EV batteries store energy in DC form, while most public and home charging stations supply AC power. The OBC also manages the charging process to ensure it is safe, efficient, and optimized for the battery's longevity.

Key functions include:

- **AC to DC Conversion:** Transforming the AC from the grid into DC required by the battery.
- **Voltage Regulation:** Ensuring the voltage is appropriate for the battery's charging needs.
- **Charging Control:** Managing the charging speed and efficiency to maintain battery health.
- **Safety Protocols:** Implementing safety measures to prevent overcharging, overheating, and electrical faults.

2.1.2 Components of the Onboard Charging System

The OBC comprises several critical components that work together to achieve its functions:

- **AC/DC Converter:** The core component that converts AC power to DC power. It typically includes rectifiers and power factor correction (PFC) circuits.

- Control Unit: Manages the charging process, including monitoring battery status, controlling power flow, and ensuring compliance with safety protocols.

- Thermal Management System: Keeps the OBC components within optimal temperature ranges to prevent overheating and ensure efficient operation.

- Communication Interface: Facilitates communication between the vehicle and the external charging infrastructure, ensuring proper handshake and charging protocol adherence.

- Connectors and Cables: High-quality electrical connections that safely transfer power from the charging station to the vehicle's OBC.

2.2 Plug Standardizing with IEC 61851-1

IEC 61851-1 is an international standard that defines the general requirements for electric vehicle conductive charging systems. This standard ensures the compatibility and safety of charging equipment and infrastructure across different regions and manufacturers.

2.2.1 Overview of IEC 61851-1

IEC 61851-1 specifies the characteristics and operational requirements for conductive charging systems. This includes the requirements for the connection between the electric vehicle and the charging station, ensuring safe and efficient charging.

2.2.2 Plug Types and Their Standards

IEC 61851-1 standardizes the connectors and plugs used for charging electric vehicles. The most common plug types include:

Type 1 (SAE J1772): Commonly used in North America, supports single-phase AC charging.

Type 2 (IEC 62196): Widely used in Europe and other regions, supports both single-phase and three-phase AC charging.

GB/T: Used in China, with separate standards for AC and DC charging.

CHAdeMO: A DC fast charging plug developed in Japan, suitable for high-speed charging.

CCS (Combined Charging System): Supports both AC and DC charging, integrating the Type 2 plug with additional DC pins for fast charging.

The Type 2 plug, as seen in Figure 2.1, is commonly used in Europe due to its versatility and robustness.

2.2.3 Communication and Safety Protocols

IEC 61851-1 also defines the communication protocols between the electric vehicle and the charging station. This ensures that the vehicle and the charger can communicate effectively to manage the charging process and ensure safety. Key protocols include:

Pilot Signal: Used to initiate and control the charging process, ensuring proper connection and power flow.

Proximity Detection: Ensures that the plug is securely connected before charging begins, preventing accidental disconnection.

Fault Monitoring: Continuously monitors the charging process for faults, such as overcurrent or overheating, and takes corrective actions to ensure safety

2.2.4 Plug Types and Their Standards

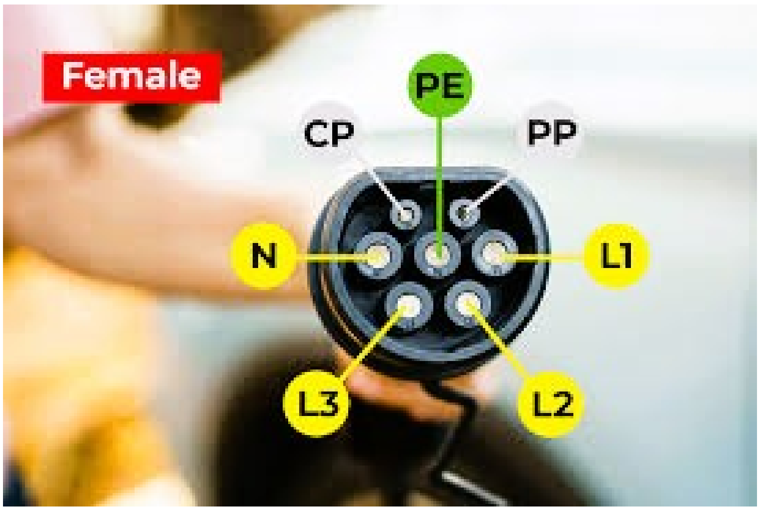

Figure 2.1: plug Type 2

Conclusion

The onboard charging system is a vital component of electric vehicles, enabling them to recharge efficiently, safely, and conveniently. By converting AC power from the grid into DC power for the battery, the OBC ensures that EVs can be integrated seamlessly into existing electrical infrastructure. The choice of the Type 2 plug reflects the need for compatibility with regional standards and charging speeds, further enhancing the flexibility and usability of EVs. As the adoption of electric vehicles continues to grow, advancements in OBC technology will play a crucial role in enhancing the performance, safety, and convenience of EV charging.

2.3 CAN bus (Controller area network)

The car performs better and is more efficient thanks to this cooperative approach. Similar to the nervous system in biological organisms, the Controller Area Network (CAN bus) is the network that enables communication in cars by serving as a conduit for data and decision-making. Similar to how neurons in the human body interact and coordinate, nodes, or electronic control units (ECUs), are connected via the CAN bus. Every ECU performs a particular task, such managing brakes or engine performance. These ECUs interact with one another across the CAN bus in a way that is similar to how biological systems work together to maintain overall functionality and gather data. 30

2.3.1 Electronic Control Units (ECUs)

Electronic control units (ECUs) in modern cars perform a multitude of tasks, including as engine and transmission management, airbag and audio system control, and more, as seen in figure 2.2. Due to the increasing sophistication of automotive systems, a single car might occasionally have up to 70 ECUs, each of which is responsible for a particular component of the vehicle's operation. The may bus technology allows these ECUs to communicate with one another, allowing them to share information and plan actions. Because each ECU no longer requires intricate and specialised wiring, the CAN bus technology streamlines the wiring of the vehicle. Rather, every ECU is linked to a solitary network, permitting inter-ECU communication using a common protocol.

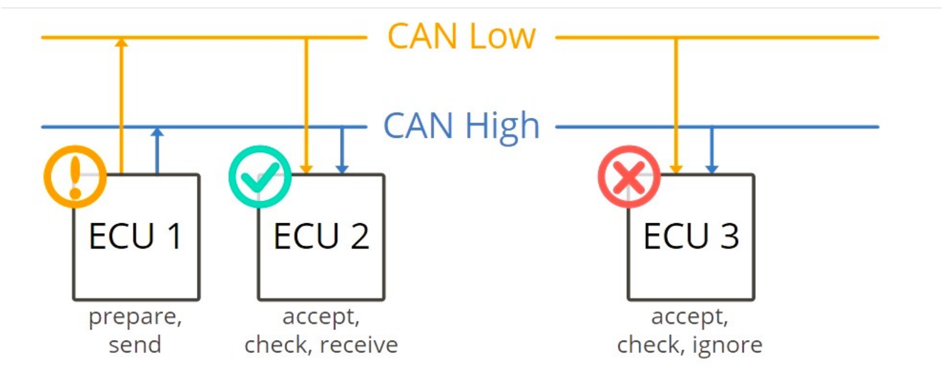

Figure 2.2: CAN communication

An ECU particularly uses the CAN bus to prepare and transmit data, such as control commands or sensor readings. There are two wires used to transport this data: CAN low and CAN high. Every other ECU connected to the CAN network receives this transmitted data. Following that, each ECU verifies the data it has received and decides, based on its own internal logic and standards, whether to accept or reject it. The great coordination of vehicle duties by the ECUs is made possible by their seamless communication via the CAN bus system, resulting in improved performance, safety, and dependability.

2.3.2 CAN frames

Over the CAN bus, communication takes place using CAN frames. An illustration of a typical CAN frame—the sort seen in most cars—with an 11-bit identity. The enlarged 29-bit identification frame is the same, save for the lengthier ID.

The CAN ID and data, which are indicated in the figure 2.3 below, are important to record when it comes to CAN bus data, as we will see.

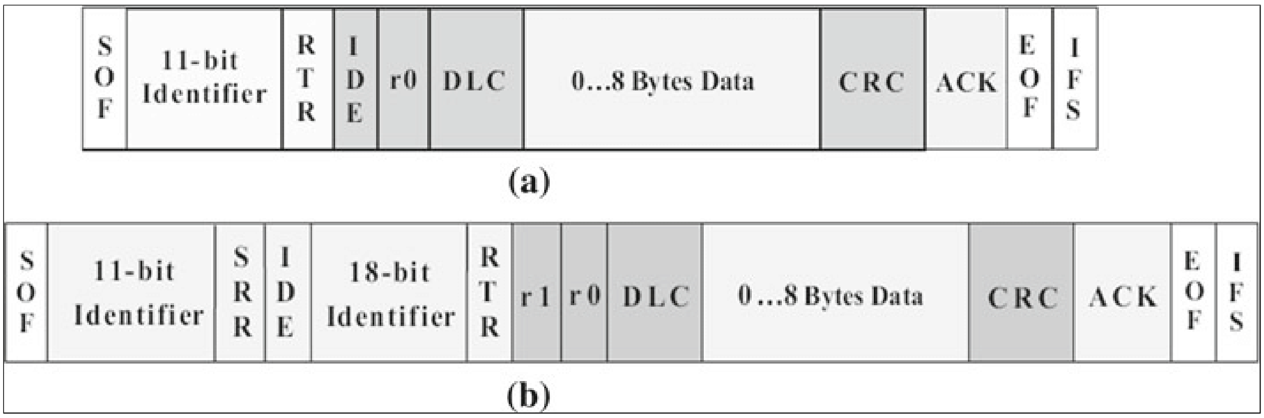

Figure 2.3: CAN frame

a: standard CAN frame **b:** Extended CAN frame

SOF: To indicate to other nodes that a CAN node is planning to communicate, the Start of Frame (SOF) is a "dominant 0".

ID: The frame identification is the ID; lower values indicate a higher priority.

RTR: A node's ability to transmit data or request specific data from another node is indicated by the Remote Transmission Request.

Control: With the exception of the Identifier Extension (IDE) bit, the control field consists of 5 bits. The identifier format in use is indicated by the IDE bit. A normal 11-bit identifier is represented by a 'dominant 0' in the IDE bit, and an extended 29-bit identifier is represented by a 'recessive 1'.

Data: The payload, or data bytes, in the data comprises may signals that may be retrieved and decoded to provide information.

CRC: Data integrity is ensured by the use of the Cyclic Redundancy Check.

ACK: The ACK slot shows whether the node has acknowledged and received the data.

EOF: The CAN frame ends at the EOF.

2.3.3 CAN characteristics

As the accompanying image 2.4 shows, a Controller Area Network (CAN) connection needs specific qualities in order to transport data reliably. Comprehending these attributes is essential for formulating and executing resilient CAN-dependent systems.

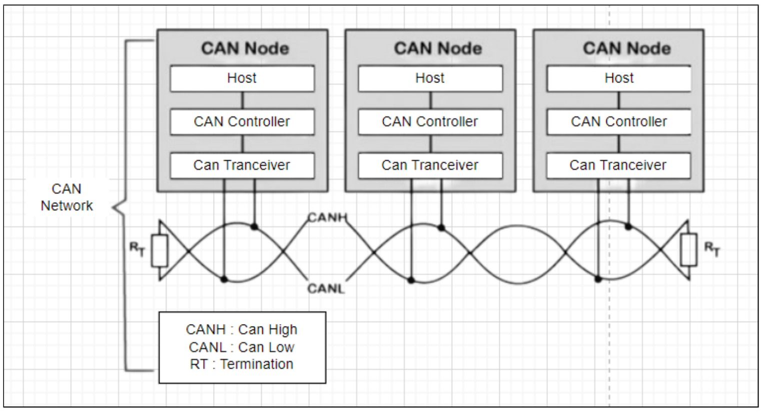

Figure 2.4: CAN node

Host: Programmes responsible for certain tasks are executed by a host, which is usually a microcontroller or CPU, within the CAN network.

CAN controller: The CAN controller manages communication activities, including initiating message transmission and reception, in compliance with the CAN standard.

CAN tranceiver: This device serves as a bridge between the CAN controller and the actual CAN bus, enabling data transmission and reception. It converts the electrical impulses on the CAN bus into a format that can be understood by CAN controllers.

Variations of voltage

CAN communication consists of two lines: CANL and CANH. The differential voltage across these lines controls the transmission of logical 0s and 1s.

High-speed CAN (HS CAN) as shown in the figure 2.5 and low-speed CAN (LS CAN) as shown in the figure 2.6, are the two basic forms of CAN buses.

High-Speed CAN (HS CAN):

Recessive state (logic 1): The definition of a logic 1 is a differential voltage of roughly 0V. This is the state of the bus when it is idle.A differential voltage is considered recessive if it is less than 0.5V.

dominant state(Logic 0): A logic 0 is indicated by a differential value of usually 2 volts. This occurs when the CANL line's voltage is less than the CANH line's voltage. It is believed that there is a dominating condition when the differential voltage is higher than 0.9V.

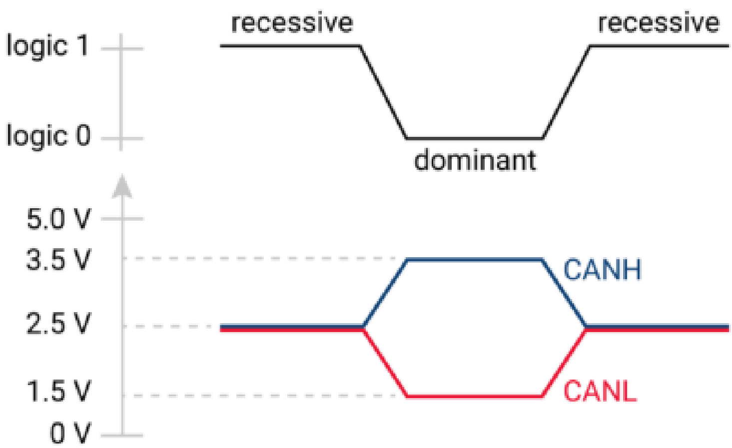

Figure 2.5: CAN high speed variation

Low-Speed CAN (LS CAN):

Logic 1: Recessive state A differential voltage of 5V is frequently used to represent a logic 1. This is the state of the bus when it is idle.

Logic 0: the dominant state A 2V standard differential value indicates a logic zero. This occurs when the CANL line's voltage is less than the CANH line's voltage.

Important things to think about:

Logic 1 can only be in its recessive state when both the CANL and CANH lines in the HS CAN are at 2.5V. When CANL is at 1.5V and CANH is at 3.5V, the dominant state, also known as logic 0, is reached.

The ideal condition for a bus is recessive; once a node reaches the dominant state, other nodes are unable to push it back into the recessive state. To provide a reliable network connection, CAN-based systems must be professionally designed and configured, requiring an understanding of unique CAN

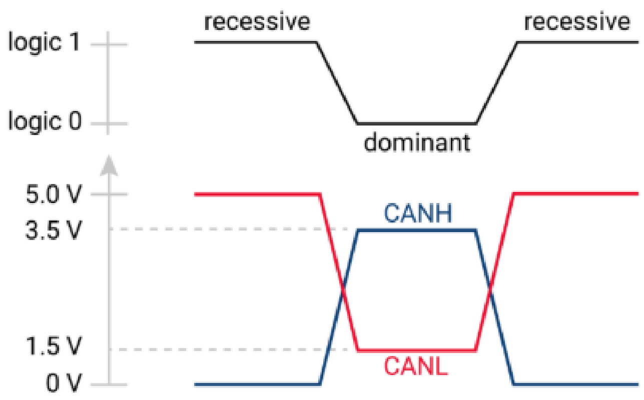

Figure 2.6: CAN low speed variation

features.

2.4 Battery management system (BMS) controller

A BMS typically gets data from the battery it is monitoring, applies an algorithm to process it, and outputs the result. Utilising a range of intricate procedures, the BMS aims to track, regulate, and enhance the performance of every single battery cell in the pack. Here, we examine the essential elements and working procedures of a standard BMS.

2.4.1 Battery management system's architecture

As seen in figure 2.7, the BMS architecture is made up of a number of parts and features that are required to oversee and control the condition and functioning of the battery pack in an electric vehicle.

The following are a BMS's primary duties:

- **Battery cells:** These are the separate components that comprise the battery pack. Every cell's performance and well-being are tracked and managed by the BMS.
- **Voltage & current sensors:**These sensors provide vital information for comprehending the battery's condition and spotting possible problems by measuring the voltage and current levels of both the battery pack as a whole and individual cells.
- **Temperature sensors:** The lifetime and proper operation of the battery depend on these

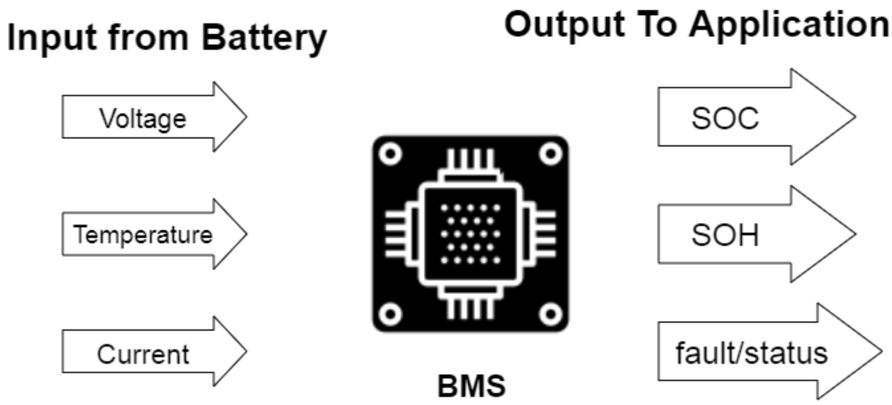

Figure 2.7: The BMS architecture

sensors monitoring the temperature of the battery pack and individual cells.

- **State of charge (SOC):**The SOC % indicates the remaining usable portion of the battery pack. To determine how much battery range is remaining, users can utilise the data supplied by the BMS, which is an estimate of SOC based on several characteristics.

- **State of health (SOH):**SOH evaluates the overall condition and capacity of the battery pack to its original specifications. Many characteristics are monitored by the BMS in order to determine SOH, or remaining battery life.

- **Faults/Status monitoring:** Potential problems with the battery pack, including overvoltage, undervoltage, overcurrent, or excessive temperature, are identified by the BMS. In order to alert customers to any battery-related issues, it also keeps an eye out for anomalies in the condition of the cells and gives fault codes or status information.

2.4.2 Battery management systems functionality

A battery management system's (BMS) many intricate procedures are designed to track, regulate, and enhance the performance of every single battery cell in the pack. Here, we examine the essential elements and working procedures of a standard BMS:

- **Cell monitoring and balancing**: Monitoring the voltage, temperature, and state of charge (SoC) of every single cell in the battery pack is one of a BMS's main responsibilities. The BMS can evaluate the cell's health and performance in real time and spot any deviations from the intended operating parameters thanks to this real-time monitoring. The BMS also uses active cell balancing techniques, as seen in the figure 2.8 below, to balance the charge levels of individual cells in order to enhance the battery pack's total energy capacity and limit the risk of overcharging or undercharging.

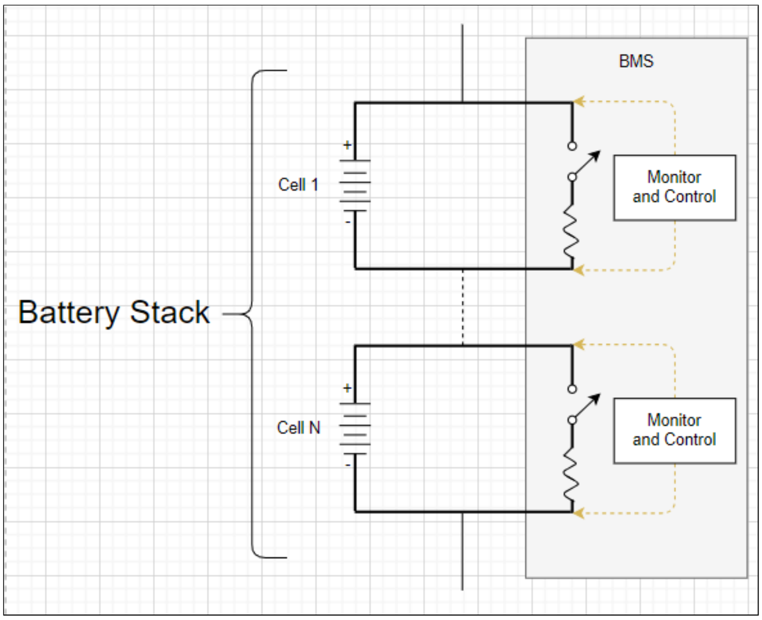

Figure 2.8: BMS monitoring process

- **Thermal management**: In order to maintain ideal working conditions and avoid overheating, the BMS may use thermal management techniques such active cooling or heating by continuously monitoring cell temperatures and identifying any thermal runaway events, as shown in the picture 2.9 below. In order to avoid dangerous situations and battery-related mishaps, the BMS also includes safety measures including overcurrent protection, short-circuit detection, and emergency shutdown methods.

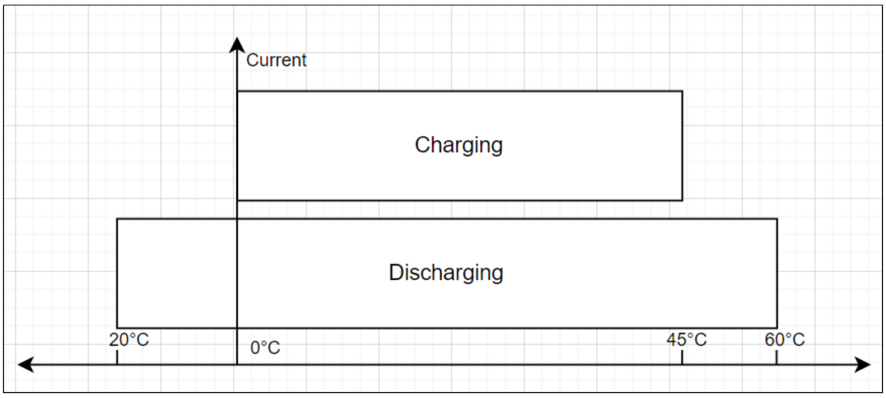

Figure 2.9: BMS thermal management

- **BMS data acquisition**:The sole purpose of the battery management system (BMS) controller is to govern energy derived from renewable sources. Traditional appliances that use

AC power cannot be powered by renewable energy sources since they have a nominal voltage of 48 volts. While one control connects or disconnects the power source—which might be a battery or a renewable energy source—the other chooses the inverter mode. This gadget has an AC inverter that can change DC energy into AC energy. To avoid power spikes, the BMS controller can transfer electricity to the inverter at a certain percentage based on user and system characteristics.

- **Communication and integration**: The car's electrical system and other onboard systems, including the powertrain, thermal management, and vehicle control unit, can communicate more easily thanks to the BMS, which acts as a central hub. In addition, the BMS may interface with other gadgets, including charging stations and diagnostic equipment, to provide useful diagnostic data and allow remote battery pack monitoring and control.

2.4.3 The BMS frame

There are several parts that make up the communication frame of the battery management system (BMS) in an electric car. Two-byte fields that come first—a Priority field and a data ID field—identify the type of data being delivered or requested. Then, the source and destination addresses each include two more bytes. The sender is identified by the source address, while the BMS is often represented by the destination address. Lastly, the eight bytes that comprise the data field contain the actual data that is sent between nodes. These bytes are initially empty if it's a request, since they wait for the BMS to provide information. An example of this frame construction may be seen in the image. 2.10.

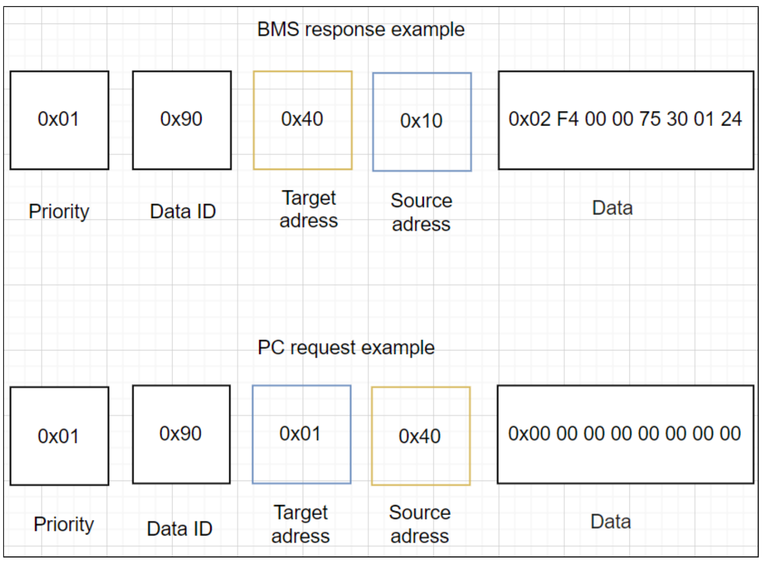

Figure 2.10: BMS frame

‘ **Priority**: Shows the message’s urgency, with lower numbers indicating a higher priority.
Data ID: The data type being sent or requested, such as the maximum and minimum voltage values, is specified in the data ID field.
Source address: The message’s sender.
Target address: The message’s target.
Data: Contains the actual data that is shared, like temperature readings.

2.5 Diagnostic systems

Figure 2.11 illustrates how much electric vehicles (EVs) depend on their diagnostic systems to guarantee optimal performance and flawless functioning. A network of sensors, controllers, and software algorithms is used in various ways to monitor, assess, and troubleshoot the car’s mechanical and electrical systems. Battery Management Systems (BMS) are critical components of EV diagnostic systems. They monitor the health and condition of the battery pack, preventing overcharging and overheating and facilitating smooth charging and draining. Furthermore, Motor Control Units (MCUs) and Power Electronics Modules (PEMs) provide an effective power supply and increase the vehicle’s range by monitoring and controlling the operation of the electric motor. Full functionality of onboard diagnostics (OBD)

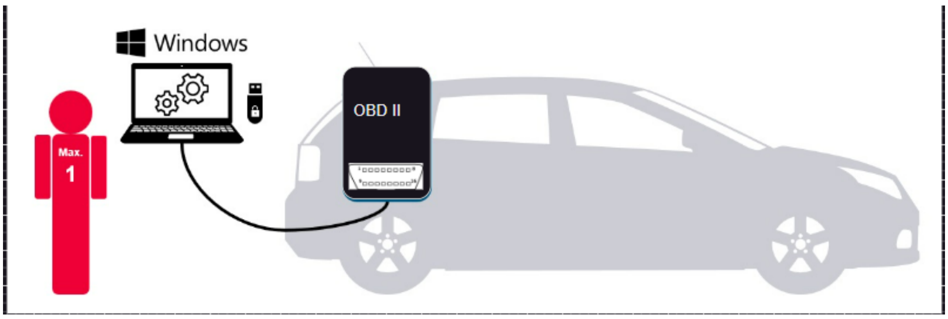

Figure 2.11: Diagnostic system

2.6 OBD-II (On-Board Diagnostics)

Has your dashboard’s failure indicator light ever been on?
That’s your car trying to alert you to a problem. When you bring your car to a repair shop, the mechanic will use an OBD2 scanner to identify the problem. He will do this by attaching the OBD2 reader to the OBD2 16 pin connector that is located next to the steering wheel. This enables him to research and solve the problem using OBD2 codes, or diagnostic trouble codes (DTCs).

2.6.1 The OBD2 connector

Fast automobile data retrieval is available with OBD2 types A and B connections. Type B is often found in medium- and heavy-duty vehicles, whereas type A is typically seen in cars. Types A and B have different power supply outputs while having identical pin layouts: type B has a 24V output, whilst type A has a 12V output.

Usually found next to the steering wheel, the OBD2 connector can occasionally be hidden by panels or covers. An illustration of a type A OBD2 pin connection, sometimes referred to as a Data Link connector (DLC), is shown in Figure 2.12 below.

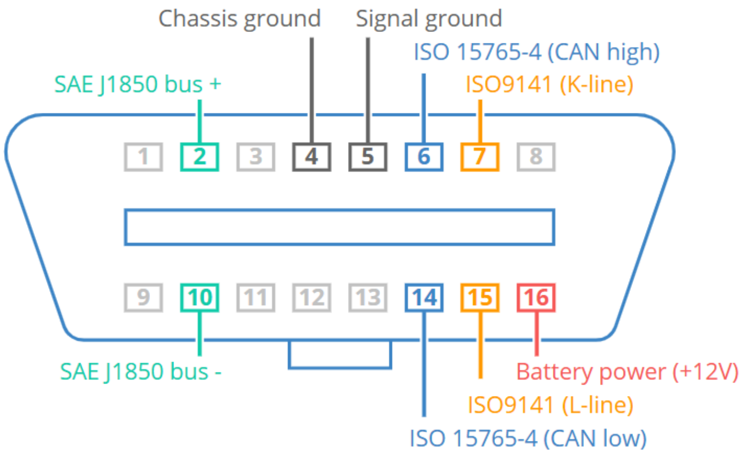

Figure 2.12: the obd2-connector-pinout

Pin 16 frequently provides battery power even in the absence of ignition. The car's electronic control units (ECUs) and external diagnostic equipment communicate serially back and forth over pins 7 (K-line) and 15 (L-line), as well as pins 2 (bus+) and 10 (bus-). In most situations, the CAN protocol is used to link pins 6 (CAN-H) and 14 (CAN-L).

2.6.2 Connection between CAN bus and OBD2

On-Board Diagnostics (OBD2), a defined communication language used in automotive diagnostics, is referred to as "higher layer protocol". It makes communication between the various electronic control units (ECUs) of a car and outside diagnostic equipment easier.OBD2 has five protocols that control data interpretation and transmission across the vehicle's network in addition to the OBD2 connection. Of these, the controller area network (CAN) protocol is the most widely used and has been required for cars built in the United States since 2008. Through effective and standardised communication between ECUs made possible by OBD2, comprehensive diagnostics and troubleshooting are made easier.

2.6.3 Integrating OBD II

The primary goal of my project is to create a direct communication channel between the On-Board Diagnostics (OBD) connector of the automobile and the CAN high and low pins of the tranceiver CAN controller, as shown in figure 2.13. The tranceiver and the vehicle's electrical systems can communicate back and forth by establishing this link. This improves the system's general performance and diagnostic capabilities by making it easier to retrieve diagnostic and other crucial data from the car's network.

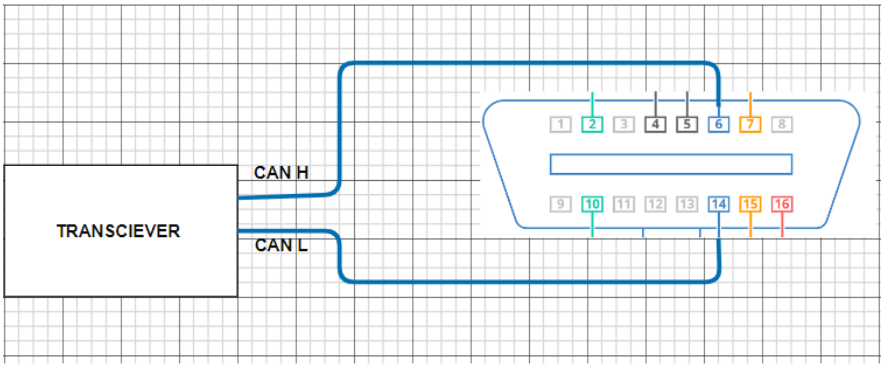

Figure 2.13: The OBD port to tranceiver

Conclusion

An extensive review of the communication technologies and protocols that are essential to our project has been given in this chapter. The following chapter will describe how our solution was created and put into practice, using the knowledge we have gained to advance the project, after this framework has been established.

Chapter 3

Problem Identification

Plan

1 **Introduction** **38**

2 **Importance of SOC in Electric Vehicles** **38**

3 **Display of SOC in the BEE Model** **39**

4 **Display of SOC in the EVAN Model** **41**

5 **the Onboard Charger** **41**

6 **lack of Diagnostic System Bako Motors Vehicles** **43**

3.1 Introduction

Electric vehicles like Bako Motors' BEE model offer eco-conscious urban drivers a sustainable transportation solution. However, the BEE model faces technical challenges affecting its performance and user experience. Two critical issues are its inaccurate State of Charge (SOC) display and onboard charger inefficiencies.

The SOC display is crucial for estimating remaining battery capacity and range, but the BEE model's display is unreliable, causing range anxiety and frustration for drivers.

The onboard charger, responsible for converting grid electricity for battery charging, suffers from inefficiencies and lacks an automatic disconnection mechanism when the battery is fully charged. This can lead to overcharging, increased energy consumption, and reduced battery lifespan.

Understanding these problems is imperative to developing solutions that enhance the performance, reliability, and user satisfaction of the BEE model. This chapter outlines the identified issues in SOC display and onboard charging, providing a foundation for the subsequent development of a robust diagnostic system aimed at addressing these challenges.

3.2 Importance of SOC in Electric Vehicles

The State of Charge (SOC) is a crucial metric in electric vehicles, indicating the battery's current energy level as a percentage of its total capacity. Here's why SOC is important:

Range Prediction SOC helps drivers know how far they can travel before needing to recharge. Accurate SOC readings prevent unexpected battery depletion and reduce range anxiety, especially in urban environments where charging stations may be sparse.

Battery Management Proper SOC monitoring ensures the battery operates within safe limits, preventing overcharging and deep discharging. This helps maintain battery health and extends its lifespan.

Efficiency Optimization SOC data allows the vehicle to optimize energy usage, balancing power delivery and regenerative braking to maximize driving range and reduce the need for frequent charging.

Charging Process Management Accurate SOC readings guide the charging process, determining when and how to charge the battery efficiently. This reduces waiting times and ensures the vehicle is ready when needed.

User Confidence and Satisfaction Reliable SOC displays build driver confidence, making them more likely to depend on their EV for daily commutes and longer trips. Trust in SOC readings enhances overall user satisfaction and promotes wider EV adoption.

3.3 Display of SOC in the BEE Model

The BEE model from Bako Motors encounters specific issues with the display of the State of Charge (SOC), impacting driver confidence and vehicle performance. The primary problem lies in the programming of the SOC display, which was originally calibrated for lead-acid batteries, not the lithium batteries used in the BEE model. This discrepancy causes significant inaccuracies in SOC readings due to the differences in discharge characteristics between the two types of batteries.

3.3.1 Discharge Characteristics: Lead-Acid vs. Lithium Batteries

The discharge curves of lead-acid and lithium batteries differ significantly, as illustrated in the provided charts.

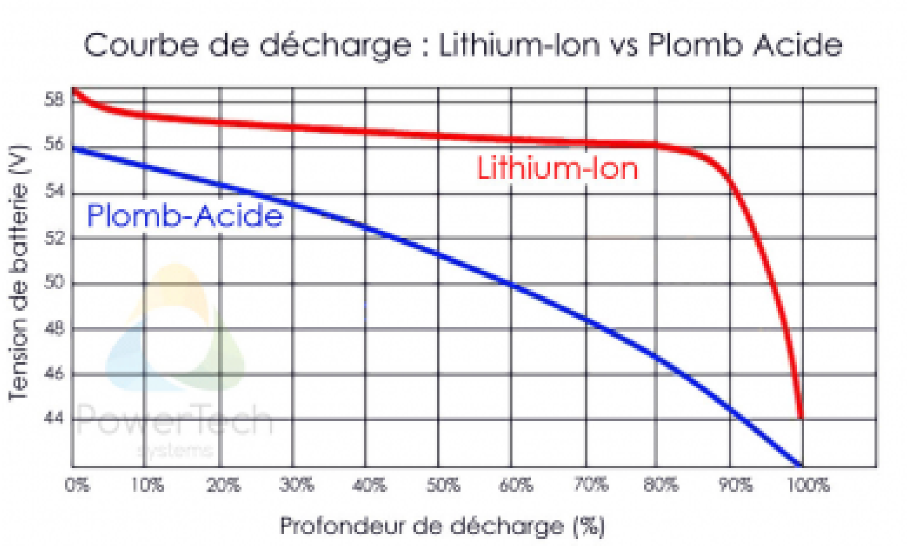

Figure 3.1: Comparison of Discharge Curve: Lithium vs Lead-Acid Battery

Lead-Acid Batteries: The voltage of lead-acid batteries decreases steadily with the depth of discharge (DOD). This means that the voltage is a relatively good indicator of the SOC, as it shows a clear, monotonic decline as the battery discharges.

Lithium Batteries: In contrast, lithium batteries maintain a relatively stable voltage over most of their discharge cycle, with a sharp decline only at the end. This flat discharge curve makes it challenging to estimate SOC based on voltage alone.

3.3.2 Problem with SOC Display in the BEE and B1 Model

The SOC display in the BEE model is problematic because it was designed to interpret the battery voltage curve of lead-acid batteries, not lithium batteries. This leads to several issues:

Figure 3.2: B1 DISPLAY

Inaccurate SOC Readings: The display often shows incorrect SOC values because it expects a steady decline in voltage as the battery discharges. However, with lithium batteries, the voltage remains almost constant until the battery is nearly depleted, leading to SOC readings that do not accurately reflect the remaining battery capacity.

Delayed and Sudden Drops: Drivers might experience sudden drops in SOC readings when the battery voltage finally begins to decline sharply, rather than a gradual decrease. This can be surprising and alarming, leading to range anxiety and mistrust in the vehicle's SOC display.

Misinterpretation of Battery Health: The incorrect SOC display can also cause drivers to misinterpret the health and performance of their battery. This misinterpretation can result in inefficient usage patterns and unnecessary concerns about battery degradation.

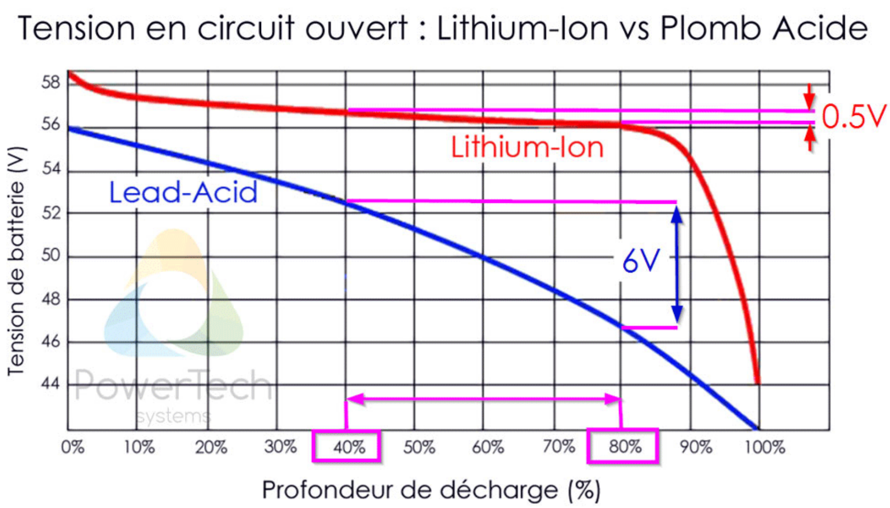

Figure 3.3: Delayed SOC Drops in Electric Vehicles

3.4 Display of SOC in the EVAN Model

The EVAN model faces significant challenges with the accuracy and reliability of its SOC display. One of the primary issues is the inability of the Battery Management System (BMS) to effectively communicate the SOC information to the vehicle's display system.

Figure 3.4: EVAN DISPLAY

3.4.1 SOC Display via CAN:

In the EVAN model, SOC information is intended to be transmitted from the BMS to the display using the Controller Area Network (CAN) protocol. This protocol is critical for ensuring reliable communication between various electronic components within the vehicle.

3.4.2 Communication Breakdown:

The current problem lies in the BMS's inability to send SOC data to the display unit via the CAN network. This communication breakdown means that the display does not receive the necessary SOC information, leading to the absence or inaccuracy of SOC readings shown to the driver.

3.5 the Onboard Charger

3.5.1 Overview of Onboard Charger (OBC) in EVs

The onboard charger (OBC) in electric vehicles is responsible for converting AC (alternating current) from external power sources into DC (direct current) to charge the vehicle's battery. It plays a crucial role in ensuring efficient and safe charging of the EV.

3.5.2 Relevant ISO Standards for OBC

Several ISO standards pertain to the design, safety, performance, and interoperability of onboard chargers in EVs, including:

ISO 15118: Covers communication between EVs and charging stations.

ISO 6469-3: Specifies safety requirements for onboard rechargeable energy storage systems.

ISO 17409: Addresses the electrical safety of the OBC.

3.5.3 Problem Identification

The issue at hand is the non-compliance of the OBC with relevant ISO standards. This non-compliance can manifest in various ways:

Safety Concerns Electrical Hazards: Non-compliance with ISO 6469-3 may lead to risks such as short circuits, overheating, or electrical shock.

Thermal Management: Inadequate thermal management can result in overheating, which is critical in preventing fires and ensuring long-term reliability.

Interoperability Issues Communication Protocols: Failure to meet ISO 15118 standards may cause issues with vehicle-to-grid communication, leading to inefficient charging or incompatibility with certain charging stations.

Standardized Interfaces: Non-compliance can result in proprietary interfaces that limit the usability of public charging infrastructure.

3.5.4 Performance Deficiencies

Charging Efficiency: If the OBC doesn't meet efficiency standards, it can lead to longer charging times and higher energy losses. Voltage and Current Ratings: Incompatibility with ISO standards may mean the OBC cannot handle the required voltage and current ranges, affecting the charging process.

3.5.5 Impact Analysis

Non-compliance with ISO standards has several potential impacts:

Safety Risks: Increased risk of accidents and liabilities. User Experience: Reduced convenience and satisfaction due to charging issues.

Market Acceptance: Negative perception among consumers and regulators.

Regulatory Penalties: Possible fines and restrictions from regulatory bodies.

3.6 lack of Diagnostic System Bako Motors Vehicles

3.6.1 Importance of a Diagnostic System:

A diagnostic system is crucial for continuously monitoring vehicle performance, detecting anomalies, and diagnosing issues in real-time. It plays a vital role in maintaining the reliability and efficiency of an EV.

Bako Motors' vehicles, including the B1 and BEE models, currently lack an integrated diagnostic system. This absence means that potential issues with SOC display, onboard charging, and other critical components are not detected promptly.

3.6.2 Introducing the Diagnostic System as a Proposed Solution

To address these challenges effectively, the implementation of a diagnostic system emerges as a strategic solution. The diagnostic system is envisioned as a sophisticated tool capable of real-time monitoring, analysis, and diagnosis of various vehicle parameters, including SOC accuracy and charging performance.

Chapter 4

Solution Design

Plan

Introduction 45

1 Display Solution Design 45

2 Plug Standardizing 47

3 diagnostic systeme 52

4 System functionality 54

5 project global archirtarchitecture 55

6 Hardware choice 57

7 supply system 64

Conclusion 65

Introduction

This chapter will begin by describing the system's general architecture and functionality, as well as its intended uses and interactions. After that, we will go over the design principles, emphasizing the importance of robustness and efficiency. Lastly, we will discuss the equipment selection procedure, including the judging standards and the reasoning behind our decisions. We will then provide a detailed architecture, including the components. This will provide a clear picture of the system's growth from beginning to end.

4.1 Display Solution Design

In this section, we will explore the design considerations for the display solution. This includes the choice of display technology, resolution, interface, and the overall integration into the system. The goal is to ensure that the display is not only functional but also user-friendly and reliable under various operating conditions.

4.1.1 architecture solution

In this section, we will delve into the design of the display solution for the B1 ,Bee and Evan Vehicle, focusing on the requirements, selection criteria, and integration into the overall system architecture.

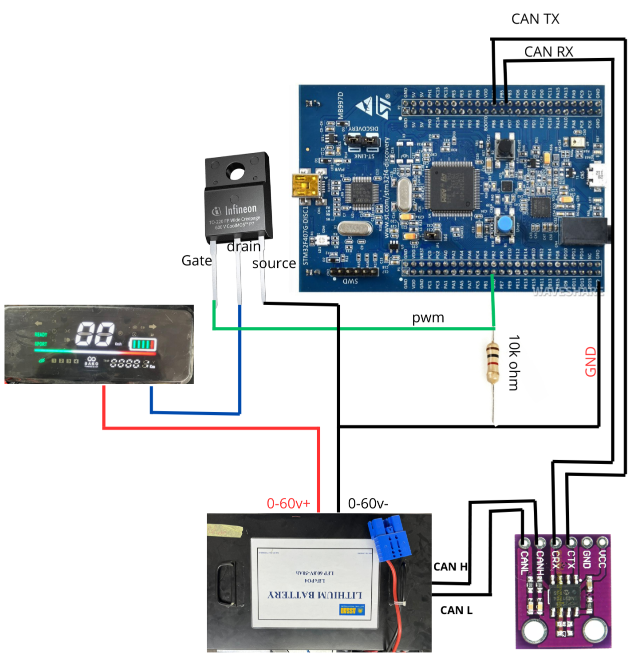

Figure 4.1: display solution architecture for BEE AND B1

This solution involves an STM32 microcontroller communicating with a Battery Management System (BMS) to read the State of Charge (SOC) and voltage levels. The SOC and voltage data are sent over the CAN bus to the STM32, which then calculates the appropriate Pulse Width Modulation (PWM) signal. This PWM signal is used to control a MOSFET, adjusting the voltage level to match the SOC. The display module, powered by the MOSFET, shows the SOC and voltage levels in real-time to the driver. The diagram illustrates the connections between the battery, BMS, CAN transceiver, STM32 microcontroller, MOSFET, and the display, ensuring accurate and dynamic feedback on the vehicle's battery status as seen above in figure 4.1 .

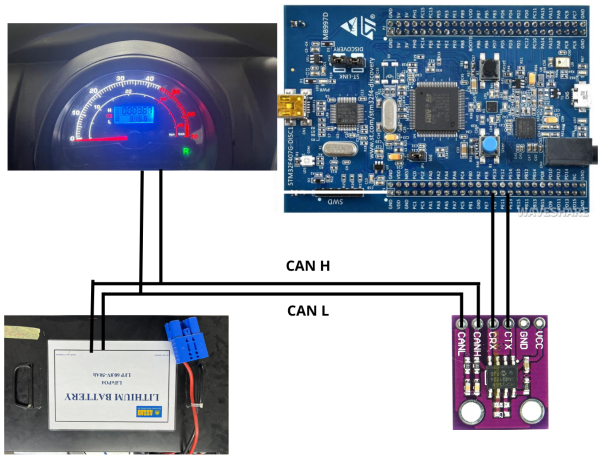

Figure 4.2: display solution architecture for EVAN

This solution for the EVAN Vehicle involves a display system that communicates via the CAN bus to show real-time battery information. The setup integrates an STM32F4 microcontroller and a CAN transceiver to read the State of Charge (SOC) from the battery. The SOC data is transmitted over the CAN bus to the display, ensuring that the driver receives accurate and timely updates on the battery status. The diagram illustrates the connections between the battery, CAN transceiver, STM32F4 microcontroller, and the display, highlighting the flow of data from the battery to the display through the microcontroller and CAN bus as seen above in figure 4.2 .

4.1.2 solution organizational chart

This diagram visually represents the sequence of steps described in figure 4.3 . Starting with the initialization of the display and CAN Bus communication, the system reads crucial battery information, processes it to generate the correct PWM signal, and updates the display in real-time. This ensures that the driver always has access to accurate and current information regarding the vehicle's battery status.

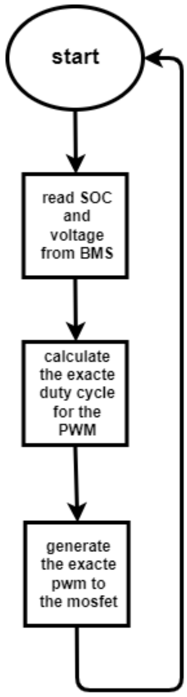

Figure 4.3: Solution organizational chart for bee and B1

4.2 Plug Standardizing

This section covers the standardization of plugs within the system. We will discuss the types of plugs used, their specifications, and the rationale behind their selection. Standardizing plugs ensures compatibility, ease of maintenance, and future scalability.

4.2.1 Type 2 Connector (Mennekes)

Description: The Type 2 connector, also known as the Mennekes connector, is predominantly used in Europe for AC charging of electric vehicles.

Applications: Suitable for both public and private charging stations, and supports single-phase and three-phase charging.

4.2.2 Specifications of Type 2 Plugs

Pin Configuration

Power Pins:

- **L1, L2, L3:** Phase conductors for AC power.

- **N:** Neutral conductor.
- **PE:** Protective Earth for grounding.

Communication Pins:

- **PP:** Proximity Pilot for cable detection and current rating.
- **CP:** Control Pilot for communication between the EV and the charging station.

Electrical Ratings

- **Voltage:** Up to 500V AC.
- **Current:** Typically supports up to 32A (single-phase) or 63A (three-phase).

Physical Characteristics

- **Dimensions:** Standardized sizes and shapes to ensure compatibility.
- **Durability:** Designed to withstand repeated use and environmental factors such as temperature and humidity.

4.2.3 Rationale Behind Selection

Compatibility

- **Interoperability:** Standardized plugs like Type 2 ensure that EVs from different manufacturers can use the same charging infrastructure.
- **Global Adoption:** Type 2 connectors are widely adopted across Europe, facilitating cross-border travel and charging.

Safety

- **Enhanced Safety Features:** Type 2 plugs incorporate safety features such as interlocks and communication protocols to prevent electrical hazards.
- **Compliance with Regulations:** Adhering to the IEC 62196 standard ensures compliance with international safety and performance regulations.

Ease of Maintenance

- **Uniform Design:** Standardized designs simplify maintenance and repair processes.
- **Availability of Spare Parts:** Standardization ensures that spare parts are readily available and interchangeable.

Future Scalability

- **Support for Higher Power Levels:** The design of Type 2 connectors allows for future upgrades to support higher charging power levels.
- **Technology Integration:** Standardized plugs facilitate the integration of new technologies such as smart charging and vehicle-to-grid (V2G) systems.

4.2.4 Pin Configuration and Communication Process

Type 2 Plug Pin Layout

Power Pins:

- **L1, L2, L3 (Line Conductors):** These carry the alternating current (AC) power.
- **N (Neutral Conductor):** Completes the circuit for the AC power.
- **PE (Protective Earth):** Provides grounding for safety.

Communication Pins:

- **PP (Proximity Pilot):** Used for cable detection and current rating.
- **CP (Control Pilot):** Facilitates communication between the EV and the charging station.

4.2.5 Communication Protocols

Proximity Pilot (PP) Communication

- **Purpose:** To detect the presence of the charging cable and determine its maximum current-carrying capacity.
- **Mechanism:** A resistor inside the plug connects between the PP pin and ground. The charging station measures the resistance value to identify the maximum current rating of the cable and adjusts the current supply accordingly.
- **Resistor Values:**

- **220 :** 32A
- **680 :** 20A
- **1.5 k:** 13A

Control Pilot (CP) Communication

- **Purpose:** To manage and control the charging process, ensuring safe and efficient power transfer between the charging station and the EV.
- **Mechanism:** The CP pin initially carries a +12V signal when the cable is not connected. When the cable is connected to the EV, the voltage on the CP pin drops to +9V, indicating to the charging station that an EV is connected.
- **PWM Signal:** The charging station generates a 1 kHz Pulse Width Modulation (PWM) signal on the CP pin to communicate with the EV.
- **PWM Duty Cycle:**
  - **10% duty cycle:** 6A
  - **50% duty cycle:** 30A
  - **90% duty cycle:** 54A
- **Voltage Levels:**
  - **+12V:** EV not connected
  - **+9V:** EV connected, not ready
  - **+6V:** EV connected, ready to charge
  - **+3V:** Fault condition

4.2.6 Communication with STM32 Microcontroller

The communication between the charging station and the STM32 microcontroller in the electric vehicle (EV) is facilitated through the Control Pilot (CP) pin using a Pulse Width Modulation (PWM) signal. This process is crucial for determining the allowable charging current and ensuring a safe and efficient charging process.

Initial Connection:

- When the EV is connected to the charging station, the CP pin initially carries a +12V signal.
- The EV signals its readiness by pulling the CP voltage down to +9V using a 2.74 k resistor in parallel with a diode.

PWM Signal Transmission:

- The charging station then generates a 1 kHz PWM signal on the CP pin.
- The duty cycle of this PWM signal indicates the maximum current the charging station can provide. For instance, a 10% duty cycle might indicate a maximum of 6A, while a 50% duty cycle might indicate up to 30A.

Signal Capture by STM32:

- The CP signal is connected to a Timer/Capture Compare input on the STM32.
- The STM32's timer module captures the high and low times of the PWM signal to determine the duty cycle.

Duty Cycle Calculation:

- The STM32 reads the duration of the high and low periods of the PWM signal.
- It calculates the duty cycle by dividing the high time by the total period (high time + low time).

Current Regulation:

- Based on the calculated duty cycle, the STM32 determines the maximum current it can draw from the charging station.
- The EV's onboard charging system adjusts the charging current accordingly to match the station's signal.

Safety Features:

- The STM32 ensures that the charging process adheres to safety protocols, such as stopping the charge if the CP signal indicates a fault or disconnection.

4.2.7 Architecture Diagram

Below is a simplified diagram illustrating the pin layout of a Type 2 plug, the communication process between the charging station and the electric vehicle (EV), and the flow of information via the Proximity Pilot (PP) and Control Pilot (CP) pins, interfacing with an STM32 microcontroller.

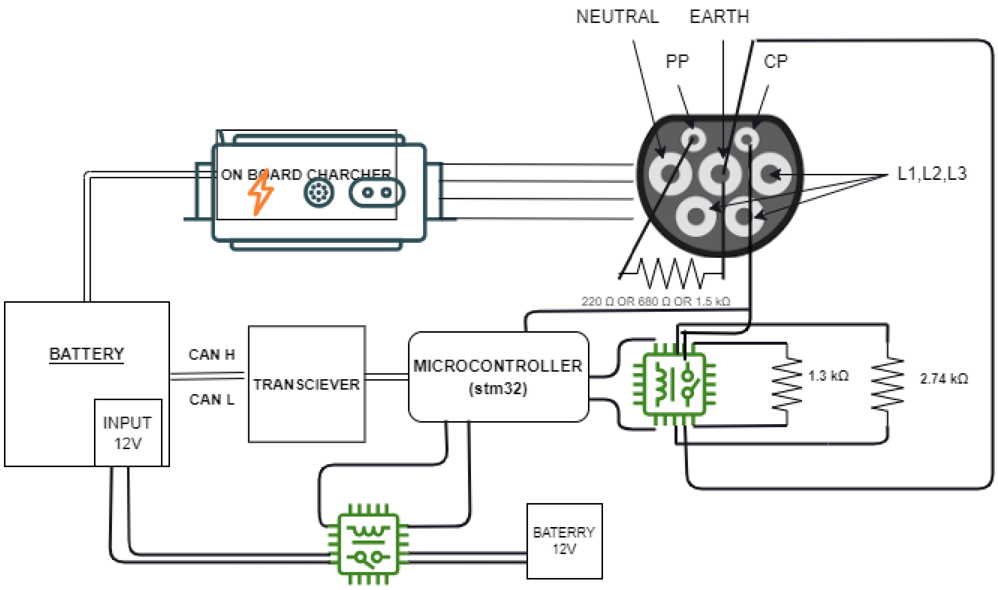

Figure 4.4: Type 2 plug architecture

The transition between the discharge mode and the charging mode of the battery is regulated by a 12V signal sent to the battery. This process is facilitated by a small 12V battery located at the front of the car, which helps control the main battery.

4.3 diagnostic systeme

In this section, we will detail the diagnostic system implemented within the architecture. This includes the methods and tools used for monitoring system health, identifying faults, and performing troubleshooting. The focus will be on creating a system that is proactive in detecting issues and efficient in addressing them.

The diagnostic system leverages the capabilities of the STM32 microcontroller integrated with a CAN (Controller Area Network) transceiver to monitor the state of the Battery Management System (BMS) and the motor controller. The diagnostic system is designed with the following key components:

CAN Bus Communication: The CAN bus is used to facilitate communication between the BMS, motor controller, and the STM32 microcontroller. The CAN bus transmits crucial data such as battery voltage, current, state of charge (SOC), temperature readings, and motor controller status. This data is continuously monitored to ensure the optimal performance and safety of both the battery pack and the motor system.

Data Acquisition and Processing: The STM32 microcontroller, equipped with an MCP2515 CAN transceiver, receives and decodes the CAN messages from both the BMS and the motor controller. The received data includes:

- **Battery Data (ID 0x90 - 0x94, 0x98):** Information such as total voltage, gathered total voltage, current, SOC, cell voltage extremes, temperature extremes, MOS status, general status, and failure status.

- **Motor Controller Data (Specific IDs):** Includes motor speed, torque, temperature, and operational status.

Real-Time Monitoring and Alerts: The system performs real-time monitoring of the battery pack and motor controller's health by analyzing the data received via the CAN bus. If any parameter exceeds predefined safe thresholds (e.g., over-voltage, under-voltage, over-temperature for the battery, or abnormal speed/torque for the motor), the system generates alerts. These alerts can trigger visual indicators on the display, send warning messages via the CAN bus, or log the events for further analysis.

Troubleshooting and Fault Isolation: When a fault is detected, the diagnostic system provides detailed information to help isolate the issue. For instance, if an over-temperature condition is detected in the battery pack, the system logs the exact cell and temperature value. Similarly, motor controller faults such as overcurrent or overtemperature are logged with precise details, enabling quick identification and resolution of the problem.

Proactive Maintenance: By continuously monitoring and logging the operational data of both the battery pack and the motor controller, the diagnostic system helps in scheduling proactive maintenance. Historical data analysis can identify patterns indicating potential future failures, allowing maintenance teams to address issues before they lead to significant problems.

4.4 System functionality

In this section, we will describe the overall functionality of the system. This includes the primary operations it performs, how users interact with it, and the expected outcomes. The functionality will be mapped to the initial requirements and intended use cases, highlighting how the system meets its objectives.

4.4.1 diagnostic architecture

The data flow from the vehicle's diagnostic system to the user interface is highlighted in this architecture, which also emphasizes the functions of the CAN bus, the microcontroller, and the display of data in accomplishing this integration.

The figure 4.5 shows the Project's global architecture.

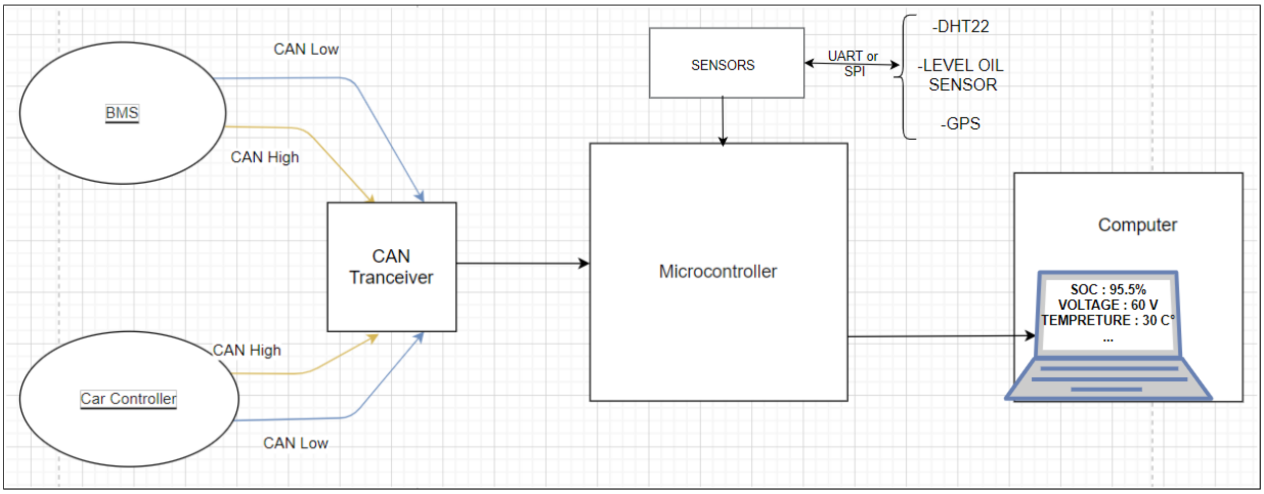

Figure 4.5: The project's global architecture

4.4.2 System organizational chart

This organizational chart shows how the microcontroller firmware for the electric car diagnostic system manages control and data flows. It demonstrates communication between the microcontroller and the CAN bus, where incoming CAN messages cause the microcontroller to transmit a response message. The microcontroller then communicates with the serial monitor via UART, displaying key data such as SOC, total voltage, current, motor speed, and any active fault conditions. This approach ensures that operators can quickly assess the status of both the battery pack and the motor controller by viewing the data on the serial monitor and taking appropriate actions.

The figure 4.6 represents the organizational chart of the microcontroller.

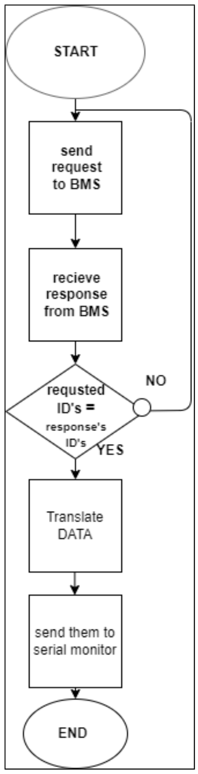

Figure 4.6: Application organizational chart

4.4.3 Perspective: User Interface Development

The system includes a user-friendly interface that displays key information such as SOC, total voltage, current, motor speed, and any active fault conditions. This interface ensures that operators can quickly assess the status of both the battery pack and the motor controller and take appropriate actions. The development of this interface will be handled by our team member, who will utilize QT for creating an intuitive and efficient user interface. This interface will play a crucial role in enabling users to monitor the system effectively and respond promptly to any alerts or issues.

4.5 project global archirtarchitecture

The project architecture addresses multiple key aspects of an electric vehicle (EV) system, focusing on display integration, standardizing EV plugs, and implementing a robust diagnostic system. The display problem is tackled by utilizing an STM32 microcontroller and a CAN transceiver to read the State of Charge (SOC) and other critical data from the Battery Management System (BMS) and the motor controller, displaying it via UART on a serial monitor. To standardize the EV plug, we ensure compatibility and reliability across various charging stations. The diagnostic system continuously monitors the health of the vehicle, identifies faults, and performs troubleshooting, enhancing the overall reliability and efficiency of the EV.as seen in figure 4.7

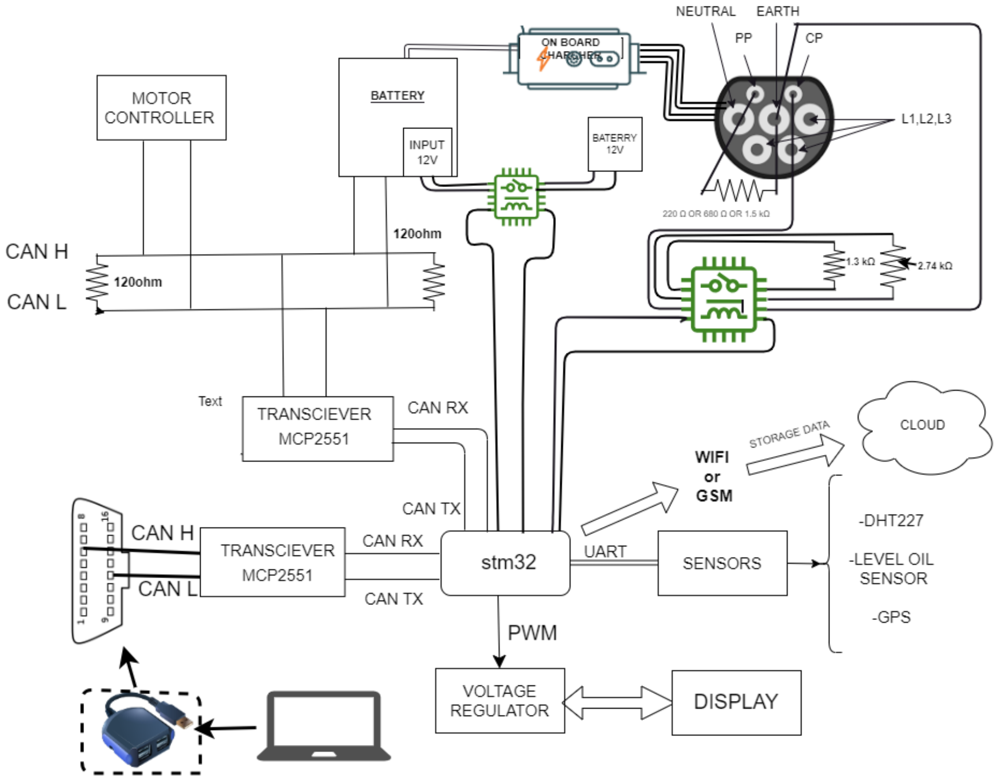

Figure 4.7: project global archirtarchitecture

Display Solution The system addresses display issues by using the STM32 to read data from the BMS and motor controller via the CAN bus. This data is displayed on the serial monitor, allowing drivers to monitor SOC, voltage, and current easily.

Standardized Plug We ensure compatibility with a wide range of charging stations by incorporating a standardized EV charging plug with proper wiring and resistance configurations.

Diagnostic System The STM32 continuously monitors the vehicle's battery and motor controller, identifying faults and logging diagnostic information. This data is sent to the serial monitor for quick assessment.

Role of STM32

- Acquires and processes data from the CAN bus.
- Communicates with the display via UART.
- Manages diagnostics and fault detection.
- Sends data to the cloud via GSM or Wi-Fi for remote monitoring.

4.5.1 Cloud Communication

GSM or Wi-Fi modules enable the transmission of data to the cloud, allowing remote monitoring of temperature, oil levels, GPS data, and overall vehicle health.

4.6 Hardware choice

This section provides a comprehensive overview of the critical hardware components required for the project's success. The selected hardware has been carefully vetted to assure compatibility and optimal performance, and it meets strict criteria such as durability, dependability, and environmental compatibility. Efficiency is important in our hardware selection process, which prioritizes performance and functionality above peripherals and processors. By emphasizing the important features critical to the project's success, we present a clear overview of the hardware's importance inside our system architecture.

4.6.1 Micocontroller

Certainly! Here is a paragraph describing the reasoning behind using microcontrollers, which you can place before your table:

Microcontrollers are integral components in modern electronic design due to their compact size, versatility, and cost-effectiveness. They serve as the brains of embedded systems, enabling complex functionalities within a minimal footprint. By integrating processors, memory, and input/output peripherals on a single chip, microcontrollers facilitate the development of efficient and reliable electronic devices. Their wide range of applications spans from simple consumer electronics to advanced industrial automation and Internet of Things (IoT) solutions. The choice of microcontroller can significantly influence the performance, power consumption, and development complexity of a project. Therefore, selecting the appropriate microcontroller requires careful consideration of factors such as processing power, memory capacity, connectivity options, and community support. The following table compares three popular microcontroller families—Arduino, ESP, and STM32—highlighting their key features and use cases to aid in making an informed decision. Table 4.1 describes the reasoning behind the microcontroller choice.

Table 4.1: Microcontroller selection and explanation

Microcontroller			
Arduino	ESP	STM32	Explanation
Arduino is a fantastic platform for novices and enthusiasts because of its simplicity and ease of usage. Numerous internet tools and a sizable community offer plenty of help. Performance might be hampered by its reduced memory and processing power, though.	ESP32 is perfect for Internet of Things applications because of its superior wireless connectivity. Real-time operations and multitasking are made possible by its dual-core processor, integrated Wi-Fi, and Bluetooth. But in contrast to Arduino, it costs more and has less community support.	High performance and versatility are well-known characteristics of STM32 microcontrollers. They provide a wide variety of devices with different features and a powerful processing capacity. Comparing them to more straightforward microcontrollers, they are more expensive and have a steeper learning curve.	**The STM32** microcontroller excelled for its powerful performance, adaptability, and ability to handle complex tasks. With diverse devices offering varying features and processing capabilities, it provided the necessary horsepower for sophisticated applications. Its robust architecture and extensive peripherals enabled seamless integration with additional hardware and sensors for comprehensive data processing. Additionally, the STM32's comprehensive ecosystem, including documentation, tools, and community support, streamlined development and ensured reliability.

Because of its extensive peripheral set and ARM Cortex-M4 CPU, the STM32F407G-DISC1 development board, which is seen in figure 4.8, is perfect for creating a variety of applications, including real-time control systems and Internet of Things devices. With a wealth of documentation and software tools[3].

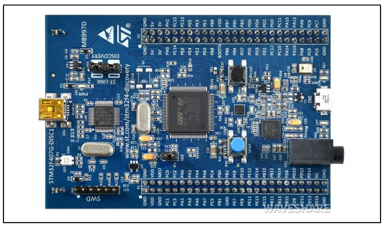

Figure 4.8: The STM32F407G-DISC1 development board

4.6.2 Tranceiver

Table 4.2 describes the reasoning behind the tranceiver choice.

Table 4.2: Tranceiver selection and explanation

Tranceiver		
MCP2551	MCP2515	Explanation
The MCP2551 is a CAN transceiver commonly used in automotive and industrial applications. It operates at 5 volts, matching the voltage levels typically used in these environments. The MCP2551 interfaces directly with the CAN protocol controller and provides differential transmit and receive capabilities for the CAN protocol..	The MCP2515 is a commonly used CAN controller with an SPI interface that works with Arduino and STM32. The 5V working voltage, low standby current (1 A), and wide temperature range (-40°C to +125°C) make it ideal for a variety of applications. The built-in termination resistor simplifies hardware setup while ensuring signal integrity.	We chose to use both the MCP2551 and MCP2515 transceivers for different purposes within our system. The MCP2551 was selected for its direct interface with the STM32 microcontroller's built-in CAN protocol, which reduces complexity and the number of components required. This choice ensures robust and reliable communication for automotive applications.

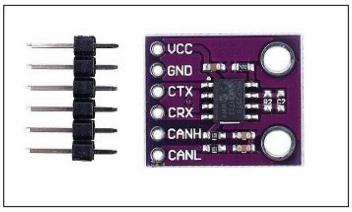

Figure 4.9: MCP2551 Tranceiver

On the other hand, the MCP2515 was utilized for testing and simulation purposes with microcontrollers that do not have an integrated CAN controller, such as the Arduino. The MCP2515's compatibility with SPI interfaces and its integrated features, such as the built-in termination resistor, make it a suitable choice for these scenarios. Figure 4.10 illustrates this. This ensures that embedded systems will integrate and communicate effectively. [4].

Figure 4.10: MCP2515 Tranceiver

4.6.3 voltage regulator

The IRF530N MOSFET was chosen to address the BEE model's voltage regulation needs and SOC display errors because of its ability to handle both current and voltage with steadiness.

Selection Criteria The decision to choose the IRF530N MOSFET was guided by specific criteria:

- **Voltage Regulation:** Ensures consistent and stable voltage output throughout the battery discharge cycle.
- **Current Handling:** Capable of managing the required current levels efficiently.
- **Reliability:** Demonstrates robust performance under varying load conditions, crucial for sustained vehicle operation.

Figure 4.11: mosfet

Chosen MOSFET: IRF530N The IRF530N MOSFET was chosen due to its key attributes:

- **Voltage Stability:** Maintains reliable voltage levels critical for accurate SOC readings.
- **Current Capacity:** Handles required current levels effectively, minimizing power losses.
- **Reliability:** Known for its durability and performance under demanding operational conditions.

Impact on SOC Display Integration of the IRF530N MOSFET has significantly improved SOC display accuracy:

- **Stable Voltage Output:** Minimizes fluctuations, ensuring precise SOC calculations.
- **Enhanced Current Management:** Supports efficient battery management and charging.
- **Improved User Confidence:** Provides reliable information for effective trip planning and usage.

4.6.4 RELAY module

In our project, we used the 5V 4-Channel Relay Module to standardize the plug connections. This relay module is crucial for switching between different resistors in the plug, ensuring the correct resistance value is selected based on the required configuration. The module operates at 5 volts, making it compatible with most microcontrollers, and it can handle multiple channels simultaneously, providing flexibility and reliability in managing the electrical connections. This allows for automated and precise switching, enhancing the overall efficiency and safety of the system. as seen in the figure below 4.12:

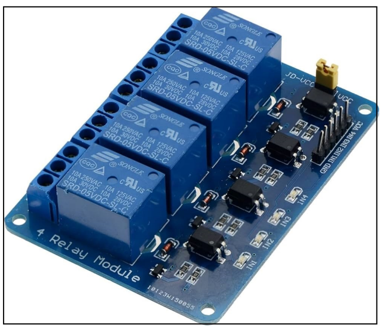

Figure 4.12: RELAY module

4.6.5 USB module

Table 4.3 describes the USB module benefits.

Table 4.3: USB module selection

USB Module (CP2102)	
Benefits	Explanation
USB connectivity	The diagnostic tool and a PC may easily connect thanks to the USB module, allowing for data exchange.
Easy integration	Its common USB interface makes integration with STM32 microcontrollers and other components simple.
Plug-and-play compatibility	This feature allows regular USB ports on PCs to be used without the need for extra drivers.
Wide compatibility	Broad interoperability with technicians' PCs is ensured by its compatibility with several operating systems, including Windows, macOS, and Linux.

The CP2102 USB module (see figure 4.13) offers seamless USB-to-serial communication in a compact size. Its plug-and-play simplicity and UART interface make it perfect for connecting microcontrollers to USB-equipped devices. [5].

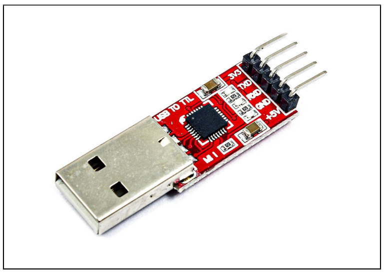

Figure 4.13: The CP2102 usb module

4.6.6 GSM module

The GSM module in the car is primarily responsible for enabling remote monitoring and real-time data transfer. It gathers information from the STM32 microcontroller and a range of sensors, including .

Figure 4.14: GSM module

Battery Status: State of Charge (SOC), voltage, current, and temperature. **Motor Controller Data:** Performance metrics and fault conditions. **Sensor Data:** Environmental conditions (e.g., temperature, humidity) and vehicle-specific information (e.g., oil level, GPS location). After that, this data is sent to a cloud server so that it may be remotely accessed, saved, and examined. In order to facilitate preventative maintenance and troubleshooting, the GSM module makes sure that all relevant parties can track the health and performance of the vehicle in real-time.

4.6.7 Detailed architecture

The diagnostic tool's communication structure is illustrated in the system architecture diagram, which also highlights the interaction between hardware components and protocols. The core of the architecture is the STM32F407G-MCU, which contains the diagnostic tool software responsible for communicating with the vehicle's electronic control units (ECUs). After the hardware selection is complete, the system architecture and wiring are shown in figure 4.15.

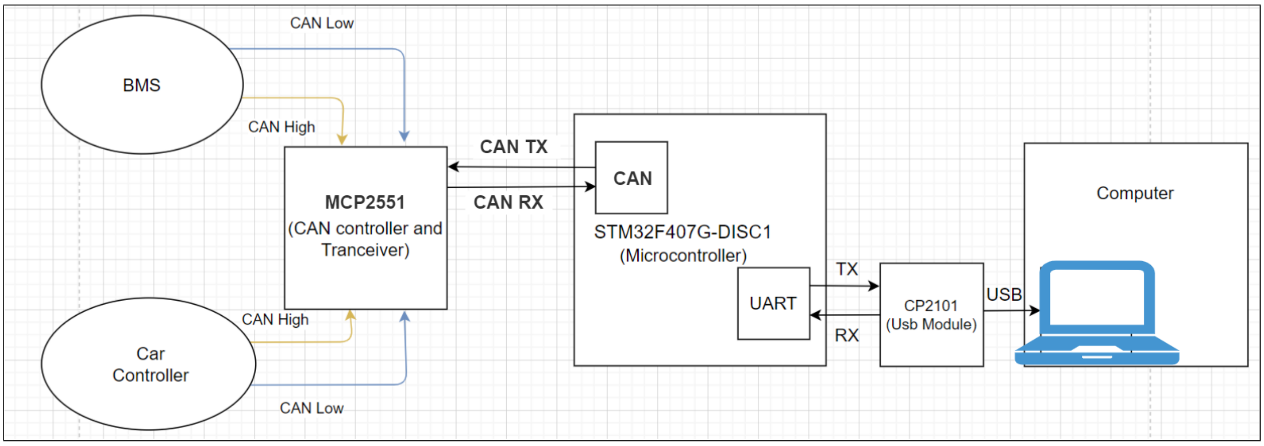

Figure 4.15: Detailed architecture of the diagnostic system

- **Transceiver for CAN:** Facilitates communication across the controller area network (CAN) bus to enable bidirectional message exchange between the diagnostic tool and various automotive ECUs.

- **Battery management system (BMS) controller and ECUs:** The CAN bus serves as the primary means of communication between the car's ECUs and the BMS controller. This connectivity enables tasks like sending diagnostic commands and retrieving sensor data.

- **Interface UART:** provides an additional communication channel that may be used to communicate with other modules or gadgets, such as a display unit or additional sensors.

- **Use CP2102 USB Module with Serial Monitor:** allows the diagnostic gadget and a PC to communicate via USB. With this connection, the serial monitor—which offers a simple interface for displaying real-time data—can be used to log and monitor data. This configuration improves user engagement and data logging capabilities by giving operators easy access to important data including SOC, total voltage, current, motor speed, and any active fault states directly through the serial display.

4.7 supply system

The power supply system is a critical component for ensuring the reliable operation of the STM32F4 microcontroller and its peripherals in the diagnostic system for the electric car. This section details the design and implementation of the power supply system.

4.7.1 Power Requirements

The STM32F4 microcontroller board used in this diagnostic system requires a 12V power supply and consumes 250mA. This voltage is provided by the car's onboard DC/DC converter. Direct Connection: The STM32F4 board can be directly connected to the 12V output of this DC/DC converter but to ensure the safe and reliable operation of the STM32F4 board, additional protection components are necessary. These include fuses, transient voltage suppression (TVS) diodes, and reverse polarity protection diodes.

Conclusion

The fundamental elements of the system, such as its functioning, design philosophies, and the equipment selection procedure, have all been covered in this chapter. We have created a strong foundation for the system's implementation by comprehending these constituents. We will examine the testing and implementation process in the upcoming chapter, where we will verify the system's functionality and make sure it satisfies the necessary requirements.

Chapter 5

Implementation and Test

Plan

Introduction 67

1 Software and hardware environment 67

2 Arduino emulation 69

3 Embedded application development 70

Introduction

We will examine the main facets of our project in this chapter, beginning with the software environment. After that, we'll emulate with an arduino board. Ultimately, we will showcase the outcomes and verification of our labor after explaining the development of the QT interface and the embedded application. Each part focuses on important development milestones that resulted in the production of an all-inclusive electric car diagnostic tool.

5.1 Software and hardware environment

Modern engineering projects require a robust combination of hardware and software tools to ensure efficient development and reliable operation. This section outlines the essential components selected for the project, focusing on their compatibility, performance optimization, and support for project-specific requirements.

5.1.1 Hardware Components

- **Oscilloscope:** Used for debugging and analyzing electrical signals in real-time, crucial for verifying signal integrity and troubleshooting.

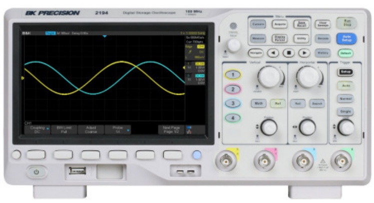

Figure 5.1: Oscilloscope

- **Multimeter:** Essential for measuring electrical quantities such as voltage, current, and resistance, aiding in hardware debugging and verification.

Figure 5.2: Multimeter

5.1.2 Software Tools

- **Cube IDE:** An integrated development environment (IDE) designed specifically for the development of STM32 microcontrollers. It streamlines the development process by offering complete code editing, debugging, and project management tools.

Figure 5.3: STM32CubeIDE

- **Visual Studio Code (VSCode):** A light-weight, multipurpose source code editor that works with many different languages. Because it can be extended using community-provided extensions, VSCode is used to increase coding productivity.

Figure 5.4: VS code

- **Draw.io:** an online tool for making flowcharts and diagrams. It is useful for displaying communication protocols, design processes, and system architecture.

Figure 5.5: Draw.io

- **Arduino IDE:**used for testing and quick prototyping with microcontrollers that are compatible with Arduino. It provides a user-friendly environment and broad library support, making software component testing and iterations quick and easy.

Figure 5.6: Arduino IDE

- **FreeRTOS:** A real-time operating system kernel for embedded devices. FreeRTOS manages tasks, memory, and timing, crucial for coordinating multiple tasks on the STM32 microcontroller efficiently.

Figure 5.7: FreeRTOS

The combination of these hardware tools and software environments ensures a robust development and testing framework, enabling smooth integration and reliable performance of the project.

5.2 Arduino emulation

This section contains a detailed description of the systematic methodology we used to construct our product. First, we replicated a basic communication situation using two Arduino microcontrollers. Next, we moved from using an Arduino microcontroller to an STM32 microcontroller. To facilitate the transition to STM microcontrollers and to better grasp the communication protocols with the car's battery management system (BMS), we redesigned our testing approach. We learned a lot from this first phase, which paved the way for us to move to the STM32 platform and get closer to the project's ultimate objectives.

- **Experiment setup:**
  - Hardware: There is a direct connection to the BMS, an Arduino microcontroller, and an MCP2551 CAN controller employed.
  - Configuration: Data retrieval and analysis were made possible by the communication setup between the Arduino and BMS. The MCP2515 ensured that the BMS protocol was compliant by allowing CAN communication.

    * Purpose: We were able to investigate the intricacies of BMS communication and assess the feasibility of a seamless transition to STM microcontrollers using this setup.

- **Insights Gained:**
    * Knowledge of BMS Communication:The direct connection to the BMS helped us better understand data transmission and reception and provided valuable insights into its communication methods.
    * Enhanced transition preparation: By collaborating closely with the BMS and building a solid framework for the integration of STM microcontrollers into the system, we expedited the changeover process.

As seen in figure 5.8, we immediately coupled an Arduino and an MCP2551 with the BMS in order to read BMS frames and analyse communication patterns. With this arrangement, we want to provide additional insight into BMS data transmission and reception by accurately modelling real-world situations.

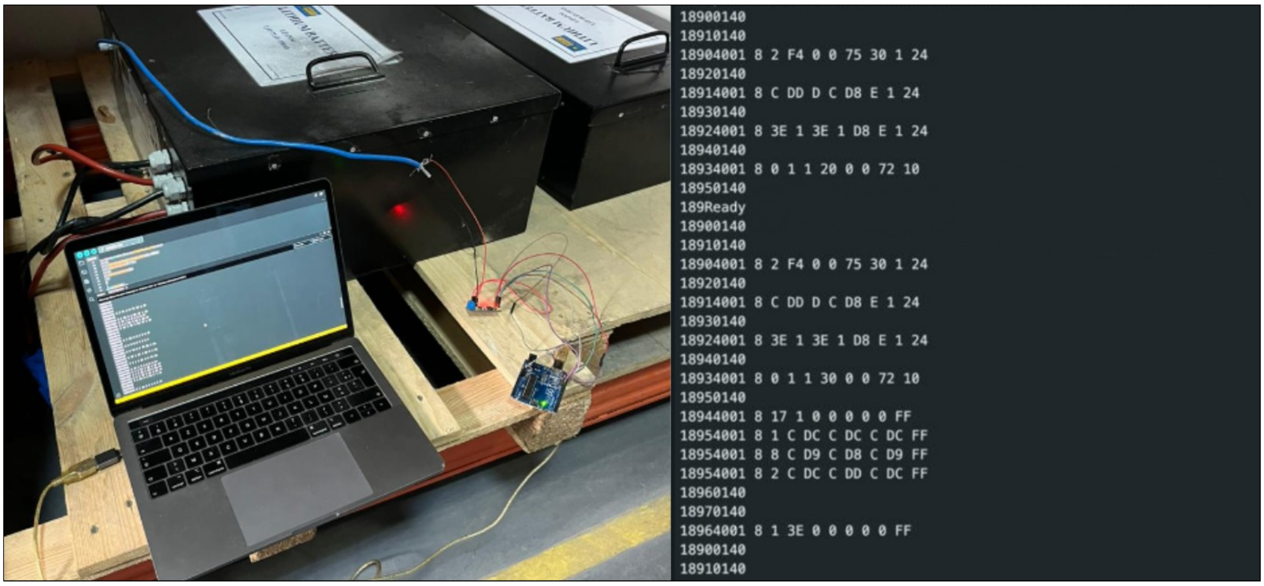

Figure 5.8: Arduino and MCP2515 connected to the BMS, and the received frames

5.3 Embedded application development

The "Vehicle Integration" section will look at how the Battery Management System (BMS) and microcontroller code are integrated. The SOC display issue, charging plug standardisation, and diagnostic system implementation will be the main topics of this part. These components are essential for permitting human control, facilitating real-time data access, and guaranteeing efficient connection with the car's electrical systems.

5.3.1 STM32 connection with BMS system

The interface between the STM32 microcontroller and the CAN transceiver allows messages to be sent and received on the CAN bus. This communication channel allows the BMS or the vehicle's Controller to send sensor data to the STM32.
Certainly! Here's a revised and completed version of your section:

5.3.1.1 CAN message reception

By serving as an essential link in the communication chain connecting the microcontroller and the CAN bus, this function makes it possible to receive CAN messages. It painstakingly gathers crucial data from communications, such as the message ID, data length, and actual content. After that, it saves the extracted data in a structured manner so that it may be processed, this is indicated by the 'tempCanMsg' variable.

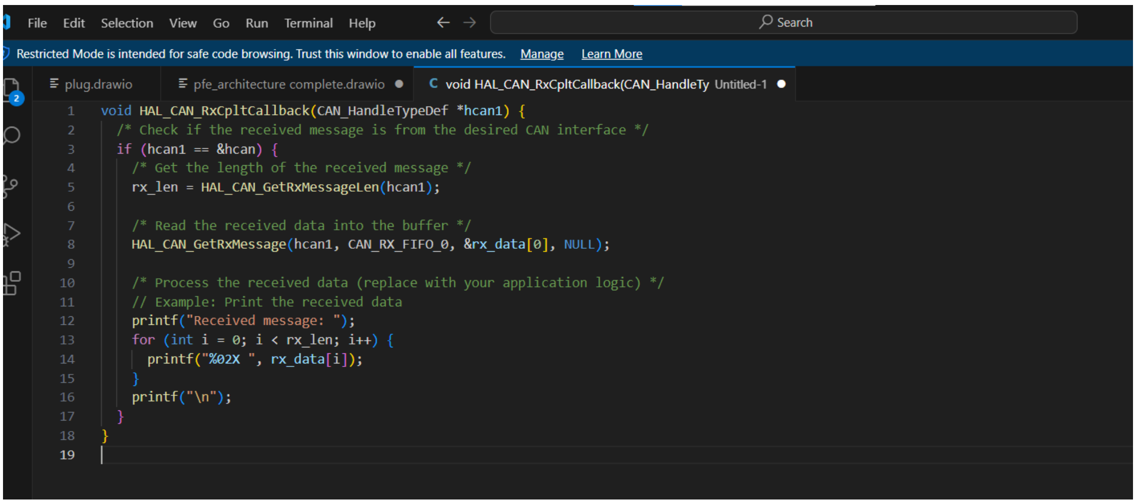

Figure 5.9: The STM32 getting the received frame

- **Message reception in buffer:** The debug environment is depicted in Figure 5.10, which also shows how data is received in the STM32 microcontroller during the debugging process. When debugging the STM32 microcontroller code, the IDE displays data structures and variable values in real time. The IDE's debug window is seen in this context, with the received CAN message displayed in the 'rxMessage' variable. The 'rxMessage' variable holds the frame ID and data bytes that were received from the BMS. Validating and troubleshooting the data reception process is made simpler by this visualisation, which allows one to look closely at the received CAN message during runtime. We were able to verify the accuracy of the data received and the appropriateness of the contact with the BMS.

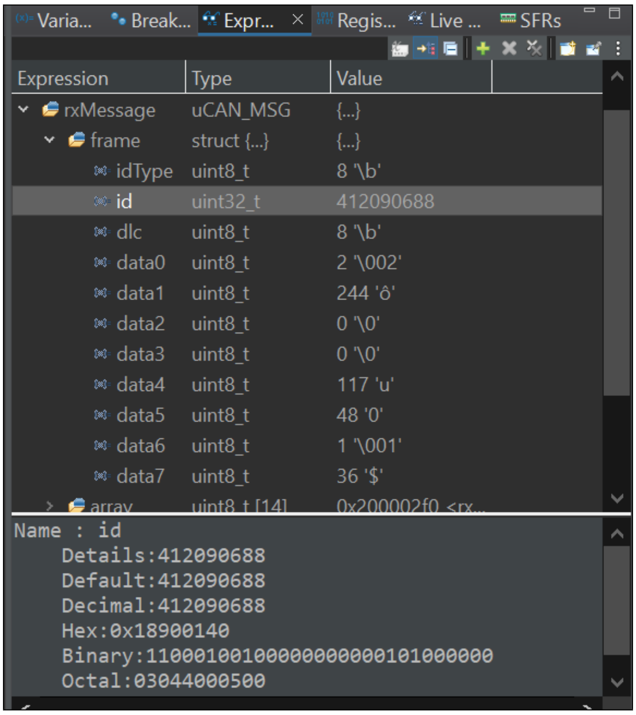

Figure 5.10: The received frame 'rxMessage' from the BMS

5.3.1.2 CAN message translation

When a CAN message is received, the data is extracted from the message frame and stored in an array for additional processing. According to the table example, each ID has a matching set of parameters, therefore the translation process varies depending on the frame ID. 5.1.

Table 5.1: Manufacturer translation manual example

The data	Data ID	Explanation
SOC of total voltage current	0x90	Byte 0 - Byte1: Cumulative total voltage (0.1V) Byte2 - Byte3: Gather total voltage (0.1V) Byte4 - Byte5: Current (30000 offset, 0.1A) Byte6 - Byte7: SOC (0.1percent)

For instance, when the frame ID is 0x18904001 as seen in the figure 5.11, the translation process entails collecting data bytes and converting them into usable information, such as cumulative voltage, aggregate voltage, current, and State of Charge (SOC). These values are formatted into strings and then delivered via UART for further processing or display. It is important to keep in mind that different frame IDs acquired from the BMS may be translated using the same procedures, which enables comprehensive battery parameter monitoring and analysis.

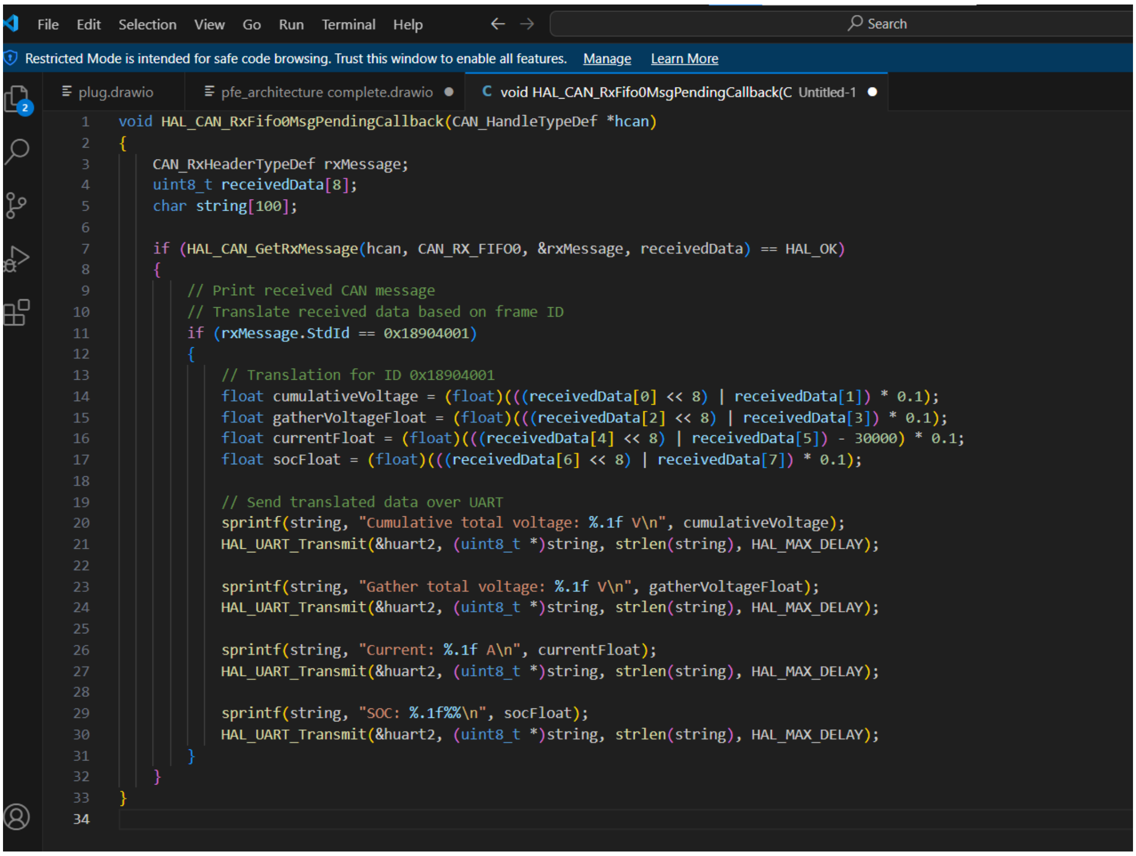

Figure 5.11: STM32 translation of the data received

5.3.1.3 CAN message transmission

This function prepares and sends a specified CAN message. More generally, it is a component of a larger system that manages the communication between different elements in an industrial or automotive environment. This code enhances the overall efficacy and usefulness of the system by facilitating real-time data exchange, control instructions, and system monitoring through the configuration and transmission of CAN messages.

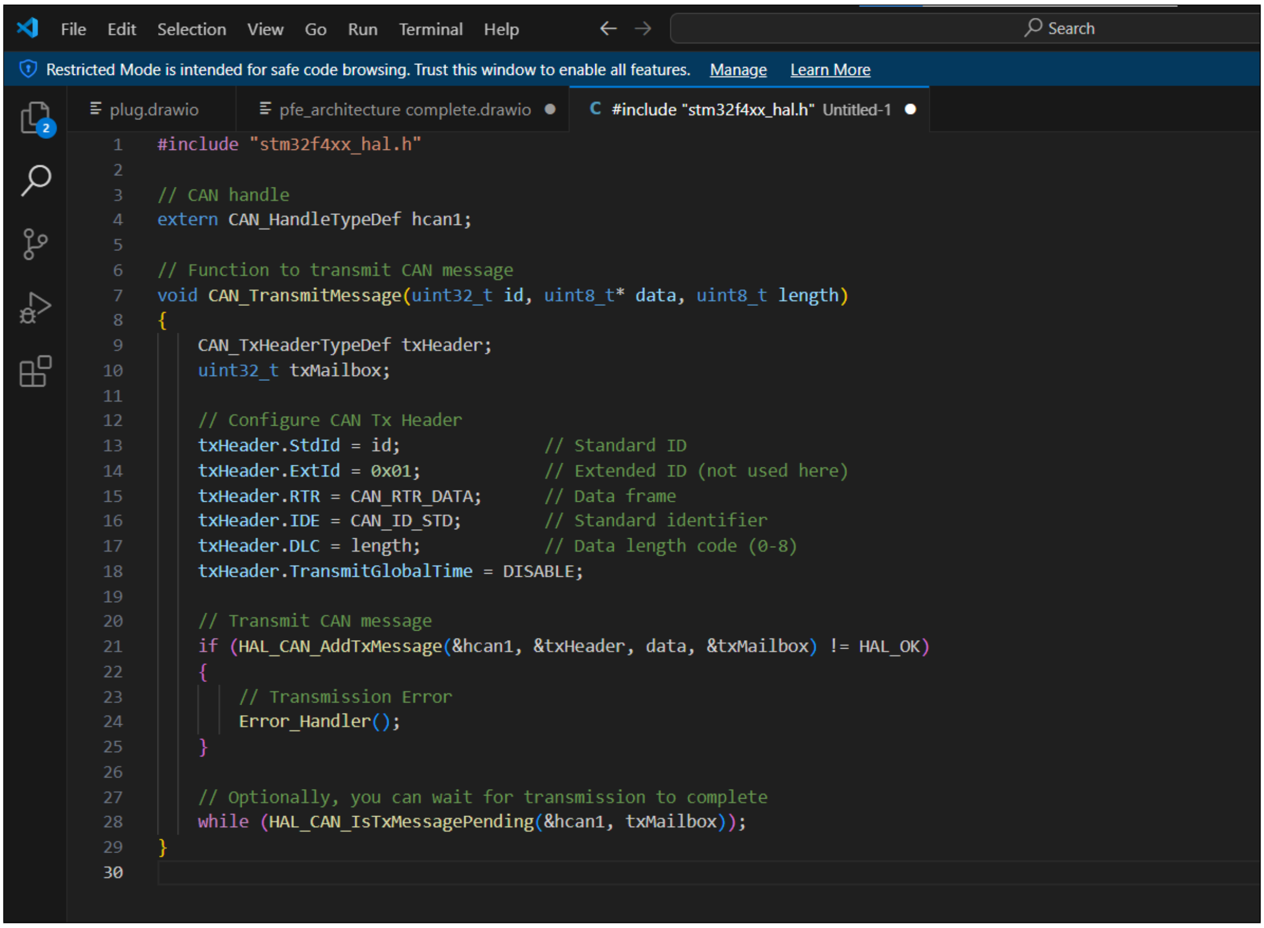

Figure 5.12: STM32 transmission to the vehicle

LoadTxSequence and RequestToSend: The Load function must load a CAN message into the transmit buffer of the MCP2551 CAN controller. It needs parameters like the message ID in the identifier register, the loading instruction, the data length code (DLC), and the actual data to be transmitted. The function initiates the transmission procedure by sending the required data via the can protocol to the MCP2551 controller.

Nonetheless, the Request function is used to send the message that has already been added to the transmit buffer. By delivering the appropriate instruction, it tells the MCP2551 controller to transfer the data kept in the transmit buffer onto the CAN bus. The microcontroller's ability to deliver CAN messages to other CAN network nodes is facilitated by the combination of these qualities.

Message transmission in buffer: We present an example of the debug environment in Figure 5.13 to show how data is sent from the STM32 microcontroller to the BMS. The IDE shows variable values and data structures in real-time when debugging the STM32 microcontroller code. The graphic illustrates the IDE's debug window in this circumstance, where the CAN message to be transmitted is shown in the ‘txMessage‘ variable. Data bytes and the frame ID that are being transferred to the BMS are stored in the ‘txMessage‘ variable (in decimal). Debugging and validating the data transmission process are made easier by this visualization, which enables a thorough analysis of the data being communicated during runtime. It allows developers to make sure that the data is transferred correctly and that there is appropriate communication with the BMS.

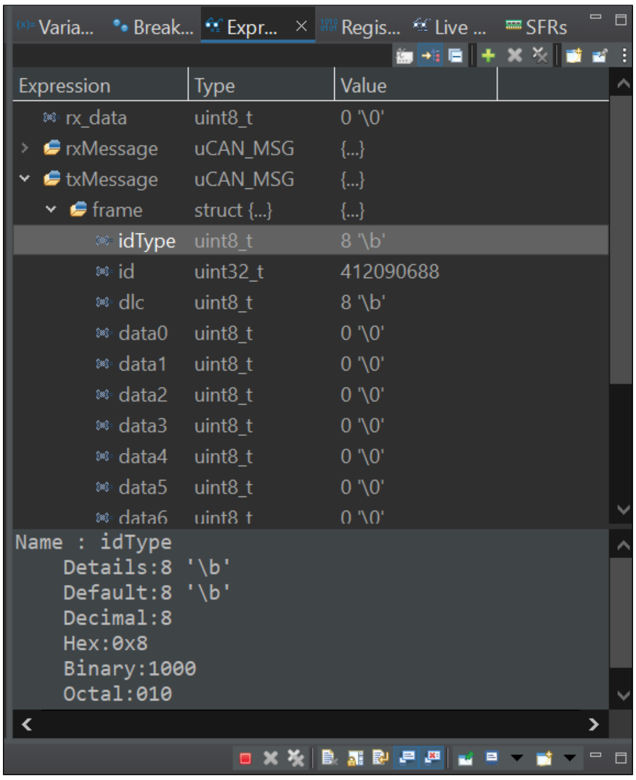

Figure 5.13: The transmitted frame ’txMessage’

5.3.2 STM32 connection with Serial Monitor

The serial monitor’s communication with the STM32 microcontroller provides an easy-to-use interface for obtaining and displaying the data collected from the car’s parts. Through a serial communication interface, such as USB or UART, the STM32 and the PC may interact with one another.

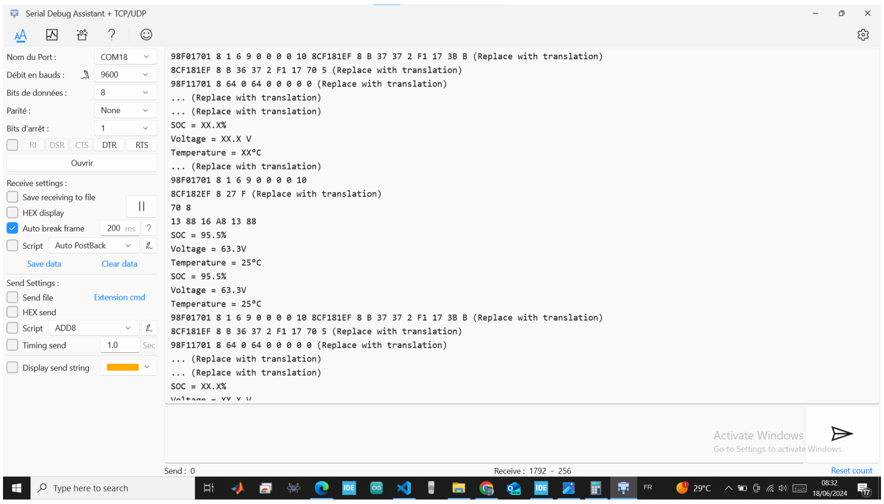

Figure 5.14: displayed data in the serial monitor

- **Receive data over rx function:** This function call initiates an asynchronous receive operation utilising the UART peripheral (USART2) when interrupt-based reception is enabled. The hardware abstraction layer (HAL) is instructed to take one byte of data and save it in the 'rxdata' variable when data is available.

```
HAL_UART_Receive_IT(&huart2, &rx_data, 1);
```

Figure 5.15: The UART receive function

General conclusion and perspectives

During this end-of-study project carried out at Bako Motors, we successfully implemented a diagnostic system for their electric vehicle. This report has comprehensively explored the development and implementation process, covering the initial company presentation and project context, system functionality, design principles, equipment selection, and the testing and implementation stages. Each chapter has detailed the methods and technologies employed to achieve our objectives.

The project began with an overview of diagnostic systems and components, including the Battery Management System (BMS) controller, OBD-II protocols, and the CAN bus. We covered the general functionality of the system, design concepts, and standards for selecting the best equipment to ensure reliable and effective performance. The subsequent testing and installation stages confirmed the system's operation and compliance with all necessary specifications.

Through rigorous planning and implementation, we have created a system that meets the original project criteria and demonstrates reliable performance. Key achievements include addressing SOC display inaccuracies and standardizing the charging plug, which have significantly improved user experience and system reliability.

Looking to the future, there are several areas for enhancement and innovation:

Adopting ISO 14229-1 Standard: Aligning our system with the ISO 14229-1 standard for Unified Diagnostic Services (UDS) on Controller Area Network (CAN) would enhance our system's diagnostic capabilities. This standardization would provide a more comprehensive and uniform approach to vehicle diagnostics, facilitate system interoperability, and ensure adherence to industry best practices.

Custom Microcontroller Development: Instead of relying on off-the-shelf STM microcontrollers, developing a custom microcontroller tailored to our project's requirements could offer several benefits. These include improved performance, reduced power consumption, and the inclusion of features and interfaces specifically designed for our system. This customization would result in a more efficient and cost-effective solution, further enhancing the system's overall reliability and functionality.

Resolving SOC Display Issues: Continued focus on improving the accuracy of the SOC display through advanced algorithms and better sensor integration will ensure drivers have reliable information about their vehicle's battery status. This improvement is crucial for alleviating range anxiety and enhancing user trust.

By incorporating these improvements, we can advance the current project and lay the groundwork for future advancements and innovations in the sector. Our commitment to continuous improvement and adherence to industry standards will keep our system at the forefront of technological advancements, meeting the evolving needs of the market.

In summary, this report documents a successful project development experience and identifies potential areas for future improvement. By embracing these perspectives, we can ensure our system's continued relevance and excellence in the ever-evolving field of automotive diagnostics and management systems.

Bibliography

[1] “Bako motors sas presentation.” [Last Access on April,1 2024]. (Bako Motors SAS), [Online]. Available: https://bakomotors.com/accueil.

[2] “Partners presentation.” [Last Access on April,1 2024]. (Bako Motors SAS), [Online]. Available: https://bakomotors.com/la-marque.

[3] “Stm32f4discovery user manual.” [Last Access on April , 25 2024]. (ST electronics), [Online]. Available: https://www.st.com/resource/en/user_manual/um1472-discovery-kit-with-stm32f407vg-mcu-stmicroelectronics.pdf.

[4] “Mcp2515 datasheet.” [Last Access on April, 19 2024]. (MICROCHIP [Microchip Technology]), [Online]. Available: https://pdf1.alldatasheet.com/datasheet-pdf/view/166906/MICROCHIP/MCP2515.html.

[5] “Usb module guide.” [Last Access on April , 26 2024]. (Silicon Labs), [Online]. Available: https://www.silabs.com/interface/usb-bridges/classic/device.cp2102.

Appendix

Appendix 1. Support

Appendix 1.1: Libraries used

Table 6.1 presents the libraries referenced in each software used in the process of working on the project.

Table 6.1: Libraries used

Software	Libraries
STM32CubeIDE	-main.h -stm32f4xx_hal.h -stdio.h
Arduino	-SPI.h -mcp2515.h

Appendix 1.2: Components pinout guide

- **STM32f407g-disc1:** Figure 6.1 illustrates the pinouts of the STM32 microcontroller used in the project.

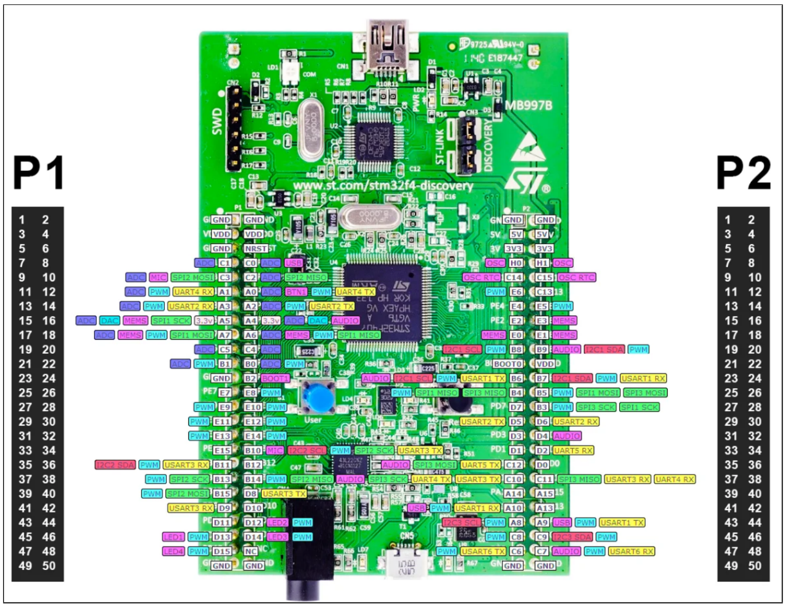

Figure 6.1: STM32 pinouts

- **Arduino uno:** Figure 6.2 illustrates the pinouts of the arduino uno microcontroller used in the testing process of the project.

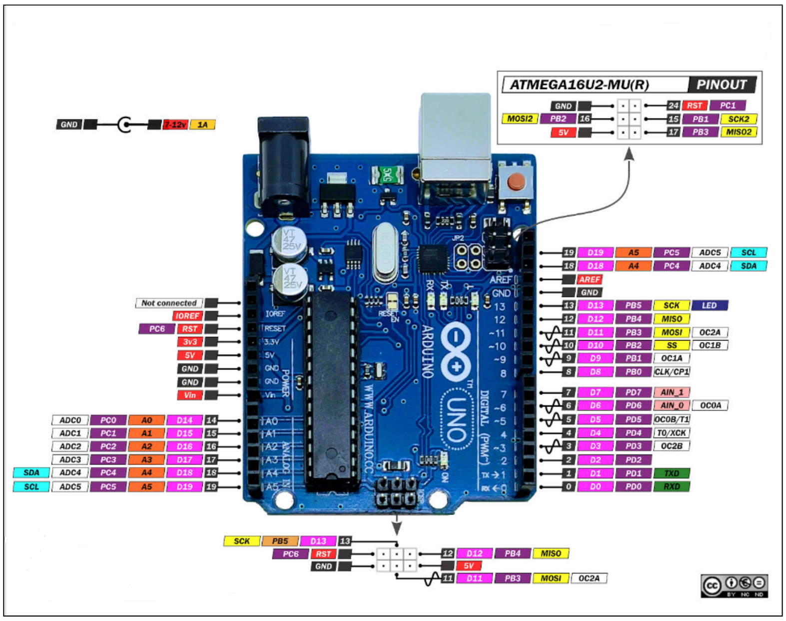

Figure 6.2: Arduino uno pinouts

- **USB module:** Figure 6.3 illustrates the pinouts of the usb module used in the project.

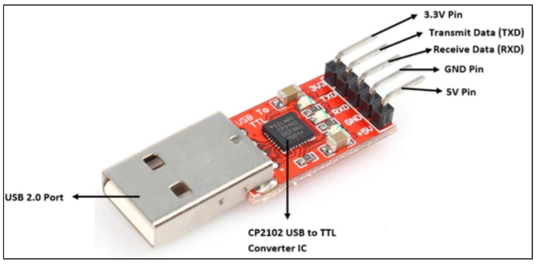

Figure 6.3: USB module pinouts

- **MCP2515:** Figure 6.4 illustrates the pinouts of the MCP2515 transceiver used in the project.

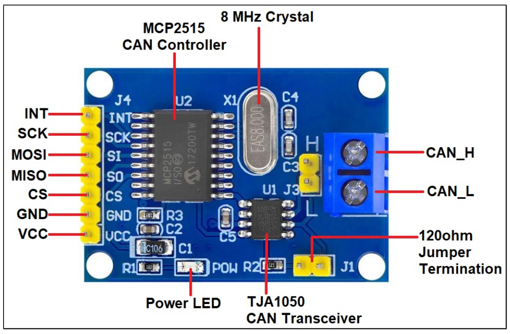

Figure 6.4: MCP2515 pinouts

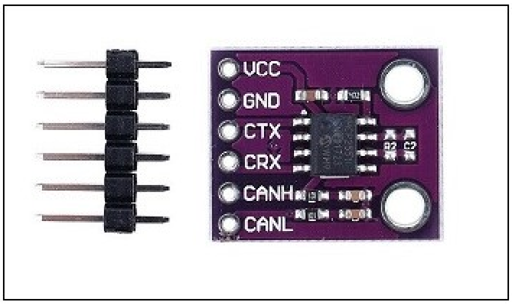

Figure 6.5: MCP2551 pinouts

Appendix 1.3: STM32f407g-disc1 configuration tool

- **Pinout and configuration:** Figure 6.6 illustrates the configuration process of the pinouts of the STM32 board.

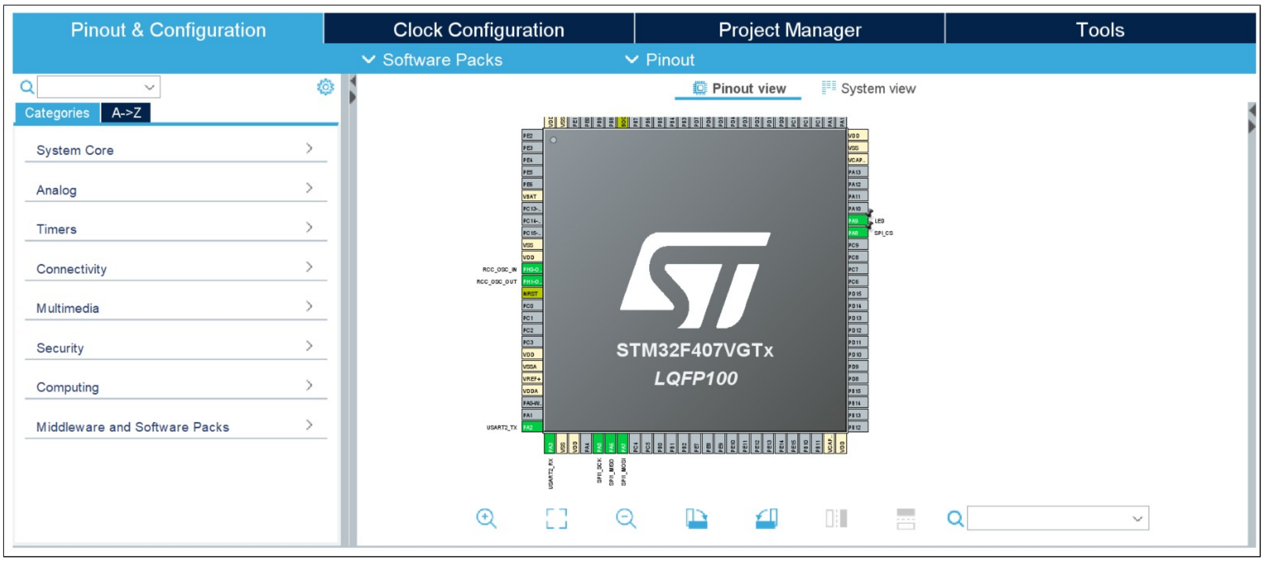

Figure 6.6: STM32 pinout configuration

- **Clock configuration:** Figure 6.7 illustrates the clock configuration of the STM32 board.

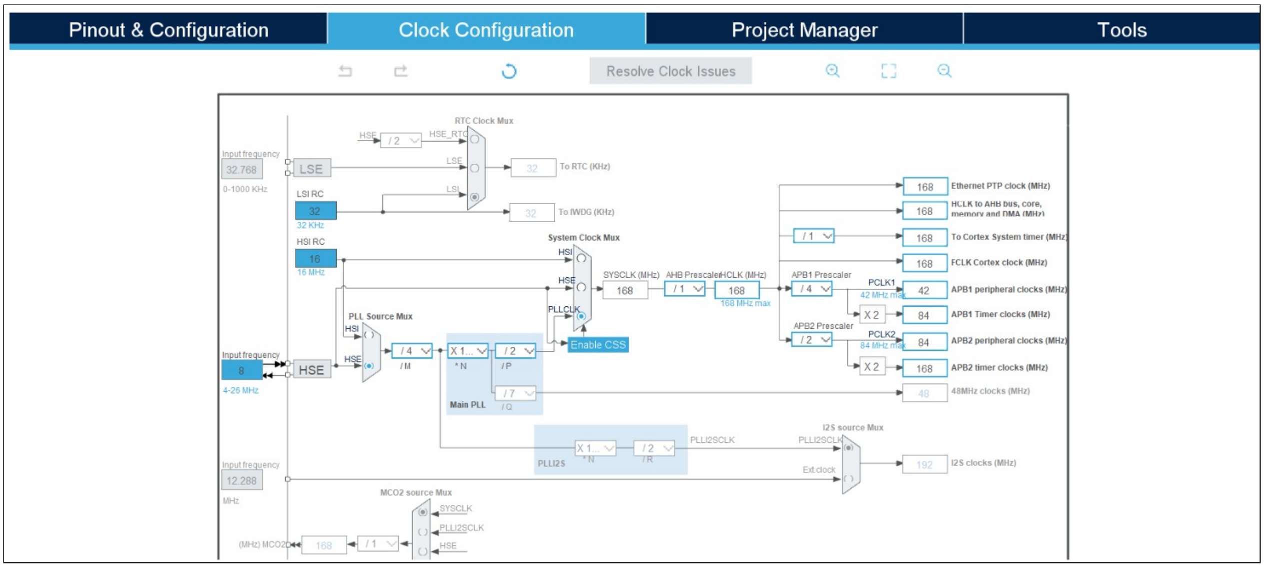

Figure 6.7: STM32 clock configuration

ملخّص

يعد مشروع "تشخيص المركبات الكهربائية" مشروع تخرج في المدرسة الوطنية العليا للمهندسين (ENSIT) في تونس. يتناول المشروع مشكلة العرض غير الدقيق لحالة الشحن (SOC) في المركبات الكهربائية لشركة Bako Motors. يتضمن المشروع تحليل مشاكل عرض SOC الحالية، وتنفيذ حلول تصحيحية، وتطوير نظام تشخيص لتحسين كفاءة صيانة المركبات. تهدف هذه المبادرة إلى تحسين الموثوقية التشغيلية ورضا العملاء لشركة Bako Motors، بما يتماشى مع معايير الصناعة وتعزيز النقل المستدام.

كلمات مفاتيح : حالة الشحن، المركبات الكهربائية، التشخيص، الأنظمة المدمجة

Résumé

Ce projet intitulé "Apporter de la clarté à la mobilité électrique : diagnostics Mise en œuvre et problèmes de rectification en électrique Véhicules" constitue un projet de fin d'études à l'École Nationale Supérieure d'Ingénieurs (ENSIT) de Tunis, Tunisie. Il aborde l'affichage incorrect de l'état de charge (SOC) dans les véhicules électriques de Bako Motors. Le projet consiste à analyser les problèmes actuels d'affichage du SOC, à mettre en œuvre des solutions correctives et à développer un système de diagnostic pour améliorer l'efficacité de la maintenance des véhicules. Cette initiative vise à améliorer la fiabilité opérationnelle et la satisfaction des clients de Bako Motors, tout en respectant les normes de l'industrie et en promouvant le transport durable.

Mots clés : État de charge, Véhicules électriques, Diagnostics, Systèmes embarqués, Bako Motors

Abstract

The project titled "Bringing Clarity to Electric Mobility: Diagnostics Implementation and problems Rectification in Electric Vehicles" is a capstone project at the Higher National School of Engineering (ENSIT) in Tunis, Tunisia. It addresses the inaccurate State of Charge (SOC) display in Bako Motors' electric vehicles. The project involves analyzing current SOC display issues, implementing corrective solutions, and developing a diagnostic system to enhance vehicle maintenance efficiency. This initiative aims to improve operational reliability and customer satisfaction for Bako Motors, aligning with industry standards and promoting sustainable transportation.

Keywords : State of Charge, Electric Vehicles, Diagnostics, Embedded Systems, Bako Motors