FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

EVs Charge Protocol Universal Control

Rodrigo Leite Nóbrega

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. PhD Adriano da Silva Carvalho

August 1, 2022

© Rodrigo Nóbrega, 2022

Resumo

A empresa AddVolt possui o primeiro sistema Híbrido Plug-in do mundo aplicado ao transporte refrigerado cuja operação consiste em eletrificar a fonte de alimentação do sistema de refrigeração de vans, camiões, carretas e contentores, ao invés de tê-lo acionado por motores de combustão interna, para armazenar e refrigerar produtos perecíveis.

Neste contexto, o objetivo deste trabalho foi implementar um protocolo de carregamento existente para Veículos Elétricos, por meio do projeto de dois controladores de carga que possam permitir tanto carregar a unidade da AddVolt através de estações de carregamento de Veículos Elétricos, como também carregar cargas compatíveis, utilizando como carregador a unidade da AddVolt.

Neste documento, são apresentadas inicialmente as motivações para o projeto e os desafios, introduzindo o trabalho. Em seguida, são apresentados os principais conceitos na área de carregamento de Veículos Elétricos, juntamente com as principais normas que os regem. Após esta apresentação, o protocolo de carregamento SAE J1772 foi selecionado para ser implementado no projeto. Em seguida, é apresentado o projeto do sistema proposto, desdobrado em seus subsistemas, em Hardware e Firmware, que implementam as funcionalidades desejadas. Em seguida, o protótipo projetado foi testado e os resultados discutidos, consolidando as principais conclusões ao final do trabalho.

Palavras-chave: Electric Vehicle Charging, Controller Area Network, Pulse Width Modulation, Analog to Digital Conversion, SAE J1772, AC Charging.

Abstract

The AddVolt company has the world's first Plug-in Hybrid system applied to refrigerated transport, which the operation consists of electrifying the power source of the refrigeration system of vans, trucks, trailers and containers, instead of having it powered using internal combustion engines, in order to store and refrigerate perishable products.

In this context, the purpose of this work was to implement an existing charging protocol for Electric Vehicles, through the design of two charge controllers that can allow AddVolt's unit to be charged through Electric Vehicle charging stations and to charge compatible loads.

In this document, the motivations for the project and the challenges are initially presented, introducing the work. Then, the main concepts in the field of charging Electric Vehicles are presented, together with the main standards that govern it. After this presentation, the SAE J1772 AC charging protocol was selected to be implemented in the project. Then, the design of the proposed system is presented, broken down into its subsystems, in Hardware and Firmware, that implement the desired features. After that, the designed prototype was tested and the results were discussed, consolidating the main conclusions in the end of the work.

Keywords: Electric Vehicle Charging, Controller Area Network, Pulse Width Modulation, Analog to Digital Conversion, SAE J1772, AC Charging.

Agradecimentos

Primeiramente, devo agradecimentos à Faculdade de Engenharia da Universidade do Porto, por me proporcionar esta oportunidade durante os últimos cinco anos, assim como também a todos os portugueses por me acolherem nesta jornada que, para além de ter me dado uma formação acadêmica, fez de mim uma pessoa melhor.

Também devo agradecimentos ao Professor Adriano Carvalho, que desde o primeiro dia se fez presente e esteve disponível para dar todo o apoio necessário ao longo da realização deste trabalho. Toda a partilha de experiências e de ideias foi fundamental.

Um agradecimento especial à AddVolt e à toda a sua equipa pela confiança depositada em mim e pela oportunidade com o projeto, com um destaque para quem o acompanhou de perto: os Engenheiros Marco Santos, Luís Encerrabodes e Ricardo Gonçalves. Igualmente muito obrigado aos diretores Bruno Azevedo e Ricardo Soares por este período bastante enriquecedor na empresa, com muita experiência e aprendizado que com certeza levarei comigo para o resto da vida.

Muito obrigado ao meu pai Mário, à minha mãe Lueli e aos meus irmãos Murilo e Henrique. Desde o primeiro dia, estar longe da família e dos amigos nunca foi fácil, muito obrigado pelo apoio e por fazerem o possível e o impossível para que essa jornada tenha ocorrido da melhor forma nos últimos anos.

A todos os amigos que trouxe comigo e aos amigos que fiz no Porto, muito obrigado por fazerem parte da minha vida, pela companhia e por todas as memórias que foram criadas e que jamais serão esquecidas.

Rodrigo Leite Nóbrega

"The way to succeed is to double your failure rate."

Thomas J. Watson

Contents

1 Introduction **1**
- 1.1 Motivation and Context . . . 1
- 1.2 Structure of the document . . . 2

2 State of the Art **5**
- 2.1 Background on Charging of Electric Vehicles . . . 5
  - 2.1.1 Electric Vehicle Charging Tecnology . . . 5
  - 2.1.2 AC Charging . . . 8
  - 2.1.3 DC Charging . . . 12
- 2.2 Main Standards . . . 16
  - 2.2.1 Generic Standards . . . 17
  - 2.2.2 Connector Standards . . . 17
  - 2.2.3 Supply Station Standards . . . 19
  - 2.2.4 Communication Standards . . . 20
- 2.3 Electric Vehicles Charging Definitions . . . 22
  - 2.3.1 Charging Levels . . . 22
  - 2.3.2 Electric Vehicle Charging around the globe . . . 26
  - 2.3.3 Charging Modes . . . 26
  - 2.3.4 Charging Cases . . . 29
  - 2.3.5 Plug Types and Charging Protocols . . . 31
- 2.4 Conclusion . . . 46

3 Design and Implementation **47**
- 3.1 Introduction . . . 47
- 3.2 Environment . . . 48
- 3.3 Workplan . . . 50
- 3.4 Hardware . . . 51
  - 3.4.1 Supply Equipment Charge Controller (SECC) . . . 51
  - 3.4.2 Electric Vehicle Charge Controller (EVCC) . . . 62
- 3.5 Firmware . . . 70
  - 3.5.1 DAVE™ . . . 70
  - 3.5.2 Timers . . . 71
  - 3.5.3 Supply Equipment Charge Controller (SECC) . . . 73
  - 3.5.4 Electric Vehicle Charge Controller (EVCC) . . . 80

4 Tests and Results **97**
- 4.1 Physical Assembly . . . 97
  - 4.1.1 Test 1 - Charging the unit from the grid . . . 97

4.1.2 Test 2 - Charging the unit from the grid using external charger 98
4.1.3 Test 3 - Charging the unit from another unit 99
4.2 Test of Use Cases 100
4.2.1 Test 1 - Charging the unit from the grid 101
4.2.2 Test 2 - Charging the unit from the grid using external charger 113
4.2.3 Test 3 - Charging the unit from another unit 120
4.3 Analysis of the Results 121

5 Conclusion **123**
5.1 Achievement of Objectives 123
5.2 Future Work 123

A Controller Area Network (CAN) **125**

B IEEE 754 **129**

C Circuit Schematics **131**
C.1 SECC Circuit CAD Schematic 132
C.2 EVCC Circuit CAD Schematic 133

References **135**

List of Figures

1.1 Simplified Diagram of AddVolt's System. 2

2.1 Conductive Charging (on the left), Inductive Charging (in the centre) and Battery Swap (on the right) [1]. 5
2.2 Inductive Charging [1]. 6
2.3 Dynamic Inductive Charging [1]. 7
2.4 Battery Swap [1]. 7
2.5 Main Conductive Charging solutions for Battery Electric Vehicles (BEV) (adapted from [2]). 7
2.6 AC Charging infrastructure composition schematic [1]. 8
2.7 Basic simplified AC Charging configuration [1]. 9
2.8 AC Charging plugs used worldwide [1]. 10
2.9 AC Charging connectors pinout: Type 1 (on the left), Type 2 (in the centre) and Tesla US connector (on the right) [1]. 10
2.10 AC Charging Type 1 and Type 2 connectors Control Pilot (CP) and Proximity Pilot (PP) schemes [1]. 11
2.11 DC Charging infrastructure composition schematic [1]. 12
2.12 Basic simplified DC Charging configuration [1]. 13
2.13 DC Charging plugs used worldwide [1]. 13
2.14 DC Charger CCS Combo 1 connector and vehicle inlet [1]. 14
2.15 DC Charger CCS Combo 2 connector and vehicle inlet [1]. 15
2.16 DC Charger CHAdeMO (on the left) and Chinese GB/T (on the right) connectors [1]. 16
2.17 DC Charger Tesla proprietary (on the left) and Type 2 in Tesla configuration (on the right) connectors [1]. 16
2.18 General Standards in Electric Vehicle Charging [3]. 17
2.19 AC Level 2 charging station [4]. 22
2.20 AC level 1 system configuration in SAE J1772 [5]. 23
2.21 AC level 2 system configuration in SAE J1772 [5]. 24
2.22 DC level 1 system configuration in SAE J1772 [5]. 25
2.23 DC level 2 system configuration in SAE J1772 [5]. 25
2.24 EV Charging Mode 1 [6]. 27
2.25 EV Charging Mode 2 [6]. 27
2.26 EV Charging Mode 3 [6]. 28
2.27 EV Charging Mode 4 [6]. 29
2.28 EV Case A connection [7]. 30
2.29 EV Case B connection [7]. 30
2.30 EV Case C connection [7]. 31

2.31 IEC 62196 Type 1 plug (SAE J1772 *J plug*) [8]. 32
2.32 IEC 62196 Type 2 plug and pinout [2]. 32
2.33 SAE J1772 signaling circuit [8]. 33
2.34 SAE J1772 Control Pilot circuit [5]. 34
2.35 Maximum available output current in function of Control Pilot PWM duty cycle, as stated in SAE J1772 [5]. 36
2.36 SAE J1772 Proximity Pilot circuit [5]. 37
2.37 CHAdeMO plug [9]. 38
2.38 CHAdeMO plug connector interface [10]. 39
2.39 CHAdeMO charging sequence [11]. 40
2.40 Control Pilot Circuit in Charging Mode 3 and Case A Connection in GB/T 18487 [12]. 41
2.41 Control Pilot Circuit in Charging Mode 3 and Case B Connection in GB/T 18487 [12]. 41
2.42 Control Pilot Circuit in Charging Mode 3 and Case C Connection in GB/T 18487 [12]. 41
2.43 GB/T plug and its pinout (adapted from [13]). 42
2.44 Control Pilot Circuit in DC Charging Mode in GB/T 18487 [12]. 42
2.45 Combo 1 and Combo 2 connectors [14]. 43
2.46 CCS Charging Sequence [5]. 45

3.1 Schema of the final working system. 47
3.2 Infineon XMC4500 Relax Kit Microcontroller development board [15]. 48
3.3 Architecture of Infineon XMC4500 Microcontroller [15]. 50
3.4 Circuit schematic representing the hardware of the SECC. 52
3.5 Power Supply block of the SECC. 53
3.6 Power Supply block of the SECC designed in CAD software. 54
3.7 Circuit schematic draft of the subsystem to drive the contactor connected between the three-phase source and the plug to the load to be charged. 55
3.8 Relay Drive block of the SECC designed in CAD software. 57
3.9 Circuit schematic draft of the subsystem of Control Pilot Interface. 58
3.10 Control Pilot Interface block of the SECC designed in CAD software. 59
3.11 Typical application of the CAN transceiver NCV7342 [16]. 60
3.12 Circuit schematic draft of the connection between the microcontroller and the CAN transceiver NCV7342, then connected externally to the insulation meter. . . 60
3.13 Connection between the microcontroller and the CAN transceiver NCV7342, designed in CAD software. 61
3.14 Final PCB view of SECC in CAD software (left) and PCB after printed (right). . 61
3.15 Final PCB of SECC after printed and with the components soldered. On the right, the SECC with the XMC 4500 Relax Kit attached. 62
3.16 Circuit schematic representing the hardware of the EVCC. 63
3.17 Power Supply block of the EVCC. 64
3.18 Adopted pinout for M12 connector in AddVolt's CAN bus. 64
3.19 Power Supply block of the EVCC designed in CAD software. 65
3.20 Circuit schematic draft of the subsystem of Proximity Pilot Interface in the EVCC. 66
3.21 Proximity Pilot Interface block of the EVCC designed in CAD software. 67
3.22 Circuit schematic draft of the subsystem of Control Pilot Interface in the EVCC. . 67
3.23 Control Pilot Interface block of the EVCC designed in CAD software. 68
3.24 Final PCB view of EVCC in CAD software (left) and PCB after printed (right). . 69

3.25 Final PCB of EVCC after printed and with the components soldered. On the right, the EVCC with the XMC 4500 Relax Kit attached. . . . 70
3.26 *Timer_struct* structure to work as a clock base for the system. . . . 71
3.27 *TimerInterrupt()* interrupt routine that calls *update_timer()* routine each 1ms. . . . 71
3.28 *update_timer()* routine to increment the variables of *Timer_struct* structure periodically each 1ms. . . . 72
3.29 *clear_timer()* routine to reset the variables of *Timer_struct* by setting all to 0. . . . 72
3.30 Finite State Machine for SECC Firmware. . . . 73
3.31 Actions executed according to the states in the Finite State Machine of SECC Firmware. . . . 73
3.32 *max_current_to_duty_cycle()* routine that converts a current value in Ampère into a duty cycle value in percentage, according to SAE J1772. . . . 74
3.33 Toggle Current auxiliary Finite State Machine for SECC Firmware to set the maximum available output current. . . . 75
3.34 *PWM Generation* App configuration windows in DAVE™. . . . 76
3.35 Duty cycle setup and PWM Generation start command. . . . 76
3.36 Principal compare blocks for PWM generation in XMC 4500 [17]. . . . 77
3.37 PWM generation in *up-count* mode [17]. . . . 77
3.38 Synchronized PWM generation to generate Compare Match Event. . . . 78
3.39 Synchronized ADC triggering for measurements of the high level voltage of the PWM signal on Control Pilot line. . . . 78
3.40 Indication of the used buttons in the XMC 4500 Relax kit board for the functionalities of Stop and Toggle Current (adapted from [15]). . . . 80
3.41 App Dependency Tree in DAVE™ IDE for the SECC. . . . 80
3.42 Finite State Machine diagram for EVCC Firmware. . . . 81
3.43 Actions executed according to the states in the Finite State Machine of EVCC Firmware. . . . 82
3.44 Simplified diagram of the charging current path from the grid to the battery pack to be charged. . . . 82
3.45 Estimation of Proximity Pilot Resistance in code. . . . 85
3.46 *void CaptureIRQHandler()* interrupt routine to handle the results from the measurements done in CAPTURE App. . . . 86
3.47 Interrupt routine to handle values after ADC measurements are done. . . . 87
3.48 Example on ADC measurement for level counting to determine duty cycle of CP signal. . . . 88
3.49 *get_duty_cycle()* function to determine the duty cycle of the PWM signal on CP line. 88
3.50 Configuration of Logical Message Objects (LMOs) in CAN_NODE App instance. 89
3.51 Declaration of structure *flint* and of the necessary variables. . . . 91
3.52 Declaration of the routine *can_transmit()*. . . . 92
3.53 Declaration of the interrupt routine *CAN_RX_LMO_01()*. . . . 93
3.54 Declaration of the routine *process_CAN_RX()*. . . . 94
3.55 Updated *update_timer()* routine to increment the variables of *Timer_struct* structure periodically each 1ms and to send periodic CAN messages. . . . 95
3.56 App Dependency Tree in DAVE™ IDE for the EVCC. . . . 96

4.1 Schematic of the assembly made for Test 1. . . . 97
4.2 Physical assembly made for Test 1. . . . 98
4.3 Schematic of the assembly made for Test 2. . . . 98
4.4 Physical assembly made for Test 2. . . . 99

4.5 Schematic of the assembly made for Test 3. 99
4.6 Physical assembly made for Test 3. 99
4.7 Graphical User Interface designed in Micrium to monitor the operation of the SECC. 100
4.8 Graphical User Interface designed in Micrium to monitor the operation of the EVCC. 100
4.9 Evolution in time of the changes in the maximum available output current of the charger. 101
4.10 Evolution in time of the voltage on Control Pilot line, sensed by the SECC. . . . 102
4.11 Evolution in time of the state of the SECC and the state of the relay and contactor. 102
4.12 Control Pilot PWM signal with 10% duty cycle when the plug is connected and the switch on EVCC is closed, causing the high level of the signal to be at 6V and the low level remaining at -12V. 103
4.13 Control Pilot PWM signal with 20% duty cycle when the plug is connected and the switch on EVCC is closed, causing the high level of the signal to be at 6V and the low level remaining at -12V. 103
4.14 Evolution in time of the duty cycle of the PWM signal on Control Pilot line. . . . 104
4.15 Evolution in time of SECC's state in its FSM when charging is interrupted. . . . 104
4.16 Evolution in time of Control Pilot PWM signal duty cycle when charging is interrupted. 105
4.17 Evolution in time of Control Pilot voltage when charging is interrupted. 105
4.18 Evolution in time of Proximity Pilot voltage when Type 2 plug is connected. . . . 106
4.19 Evolution in time of Control Pilot duty cycle when Type 2 plug is connected. . . 106
4.20 Evolution in time of EVCC's state when Type 2 plug is connected. 107
4.21 Evolution in time of the charging current in the beginning of the charging procedure, when the Type 2 plug is connected. 107
4.22 Evolution in time of the charging power in the beginning of the charging procedure, when the Type 2 plug is connected. 108
4.23 Evolution in time of the *State of Charge (SoC)* in the beginning of the charging procedure, when the Type 2 plug is connected. 108
4.24 Control Pilot PWM signal duty cycle change in response to changing the maximum available AC current from the SECC. 109
4.25 Charging current response to the change on maximum available AC current. . . . 109
4.26 Charging power response to the change on maximum available AC current. . . . 110
4.27 Evolution in time of EVCC's state in the end of the charging procedure, when the stop button of the SECC is pressed and the Type 2 plug is disconnected. 111
4.28 Evolution in time of Control Pilot PWM duty cycle in the end of the charging procedure, when the stop button of the SECC is pressed and the Type 2 plug is disconnected. 111
4.29 Evolution in time of Proximity Pilot voltage in the end of the charging procedure, when the stop button of the SECC is pressed and the Type 2 plug is disconnected. 112
4.30 Evolution in time of the charging current in the end of the charging procedure, when the stop button of the SECC is pressed and the Type 2 plug is disconnected. 112
4.31 Evolution in time of the charging power in the end of the charging procedure, when the stop button of the SECC is pressed and the Type 2 plug is disconnected. 113
4.32 Evolution in time of Proximity Pilot voltage when Type 2 plug of the external EVSE is connected. 114
4.33 Evolution in time of Control Pilot PWM duty cycle in the beginning of the charging procedure, when Type 2 plug of the external EVSE is connected. 114

4.34 Evolution in time of EVCC's state in the beginning of the charging procedure, when Type 2 plug of the external EVSE is connected. 115
4.35 Evolution in time of the charging current in the beginning of the charging procedure, when Type 2 plug of the external EVSE is connected. 116
4.36 Evolution in time of the charging power in the beginning of the charging procedure, when Type 2 plug of the external EVSE is connected. 116
4.37 Evolution in time of the battery *State of Charge (SoC)* in the beginning of the charging procedure, when the Type 2 plug from the EVSE is connected. 117
4.38 Evolution in time of EVCC's state in the end of the charging procedure, when the Type 2 plug is partially and fully disconnected. 118
4.39 Evolution in time of Control Pilot PWM duty cycle in the end of the charging procedure, when the Type 2 plug is partially and fully disconnected. 118
4.40 Evolution in time of Proximity Pilot voltage in the end of the charging procedure, when the Type 2 plug is partially and fully disconnected. 119
4.41 Evolution in time of the charging current in the end of the charging procedure, when the Type 2 plug is partially and fully disconnected. 119
4.42 Evolution in time of the charging power in the end of the charging procedure, when the Type 2 plug is partially and fully disconnected. 120

A.1 CAN node connection to CAN bus network schematic. 125
A.2 CAN data message structure for standard and extended frame [18]. 126
A.3 CAN bus signaling logic [19]. 128
A.4 CAN bus electrical topology [19]. 128

B.1 Representation of floating-point numbers in IEEE 754 standard for Single Precision [20]. 129
B.2 Representation of floating-point numbers in IEEE 754 standard for Double Precision [20]. 130

C.1 Final circuit schematic design of SECC in CAD software. 132
C.2 Final circuit schematic design of EVCC in CAD software. 133

List of Tables

2.1 AC charging levels [5]. 23
2.2 DC charging levels 1 and 2 in J1772 [5]. 24
2.3 Charging states for the Control Pilot in AC charging [8]. 35
2.4 PWM duty cycle in Control Pilot pin [8]. 36
2.5 EVSE Proximity Pilot Resistance [5]. 37
2.6 CHAdeMO plug pin description [9]. 38

3.1 Overview of the *start/stop* message. 90
3.2 Overview of the *set-point* message. 90
3.3 Overview of the *variables* message. 90

4.1 Calculated DC Charging Power for different charging intervals. 122

A.1 Bit fields of CAN data transmit messages [18]. 127

B.1 Representation of the number 257.75 in IEEE 754 standard for Single Precision. . 130

Abbreviations and Symbols

EV	Electric Vehicle
BEV	Battery Electric Vehicle
PHEV	Plug-in Hybrid Electric Vehicle
EVSE	Electric Vehicle Supply Equipment
EVCC	Electric Vehicle Charge Controller
SECC	Supply Equipment Charge Controller
PCB	Printed Circuit Board
AC	Alternating Current
DC	Direct Current
OBC	On-board Battery Charger
BMS	Battery Management System
SoC	State of Charge
CP	Control Pilot
PP	Proximity Pilot
PE	Protective Earth
PWM	Pulse Width Modulation
CAN	Controller Area Network
PLC	Power Line Communication
CCS	Combined Charging System
IEC	International Eletrotechnical Comission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
SAE	Society of Automotive Engineers
HLC	High Level Communication
V2G	Vehicle-to-Grid
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
CAD	Computer-aided Design
IDE	Integrated Development Environment
GUI	Graphical User Interface
API	Application Programming Interface
ADC	Analog to Digital Converter
FSM	Finite State Machine
ISR	Interrupt Service Routine
GPIO	General Purpose Input Output
LMO	Logical Message Object
MSB	Most Significant Byte
LSB	Least Significant Byte

Chapter 1

Introduction

Electric Vehicle (EV) charging is far beyond plugging a cable between a car and a socket outlet from the electrical grid. There are many concepts, rules and standards that must be followed, those that can vary by country and continent. Nevertheless, along with time many standards were developed in order to unify the charging infrastructure and network, so being easier, more comfortable and encouraging for the customer to own an EV and to contribute for the reduction on greenhouse gases emission.

1.1 Motivation and Context

The purpose of this dissertation is to come up with a solution, implementing an existing charge protocol, in order to develop Charging Controllers to be *"black-boxes"* between the electrical grid and the EV going to be charged. The goal is to develop two hardware control systems, one that works on the Electric Vehicle Supply Equipment (EVSE) side, which is the Supply Equipment Charge Controller (SECC), and the other that works on the EV side, which is the Electric Vehicle Charging Controller (EVCC). In addition, it is also necessary to implement a software stack in order to control each one of these systems. The final designed system is intended to work with the Plug-in Electric System of the company AddVolt, charging it or discharging from it.

AddVolt proposed the world's first Plug-in Electrical System applied to refrigerated transport. Its operation consists of electrifying the power source of the refrigeration system of vans, trucks, trailers and containers, instead of having it powered using internal combustion engines, in order to store and refrigerate perishable products. This solution, in addition to reducing the emission of polluting gases into the atmosphere, also saves money for the user, who no longer spends money on fuel for this purpose. A simplified diagram, but that well illustrates the operation of the AddVolt system, is presented in the Figure 1.1.

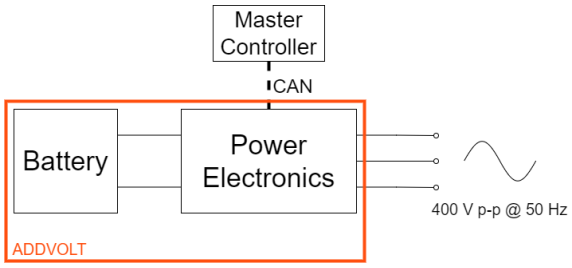


Figure 1.1: Simplified Diagram of AddVolt's System.

The system works as follows. Using a battery pack, through power electronics and filters, a three-phase AC network is created that is used to power the refrigerating machine that preserves food and other types of goods in refrigerated transport. In addition, through a mechanical coupling connected to the vehicle's crankshaft, energy is recovered using an electrical generator through Regenerative Braking, in order to recharge the battery. The system has a rated power of approximately 19kW and a battery capacity of around 19kWh.

However, it is still necessary a system that allows charging the battery pack from the electrical grid, in a controlled way. It is also desirable that this system allows using the AddVolt system to charge other loads, such as an Electric Vehicle. It is in this context that this Dissertation Project is located.

Then, in this document, it was done a deep study in the field of EV charging, covering the main standards that must be followed, as well as the charging protocols that are present in the market today, in order to have a system that works with most of EVs and EVSEs. Some of the protocols and standards, more relevant to the work, are discussed in more detail in order to provide the theoretical basis necessary for the perception of the developed project. In the end of the study and analysis, one of the charging protocols is chosen and implemented, in order to have, in the end of the work, a final working prototype.

1.2 Structure of the document

In addition to the Introduction, this document contains 4 more chapters.

In Chapter 2, the State of the Art is described and the many base concepts and related works are presented, that is, a deep dive was made into the concepts already consolidated on the subject. In this case, the main definitions, standards and terms that govern the field of Electric Vehicle Charging are described. A summary of the Charging Types, Charging Levels, Charging Modes, Plug Types, Communication Protocols and the implemented Charging Protocols is presented.

In Chapter 3, the entire implementation of the project is presented in detail. The chapter is divided between Hardware and Firmware approaches. In Hardware, all the sizing of the necessary components was presented, followed by the design of a Printed Circuit Board (PCB), which would become the physical prototype. In Firmware, the development of the algorithm that executes the system charging sequence was presented and the developed code was loaded into a microcontroller, which is what controls each system. In each of the Hardware and Firmware approaches, an

internal division was made between SECC and EVCC, in order to separate what was developed for each subsystem.

In Chapter 4, the tests performed with the final prototype were presented, as well as the analysis and discussion of the test results.

In Chapter 5 are presented the main conclusions that were taken with the accomplishment of this work, the adequacy of the prototype to the needs and also the indications for a Future Work following the work already carried out.

Chapter 2

State of the Art

The purpose of this chapter is to present and describe the main definitions, standards and terms that rule the Electric Vehicle Charging domain. It is an overview on Electric Vehicle Charging Types, Charging Levels, Charging Modes, Plug Types, Communication Protocols and the implemented Charging Protocols. The idea is to make a short introduction in the field of EV Charging, in order to have a better understanding of the developed work.

2.1 Background on Charging of Electric Vehicles

2.1.1 Electric Vehicle Charging Tecnology

As said before, there are many concepts that rule the EV charging field. First, it must be distinguished between different charging solutions available for Battery Electric Vehicles (BEV), which can refer to Electric Vehicles (EV) and Plug-in Hybrid Electric Vehicles (PHEV). Another possible definition for this type of vehicle is Plug-in Electric Vehicles (PEV). Depending on the type of vehicle, there are three ways to recharge a car battery, those are Conductive Charging, Inductive Charging and Battery Swap. An illustration of the three methods is shown in Figure 2.1 [1].

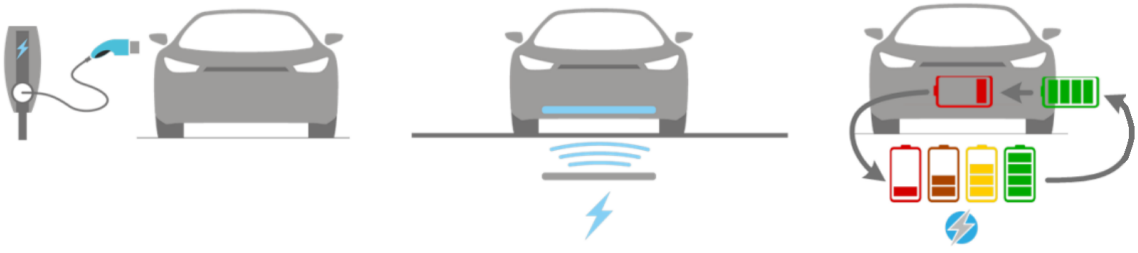


Figure 2.1: Conductive Charging (on the left), Inductive Charging (in the centre) and Battery Swap (on the right) [1].

Inductive Charging is the name given to the process of transferring energy from the source to the Electric Vehicle using an inductive magnetic coupling, based on the principle of electromagnetic induction at high frequencies. The solution proposed by Inductive Charging consists of the

use of two electromagnetically coupled coils, with the primary coil located on the street surface, in a pad-lilke construction, connected to the electrical grid. The secondary coil is located in the vehicle, usually on the underside of the car, at a safe distance from passengers. The current coming from the AC grid is then rectified and converted into high frequency AC power at the output of the charging station, and this high frequency power is then transferred to the Electric Vehicle by electromagnetic induction. An illustration of an example of such a solution is shown in the Figure 2.2 [1].

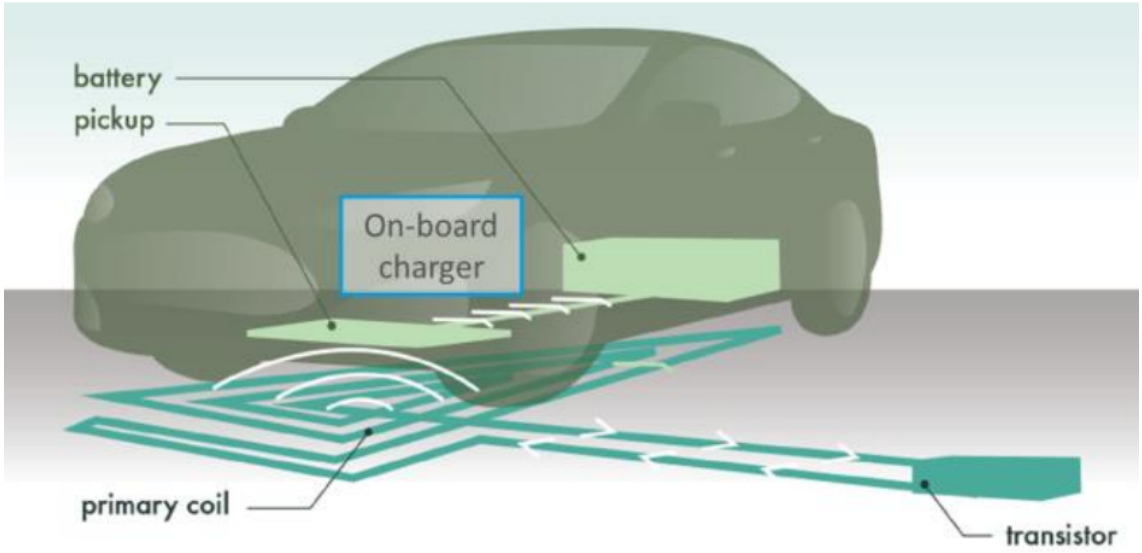


Figure 2.2: Inductive Charging [1].

In addition to the solution presented, Inductive Charging can also be performed dynamically. That is, the transfer of energy from the charger to the car is also carried out by means of inductive magnetic coupling, but with the difference that this charging can be carried out with the vehicle in motion. This is due to the fact that the coils connected to the electrical cables used to supply the necessary power are buried under the road. The coils emit an electromagnetic field that is picked up by vehicles as they are driven over these fields, then converting this field into electrical current to recharge the batteries. An illustration of the idea is presented in the Figure 2.3 [1].

Another method is Battery Swap. Its operation is as the name describes, it basically consists of replacing the depleted battery with a fully charged one. This process involves taking the car to a battery replacement station and an automatic process positions the vehicle, removes the current battery and replaces it with a fully charged battery. Depleted batteries are recharged at the station for later use. An illustration of the process is shown in the Figure 2.4 [1].

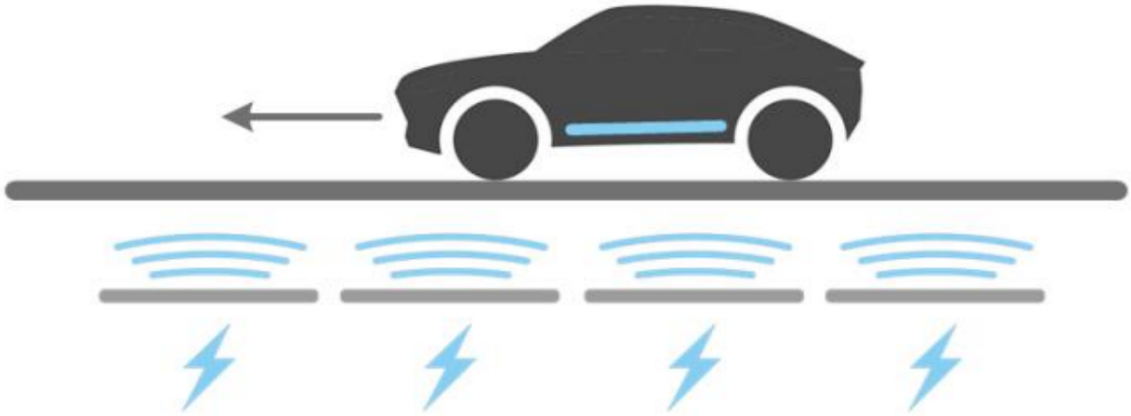


Figure 2.3: Dynamic Inductive Charging [1].

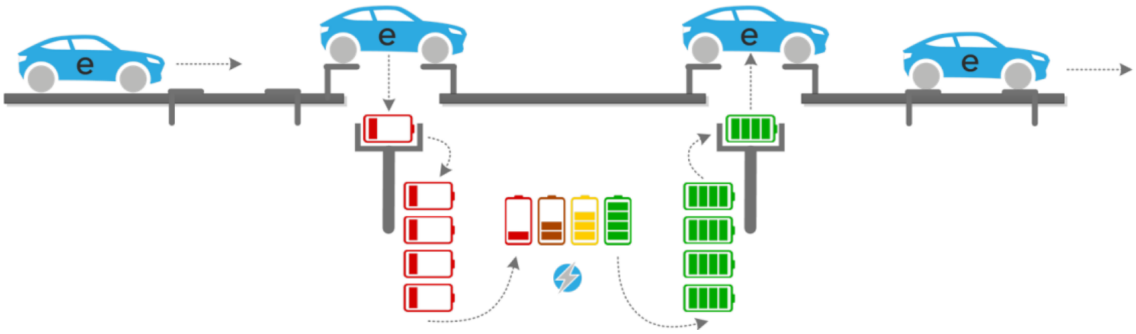


Figure 2.4: Battery Swap [1].

Last, but not least, Conductive Charging is the most common charging method, being divided into AC Charging and DC Charging. In Figure 2.5, it is illustrated the division between the two approaches in Conductive Charging.

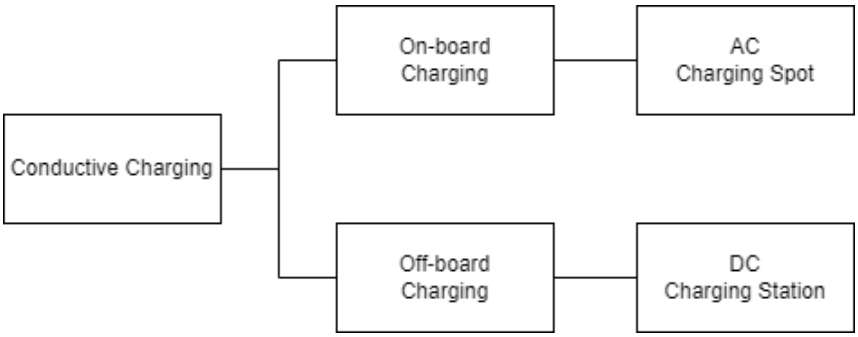


Figure 2.5: Main Conductive Charging solutions for Battery Electric Vehicles (BEV) (adapted from [2]).

The first method of Conductive Charging is the most popular of all, which is AC Charging. The main feature of this type of charging is that the car battery can be recharged anywhere with a standard electrical outlet. This is due to the fact that AC Charging is performed On-board, that is, the charging itself takes place within the vehicle's structure, in other words, there is a charging

system (i.e., power electronics converters, controllers and filters) inside the vehicle, which is called the On-board Battery Charger (OBC), and with the charging spot being simply an AC socket outlet (single or three-phase). The OBC can easily communicate with the Battery Management System (BMS) thanks to the vehicle's internal network of connections. This property guarantees greater performance at a lower cost. However, since AC power needs to be converted to DC power inside the vehicle, there is a limitation on charging power as there are size and weight restrictions for the OBC inside the vehicle. Because of this, AC Charging usually takes a relatively long time [1][2].

The other method of Conductive Charging is DC Charging or, as it is commonly called, *Fast Charging*. Unlike AC Charging, this is performed off-board, that is, the energy conversion takes place completely outside the vehicle. This charging method supplies direct current (DC) straight to the vehicle's battery, rather than powering a charger which in turn charges the battery. This solution is suitable for high power designs as the output power of these chargers is limited only by the battery's ability to accept the charge. This is due to the fact that there are no dimensional restrictions like those presented by the OBC. The charging time for a fast charger is usually less than an hour. However, as the off-board chargers and the BMS are physically separated, reliable communication is necessary to ensure charging conditions, and the adopted charging method depends on the battery type, voltage, temperature and of the State of Charge (SoC) provided by the BMS [1][2].

Both of the solutions for BEV charging are supplied by the AC grid. In the first case, the AC grid supplies the on-board charger of the vehicle and in the off-board charging the rectifying and conversion functions happen in the Electric Vehicle Supply Equipment (EVSE). Due to this off-board construction of the device (i.e. almost no dimensional limitations, when compared to OBCs), the charging capability is much higher than in OBCs. This means less idle time for the user when waiting to fully charge the vehicle and that's the reason why this solution is called a *Fast Charging* solution [2].

2.1.2 AC Charging

The composition of the AC Charging procedure is presented in Figure 2.6.

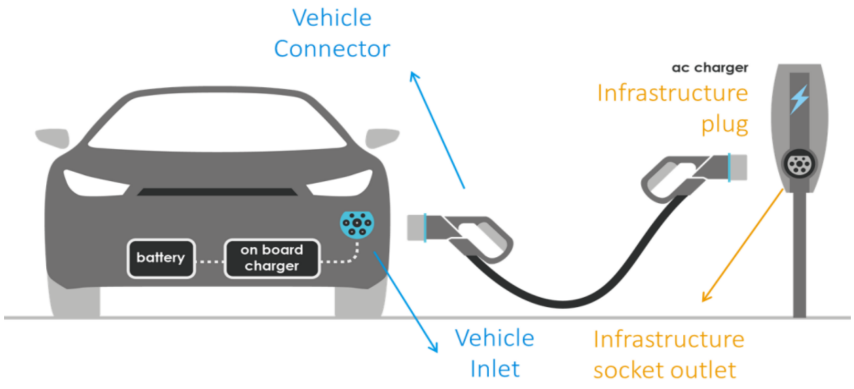


Figure 2.6: AC Charging infrastructure composition schematic [1].

As illustrated, AC Charging uses an OBC to convert AC power from the conventional mains into DC power to charge the vehicle's battery. Vehicles have a standard inlet and a charging cable is used to connect to the vehicle and to the EVSE. In many cases, this cable is permanently connected to the charging station. For safe and reliable charging, there must be many essential components in an AC charging station [1].

The moment the charging station and the vehicle are connected, the Charge Controller on the charging station begins to communicate with the vehicle. Information regarding connectivity, fault conditions and current limits is exchanged between the charger and the vehicle. Contactors and safety interlocks are used to ensure a safe charging process between the vehicle and the charging station. When AC power is transferred to the vehicle, the OBC uses a rectifier that converts that AC power into DC power [1].

While charging is in progress, the power control unit adjusts the voltage and current of a DC/DC converter appropriately to control the charging power delivered to the battery. The power control unit, in turn, receives inputs from the BMS to control battery charging. The BMS monitors key operating parameters such as voltage, current and temperature. Therefore, inputs to the power control unit are provided to control the charging power delivered by the DC/DC converter. In addition, there is a protection circuit inside the OBC. The BMS then triggers the protection circuits if battery operating limits are exceeded, isolating the battery from the system if necessary. This configuration and the power flow is illustrated in Figure 2.7 [1].

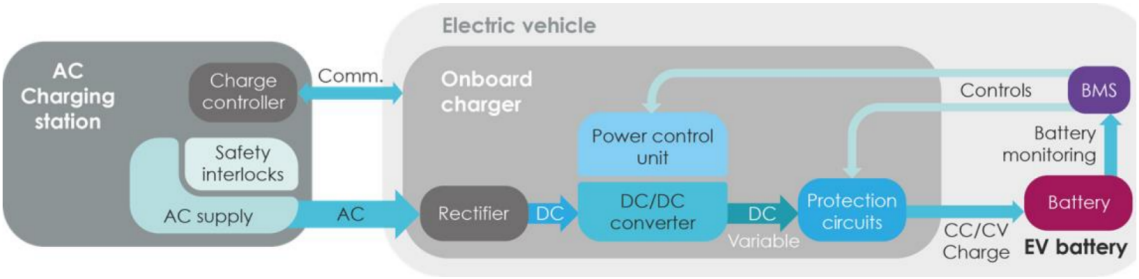


Figure 2.7: Basic simplified AC Charging configuration [1].

In order for the power transfer presented above to occur, there must be compatibility between the connector of the charging cable and the vehicle's inlet socket. In this context, there are four types of AC Charging connectors worldwide, as the industry has not yet reached an agreement on the use of the same specific connector for AC charging worldwide. Therefore, depending on the car manufacturer and the country in question, the connector may vary in shape, size and pinout configuration. The main reason is due to the difference in AC voltage and frequency. In the United States, power is supplied through a 120V, 60Hz single-phase or 240V, 60Hz two-phase grid. On the other hand, in Europe these values correspond to 230V, 50Hz single-phase or 400V, 50Hz three-phase. This difference naturally creates differences between the chargers in the two regions. In general, AC connectors have two or more larger pins for transmitting power and smaller pins for communication. In the Figure 2.8, the four types of AC connectors used internationally can be seen [1].

Figure 2.8: AC Charging plugs used worldwide [1].

According to Figure 2.8, the connectors are:

- Type 1, used in USA & Japan;
- Type 2, used in Europe;
- Type 3, used in Europe but falling into disuse, being replaced by Type 2 connectors;
- Tesla proprietary connector, used in USA but only with Tesla vehicles.

In addition, there is also the AC connector used in China, which is very similar to the Type 2 connector. In more detail, the pinout of the Type 1, Type 2 and Tesla connectors are shown in Figure 2.9.

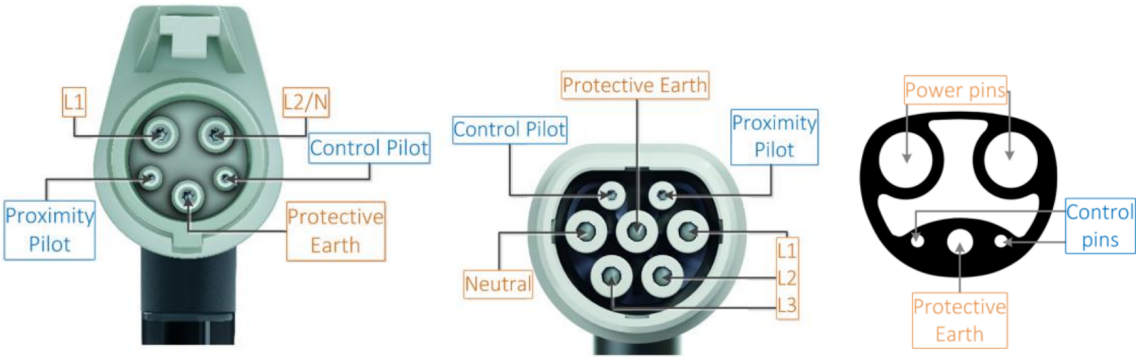


Figure 2.9: AC Charging connectors pinout: Type 1 (on the left), Type 2 (in the centre) and Tesla US connector (on the right) [1].

As shown, the Type 1 connector, which is used specifically for charging at AC single-phase, has a round housing and consists of five pins. Those pins are two for AC power, being one L1 (phase 1) and the other L2 (phase 2) or N (Neutral), a pin for Protective Earth and two communication signal pins, which are Proximity Pilot (PP) and Control Pilot (PP). The PP pin is used to

ensure connectivity between the vehicle and the charging station, while the CP is used to control the charging current. The specified maximum voltage and current values for this charger connector are 120V single-phase or 240V double-phase AC with a current not greater than 80A [1].

The Type 2 connector, also called the Mennekes connector, is circular in shape, but with a flattened top edge. The pins are divided into rows, the first row being composed of the smaller pins for communication, also called Proximity Pilot (PP) and Control Pilot (CP). The next two rows correspond to the 5 pins for three-phase AC power transfer, being three pins for the three-phase AC connection (L1, L2 and L3) and two pins for Neutral and Protective Earth. Specifications for maximum voltage and current values are $230V_{AC}$ single-phase up to 80A and $400V_{AC}$ three-phase up to 63A [1].

Regarding Tesla, a proprietary connector is used in the United States. However, in Europe Tesla vehicles use the Type 2 connector for AC Charging. Unlike other vehicle manufacturers, Tesla uses the same connector for both AC and DC Charging. Using this connector, a maximum charging power of 17.2kW is possible from a $240V_{AC}$ outlet [1].

It is to be noted that Type 1 and Type 2 connectors have two identical communication pins: the Control Pilot (CP) and the Proximity Pilot (PP). Proximity Pilot verifies that the connector is correctly connected to the vehicle's inlet. If the connection is not properly established, Proximity pilot detects it and the charging process is aborted for safety. The second communication pin is the Control Pilot (CP), which is used to control the charging current. The Control Pilot continuously sends a Pulse Width Modulation (PWM) signal to the vehicle. In this way, information about the maximum I_{max} current that can be drawn from the charging station is sent to the vehicle. With this value, the vehicle can draw the required current I_{ac}, as long as this value is less than the maximum current I_{max}. A schema of this communication process is illustrated in Figure 2.10 [1].

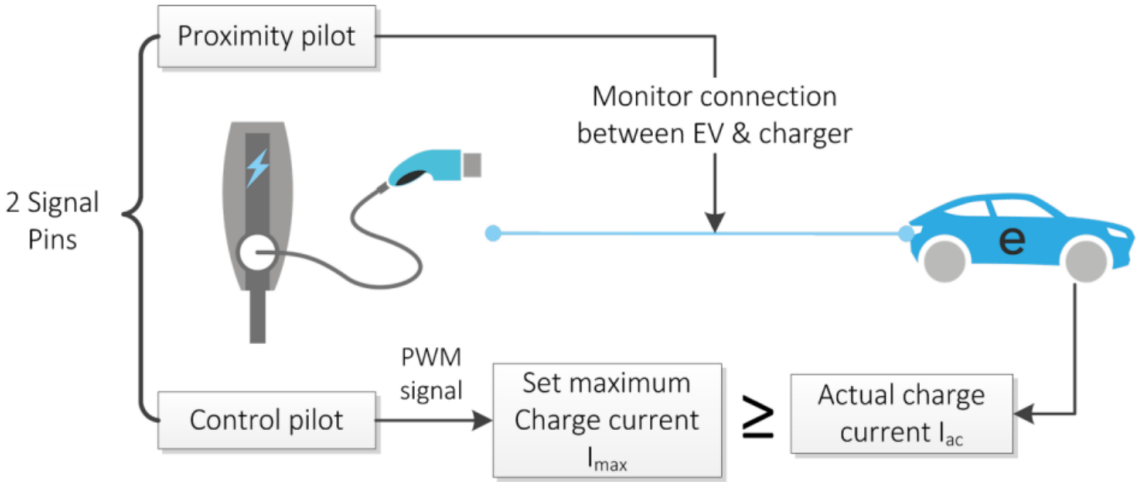


Figure 2.10: AC Charging Type 1 and Type 2 connectors Control Pilot (CP) and Proximity Pilot (PP) schemes [1].

Therefore, it is possible to calculate the power of AC charging. The calculation of single-phase AC charging power is the product of single-phase AC voltage V_{ac} and line current I_{ac}, then [1]:

$$P_{ch} = V_{ac} I_{ac} \tag{2.1}$$

The three-phase AC charging power is calculated as the product of the three-phase AC line-to-line voltage V_{3ac} and the line current I_{ac}, multiplied by the square root of 3, then [1]:

$$P_{ch} = \sqrt{3} V_{3ac} I_{ac} \tag{2.2}$$

In AC power calculations, the values used for voltages and currents are the root mean square (RMS) values. It must also be taken into account that the power supplied by the grid is not the same as that delivered to the battery, due to existing losses in the charging system. Typically, OBCs have an efficiency between 90% and 95% [1].

Based on the presented AC charging power and assuming a lossless system, the charging time t_{ch} can be calculated by dividing the E_{bat} battery capacity by the charging power P_{ch}, therefore [1]:

$$t_{ch} = \frac{E_{bat}}{P_{ch}} \tag{2.3}$$

2.1.3 DC Charging

The composition of the DC Charging procedure is presented in Figure 2.11.

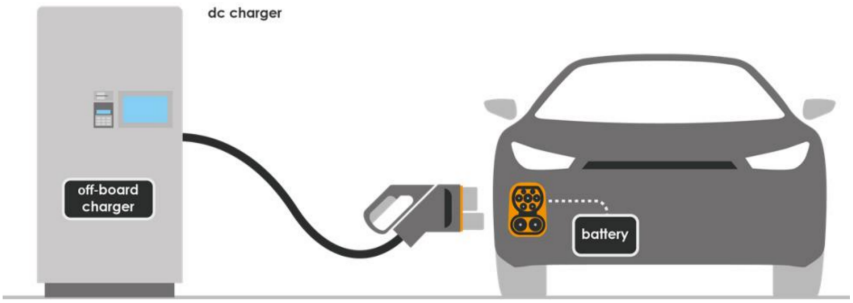


Figure 2.11: DC Charging infrastructure composition schematic [1].

DC fast chargers are proposed to charge Electric Vehicles with a power in the range between 50kW and 350kW. With high power operation, the AC/DC converter, DC/DC converter and power control circuits become larger and more expensive. This is why DC fast chargers like the one in Figure 2.11 are implemented as off-board chargers instead of on-board chargers, since in this way there is not the same dimensional restriction when compared to an OBC and also because in this way the same charging station can be shared by more than one user [1].

In DC Charging, power flows from the DC charger straight to the EV battery. First, the AC power supplied by the mains is converted into DC power using a rectifier inside the DC charging station. The power control unit then adjusts the voltage and current of a DC/DC converter to control the variable DC power delivered to charge the battery. There are also contactors, interlocks and protection circuits that are used to de-energize the connector present in the plug and thereby

interrupt the charging process whenever there is a fault condition or an improper connection between the vehicle and the charger. The BMS plays a key role in communicating with the charging station to control the voltage and current delivered to the battery. The BMS also activates the protection circuits in the event of an unsafe situation. Communication between the vehicle and the charger is carried out at a high level, using communication protocols such as Controller Area Network (CAN) or Power Line Communication (PLC). An illustration of this operating principle is shown in the Figure 2.12 [1].

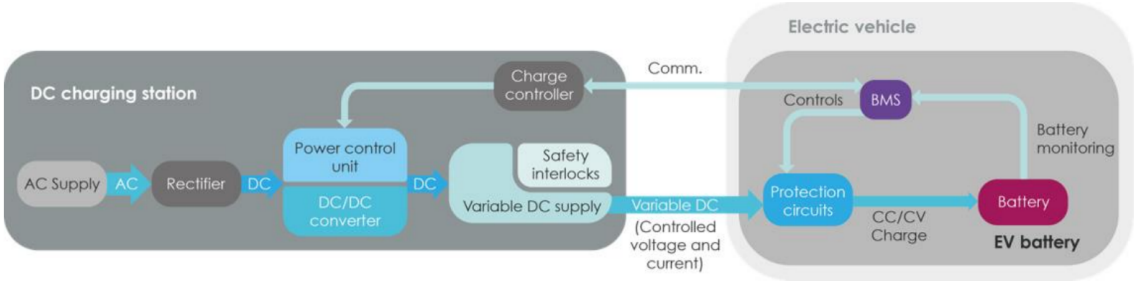


Figure 2.12: Basic simplified DC Charging configuration [1].

Similarly to AC Charging, in DC Charging there are also differences in the connectors used. However, in this case, the adoption of different connectors in different regions of the world is not due to the characteristics of the local electrical grid, but to preferences for charging protocols or government impositions. The different connectors used for DC Charging around the world are shown in the Figure 2.13.

Figure 2.13: DC Charging plugs used worldwide [1].

As shown in Figure 2.13, there are five DC Charging connectors used worldwide, those are:

- CCS - Combo 1, mainly used in the US;
- CCS - Combo 2, mainly used in Europe;

- CHAdeMO connector, used globally in cars built by japanese manufacturers;
- Tesla DC connector, used in the US for Tesla vehicles;
- Chinese GB/T connector, used only in China.

Combined Charging System (CCS) connectors, also called Combo, are integrated connectors for both AC and DC charging. These are derived from Type 1 and Type 2 AC charging connectors, the difference is the addition of two extra large pins for high current DC charging. The connectors are called Combo 1 for the Type 1 derivation and Combo 2 for the Type 2 derivation [1].

The CCS Combo 1 connector pinout is shown in the Figure 2.14.

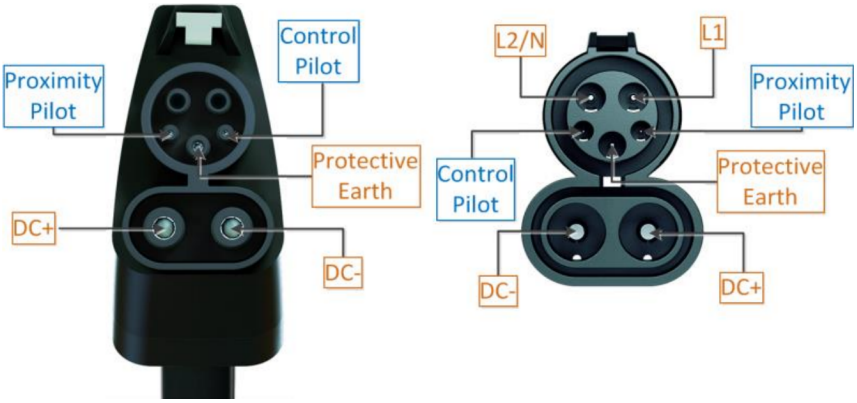


Figure 2.14: DC Charger CCS Combo 1 connector and vehicle inlet [1].

The CCS Combo 1 connector shown in the Figure 2.14 corresponds to an update of the Type 1 connector used in AC Charging. The difference consists in removing the phase pins, keeping the communication pins (Control Pilot and Proximity Pilot) and Protective Earth. In addition, the removed phase pins are replaced with two DC power pins for fast DC charging that are added to the underside of the connector. Thus, a vehicle that presents a Combo 1 inlet can be charged either by DC via CCS or by AC via Type 2 Connector [1].

Likewise, the CCS Combo 2 connector is an update of the Type 2 connector used in AC Charging and similarly also retains the Protective Earth and communication pins, which are the Control Pilot and the Proximity Pilot. In exactly the same way as the Combo 1 connector, two DC power pins are added on the underside of the connector to allow fast high-power DC charging. In this way, a vehicle that presents the Combo 2 inlet can also be charged in AC or DC. The Combo 2 connector, together with the Combo 2 inlet and its pinout, are shown in the Figure 2.15 [1].

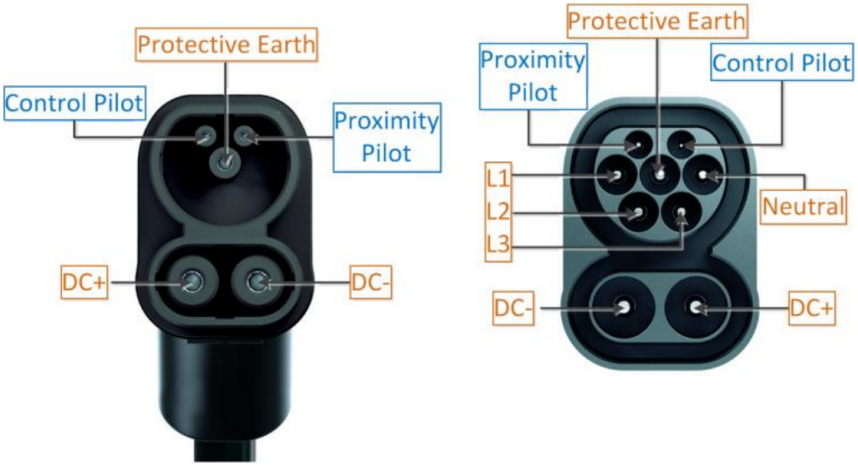


Figure 2.15: DC Charger CCS Combo 2 connector and vehicle inlet [1].

Differently from AC Charging performed using Type 1 and Type 2 connectors, which communicate only at a low level using Pulse Width Modulation (PWM), in Combo 1 and Combo 2 connectors the signaling is done using Power Line Communication (PLC) over the Control Pilot line, in order to establish high-level communication. PLC is a technology that allows the transmission of data for communication over power lines that are used for power transmission. A CCS charger is capable of delivering a current of 350A at a voltage between 250V and 1kV, which allows a maximum output power of 350kW [1].

Another type of DC charger is the CHAdeMO, which in turn is a Type 4 connector and features three power pins and six signaling/communication pins. Communication pins use Controller Area Network (CAN) for signaling. CAN is a robust communication protocol used to interconnect vehicle components, designed to allow microcontrollers and other devices to communicate with each other in real-time without the need for a host computer. An overview of CAN is presented in Appendix A. Currently, the maximum voltage and current values of CHAdeMO are 500V, 400A, which allows a maximum of 200kW of output power [1].

In China, DC Charging is established by the local GB/T standard, which has its own connector, which in turn uses CAN bus for communication. The connector has five power pins, two for DC power, two for low voltage auxiliary power and Protective Earth. This connector features four signal pins, two for Proximity Pilot and two for CAN bus communication. Currently, the rated values for voltage are 750V or 1kV, supporting a current of up to 250A [1].

The CHAdeMO and GB/T connectors and their respective pinouts for DC Charging are shown in the Figure 2.16.

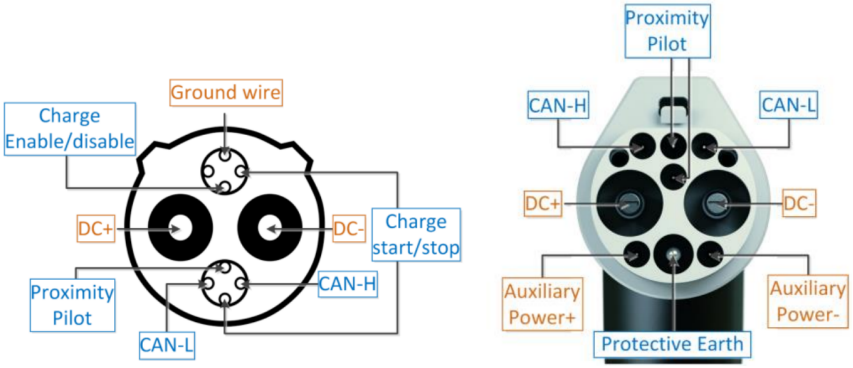


Figure 2.16: DC Charger CHAdeMO (on the left) and Chinese GB/T (on the right) connectors [1].

The other connector to be referred to is the Tesla connector. Tesla superchargers in the US use their own charging connector, while in Europe the Type 2 connector is used, however, designed for DC charging. The differential of the Tesla connector is that it is used for both AC charging and DC charging. Tesla superchargers are capable of delivering up to 120kW. The connectors used for DC charging in Tesla vehicles are shown in Figure 2.17 [1].

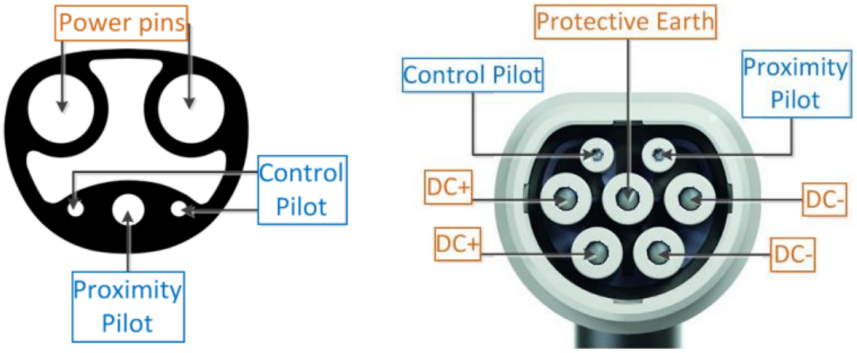


Figure 2.17: DC Charger Tesla proprietary (on the left) and Type 2 in Tesla configuration (on the right) connectors [1].

2.2 Main Standards

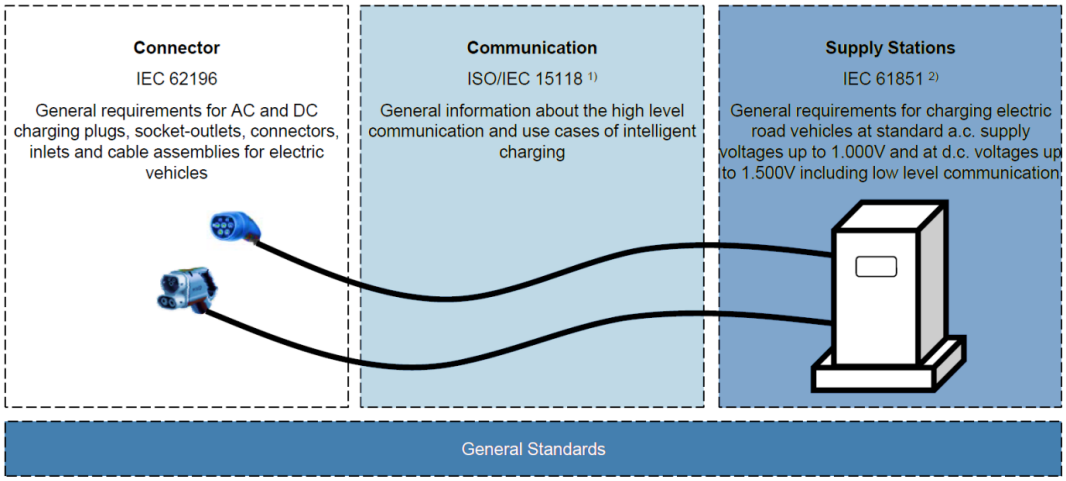
Around the different charging solutions for BEVs, there are many standards that govern it. To be simple and summarized, the three main international standards are presented below.

- Connectors - **IEC 62196:** *General requirements for AC and DC charging plugs, socket-outlets, connectors, inlets and cable assemblies for electric vehicles* [21].

- Communication - **ISO/IEC 15118:** *General information about the high level communication and use cases of intelligent charging* [22].

- Supply Stations - **IEC 61851:** *General requirements for charging electric road vehicles at standard AC supply voltages up to 1000V and DC voltages up to 1500V including low level communication* [7].

Those standards are illustrated in Figure 2.18, aimed at their appropriate applications in the field of electric vehicle charging.



1) In conjunction with ISO/IEC 15118 please note also DIN SPEC 70121
2) Note that IEC 61851-22 as part of IEC 61851 has been withdrawn in 2017.

Figure 2.18: General Standards in Electric Vehicle Charging [3].

2.2.1 Generic Standards

The main standards governing the charging of Electric Vehicles are presented in the next sections. However, there are also other standards proposed by the International Electrotechnical Commission (IEC) and also by the International Organization for Standardization (ISO), not equally discussed (but referenced by them), which refer to particular characteristics, for example: switch gear and control gear assemblies (IEC 61439), standard voltages (IEC 60038), Electromagnetic Compatibility (EMC) (IEC 61000), electrical safety (IEC 61557), protection of enclosures (IEC 60529), electrical installations (IEC 60364), insulation and coordination (IEC 60664), environmental conditions (ISO 16750), test methods for vehicles (ISO 11451) and components (ISO 11452), electrical disturbances (ISO 7637) and test methods for electrical disturbances (ISO 10605). However, as these standards do not fit directly into the scope of the developed work, they are not discussed in depth.

2.2.2 Connector Standards

2.2.2.1 IEC 62196

The main standards that govern Connectors for charging Electric Vehicles are those that make up **IEC 62196: Plugs, Socket-Outlets, Vehicle Connectors And Vehicle Inlets - Conductive**

Charging Of Electric Vehicles, more precisely, in its following parts:

- Part 1: General requirements;
- Part 2: Dimensional compatibility and interchangeability requirements for AC pin and contact-tube accessories;
- Part 3: Dimensional compatibility and interchangeability requirements for DC and AC/DC pin and contact-tube vehicle couplers.

The IEC 62196 standard refers to the ”accessories” in conductive charging systems such as plugs, socket-outlets, vehicle connectors, vehicle inlets and cable assemblies.

Part 1 is where the general requirements for AC and DC charging are dictated, with a rated operating voltage not exceeding [21]:

- $690V_{AC}$, 50-60Hz, at a rated current not exceeding 250A;
- $600V_{DC}$, at a rated current not exceeding 400A.

The main focus is related to the accessories and cable parts being intended for use in an ambient temperature of -30 to +50 °C.

The Part 2 states that the AC plugs, socket-outlets, vehicle connectors, vehicle inlets and cable assemblies for electric vehicle should not exceed the values $480V_{AC}$, 50/60Hz for the nominal rated operating voltage, and 63A three phase or 70A single phase for the nominal rated current, when used in conductive charging of electric vehicles [23].

In Part 2, requirements are presented and the plug types are categorized on their use in AC charging:

- Type 1 - single phase vehicle couplers;
- Type 2 - single and three phase vehicle couplers;
- Type 3 - single and three phase vehicle couplers with shutters.

The Part 3 applies to high power DC interfaces and also the combined AC/DC interfaces of vehicle connectors specified in Part 1 of the IEC 62196 standard, and intended for use in conductive charging systems for the circuits specified in the IEC 61851 standard, precisely in parts 1 and 23 [24].

Part 3 also sets out specifications for high power DC connectors, assuming a maximum value of $1000V_{DC}$/400A at the plugs:

- Combo 1 and Combo 2 (850V, 200A DC);
- Japan CHAdeMO Type 1 (600V, 200A DC);
- China DC Type 2 (750V, 250A DC).

This standard incorporates the signaling from SAE J1772 standard for control and communication issues.

2.2.2.2 SAE J1772

The Society of Automotive Engineers (SAE), which is responsible for some studies in automotive engineering and car manufacturing industries, proposed SAE J1772, which is a North American standard for electrical connectors for electric vehicles and is titled ”SAE Surface Vehicle Recommended Practice J1772, SAE Electric Vehicle Conductive Charge Coupler”. It is covered in it the general physical, electrical, communication protocol, and performance requirements for the electric vehicle conductive charge system and coupler. The purpose is to define a common electric vehicle conductive charging system architecture including operational requirements and the functional and dimensional requirements for the vehicle inlet and mating connector. The different charging solutions are defined in levels in SAE terminology. Those are level 1, level 2 and level 3 charging. The last one refers to DC fast charging [5].

2.2.3 Supply Station Standards

2.2.3.1 IEC 61851

The standard IEC 62196 is based on **IEC 61851: Electric vehicle conductive charging system**, which is the most important standard for the work to be developed on this Dissertation. It sets up general characteristics, including charging modes and connection configurations, and requirements for specific implementations (including safety requirements) of both EV and EVSE in a charging system. For example, it specifies mechanisms such that, power is only supplied if the vehicle is connected and stopped (immobilized), otherwise, the system does not charge [7].

The most relevant parts of this standard for the work carried out are presented below:

- Part 1: General requirements;
- Part 21: EMC Requirements for on-board and off-board chargers;
- Part 23: DC electric vehicle charging station.

It refers to EV supply equipment for charging electric vehicles, setting a supply voltage of up to $1000V_{AC}$ or $1500V_{DC}$ and an output voltage of up to $1000V_{AC}$ or $1500V_{DC}$. Electric Vehicles refers to all road vehicles, including also Plug-in Hybrid Electric Vehicles, which have all or part of their energy coming from on-board Rechargeable Energy Storage Systems (RESS). The main aspects governed by this standard, in this part, include [7]:

- Characteristics and operating conditions of the EV supply equipment;
- Specification of the connection between the EVSE and the EV;
- Requirements for electrical safety for the EVSE.

Part 21 is internally divided into two parts, with Part 21-1 referring to the EMC requirements for on-board chargers, i.e. chargers that are located inside vehicles and Part 21-2 referring to

the EMC requirements for chargers off board, those that are outside the vehicles, in a dedicated station. It gives requirements for conductive connection of an EV to an AC or DC supply. The standard covers the EMC for EVs in any charging mode while connected to the mains supply. It also covers the charging equipment for the charging modes and cases defined Part 1 [25][26].

Part 23 gives requirements for DC EVSE, also referred to as "DC charger", for connection to the vehicle, with an AC or DC input voltage up to $1000V_{AC}$ and up to $1500V_{DC}$ in accordance with the standard IEC 60038, which refers to electrical installations. It also presents the general requirements for control communication between a DC EVSE and an EV [27].

2.2.4 Communication Standards

2.2.4.1 ISO/IEC 15118

ISO/IEC 15118: Road vehicles — Vehicle to grid communication interface was first released in 2014 and defines the digital (high level) communication between an EV and the EVSE. It covers both AC and DC charging modes. One of it's features is *Plug & Charge*, which enables an EV to automatically identify and authorize itself to a charging station on behalf of the driver, to receive energy for recharging its battery. The only action required by the driver is to plug the charging cable between the EV and EVSE. It can be used for both wired (AC and DC) and wireless charging for electric vehicles [22].

As the standards presented in the previous sections, this one also has parts that are of greater relevance to this study. These parts are as follows:

- Part 1: General information and use-case definition;
- Part 2: Network and application protocol requirements;
- Part 3: Physical and data link layer requirements;

Part 1 is the basis for the other parts of the ISO 15118 standard, it is where terms and definitions, general requirements and use cases are specified for conductive and wireless High-Level Communication (HLC) between the Electric Vehicle Charge Controller (EVCC) and the Supply Equipment Charge Controller (SECC). It applies to HLC involved in conductive and wireless power transfer technologies in automatic or manual connection devices. It is also applicable to the transfer of energy either from an EVSE to charge an EV battery or from an EV battery to an EVSE in order to supply power to the home, to loads or even to the grid. It presents an overview and common perception of aspects influencing identification, association, control and optimization of charge or discharge, payment, load balancing, cybersecurity and privacy. This standard provides an interface between EVSEs for all e-mobility actors involving SECCs [22].

In Part 2, communication between EVs and EVSEs is specified. The application layer message set that is defined in ISO 15118-2 is intended to support the energy transfer from an EVSE to an EV. Part 1 contains additional elements of use cases that describe bidirectional energy transfer. Implementing these use cases requires improvements to the application layer message set defined

in Part 2. The purpose of Part 2 is to present the details of communication between an EV and an EVSE. Aspects for detecting a vehicle on a communication network and enabling Internet Protocol (IP) based communication between an EVCC and a SECC are specified. The standard defines messages, data models, XML/EXI based data representation format, use of Vehicle-to-Grid Transfer Protocol (V2GTP), Transport Layer Security (TLS), Transmission Control Protocol (TCP) and IPv6. In addition, it also describes how data link layer services can be accessed from a layer 3 perspective. Data link layer and Physical Layer functionalities are described in Part 3 [28].

The Part 3 specifies the physical layer and data link layer requirements for high-level communication directly between EVs, based on a wired communication technology, and the fixed charging station (EVSE), in addition to basic signaling. This part encompasses the entire exchange of information between all the actors involved in the exchange of electrical energy. The ISO 15118 standard is applicable to manually connected conductive charging, but only to EVSEs of charging modes 3 and 4 defined in IEC 61851-1 [29].

2.2.4.2 DIN SPEC 70121

There is also another standard for communication, the German technical specification DIN SPEC 70121, which is based on a draft version of ISO 15118 standard and defines digital communication between an EV and a DC EVSE. The standard was released in 2012 and updated in 2014, while the first version of ISO 15118 was released in 2014 and its update to version 2 was released in 2018. While the standard only addresses DC charging mode, ISO 15118 covers both AC and DC charging modes. ISO 15118 has evolved a lot and this has resulted in technical differences between DIN SPEC 70121 and ISO 15118. In fact, it is as if they are dialects of the same language. This means that an EVSE that is only ISO 15118 compliant cannot charge an EV that only supports DIN SPEC 70121. However, if the EV is able to "speak both dialects", it can successfully start a charging session. Similarly, the same is true for the EVSE [2][22][30].

The main differences between the standards ISO 15118 and DIN SPEC 70121 reside in the fact that, differently from ISO 15118, DIN SPEC 70121 does not support the Plug & Charge (PnC) feature, this is, there is no secure communication via TLS, no digital certificated and no XML-based digital signatures. So, authenticity and data integrity can't be ensured. The Plug & Charge feature that comes with ISO 15118 enables an EV to automatically identify and authorize itself to a charging station on behalf of the driver, to receive energy for recharging its battery. The only action required by the driver is to plug the charging cable between the EV and EVSE. It can be used for both wired (AC and DC) and wireless charging for electric vehicles. Another distinction is that DIN SPEC 70121 does not support smart charging and Vehicle-to-Grid (V2G), while ISO 15118 supports, which is the concept/technology of bidirectional energy flow between EVs and the grid. This can allow the customer to sell energy to the grid and contribute to electricity demand management [2][22][30].

2.3 Electric Vehicles Charging Definitions

2.3.1 Charging Levels

There is a common definition, adopted in North America and Japan, that categorizes the EVSE into one of three levels, that are presented below [2]:

2.3.1.1 Level 1

In Level 1, is rated for the EVSE to provide charging through a $120V_{AC}$ plug and requires a dedicated circuit. So, in order to have Level 1 charging, just a standard household outlet (in USA and Japan) is needed. As Level 1 charging is on-board, it does not require installation of charging equipment. This level of charging generally takes 8 to 12 hours to fully charge an EV battery. Level 1 charging is not applicable to Europe as the rated voltage of the grid is $240V_{AC}$.

2.3.1.2 Level 2

For Level 2, the equipment offers charging through $240V_{AC}$ plug and requires installation of charging equipment. These units require a dedicated 40A circuit. The connector that plugs into the vehicle for Level 2 charging is the same as for Level 1 and so it is compatible with all EVs and PHEVs. Level 2 charging takes generally 4 to 6 hours to fully charge a battery of an EV. Those chargers can be found in residential setups, public parking zones and commercial locations.

An example of AC charging station is shown in Figure 2.19.

Figure 2.19: AC Level 2 charging station [4].

In Table 2.1, the quantitative aspects of charging levels 1 and 2 are shown.

Table 2.1: AC charging levels [5].

	Voltage	Phase	Peak Current	Power
AC Level 1	120V	Single Phase	16A	1.92kW
AC Level 2	208V	Leg to Leg of a 208T/120V 3-Phase	48A	9.98kW
AC Level 2	240V	Split Phase	30A 32A (2001) 80A (2009)	7.20kW 7.68kW 19.20kW

The SAE J1772 standard defines system configurations for the charging levels 1 and 2. Those are presented in Figure 2.20 and Figure 2.21.

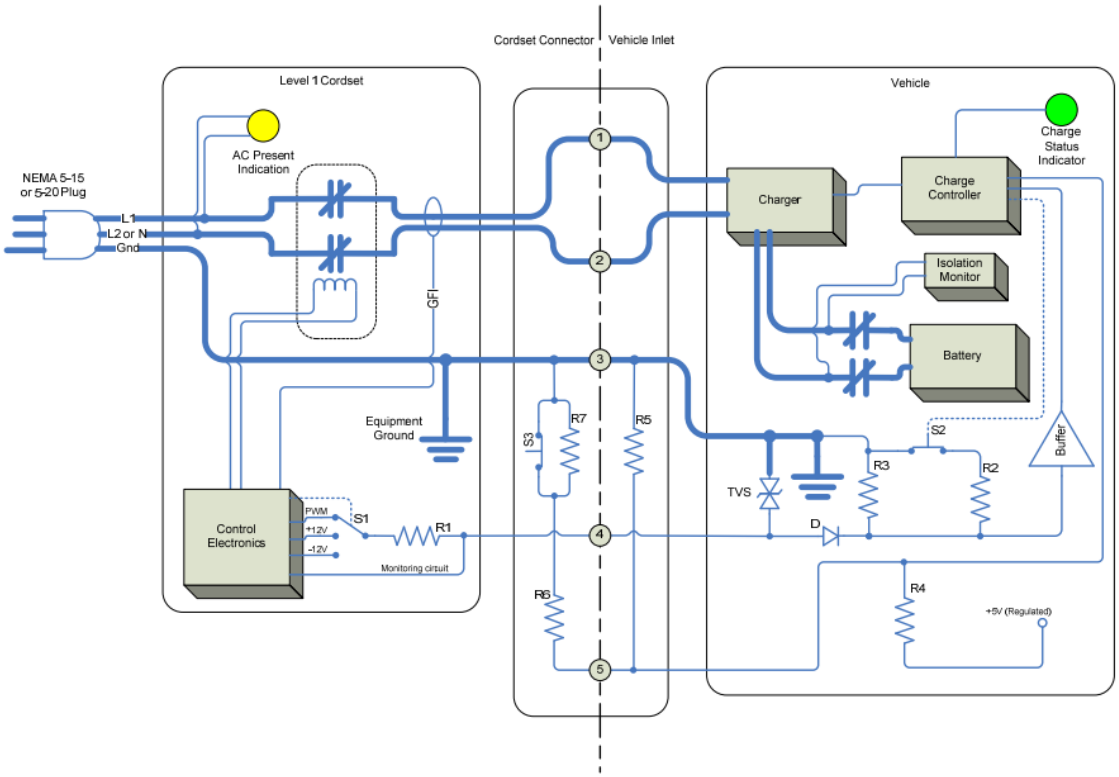


Figure 2.20: AC level 1 system configuration in SAE J1772 [5].

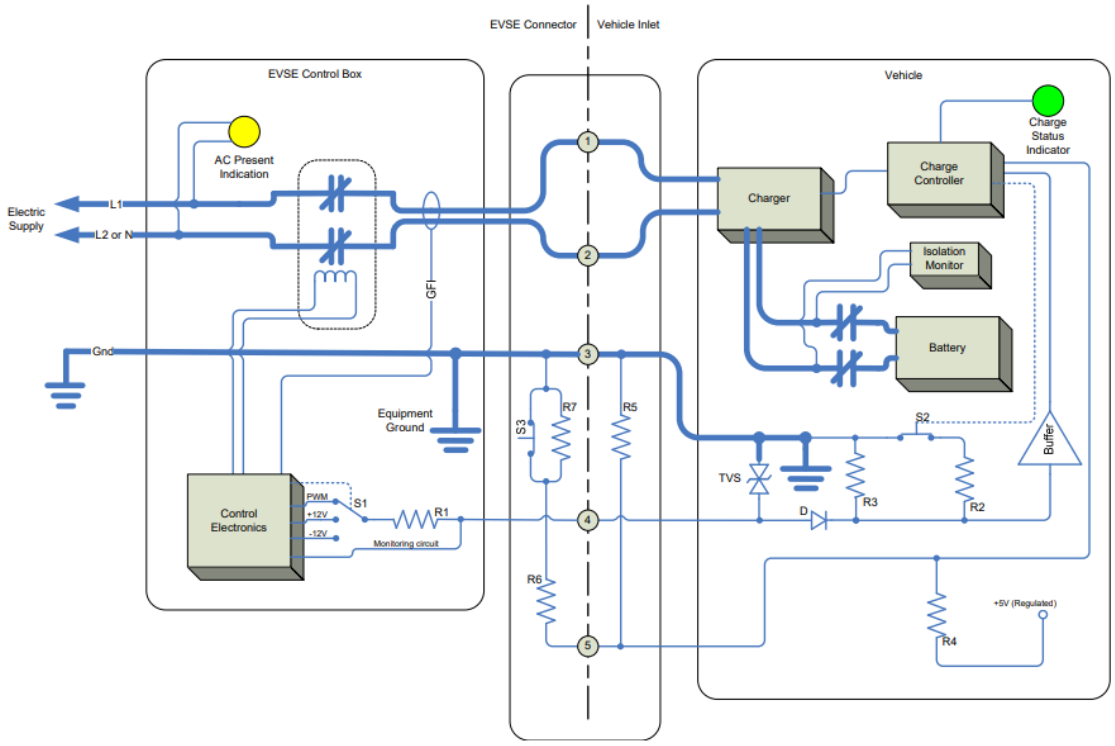


Figure 2.21: AC level 2 system configuration in SAE J1772 [5].

2.3.1.3 Level 3

In Level 3 charging, the solution implies a bypass over the On-board Charger (OBC) and the energy transfer is done in DC. This level is also referred as DC fast charging because it provides an 80% charge in around 30 minutes, with possible variations due to cold weather. Due to it's off-board nature, Level 3 chargers need the installation of robust equipment to be operated.

In SAE J1772-2017, DC Charging is divided into DC level 1 and DC level 2, as described in Table 2.2.

Table 2.2: DC charging levels 1 and 2 in J1772 [5].

Charge Method	**EVSE DC output voltage (V_{DC})**	**Max Current (Amps-continuous)**
DC level 1	50-1000	80
DC level 2	50-1000	400

The DC level 1 and 2 system configurations according to SAE J1772 can be seen in Figure 2.22 and Figure 2.23, respectively.

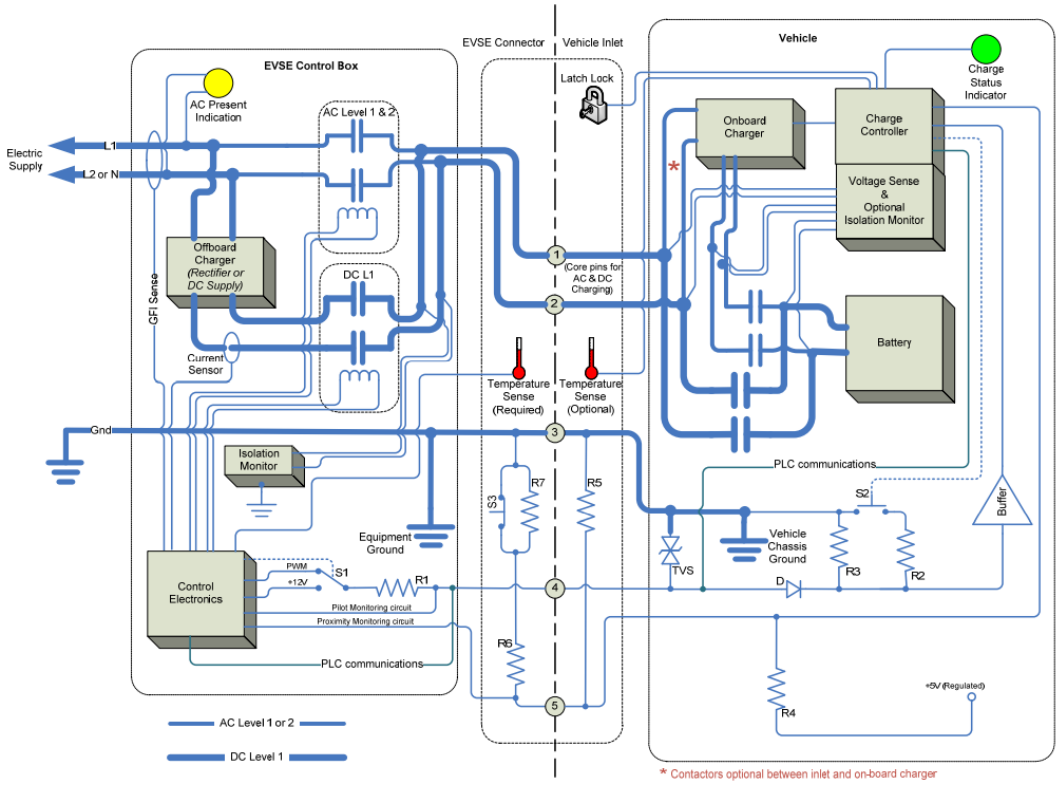


Figure 2.22: DC level 1 system configuration in SAE J1772 [5].

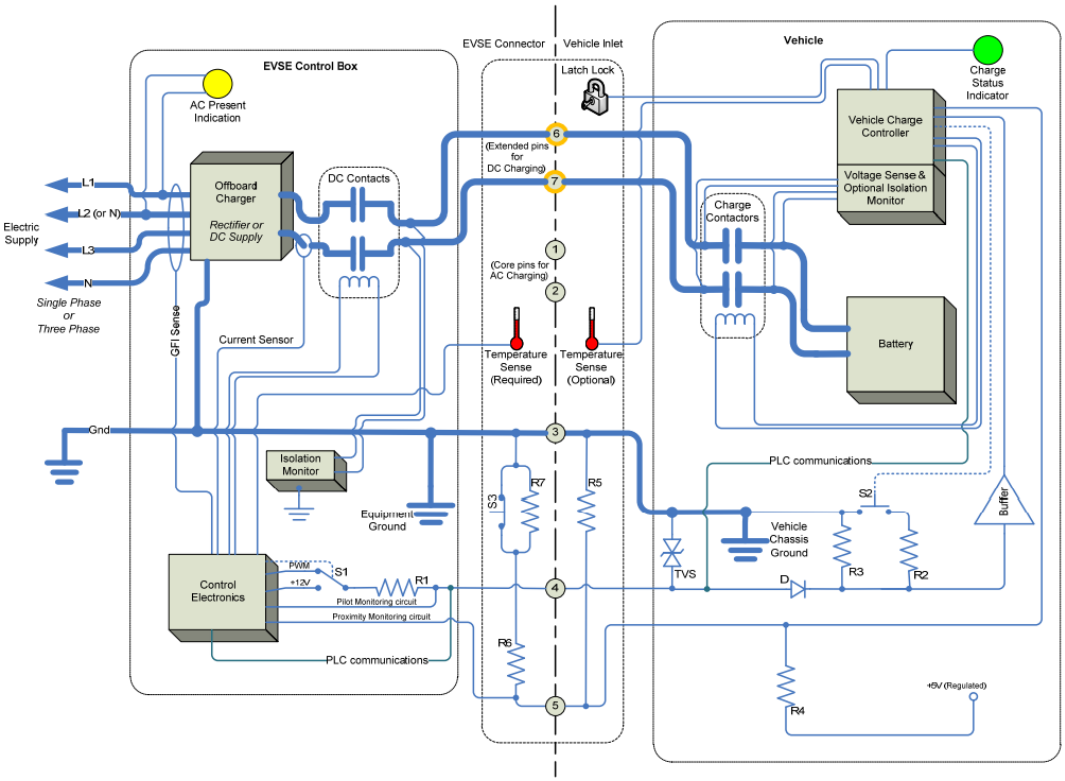


Figure 2.23: DC level 2 system configuration in SAE J1772 [5].

2.3.2 Electric Vehicle Charging around the globe

2.3.2.1 AC Charging

In Europe, for AC charging, the IEC 62196 Type 2 connector was adopted by many countries, also referred to as the *Mennekes* connector, the name of the company that originated the design. Due to restrictions imposed by distribution system operators (DSOs), the consumable current per consumer in low voltage AC distribution systems is limited to $32A_{RMS}$. Therefore, as the standard value for low voltage consumption is $230V_{RMS}$, the AC charging power in Europe is limited to 22kW. In Europe, electrical power is available through single-phase or three-phase connections. It should be noted that in some European countries 1-phase consumption is limited to $16A_{RMS}$, so single-phase AC charging is limited to a power transfer of 3.6kW in these countries [2].

In China, EV charging follows the guidelines of the Guobiao standard GB/T 20234.2-2015, which specifies a modified Type 2-style connector, with a control signaling a bit different from SAE J1772 [12][5].

In China and Europe (with some exceptions), AC charging is realized through single-phase and three-phase. Due to the charging realized via the OBC of the vehicle, the OBC needs to be rated for the grid voltage. For example, in Europe it should be rated as $230V_{LN}$, $400V_{LL}$ [2].

2.3.2.2 DC Charging

For DC fast charging, there is a "battle" over connector and protocols between the car manufacturers in North America, Japan and Europe. The Japanese carmakers prefer and support CHAdeMO, while European and North American carmakers stand for SAE Combined Charging System (CCS1 and CCS2), and Tesla goes on it's own way with it's proprietary Supercharger. The Chinese carmakers employed their own DC fast charging standard, GB/T 20234, commonly known as GB/T DC fast charging [2].

2.3.3 Charging Modes

In the first part of the standard IEC 61851, there are 4 charging modes defined [7].

2.3.3.1 Mode 1 - Slow charging

In this charging mode, the vehicle is connected to the electrical grid through a standard outlet existing in homes, which in turn can be rated around 10A, depending on the country. In order to be able to use Mode 1, the electrical installation must comply with safety regulations and must have a grounding system, a circuit breaker to protect against overload and an earth leakage protection. The outlets have blanking devices to prevent accidental contacts [7]. An illustration of Charging Mode 1 is presented in Figure 2.24.

One of its limitations is the available power, in order to avoid risks of:

- Heating of sockets and cables due to intensive use for many hours at or near maximum power, which can correspond to currents of 8A to 16A, depending on the country;

- Fire or electric shocks if the electrical installation is obsolete or if protective devices are missing;

- As the charging socket is connected to the same distribution board as all the other sockets in the place, if the combined consumption exceeds the limit established by the protection (usually 16A), the circuit breaker will trip, interrupting charging.

The other limitation is related to installation's power management.

- As the charging socket is connected to the same distribution board as all the other sockets in the place, if the combined consumption exceeds the limit established by the protection (usually 16A), the circuit breaker will trip, interrupting charging.

Figure 2.24: EV Charging Mode 1 [6].

Due to these factors, the load power is very limited in Mode 1, for security and quality of service reasons.

This charging mode implies a direct and passive connection from the EV to the AC bus at 250V single-phase or at 480V three-phase, with ground, at a maximum current of 16A. The connection does not have extra pins for control. For electrical protection, it is necessary that the EVSE provides ground to the EV and also that it has protection against ground fault.

2.3.3.2 Mode 2 - Slow charging with protection

In Mode 2, the vehicle is connected to the AC electrical grid also via residential or industrial outlets. Charging is carried out through a single-phase or three-phase network with the installation of a grounding cable. Unlike Mode 1, this one has a protection device embedded in the cable and is therefore more sophisticated and more expensive [7]. An illustration of Charging Mode 2 is presented in Figure 2.25.

Figure 2.25: EV Charging Mode 2 [6].

This mode features a direct and semi-active connection of the EV to the AC mains, at 250V single-phase or at 480V three-phase including earth with a maximum current of 32A. Between the AC mains and the EVSE, there is a passive connection, which must be 0.3 meters from the AC outlet. There is an active connection between the EVSE and the EV, with the Control Pilot terminal performing low-level communication. The EVSE performs protective earth presence detection and monitoring, ground fault, over-current, and over-temperature protection, and functional switching, depending on vehicle presence and charging power demand.

2.3.3.3 Mode 3 - Slow or fast charging with protection and control functions

In Mode 3 charging, the connection between the EV and the AC power supply is carried out using a dedicated EVSE, where the Control Pilot function is intended to control the equipment on the EVSE, permanently connected to the AC electrical grid [7].

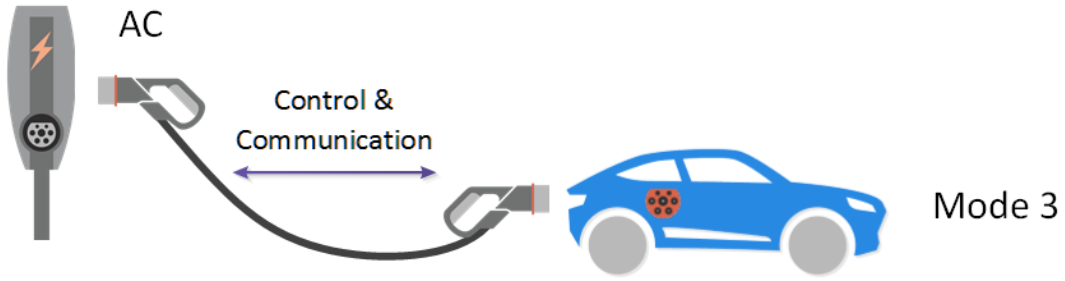


Figure 2.26: EV Charging Mode 3 [6].

In this mode, there is an active connection between the EV and a fixed EVSE, either at 250V single-phase or at 480V three-phase, including ground and Control Pilot. Connection can be via a cable with extra conductors, with a maximum current of 250A or, in order to also support Mode 2 charging, in an optional cable with a maximum current of 32A. Charging is not active by default, and requires proper communication through Control Pilot to enable it. The communication wire between the car's electronics and the charging station allows integration into smart grids.

2.3.3.4 Mode 4 - Fast charging

The connection between the EV and the AC power grid is made using an off-board charger, where the Control Pilot function extends to equipment permanently connected to the AC grid [7].

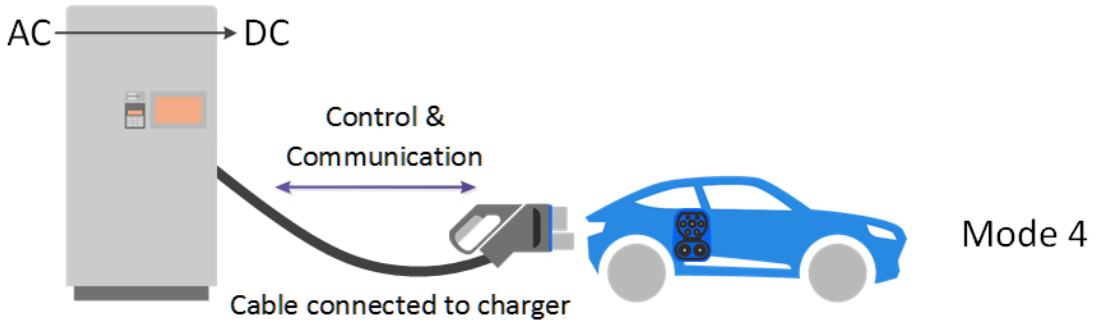


Figure 2.27: EV Charging Mode 4 [6].

In this mode, there is an active connection between the EV and a fixed EVSE, but this time at $600V_{DC}$, including ground and Control Pilot, at a maximum current of 400A. The DC charging power is rectified from the AC power coming from the mains in the EVSE, which makes the charging station more complex and more expensive than a Mode 3 station. However, the fact that the charger is off-board rather than on-board allows the charging station to be larger and more robust, allowing greater charging powers that are impossible in an on-board charger due to dimensional limitations when being inside the vehicle.

2.3.4 Charging Cases

According to IEC 61851-1, the connection of EVs using cables may be carried out in one or more of three different ways [7]:

2.3.4.1 Case A

The connection of an EV to the AC supply grid is made using a supply cable and plug permanently attached to the EV. The cable assembly is part of the vehicle. In Case A, the charging cable can be connected to a domestic or industrial socket or to a specific charging station [7].

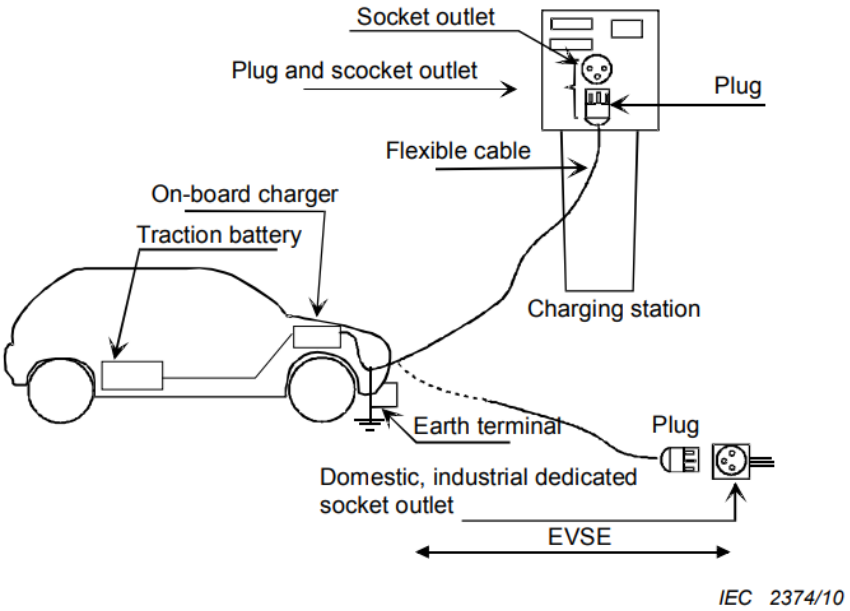


Figure 2.28: EV Case A connection [7].

2.3.4.2 Case B

The connection of an EV to the AC supply grid is made using a detachable cable assembly with a vehicle connector and AC supply equipment. As in Case A, in Case B the charging cable can be connected to a domestic or industrial socket (Case B1) or to a specific charging station (Case B2) [7].

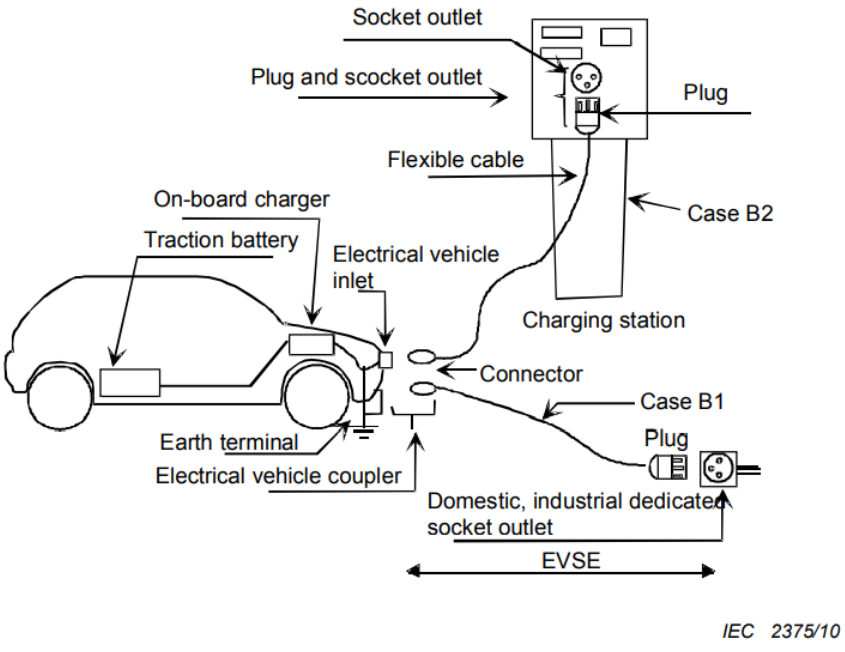


Figure 2.29: EV Case B connection [7].

2.3.4.3 Case C

The connection of an EV to the AC supply grid is made using a supply cable and vehicle connector permanently attached to the supply equipment. Only the case C is allowed for charging in Mode 4 [7].

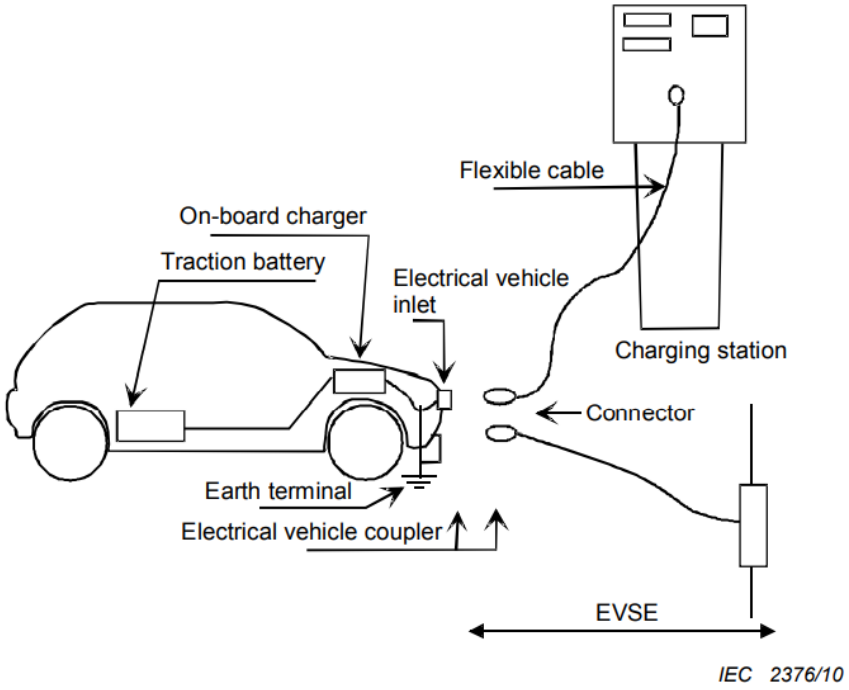


Figure 2.30: EV Case C connection [7].

2.3.5 Plug Types and Charging Protocols

The combination of the different charging cases and charging modes presented above is what gives rise to the different charging protocols that are adopted in different parts of the world, differing mainly in their respective plug types and their signaling and communication. Some plug types are standardized regionally, or internationally (in IEC 62196), while others are established by consortia of companies. The connector is the element of the charging protocol that most characterizes and differentiates it from the others.

2.3.5.1 Type 1 (AC)

The plug model proposed by the SAE J1772 standard for AC charging, commonly known as *J plug*, was established as IEC 62196 Type 1. This is the plug adopted as standard in AC charging in the United States of America and is maintained by the Society of Automotive Engineers (SAE) International [21].

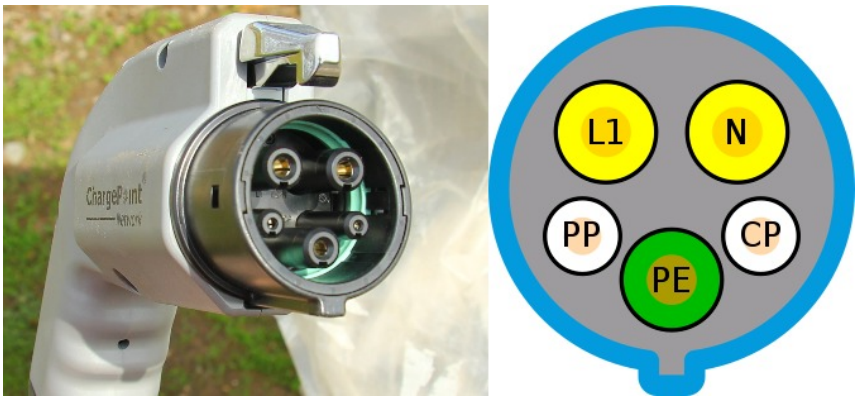


Figure 2.31: IEC 62196 Type 1 plug (SAE J1772 *J plug*) [8].

2.3.5.2 Type 2 (AC)

IEC 62196 also defines the Type 2 connector, also referred to as the Mennekes connector in reference to the company that developed the design. The IEC 62196 Type 2 connector is used in AC charging mainly in Europe. This connector has been selected by the European Commission as the official charging plug in the European Union. Since then, it is also adopted as the recommended connector in some countries outside Europe. This same connector was also released in North America under the name SAE J3068. An illustration of the *Mennekes* connector can be seen in Figure 2.32 [21].

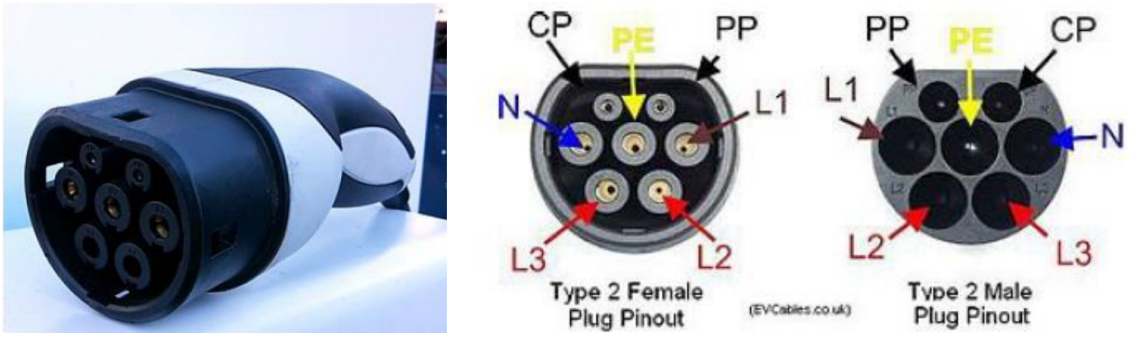


Figure 2.32: IEC 62196 Type 2 plug and pinout [2].

2.3.5.2.1 Signaling and Communication

Originated alongside the IEC 62196 Type 1 connector, SAE J1772 signaling is used on both Type 1 and Type 2 connectors. The signal pins and their respective functions were defined in the SAE J1772 standard, which in turn were included in the IEC 61851. All the plugs defined in the IEC 62196 have the two signals Control Pilot and Proximity Pilot in addition to the power pins (phase and neutral pins, and protective earth) [8].

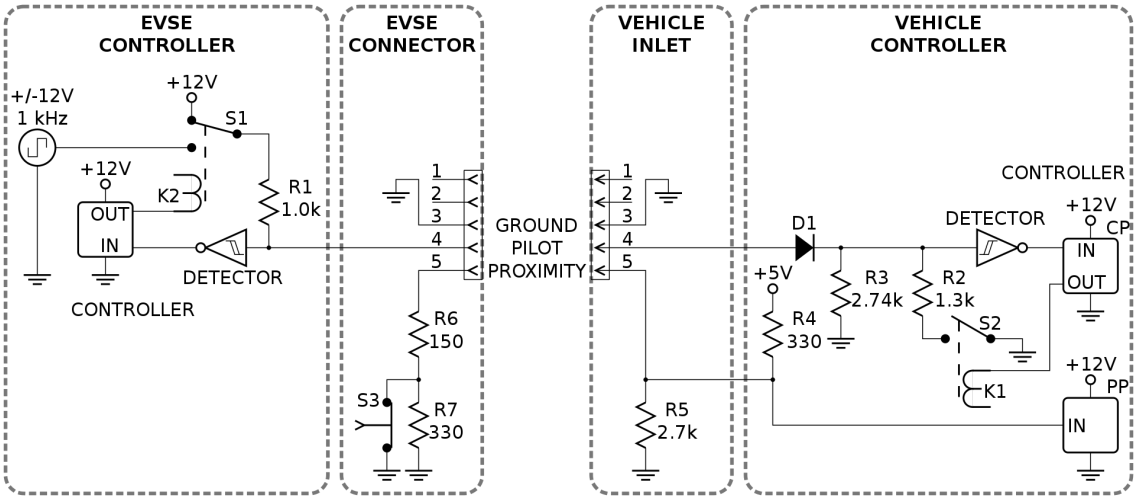


Figure 2.33: SAE J1772 signaling circuit [8].

The signaling protocol has been designed so that [8]:

1. supply equipment signals presence of AC input power;
2. vehicle detects plug via proximity circuit (thus the vehicle can prevent driving away while connected);
3. Control Pilot (CP) functions begin:
   - 3.1 supply equipment detects plug-in electric vehicle (PEV);
   - 3.2 supply equipment indicates to PEV readiness to supply energy;
   - 3.3 PEV ventilation requirements are determined;
   - 3.4 supply equipment current capacity provided to PEV;
4. PEV commands energy flow;
5. PEV and supply equipment continuously monitor continuity of safety ground;
6. charge continues as determined by PEV;
7. charge may be interrupted by disconnecting the plug from the vehicle.

The great advantage of this protocol is that it does not require integrated circuits dedicated for its implementation, which would be required for other charging protocols, making the SAE J1772 robust and operable through a temperature range of -40 °C to +85 °C [8].

2.3.5.2.2 Control Pilot

The control pilot circuit is the primary control means to ensure proper operation when connecting an EV/PHEV to the EVSE.

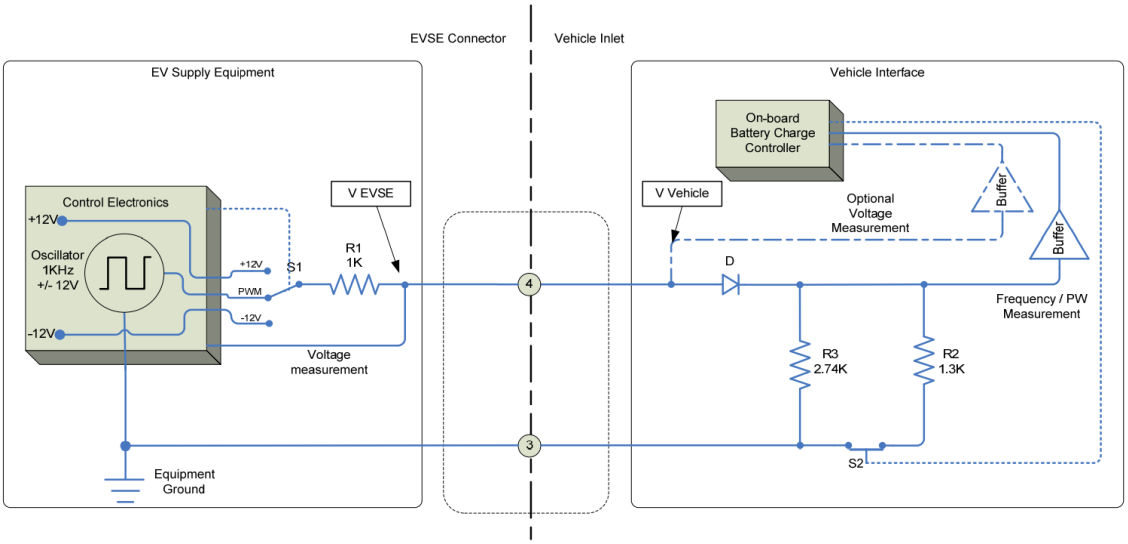


Figure 2.34: SAE J1772 Control Pilot circuit [5].

When the vehicle and the charging station are connected, the EVSE sends a 1kHz square wave signal on the control pilot line, which is connected back to the protected earth on the vehicle side by means of a resistor and a diode (voltage range ±12.0±0.4V). If the circuit is closed, then the EVSE can also test the protective earth to be functional. The vehicle can indicate readiness and request a charging state by setting a resistor and adding it to the voltage divider on the Control Pilot line; with a 2.7kΩ equivalent resistance, a Mode 3 compatible vehicle is announced (vehicle detected) which does not require charging. When a 1.3kΩ resistor is added to the voltage divider, the equivalent resistance now becomes 880Ω indicating the vehicle is ready to be charged and if instead of 1.3kΩ the added resistance is 270Ω, the equivalent resistance is now 240Ω, which indicates the vehicle requests charging with ventilation in which case charging power is only supplied if the area is ventilated [8].

The circuit examples for the Control Pilot line in SAE J1772 show that the current loop between CP and PE is always connected on the vehicle side via a 2.74kΩ resistor, causing the voltage to drop from +12V to +9V when EVSE and PEV are connected, which activates the wave generator. Charging is activated by the vehicle by adding a 1.3kΩ resistor in parallel, resulting in a voltage drop to +6V, or by adding a 270Ω resistor in parallel to indicate a need for ventilation, which results in a voltage drop to +3V. Therefore, the EVSE can respond only by checking the voltage range present in the CP-PE loop. It should be noted that the diode only allows voltages in the positive range; any negative voltage on the CP-PE loop will shut off the current as it is considered a fatal error [8].

Table 2.3: Charging states for the Control Pilot in AC charging [8].

Base Status		**Charging Status**	**Resistance CP-PE**	**Resistance R2**	**Voltage, CP-PE**
State A		Standby	Open Circuit		+12V
State B		Vehicle Detected	2740Ω		+9±1V
State C	C1	Vehicle Ready - EVSE Not Ready	882Ω	1300Ω	+6±1V
	C2	Vehicle Ready - EVSE Ready			
State D	D1	Vehicle Ready - EVSE Not Ready with Ventilation	246Ω	270Ω	+3±1V
	D2	Vehicle Ready - EVSE Ready without Ventilation			
State E		No Power (Shut-off)			0V
State F		Error			-12V

The EVSE can use the signal to describe the maximum current available using Pulse-Width Modulation (PWM): a PWM signal with a duty cycle of 16% represents a maximum of 10A, a PWM signal with a duty cycle of 25% represents a maximum of 16A, a PWM signal with a duty cycle of 50% represents a maximum of 32A, and a PWM signal with a duty cycle of 90% triggers a fast charge option [8].

The duty cycle of the CP PWM signal of 1kHz indicates the maximum possible drawable current. According to SAE, this includes the socket outlet, cable, and the inlet of the vehicle. In the United States, the definition of ampacity (ampere capacity, or current capacity) is divided into continuous and short-term operation. SAE defines the ampacity value as being derived from a formula based on a 1ms full cycle (from the 1 kHz signal) with the maximum current in continuous operation being 0.6A for each 10μs up to 850μs (the lowest value being $100\mu s * 0.6A = 6A$). Above 850μs, the formula requires subtracting 640μs and multiplying that difference by 0.25. For example, ($960\mu s - 640\mu s$) * 0.25A = 80A [8]. In Figure 2.35, it is shown graphically the relation between the maximum available current and the duty cycle of the PWM signal on Control Pilot line.

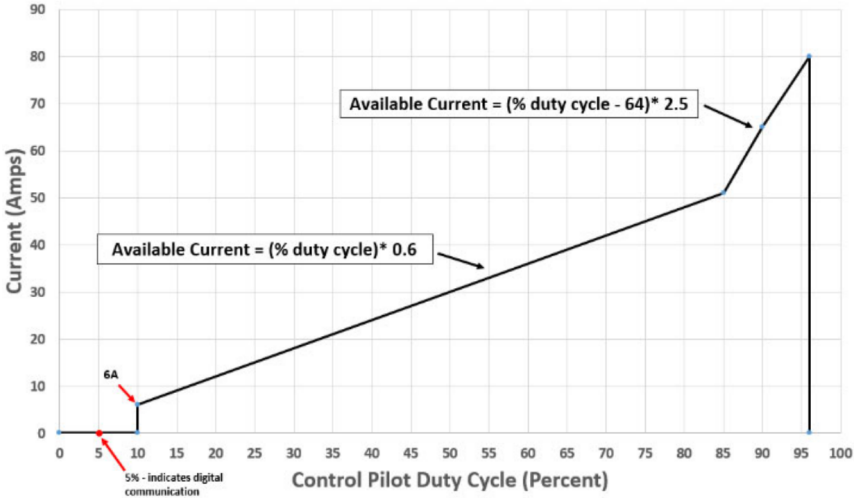


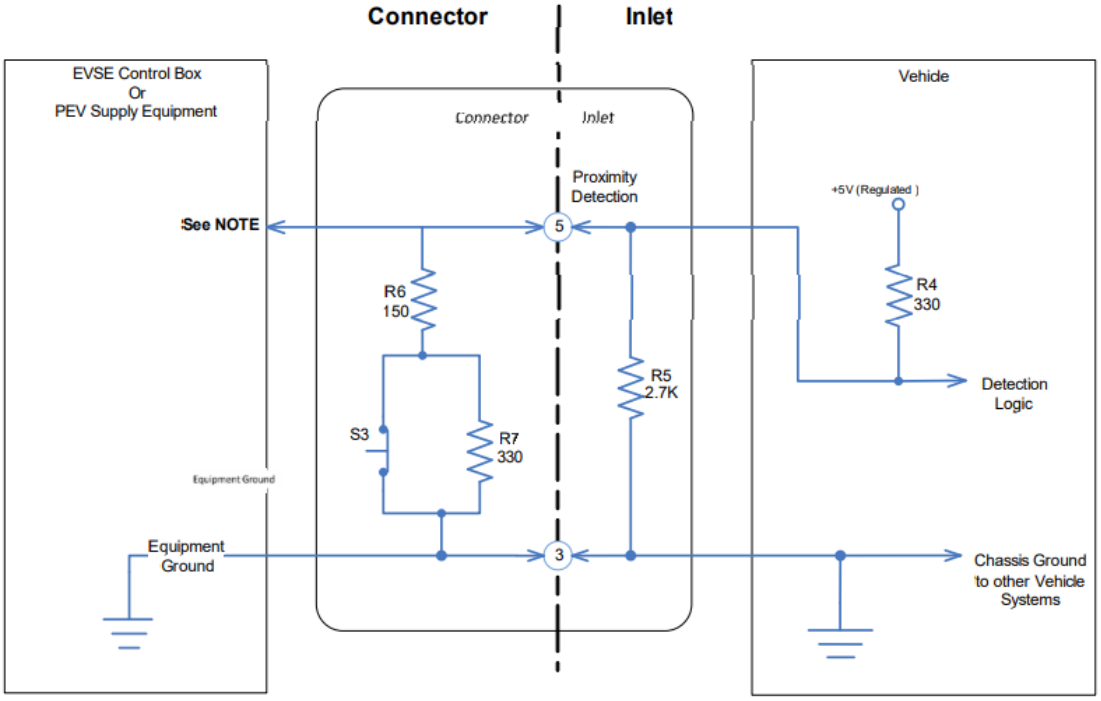
Figure 2.35: Maximum available output current in function of Control Pilot PWM duty cycle, as stated in SAE J1772 [5].

Table 2.4: PWM duty cycle in Control Pilot pin [8].

PWM	SAE Continuous	SAE Short Term
50%	30A	36A
40%	24A	30A
30%	18A	22A
25%	15A	20A
16%	9.6A	
10%	6A	

2.3.5.2.3 Proximity Pilot

The Proximity Pilot pin, PP (also known as Plug Present), as shown in the example circuit on SAE J1772, describes the switch S3 as being mechanically linked to the connector latch release actuator. During charging, the EVSE side connects to the PP-PE loop through S3 and a 150Ω resistor R6; when the actuator is released, a 330Ω resistor R7 is added to the PP-PE loop on the EVSE side, which leads to a voltage shift on the Proximity Pilot line to allow the electric vehicle to initiate a controlled shutdown before the power pins are disconnected. However, many low-power adapter cables do not provide this detection of the state of the locking actuator on the PP pin [5].



NOTE: R5 is required for both AC and DC charging. Monitoring of the Proximity Detection Circuit at the EVSE is optional for AC charging and mandatory for DC charging.

Figure 2.36: SAE J1772 Proximity Pilot circuit [5].

In the IEC 62196 standard, the Proximity Pilot pin is also used to indicate cable capacity, this is relevant for untethered EVSEs. The resistor is coded for the maximum current carrying capacity of the cable assembly. The EVSE stops the current supply if the current capacity of the cable is exceeded. This is detected by measuring Rc (shown as R5 in the circuit above). Values in the recommended interpretation range are shown in the Table 2.5. Rc is placed between PP and PE, with the detachable cable assembly [5].

Table 2.5: EVSE Proximity Pilot Resistance [5].

PWM	**Max. Current**	**Conductor Size**
Open or $\infty\Omega$	6A	$0.75mm^2$
1500 Ω	13A	$1.5mm^2$
680Ω	20A	$2.5mm^2$
220Ω	32A	$6mm^2$
100Ω	63A	$16mm^2$
50Ω or < 100Ω	80A	$25mm^2$

2.3.5.3 CHAdeMO (DC)

In Japan, AC charging for EVs is done through the same plug that is adopted in North America, the J1772. In DC fast charging, the CHAdeMO standard is used as the national standard for charging EVs.

CHAdeMO is the trade name for a fast charging method for electric vehicles, providing up to 62.5kW by 500V, $125A_{DC}$ via a special electrical connector. In its version 2.0, the CHAdeMO specification allows up to 400kW per 1000V, $400A_{DC}$. Its charging plug and pinout are presented on Figure 2.37 [9].

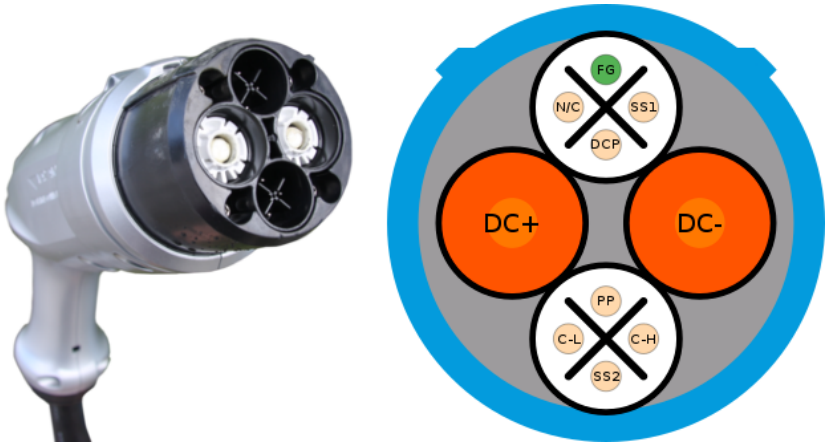


Figure 2.37: CHAdeMO plug [9].

Table 2.6: CHAdeMO plug pin description [9].

Pin	Name	Description
FG	Gound	Reference for control lines
SS1/SS2	Charge sequence signal	Start/Stop charging
N/C	(not connected)	
DCP	Charging enable	Vehicle grants EVSE permission to connect power
DC+/DC-	DC power	Supplied power
PP	Connector proximity detection	Charge interlock, disables drivetrain while connected
C-H/C-L	CAN bus	Communication with vehicle bus

2.3.5.3.1 Signaling and Communication The basic communication between EVSE and EV in the CHAdeMO standard is performed via a Controller Area Network (CAN) bus. The simple circuit illustrating the plug connections is shown in Figure 2.38.

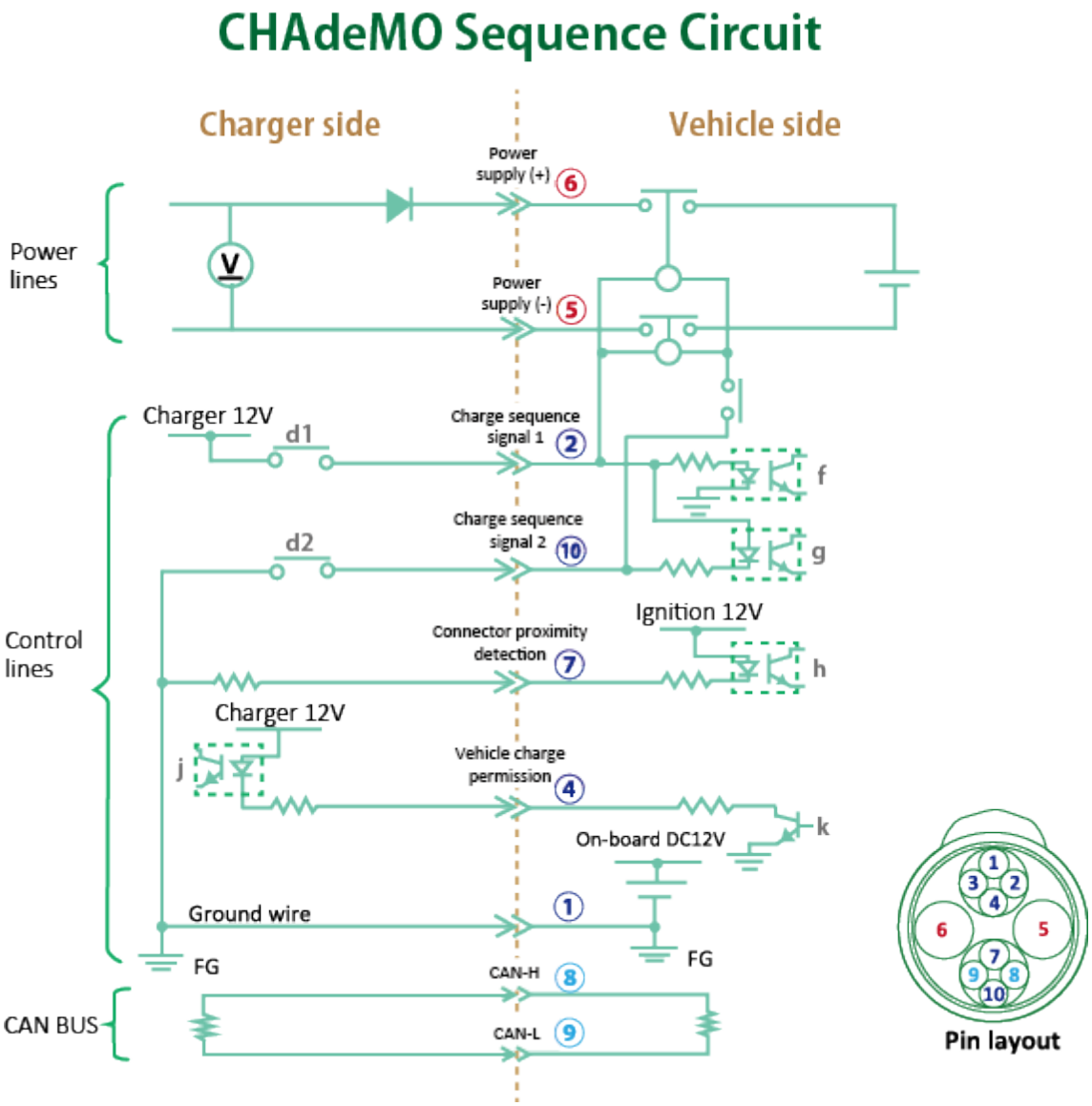


Figure 2.38: CHAdeMO plug connector interface [10].

The charging sequence in CHAdeMO from the start to the end of it is presented as follows.

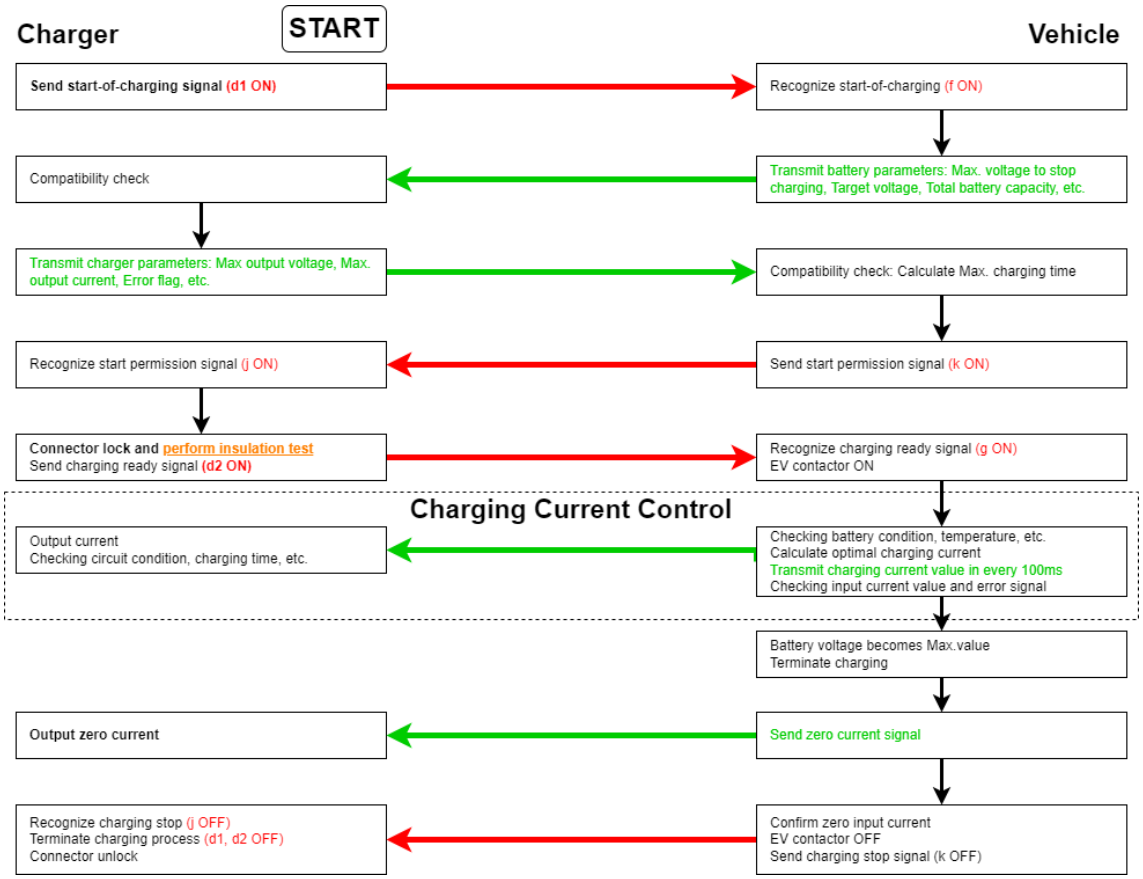


Figure 2.39: CHAdeMO charging sequence [11].

2.3.5.4 GB/T (AC and DC)

As mentioned above, in China, the characteristics of EV charging are defined by GB/T standard sets, mainly by GB/T 18487, GB/T 20234 and GB/T 27930. Among these characteristics are the system requirements, plug and connection types, as well as communication and signaling for conductive charging.

2.3.5.4.1 GB/T - AC For AC charging, the plug is identical to the IEC 62196 Type 2 used in the rest of the globe, with a control pilot communication a bit different from the one proposed in IEC 61851-1. The system diagram for the charging cases A, B and C are presented on the Figures 2.40, 2.41 and 2.42, respectively.

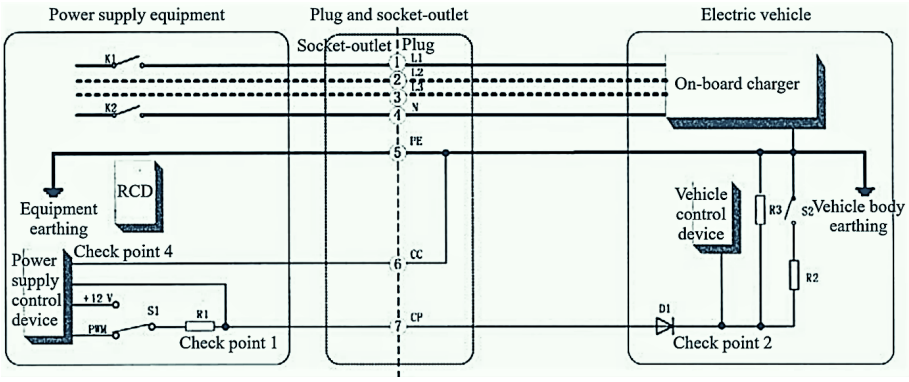


Figure 2.40: Control Pilot Circuit in Charging Mode 3 and Case A Connection in GB/T 18487 [12].

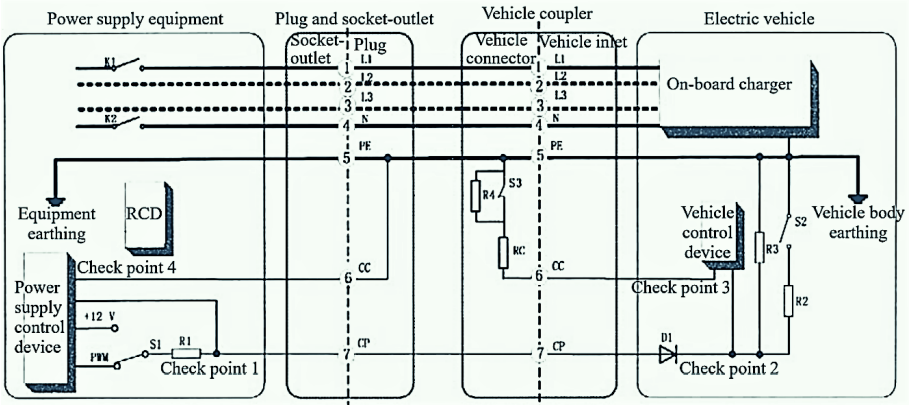


Figure 2.41: Control Pilot Circuit in Charging Mode 3 and Case B Connection in GB/T 18487 [12].

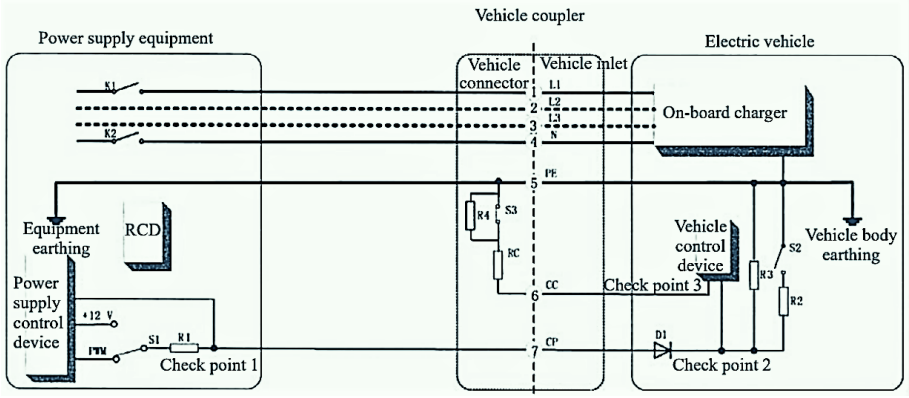


Figure 2.42: Control Pilot Circuit in Charging Mode 3 and Case C Connection in GB/T 18487 [12].

2.3.5.4.2 GB/T - DC For DC Charging, the system counts with High Level Communication, which is carried out through a CAN bus. This communication is supported by the Chinese standard GB/T 27930. The GB/T plug and its pinout for DC charging is presented on the Figure 2.43.

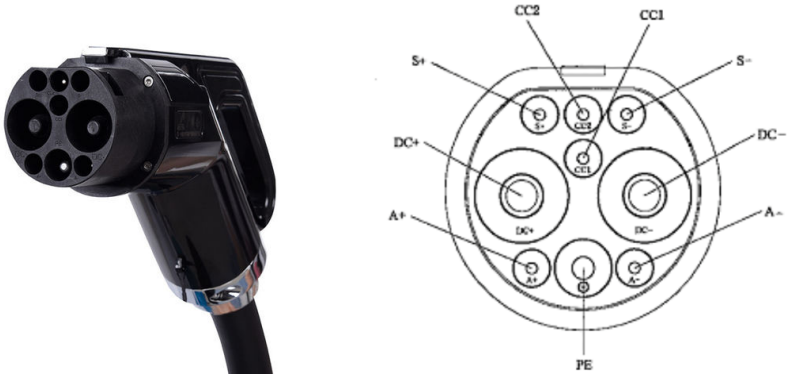


Figure 2.43: GB/T plug and its pinout (adapted from [13]).

The system diagram for the DC Charging, according to GB/T 18487, is presented on Figure 2.44.

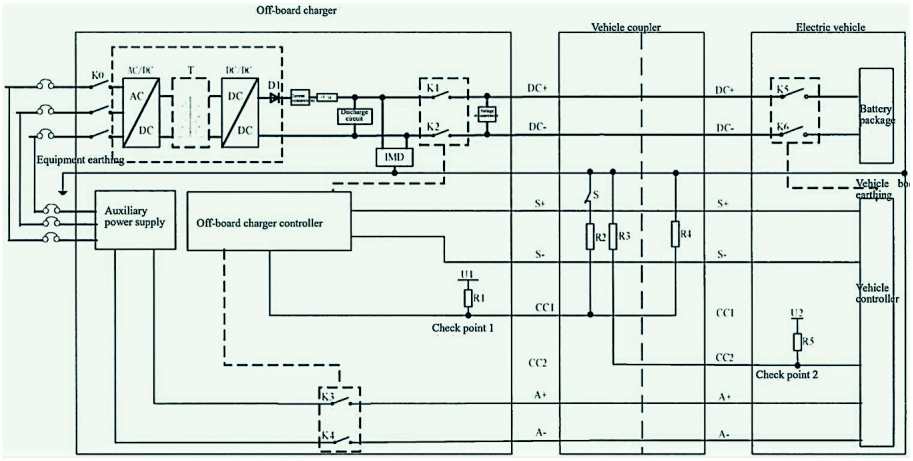


Figure 2.44: Control Pilot Circuit in DC Charging Mode in GB/T 18487 [12].

Due to the fact that this set of standards is implemented only in the People's Republic of China, its particularities are not discussed in depth in the scope of this work.

2.3.5.5 Combined Charging System

In order to unify the EV charging system in the world, the Society of Automotive Engineers (SAE) and the European Automobile Manufacturers Association (ACEA) have proposed a standard that adds DC connections to existing and already used AC connectors around the world, then there is a "global envelope" in cars that can be charged into any EVSE. In this new standard, known as Combined Charging System (CCS) DC charging, two extra connectors are added at the bottom of the IEC 62196 Type 1 and Type 2 connectors, allowing charging at high voltage DC charging stations, with a direct connection to the battery, bypassing the vehicle's on-board charger (OBC). These new plugs are commonly known as Combo 1 and Combo 2 connectors. The choice between one or the other is established based on country/region. As in North America the connector established for AC charging is IEC 62196 Type 1, the associated CCS connector is Combo 1.

Likewise, in Europe and in countries that adopt IEC 62196 Type 2, the CCS Combo 2 connector is used [31].

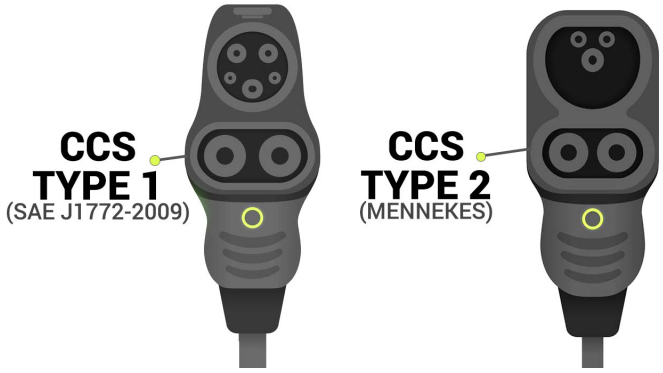


Figure 2.45: Combo 1 and Combo 2 connectors [14].

2.3.5.5.1 Signaling and Communication

For signaling and communication, the CCS uses exactly what is implemented in the SAE J1772 standard for Control Pilot and Proximity Pilot as shown in Section 2.3.5.2. However, CCS also takes advantage of high level communication, using Power Line Communication PLC), according to HomePlug Green PHY (HPGP) [32].

2.3.5.5.2 Power Line Communication

Using an off-board charger requires establishing a digital communication link between the Battery Energy Control Module (BECM) and the off-board charger. An obvious and predictable candidate for this type of communication is a Controller Area Network (CAN) bus. However, CAN requires a differential pair (CAN_H & CAN_L) to implement, which would require two additional pins on connector J1772. In order to overcome this physical impediment, the SAE Hybrid Communication Task Force decided to implement Power Line Communication (PLC) on the Control Pilot line. The specific technology chosen for communication between EV and EVSE is HomePlug Green PHY (HPGP). Therefore, in a standard updated in 2012, SAE proposes the use of Power Line Communication, specifically IEEE 1901, between the vehicle, the off-board charging station, and the smart grid, without requiring an additional pin [32].

2.3.5.5.3 HomePlug Green PHY

HomePlug Green PHY is a Broadband PLC (BB PLC) technology in which digital data is modulated on multiple carrier frequencies via Orthogonal Frequency-division Multiplexing (OFDM). Digital communication between the BECM and the off-board charger is established by coupling the HPGP signal to the PWM square wave on the Control Pilot line. Therefore, only one pin is used for communication for both AC level 1 and AC level 2 (PWM) charging and DC level 1 and DC level 2 (BB PLC) charging [32].

2.3.5.5.4 SLAC

In order to make a reliable association between PEV and EVSE, a special feature called Signal Level Attenuation Characterization (SLAC) was added to the HomePlug GP specification. When SLAC is invoked by a HomePlug GP client, it transmits a series of unknown *sounding* packets. Stations within the listening range compute the average power received by each packet and report this information to a designated Media Access Control (MAC) address. The results are compiled and the station with the highest received signal strength is identified [32].

2.3.5.5.5 Charging Sequence

In Figure 2.46 it is illustrated the Charging Sequence for the Combined Charging System, which is presented and described in SAE J1772.

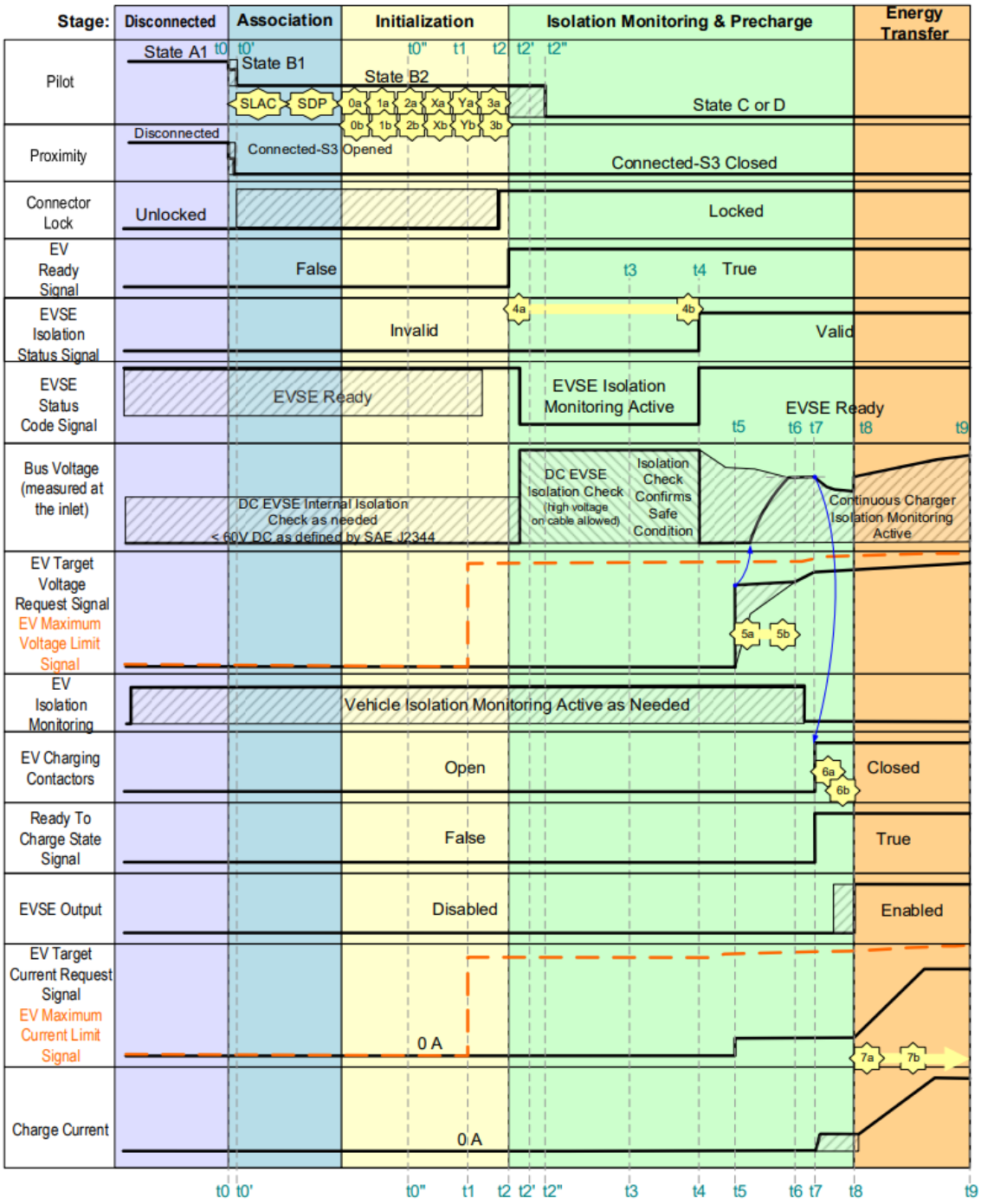


Figure 2.46: CCS Charging Sequence [5].

2.3.5.6 Tesla Supercharger

The Tesla Supercharger is a $480V_{DC}$ fast charging technology developed by North American vehicle manufacturer Tesla.

Tesla Supercharger V1 and V2 charging stations offer up to 150kW of power distributed between two vehicles with a maximum of 150kW per vehicle, depending on version and model. The

Model S, with an 85kWh battery, takes around 20 minutes to charge half of its charge, 40 minutes for 80% and 75 minutes for 100%. These charging stations provide high power DC charging directly to the battery, bypassing the internal On-board Charger (OBC) [33].

Tesla Supercharger technology is unique to Tesla vehicles and patented by Tesla. Therefore, its technical details are not deeply discussed in this study.

2.4 Conclusion

After researching the concepts in the field of EV charging and addressing the standards and protocols implemented in the market, it can be concluded that the best option to be implemented in this work is AC charging, following the SAE J1772 standard. Its simplicity and the potential for system evolution to a later implementation of greater power are the attractive factors of this option. The use of the three-phase grid already generated by the company's system is also positive. The main advantage of this approach is the interoperability provided by the AC charging proposed by SAE J1772, allowing the prototype to work with most EVSEs in the market. In addition, its ease of implementation compared to other charging protocols was also a factor of high importance.

Chapter 3

Design and Implementation

3.1 Introduction

For the development of the intended solution, an idealization of what would work as the final system was first made. Therefore, a System Level schema was made in order to represent a prediction of the resulting functional prototype. This schema is presented in Figure 3.1.

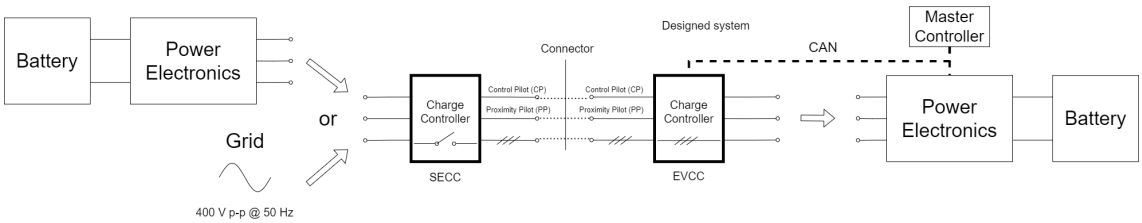


Figure 3.1: Schema of the final working system.

As shown in the diagram, the idea is to have two charge controllers, the Supply Equipment Charge Controller (SECC) on the charging station side and the Electric Vehicle Charge Controller (EVCC) on the vehicle/load side. The designed system allows AddVolt's unit to act both as an EVSE and as an EV to be charged, ensuring a proper and safe power flow between the two entities involved in this charging process.

It should be noted that, only with the SECC connected to the grid or to the AddVolt's unit, it should be possible to charge an Electric Vehicle, since, internally, it already has an EVCC and an On-board Charger (OBC).

The SECC is the controller that, in addition to controlling the opening and closing of the contactors connected to the power bus, also sends information on the maximum output current capacity, through the duty cycle of a PWM signal at the Control Pilot terminal, to be later processed by the EVCC. The SECC was initially intended also to communicate with an insulation meter via a CAN bus to ensure safe power flow. However, due to availability of modules, this feature was not implemented. On the other hand, the system was still designed to receive this addition in

the future, having its Hardware with the necessary components and its Firmware with the space provided for adding code.

Regarding the EVCC, firstly it monitors the plug presence through the Proximity Pilot terminal before any charging step, then it reads the PWM signal generated by the SECC at the Control Pilot terminal and, in response, acts on a voltage divider in order to establish a low-level communication with the charging station. In addition, it also translates the information contained in the duty cycle of the PWM signal into a current value in Ampère, to be sent through messages on a CAN bus to the Master Controller of the system, which in turn passes this information on to the Battery Management System (BMS), in order to, through its charging algorithm, determine the current set-points to be drawn by the OBC. However, the BMS, the charging algorithm of the BMS and the OBC did not make up the work developed since it is already implemented by the battery manufacturer. The EVCC only sends CAN messages and receives information on the actual values of charging current, battery voltage and State of Charge (SoC) of the battery pack.

The two systems make up State Machines, having their low-level communication established in accordance with what is described in the SAE J1772 standard, making up the whole charging sequence.

3.2 Environment

As the project itself is based on a system commanded by microcontrollers, an option must be chosen for it. The microcontroller selected to control the system and communicate with the other blocks is the Infineon XMC4500. More precisely, the Relax Kit development board, which has the Infineon XMC4500 Microcontroller attached to it. This device is benefited of being able to be programmed through DAVE™ development environment, which greatly facilitates software development, using low-level drivers, application libraries, an automatic code generation engine and interfaces to generic or customer-oriented development tools. In Figure 3.2, the XMC4500 microcontroller is shown.

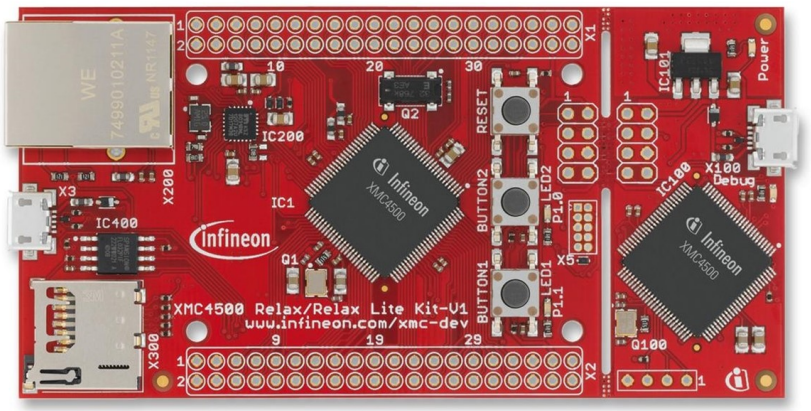


Figure 3.2: Infineon XMC4500 Relax Kit Microcontroller development board [15].

It has the following relevant features:

- 512 - 1024 KB Flash, 128-160 KB RAM;
- Supply voltage range: 3.13 - 3.63V;
- USIC 6ch [Quad SPI, SCI/UART, I^2C, I^2S, LIN];
- 2x PWM Timers (CCU8), 16-64Bit 8ch+ Dead-Time;
- 3x CAN, 64 MO;
- Peripherals Clock: 120 [MHZ];
- Core frequency: 120 [MHZ];
- 4x $\Delta\sum$-Demodulator;
- External Memory Interface (EBU);
- Temperature range: -40° .....85°/125°;
- 10/100 Ethernet MAC (/w IEEE 1588).

In the field of EV charging, the microcontroller has the following benefits:

- Platform concept to allow extensive customization;
- Performance, efficiency and cost competitiveness;
- Accurate analog-mixed signal peripherals;
- Fast timer/PWM peripheral.

The features presented can be summarized in the microcontroller architecture, which is shown in Figure 3.3.

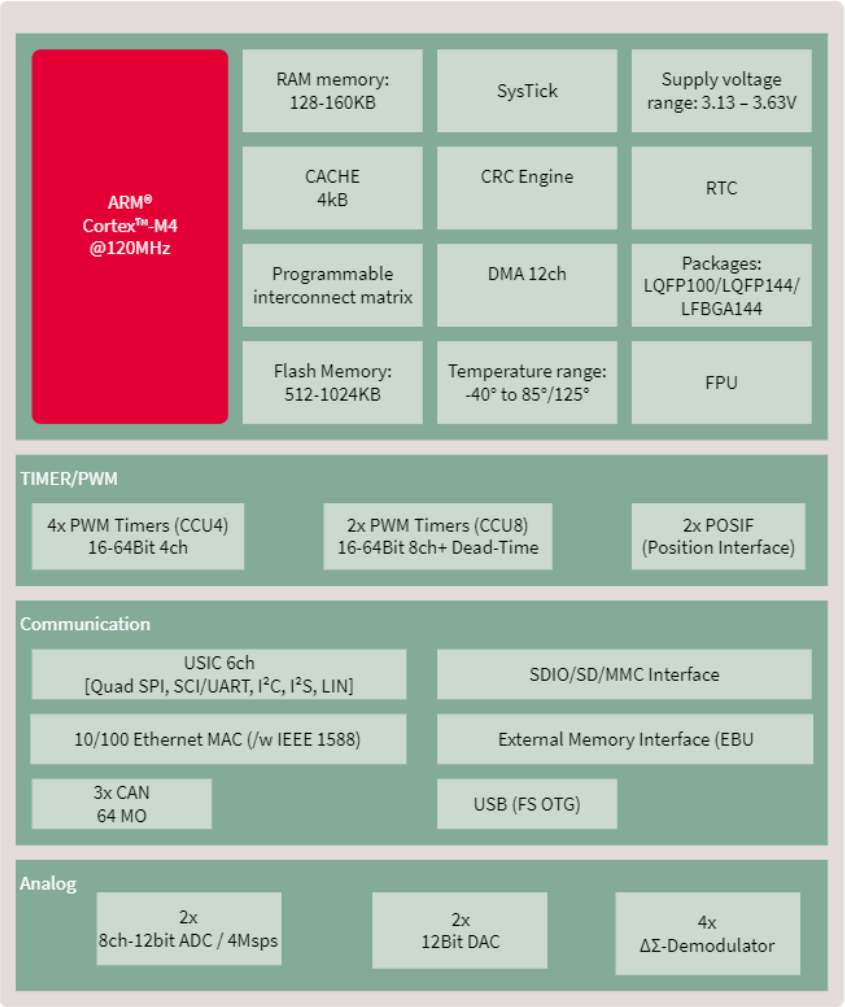


Figure 3.3: Architecture of Infineon XMC4500 Microcontroller [15].

3.3 Workplan

A division was made into two implementation domains, Hardware and Firmware. Regarding Hardware, a circuit was designed and later printed to be connected as a "shield" to the Relax Kit development board. Regarding the development of the Firmware, the IDE DAVE^{TM} was used in order to implement the functions that were necessary for the proper communication between the charge controllers, thus allowing the safe activation of contactors, together with an efficient power flow between the systems involved.

In addition to the division made between the Hardware and Firmware approaches, what would be implemented in the SECC and in the EVCC was also divided, since each of these controllers has different functions and, therefore, has both their circuit in their Printed Circuit Board (PCB) and their Firmware different from each other.

3.4 Hardware

For the hardware approach, the circuits involved in the low-level communication process and charging control were first designed and idealized. Then, a PCB was drawn in order to have a final prototype.

On the power source side, the controller was designed in order to control the opening of contactors between the electrical grid (or AddVolt's unit) and the load (an Electric Vehicle or AddVolt's unit). Besides that, this same controller has to communicate with the EVCC via a PWM signal through the Control Pilot terminal, the duty cycle of this signal is a representation of the maximum current available for charging.

On the load side, the EVCC was designed in order to communicate with the SECC through the Control Pilot terminal and with AddVolt's unit via a Controller Area Network (CAN) bus, which in its way communicates with the OBC, which basically consists of power electronics, sending the proper current set-points determined by the Battery Management System (BMS), which is drawn by the OBC in order to charge the batteries.

3.4.1 Supply Equipment Charge Controller (SECC)

3.4.1.1 Overview

After checking the main functionalities and characteristics presented in the SAE J1772 Charging Protocol standard, a schematic of the needed circuit was drawn in order to illustrate what would be present on the PCB to be designed. The schematic for the circuit of the SECC as a block diagram is shown in Figure 3.4.

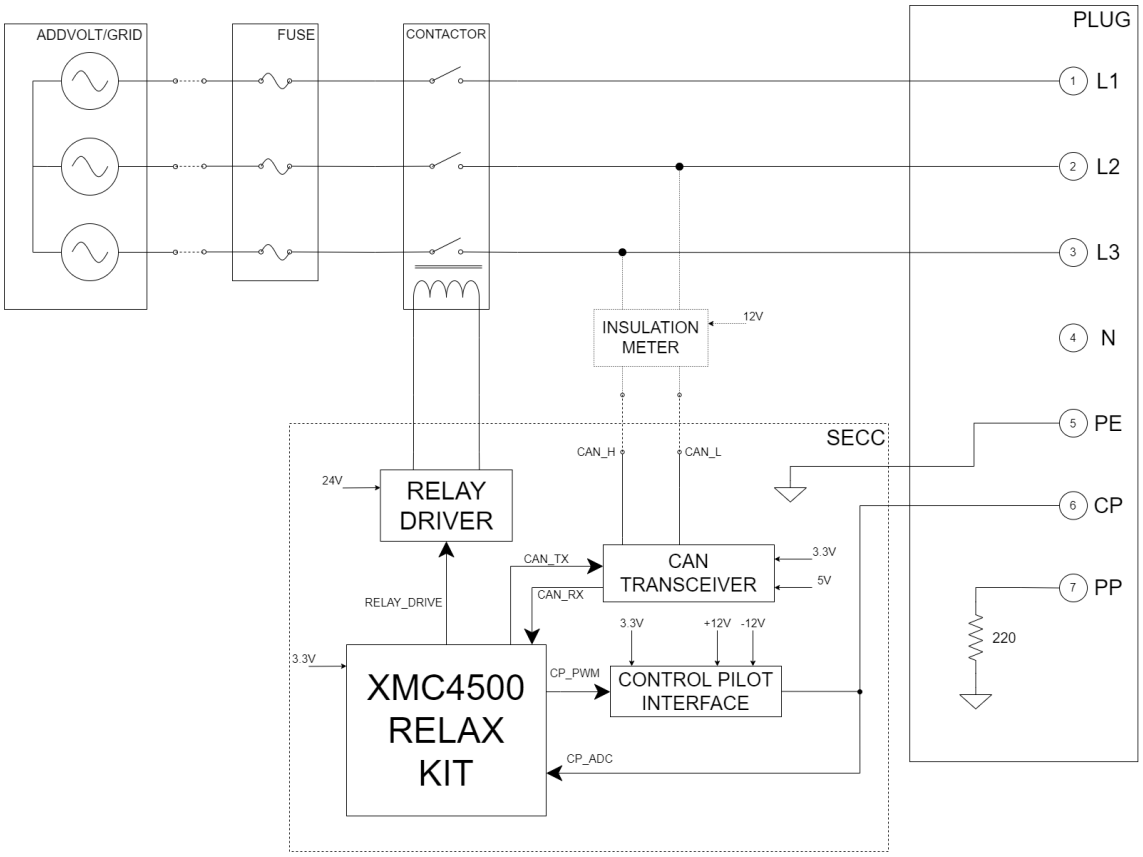


Figure 3.4: Circuit schematic representing the hardware of the SECC.

As it can be seen, the main components of the SECC, which are subsystems of it, are the Relay Driver, the Control Pilot Interface and the CAN Transceiver. Those subsystems are connected to the XMC 4500 Relax Kit development board and to external systems, being commanded by the microcontroller.

The schematic also presents the terminals in the IEC 62196 Type 2 (or *Mennekes*) plug, being its pinout as follows:

- L1: Line Phase 1 terminal;
- L2: Line Phase 2 terminal;
- L3: Line Phase 3 terminal;
- N: Line Neutral terminal (not used since the load is ballanced three-phase);
- PE: Protective Earth terminal;
- CP: Control Pilot terminal;
- PP: Proximity Pilot (or *Plug Present*) terminal, being basically a shunt resistance between PP and PE.

3.4.1.2 Power Supply

The first step to design a control circuit board is to think on the power supply of its components, how the power transfer is done and where the energy comes from. In this case, it started with the sizing of the power sources and converters that can successfully power all the components.

Initially, the datasheets of all the components that make up the SECC were consulted. The maximum value of current consumed by each one was sought, as well as the voltage levels at which these components work. In this way, it was possible to size and predict the entire power flow through different converters with different specifications, which provide the buses with the necessary voltages, as well as the currents that each component draw. The Power Supply block of the system is presented in Figure 3.5.

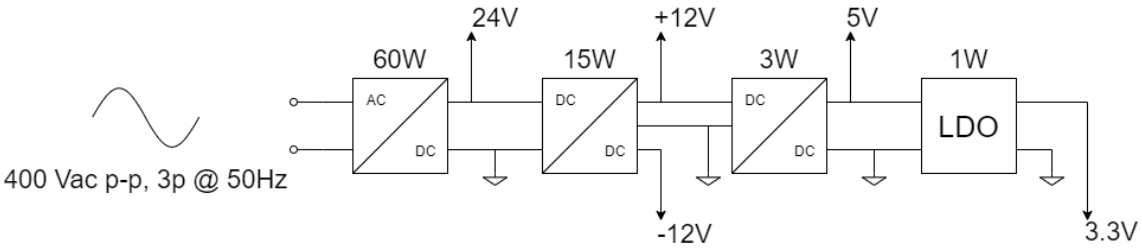


Figure 3.5: Power Supply block of the SECC.

To select the power source that supplies the SECC, the role of the SECC in the working system must be taken into account. Since the SECC is the grid-side charge controller, it is coherent that it is powered by the grid itself. Because of this, this controller is powered directly by the phase-to-phase terminals of the three-phase source in question. Given the characteristics of the system, the input is $400V_{AC}$, 50Hz. In this context, the first converter of the chain is an AC/DC, which generates a $24V_{DC}$ bus at the output, to supply the loads at $24V_{DC}$, which in this case are basically the contactor and the complementary relay. In sequence, this $24V_{DC}$ bus is the input to the next converter in the chain. To supply the contactor and relay set, together with the next 15W converter, a 20W supply would be enough. However, due to the availability of components, the AC/DC converter found and selected has a nominal power of 60W [34].

In sequence, $+12V_{DC}$ and $-12V_{DC}$ buses are required. The main load to be powered is the Insulation Meter which, according to its datasheet [35], has a maximum operating current of 300mA, which at $12V_{DC}$ results in a maximum consumed power of 3.6W which, when added with the 3W of the following converter totals 6.6W. In addition, a $-12V_{DC}$ bus is also required to supply the operational amplifier at $+12V_{DC}$ and $-12V_{DC}$, in order to generate the PWM signal at the Control Pilot Interface. Furthermore, because the 2-output converters have their power divided in the two buses, it was necessary to have a converter with at least twice the power that is needed on the $+12V_{DC}$ bus. Therefore, the most interesting option found was a two-output $+24V_{DC}$ to $+/-12V_{DC}$ 15W DC/DC converter [36].

Then, the next necessary bus is the 5V, which only supplies the CAN transceiver and the next

subsystem, which consumes 1W. According to the transceiver datasheet [16], it consumes a maximum of 75mA, which results in a maximum 375mW on the 5V bus. Therefore, a 1.5W converter would be enough. However, due to availability and also in order to foresee future additions of components, it was decided to use a DC/DC converter from $12V_{DC}$ to $5V_{DC}$ of 3W [37].

Finally, to have a stable voltage signal to feed the Microcontroller and the other loads at $3.3V_{DC}$, a Low-Dropout Regulator (LDO) was used, since its input was only $5V_{DC}$. This $3.3V_{DC}$ bus is used, in addition to powering the Microcontroller, also to supply the reference voltage to the CAN transceiver and also to power the voltage divider and the operational amplifier used as a buffer in the Control Pilot Interface. Therefore, as the consumption of these components are all low and in the order of mW, an LDO with a maximum consumption of 1 W is sufficient [38].

In Figure 3.6, the described power conversion chain is presented as it was drawn in CAD software, in order to transfer it to a PCB.

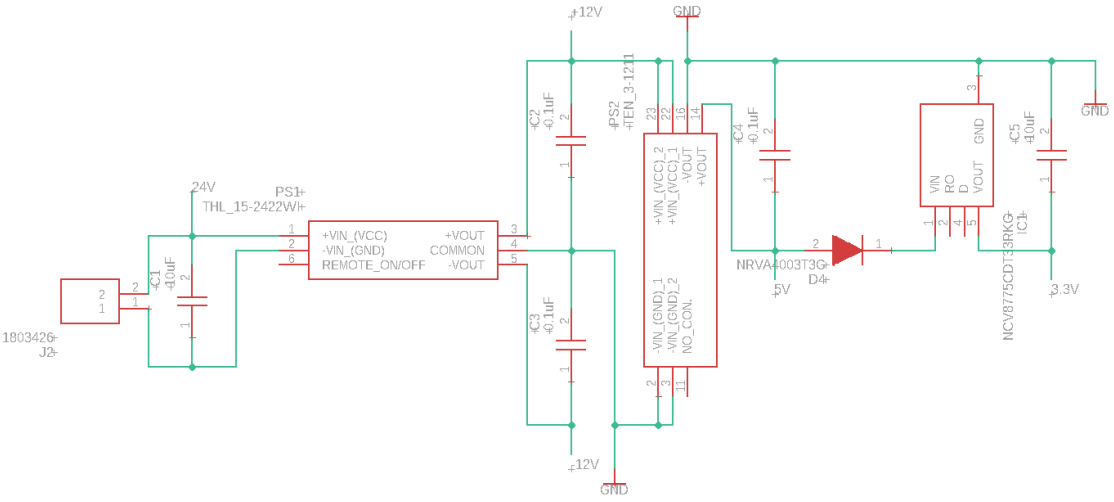


Figure 3.6: Power Supply block of the SECC designed in CAD software.

Some filter capacitors of 0.1μF and 10μF were added to the subsystem on the different DC buses in order to smooth the DC bus and because the datasheet of the LDO suggests it.

In addition, a diode at the LDO input was also added to ensure unidirectional power flow from the supply to the microcontroller and not the other way around.

3.4.1.3 Relay Driver

To ensure the load charging through a three-phase connection, the coupling is carried out by acting on a three-phase contactor, which in turn is activated using a complementary relay, which is activated by the microcontroller through a Metal Oxide Semiconductor Field Effect Transistor (MOSFET) drive. The idealized schematic of this subsystem is shown in Figure 3.7 [39][40].

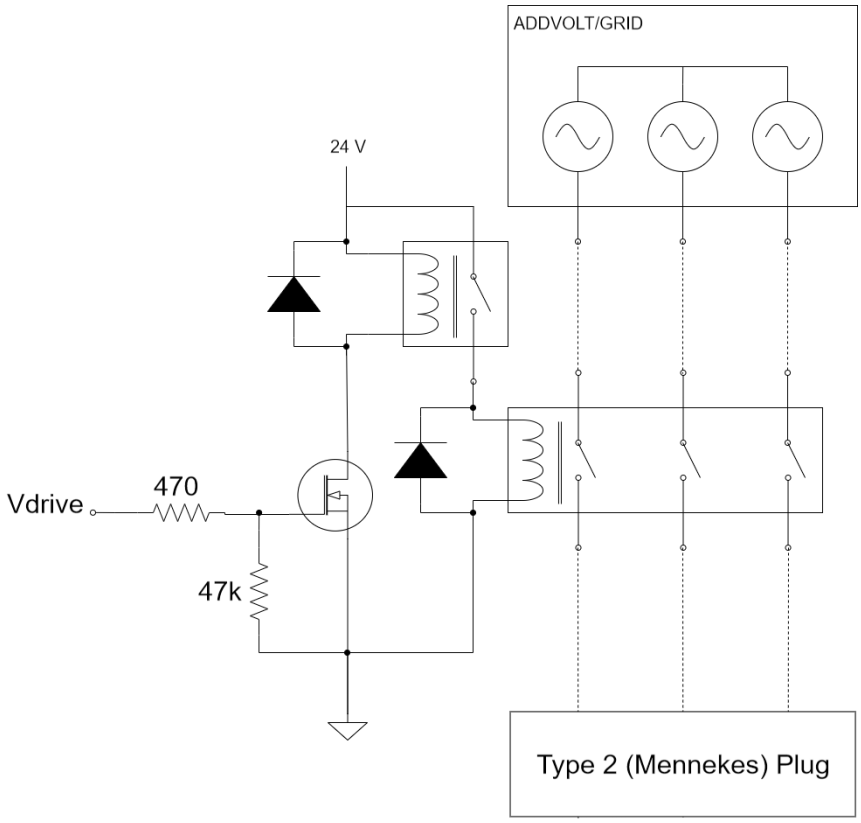


Figure 3.7: Circuit schematic draft of the subsystem to drive the contactor connected between the three-phase source and the plug to the load to be charged.

In order to drive the relay that drives the contactor, a MOSFET was used. To ensure fast switching intervals, a MOSFET Driver Integrated Circuit would be used to drive proper current at the gate input of the MOSFET, in order to charge the internal C_{ISS} capacitance of it and commutate the state of the MOSFET. However, as in this application the MOSFET is used only to close the contact of a relay for long periods, it is not necessary to size it for a high switching frequency, the current being supplied directly from the microcontroller. Besides that, to avoid floating voltage values at the gate of the MOSFET, a resistance was connected between the ground and the gate input of the MOSFET. To have the necessary voltage at the gate input of the MOSFET and without damaging the microcontroler or even the transistor itself, the two resistances named R_1 and R_2 in this draft were sized [41].

The first one, R_1, is responsible for limiting the current requirement from the microcontroller. So, having a Vdrive of 3.3 V and limiting Idrive to no more than 10mA, the following calculations can be carried out.

$$R_1 > \frac{3.3}{0.010} \implies R_1 > 330\Omega \implies R_1 = 470\Omega \tag{3.1}$$

As the resistance of R_1 is higher than the 330Ω of the maximum current, another calculation can be done.

$$\frac{I_{470}}{Idrive_{MAX}} = \frac{\frac{3.3}{470}}{0.01} = 70.2\% \tag{3.2}$$

Then, the current drawn from the microcontroller is no more than 70% of the maximum current established for it, being then around 7mA, a safe value for a digital output of the microcontroller [17].

Regarding R_2, it is responsible for limiting the voltage drop on R_1 and then keeping most of the voltage of the branch right on the MOSFET gate terminal. To do so, it was selected a resistance a hundred times higher than R_1. Then,

$$R_2 = 100R_1 = 100 * 470 = 47000 \implies R_2 = 47k\Omega \tag{3.3}$$

Thus, the voltage across R_2 is:

$$V_{R_2} = Vdrive\frac{R_2}{R_1 + R_2} = 3.3\frac{4700}{470 + 4700} = 3.267V \tag{3.4}$$

Therefore, the gate threshold voltage of the selected MOSFET must be much lower than 3.267V.

With both of the resistances R_1 and R_2 sized and selected, the expected power dissipation on both of them can be estimated.

Power dissipation in R_2:

$$P_{R_2} = \frac{V_{R_2}^2}{R_2} = \frac{3.267^2}{47000} = 0.227mW \tag{3.5}$$

Power dissipation in R_1:

$$P_{R_1} = \frac{(Vdrive - V_{R_2})^2}{R_1} = \frac{(3.3 - 3.267)^2}{47000} = 2.3\mu W \tag{3.6}$$

Then, a 100mW resistor is good enough for both of the resistors R_1 and R_2.

Additionally, a freewheeling diode was added in parallel with the relay and another one in parallel with the contactor in order to keep the MOSFET safe from high peak currents at the turn off of the relay [42].

In Figure 3.8, the described relay and contactor drive is presented as it was drawn in CAD software, in order to transfer it to a PCB.

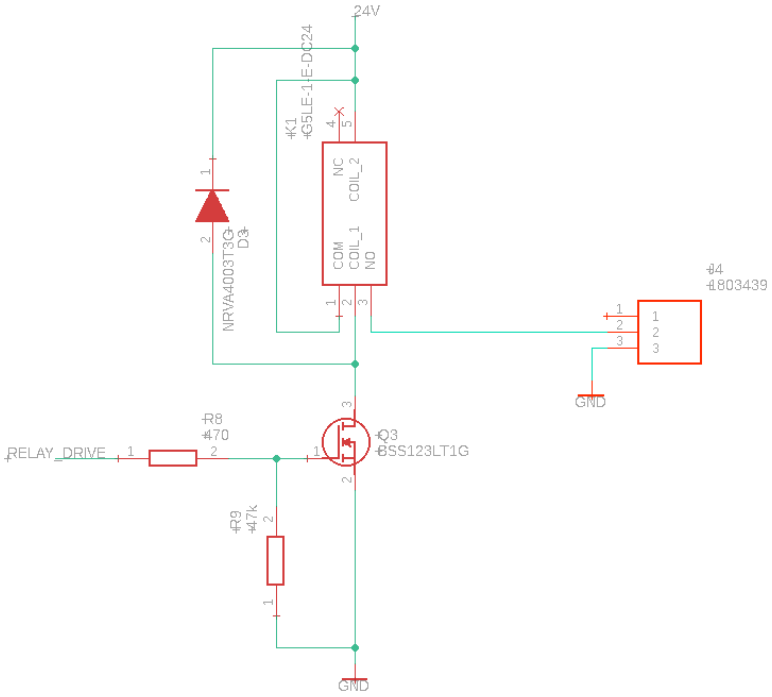


Figure 3.8: Relay Drive block of the SECC designed in CAD software.

As can be seen in the Figure 3.8, the three-phase contactor is an element external to the PCB, being then connected through an external connector that connects it to the relay *Normally Open* (NO) terminal and to the ground reference.

3.4.1.4 Control Pilot Interface

In order to provide a low level communication between EV and EVSE, the SECC must generate a 1kHz controlled Pulse Width Modulated (PWM) +12/-12V_{DC} signal in the Control Pilot (CP) terminal that is read by the EVCC at the other side of the system. The idealized circuit schematic of this subsystem is shown in Figure 3.9.

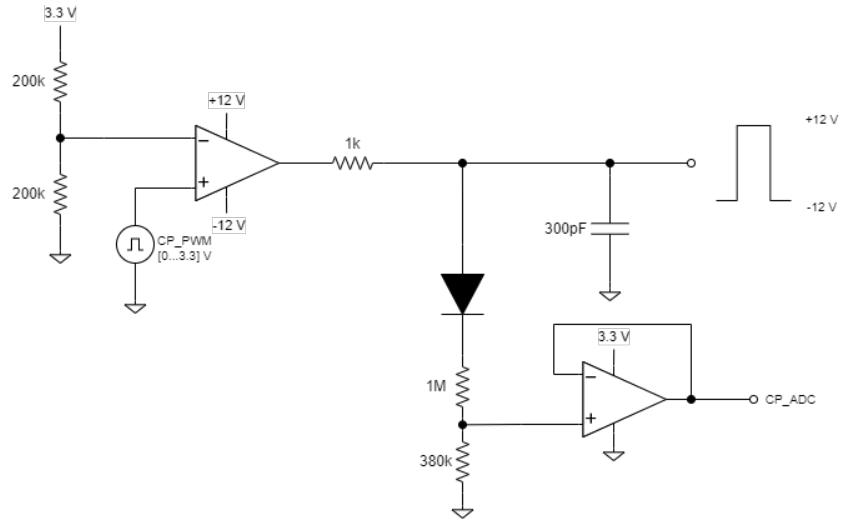


Figure 3.9: Circuit schematic draft of the subsystem of Control Pilot Interface.

To do so, the microcontroller can output a controlled PWM signal with 0V offset and a peak value of 3.3V. However, this signal must have the maximum and minimum values of $+12V_{DC}$ and $-12V_{DC}$, respectively. Thus, the solution found was to take the PWM output of the microcontroller as the positive input of an Operational Amplifier in Comparator configuration [43]. On the negative input of the comparator, a voltage divider is used in order to have the half of the maximum value of the PWM output from the microcontroller as a reference for comparisson. Then, every time the PWM from the microcontroller presents the values of 3.3V and 0V, the comparator outputs the values +12V and -12V, respectively. Additionally, a 1kΩ resistor and a 300pF capacitor were added in accordance to IEC 61851-1 and SAE J1772 [7][5].

Furthermore, in order to measure the voltage level in the Control Pilot terminal, another branch was added to it. It consists of a diode in series with a voltage divider that scales the 0V to 12V interval back to 0V to 3.3V. This voltage value passes through a buffer and then it is read by the Analog to Digital Converter (ADC) in the microcontroller [44].

For the selection of resistance values, on the negative input of the precision comparator, it were needed two high and equal resistances. Resistances of 200 kΩ seemed high enough.

For the voltage divider in series with the diode, the choice had to be made more carefully. Then, the following calculations were carried out:

$$V_{CP_{MAX}} \frac{R_2}{R_1+R_2} = V_{\mu C_{MAX}} \Rightarrow 12\frac{R_2}{R_1+R_2} = 3.3V \tag{3.7}$$

$$\Rightarrow R_2 = \frac{3.3}{12}(R_1+R_2) \Rightarrow R_2\left(1-\frac{3.3}{12}\right) = \frac{3.3}{12}R_1 \Rightarrow R_2 = \frac{\frac{3.3}{12}}{1-\frac{3.3}{12}}R_1 \tag{3.8}$$

So, if R_1 is established as 1MΩ,

$$R_1 = 1M\Omega \Rightarrow R_2 = \frac{\frac{3.3}{12}}{1-\frac{3.3}{12}}10^6 \Rightarrow R_2 = 379.31k\Omega \approx 380k\Omega \tag{3.9}$$

Due to availability of components, the selected value for R_2 was 383kΩ

Therefore, this subsystem can output a controlled PWM signal with minimum and maximum values of -12V and +12V, respectively. In addition, it can measure the voltage level of the PWM signal, in order to sense the connection with the EV.

In Figure 3.10, the described Control Pilot Interface is presented as it was drawn in CAD software, in order to transfer it to a PCB.

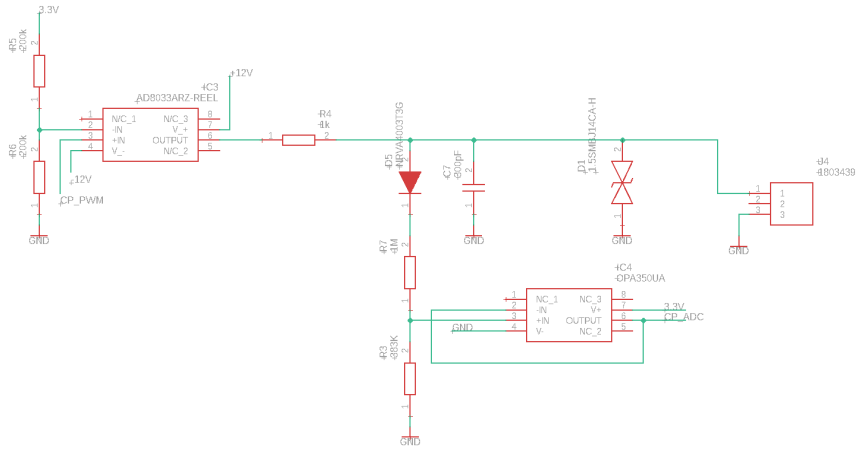


Figure 3.10: Control Pilot Interface block of the SECC designed in CAD software.

As it can be noted, the Control Pilot PWM signal is linked to an external connector on the PCB, which is wired to the Control Pilot terminal in the IEC 62196 Type 2 connector. In addition, this same signal is scaled back to the [0;3.3] V range, buffered and read by the microcontroller ADC. Furthermore, a TVS Supression Diode with a rated voltage of 14V was connected between the terminals CP and PE, in order to prevent any high voltages on the Control Pilot line [45].

3.4.1.5 Interface with Insulation Meter

The communication with the insulation meter device would done through a CAN bus. Then, because the microcontroller has the capability to communicate via CAN, a transceiver is needed. The goal is to modulate the information sent and received by the microcontroller into CAN messages that are changed in this bus. Then, due to availability, the integrated circuit NCV7342 was used. A typical application of it, taken from its datasheet, is shown in Figure 3.11[35][16].

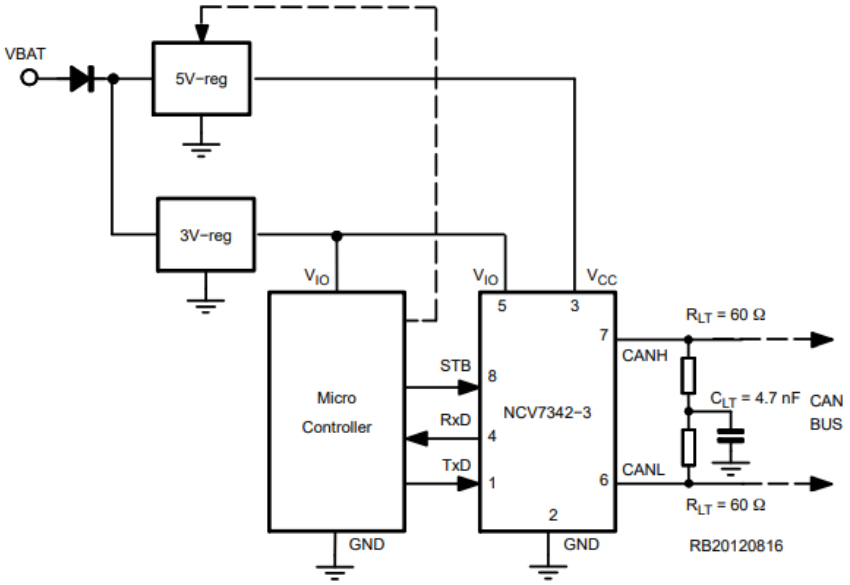


Figure 3.11: Typical application of the CAN transceiver NCV7342 [16].

The implemented solution was not much different from the typical application and it is shown in Figure 3.12.

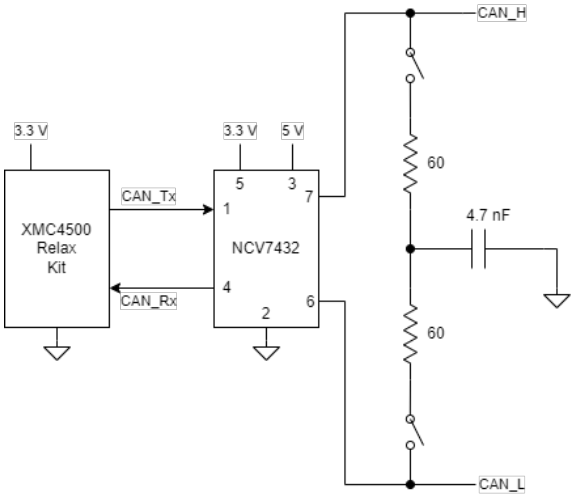


Figure 3.12: Circuit schematic draft of the connection between the microcontroller and the CAN transceiver NCV7342, then connected externally to the insulation meter.

The main difference between Figures 3.11 and 3.12 is the presence of the switches in series with the 60Ω termination resistances. The purpose of it is to disable the connection with the resistances in case the SECC is connected to a CAN node that already has those termination resistances on it. However, this feature was not used throughout this implementation. Its use is intended for future improvements and component additions.

This draft was then drawn in CAD software in order to transfer it to a PCB. The design is shown in Figure 3.13.

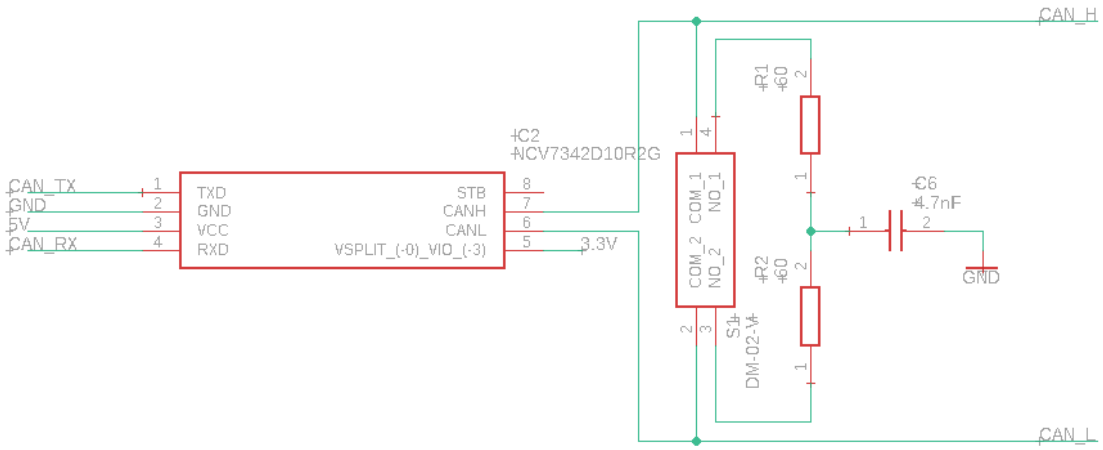


Figure 3.13: Connection between the microcontroller and the CAN transceiver NCV7342, designed in CAD software.

Then, the final circuit schematic after design in CAD software can be seen in Figure C.1 in the Appendix.

From the circuit schematic the connections between terminals were routed, generating the PCB footprint shown in Figure 3.14, together with the PCB itself after printed.

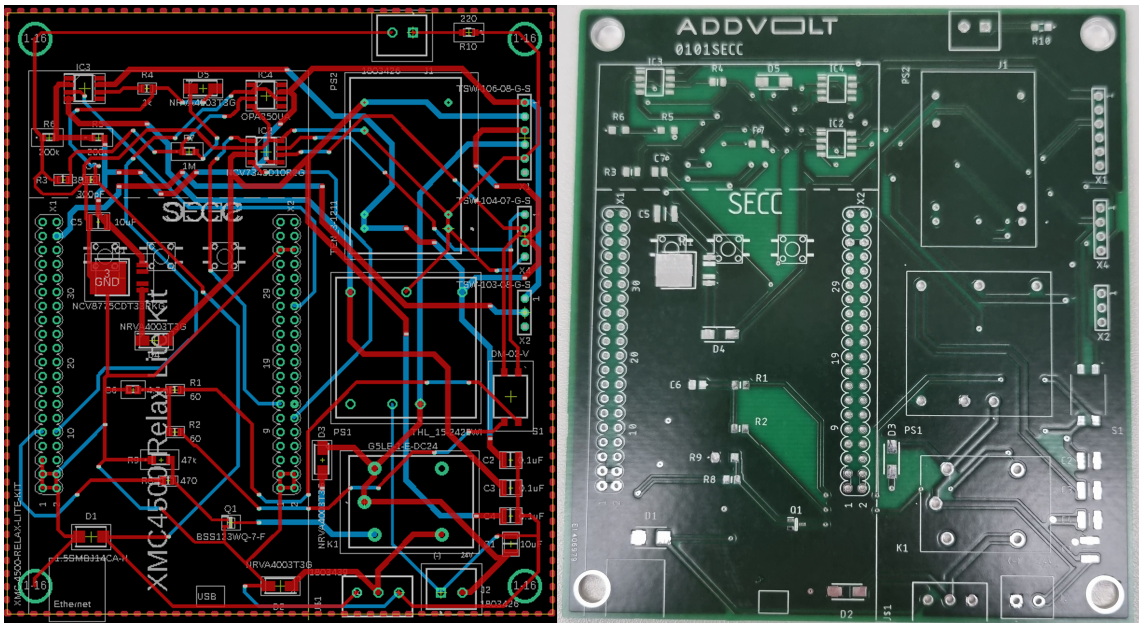


Figure 3.14: Final PCB view of SECC in CAD software (left) and PCB after printed (right).

After that, the components were soldered and the SECC was assembled, as it can be seen in Figure 3.15 with and without the XMC 4500 Relax Kit development board attached to it.

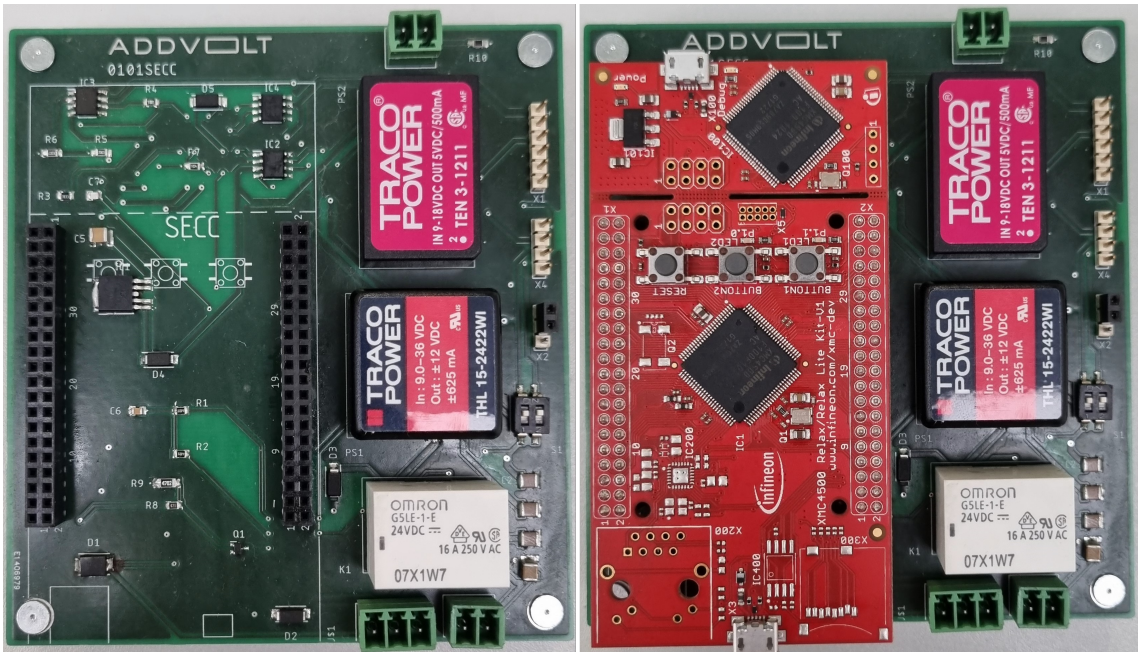


Figure 3.15: Final PCB of SECC after printed and with the components soldered. On the right, the SECC with the XMC 4500 Relax Kit attached.

3.4.2 Electric Vehicle Charge Controller (EVCC)

3.4.2.1 Overview

As for the SECC and following the recomendations in SAE J1772, another schematic of the needed circuit was drawn in order to illustrate what would be present on the designed PCB. The schematic for the circuit of the EVCC as a block diagram is shown in Figure 3.16.

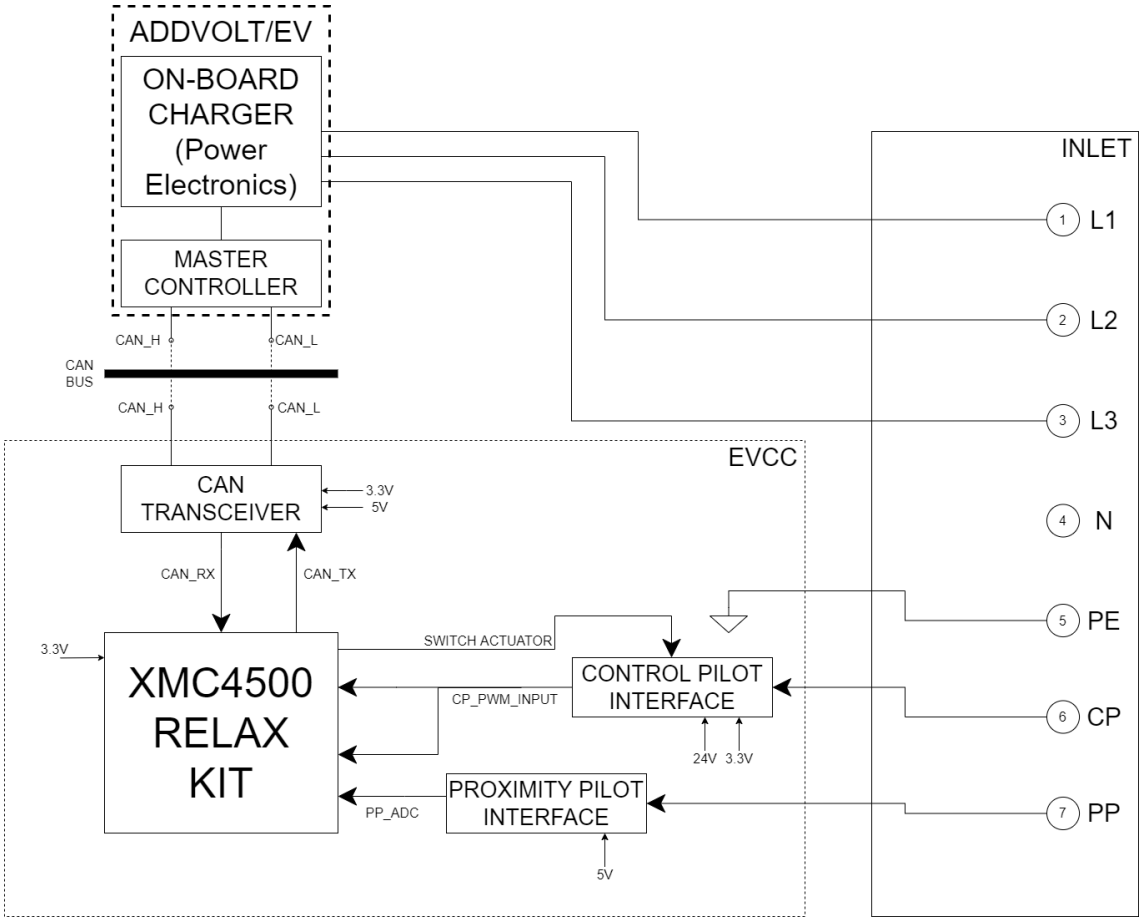


Figure 3.16: Circuit schematic representing the hardware of the EVCC.

As it is shown, the main subsystems that compose the EVCC are only the Proximity Pilot Interface, the Control Pilot Interface and the CAN Interface with AddVolt's unit. Those subsystems are connected to the XMC 4500 Relax Kit development board and to external systems, being commanded by the microcontroller.

Same as presented in Section 3.4.1.1, the schematic also considers the terminals in the IEC 62196 Type 2 (or *Mennekes*) plug.

3.4.2.2 Power Supply

As for the SECC, the EVCC also needs a Power Supply module in order to power up all the circuit board. For that, power sources and converters were also sized and selected.

Following the same steps as in Section 3.4.1.2, the maximum value of the current consumed by each component on the EVCC was searched in the datasheets of them, as well as their working voltage levels. Then, the power flow through the converters was sized and predicted, providing the necessary voltage levels, as well as being able to provide the required drawable currents for each component. The Power Supply subsystem block of the EVCC is presented in figure 3.17

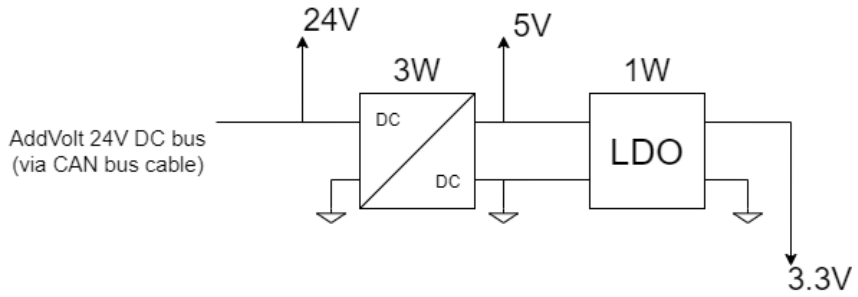


Figure 3.17: Power Supply block of the EVCC.

As in Section 3.4.1.2, the role of the EVCC in the working system must also be taken into account in order to select a proper power source for it. Therefore, as this system is much simpler in hardware than the SECC, its Power Supply block is also smaller and simpler. Since the EVCC is the vehicle-side controller, it is expected to be powered by a DC bus from the vehicle (the vehicle being AddVolt's unit in this implementation). Then, as the pinout of the CAN wiring of AddVolt (presented in Figure 3.18) provides a $24V_{DC}$ bus, it is used to power up the system.

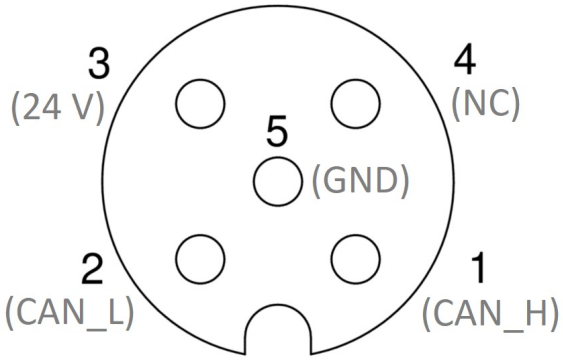


Figure 3.18: Adopted pinout for M12 connector in AddVolt's CAN bus.

In this context, as this system only needs 5V and 3.3V buses, only a small DC/DC converter and a Low-Dropout Regulator (LDO) were needed to provide the necessary DC buses.

The first conversion is from $24V_{DC}$ to $5V_{DC}$, which is only needed for the voltage divider in the Proximity Pilot Interface subsystem and to supply the CAN transceiver and the following LDO, which is rated at a maximum power of 1W. As mentioned in Section 3.4.1.2, according to the CAN Transceiver datasheet [16], it consumes a maximum current of 75mA, which results in a maximum of 375mW on the 5V bus. Then, a 1.5W converter would be enough. However, due to availability and also in order to forsee future additions of components/subsystems, the DC/DC converter was oversized to a power of 3W [46].

Therefore, as for the SECC, to have a stable voltage signal to supply the Microcontroller and the other loads at 3.3V, a Low-Dropout regulator (LDO) was used, since its input is only 5V. The provided 3.3V stable DC bus is used, in addition to powering the microcontroller, also to supply the reference voltage to the CAN transceiver, to power the buffer operational amplifier at the output of the Proximity Pilot Interface subsystem and to limit the voltage at the output of the MOSFET

driver in the Control Pilot Interface, which is presented further. Then, as the current requirements of these components are all low and in the order of mW or even less, an LDO with a maximum consumption of 1W is enough [38].

In Figure 3.19, the described power conversion sequence is presented as it was drawn in CAD software, in order to transfer it to a PCB.

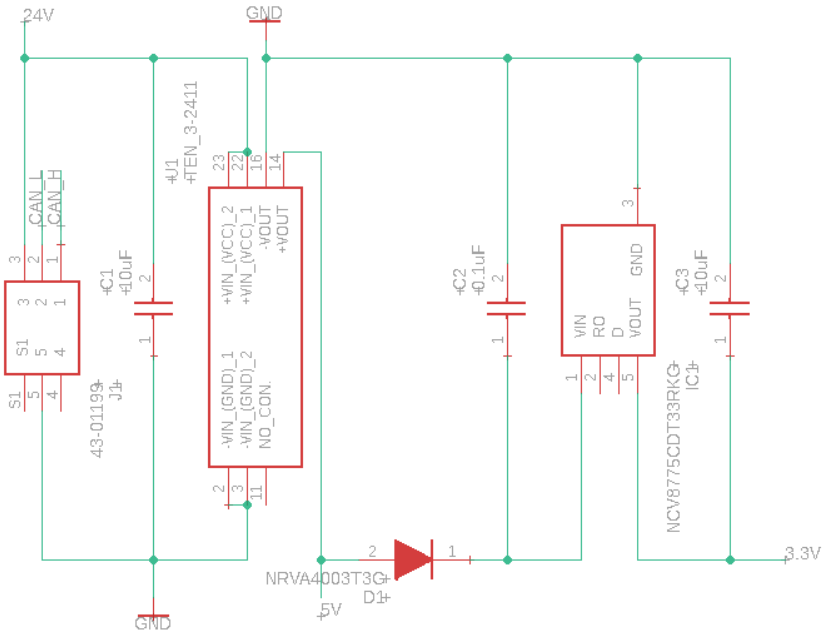


Figure 3.19: Power Supply block of the EVCC designed in CAD software.

As for the SECC, some filter capacitors of 0.1μF and 10μF were added to the subsystem on the different DC buses in order to smooth the DC bus and because the datasheet of the LDO suggests it. Besides that, the same diode was added at the LDO input to ensure unidirectional power flow from the supply to the Microcontroller and not the other way around [42].

3.4.2.3 Proximity Pilot Interface

The first stage in the charging sequence for the vehicle-side controller, according to SAE J1772 [5], is to check, through the Proximity Pilot (PP, or *Plug Present*) terminal for the presence of a plug connected to the vehicle. This is done by a simple voltage divider, which is sensed by the EVCC. The idealized circuit schematic of this subsystem is shown in Figure 3.20.

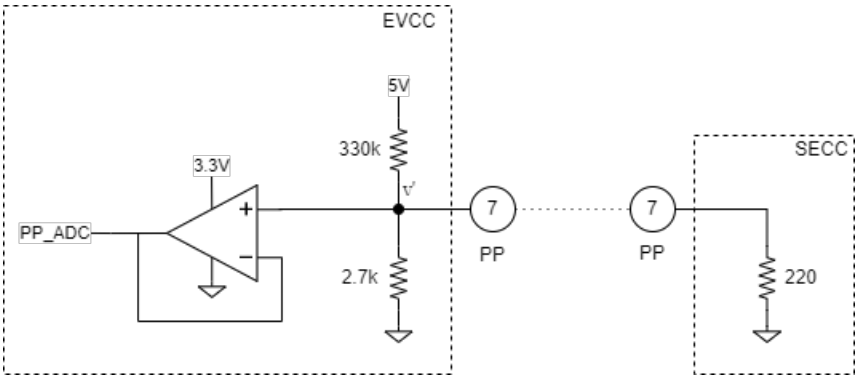


Figure 3.20: Circuit schematic draft of the subsystem of Proximity Pilot Interface in the EVCC.

As stated in SAE J1772 [5], the objective is to sense the presence of a plug (from an Electric Vehicle Supply Equipment) connected to the vehicle charging inlet. This is done through a voltage divider sensing. As shown in Figure 3.20, there is a voltage divider on the PP terminal of the plug, on the vehicle side of the connection. Then, when the vehicle is unplugged the voltage on v' is:

$$v' = 5V \frac{2700}{330 + 2700} \approx 4.46V \tag{3.10}$$

Thus, when the the plug is connected to the vehicle, the 220Ω resistance is in parallel with the $2.7k\Omega$ resistance, the new value of v' is:

$$v' = 5V \frac{2700 \parallel 220}{330 + 2700 \parallel 220} \approx 1.91V \tag{3.11}$$

This DC voltage signal is then buffered through an operational amplifier, which saturates voltages not much greater than 3.3 V due to its unipolar regulated $3.3V_{DC}$ supply and scales it to be suitable in order to be read by the Analog to Digital Converter (ADC) of the microcontroller, since it works with voltages not greater than 3.3V. Then, the 4.46V signal dictating an *unplugged state* is sensed by the microcontroller as 3.3V, being the 1.91V signal (meaning a *plugged/connected state*) kept the same as is at the v' node, since it is lower than 3.3V [44].

In Figure 3.21, the described Proximity Pilot Interface is presented as it was drawn in CAD software, in order to transfer it to a PCB.

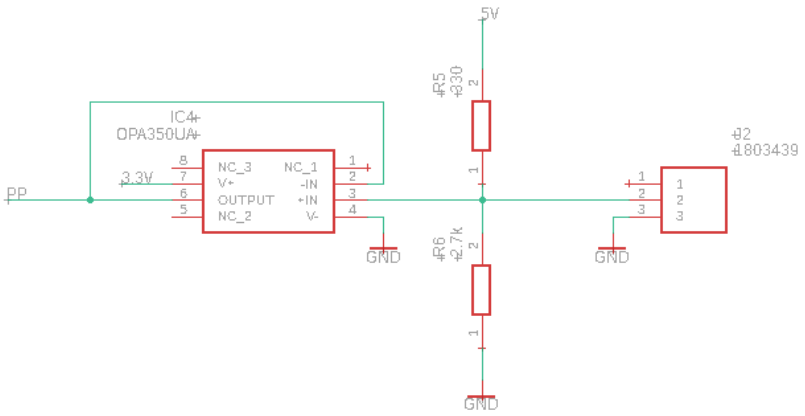


Figure 3.21: Proximity Pilot Interface block of the EVCC designed in CAD software.

The circuit looks exactly like the draft presented in Figure 3.20, with the exception of the resistance on the SECC side, which is part of the plug of the charger, being a resistance connected between the terminals PP and PE (Protective Earth).

3.4.2.4 Control Pilot Interface

Complementary to what was done in Section 3.4.1.4, on the vehicle side of the charging it must have a proper Control Pilot Interface in order to sense the generated PWM signal at the Control Pilot (CP) terminal and to read its duty cycle value which is translated into a value of maximum available output current, according to what is stated in SAE J1772 [5]. The idealized circuit schematic of this subsystem is shown in Figure 3.22.

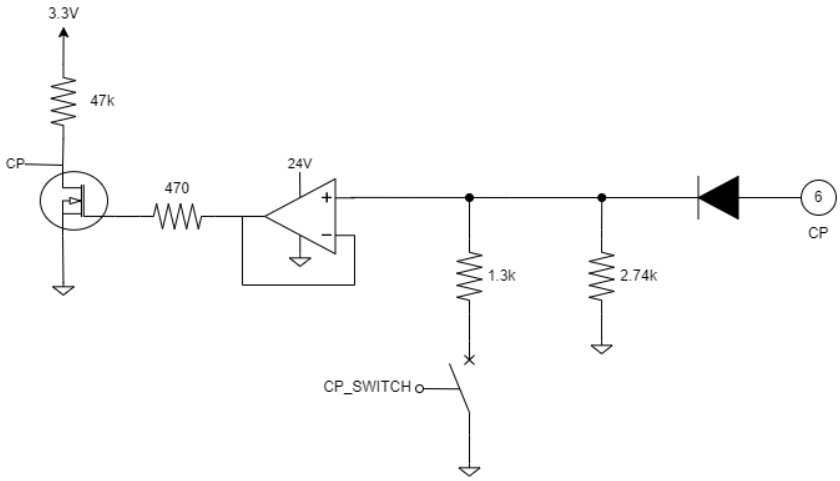


Figure 3.22: Circuit schematic draft of the subsystem of Control Pilot Interface in the EVCC.

Following the principle presented in Section 3.4.1.4 the +/-12V_{DC} PWM must be sensed by the microcontroller. For that, the SAE J1772 standard states first a diode connected right to the CP input of the EV charging inlet and then a controllable voltage divider using two resistances and a

switch. This voltage divider is then sensed by the SECC side and dictates the flow of states in the SECC. For the EVCC, it only matters the frequency and duty cycle of this signal.

For that, this subsystem has an operational amplifier which buffers the signal and, through the MOSFET driver presented in Figure 3.22, scales it to the range from 0V to 3.3V, with variable duty cycle. However, this driver reverses the duty cycle of the signal, i.e., if the duty cycle of the generated PWM Control Pilot signal by the SECC is 30%, the EVCC reads 70% on its inputs. Therefore, this issue had to be solved in software. The strategy used to measure the frequency and duty cycle of the PWM signal is presented in Section 3.5.4 [43][41].

This subsystem communicates in low-level with the SECC by acting on a switch, which changes the voltage divider resistance (and thus the CP voltage) and then sends the *ready to charge* information to the SECC via the CP terminal.

In Figure 3.23, the described Control Pilot Interface is presented as it was drawn in CAD software, in order to transfer it to a PCB.

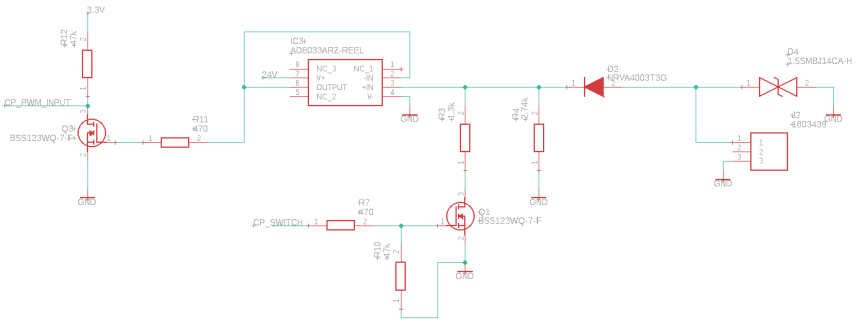


Figure 3.23: Control Pilot Interface block of the EVCC designed in CAD software.

As shown, the voltage divider switch was implemented using a MOSFET, using the same gate and gate-source resistances as the ones used in the Relay Driver in Section 3.4.1.3. Besides that, another MOSFET was used to limit the voltage to 3.3V, in order to be compatible with the microcontroller. Furthermore, a TVS Supression Diode with a rated voltage of 14V was connected between the terminals CP and PE, in order to prevent any high voltages on the Control Pilot line [41][45].

3.4.2.5 CAN Interface

The communication with AddVolt's unit is done through a CAN bus in accordance with AddVolt's CAN Protocol. Then, just like it is done in Section 3.4.1.5, a transceiver is needed between the CAN terminals of the Microcontroller and the AddVolt's CAN bus [16]. The goal is to modulate the information sent and received by the microcontroller into CAN messages that are changed in this bus. The working principle, the draft and the CAD drawing are exactly the same as presented in Section 3.4.1.5.

Then, the final circuit schematic after design in CAD software can be seen in Figure C.2 in the Appendix.

From the circuit schematic the connections between terminals were routed, generating the PCB footprint shown in Figure 3.24, together with the PCB itself after printed.

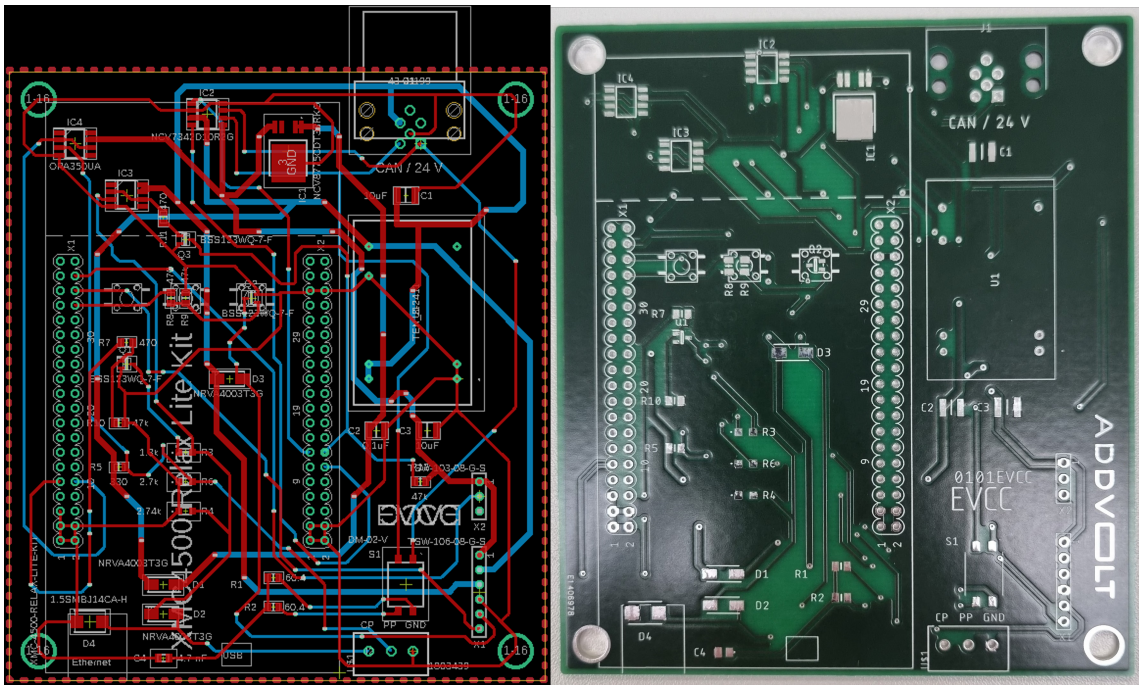


Figure 3.24: Final PCB view of EVCC in CAD software (left) and PCB after printed (right).

After that, the components were soldered and the EVCC was assembled, as it can be seen in Figure 3.15 with and without the XMC 4500 Relax Kit development board attached to it.

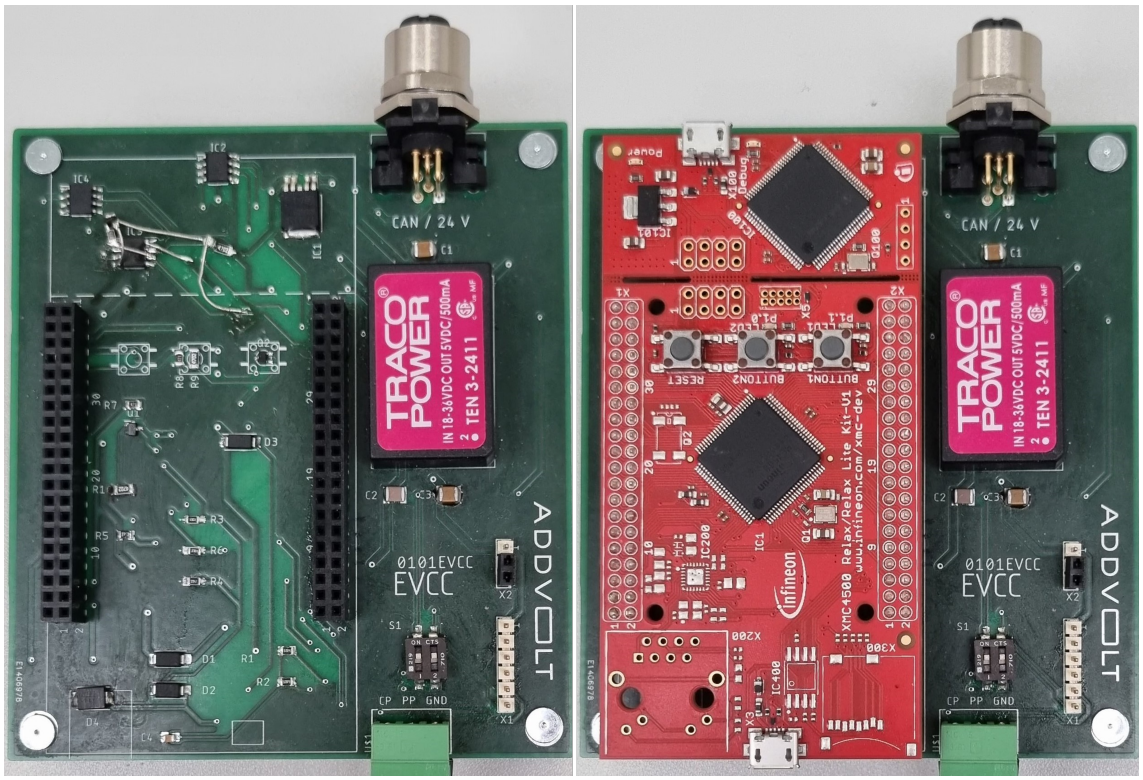


Figure 3.25: Final PCB of EVCC after printed and with the components soldered. On the right, the EVCC with the XMC 4500 Relax Kit attached.

It is to be noted that in the PCB version shown in Figure 3.25, the MOSFET driver presented in Figures 3.22 and 3.23 was not yet added to the circuit at the time it was printed. Then, this component, as well as the resistances wired to it, had to be externally soldered to the prototype.

3.5 Firmware

A great advantage of using Infineon™ XMC microcontrollers is that those devices have a dedicated Integrated Development Environment (IDE), which is DAVE™. The software has a Graphical User Interface (GUI) to configure applications that provide an Application Programming Interface (API) with the necessary variables and functions already implemented according to the desired configuration characteristics, which eases and accelerates the development process.

In the Firmware approach, the implemented solution consists in two Finite State Machines (FSMs), one for each controller that represents a side of the charging procedure. The transitions that command the flow of the State Machines follow the same conditions that are presented in the SAE J1772 standard [5].

3.5.1 DAVE™

Digital Application Virtual Engineer (DAVE™) is a software development and code generation tool in the C/C++ language for application in microcontrollers. DAVE™ is a standalone

system with automatic code generation modules and is suitable for developing software drivers for Infineon™ microcontrollers and provides the developer with automatically generated C templates with the functionality desired by the user [47].

This automatic code generation is done through DAVE™ Apps, which is an easily configurable GUI that setup many of the possible features of a system, such as Timers, Interrupts, Pulse Width Modulation (PWM) Signal Generation, Analog to Digital Converters (ADC), Serial Communication Interfaces, Controller Area Network (CAN) Interfaces and more, without having to write down lines of code to setup registers of the microcontroller.

3.5.2 Timers

The first needed functionality for both of the systems is an internal clock to work as a time base for it, being a reference for functionalities and for some FSM transitions. Then, in order to implement it, first a time structure called *Timer_struct* was declared. This structure is composed of many units of time, from milliseconds to days, as it is shown in Figure 3.26.

```
struct Timer_struct {
    uint16_t milliseconds; // milliseconds
    uint8_t seconds; // seconds
    uint8_t minutes; // minutes
    uint8_t hours; // hours
    uint16_t days; // days
};
struct Timer_struct Timer_struct;
```

Figure 3.26: *Timer_struct* structure to work as a clock base for the system.

Then, the next step is to iterate the value contained in each variable of the structure. For that, two DAVE™ Apps were needed, those are TIMER and INTERRUPT. The TIMER App can be configured to generate a periodic event with a previously set period. On the other hand, the INTERRUPT App is basically an Interrupt Service Routine (ISR) with a name set by the user and triggered by any generated event from the system.

Therefore, the TIMER App was configured to generate an event each 1 ms that was passed. Hence, this event was used as a trigger input to the INTERRUPT App that was implemented and its ISR **void TimerInterrupt()**, presented in Figure 3.27, calls the implemented **void update_timer()** function, which is shown in Figure 3.28.

```
void TimerInterrupt()
{
    update_timer();
}
```

Figure 3.27: *TimerInterrupt()* interrupt routine that calls *update_timer()* routine each 1ms.

```
void update_timer()
{
    Timer_struct.milliseconds++;
    if(Timer_struct.milliseconds >= 1000)
    {
        Timer_struct.milliseconds = 0;
        Timer_struct.seconds++;
    }
    if(Timer_struct.seconds >= 60)
    {
        Timer_struct.seconds = 0;
        Timer_struct.minutes++;
    }
    if(Timer_struct.minutes >= 60)
    {
        Timer_struct.minutes = 0;
        Timer_struct.hours++;
    }
    if(Timer_struct.hours >= 24)
    {
        Timer_struct.hours = 0;
        Timer_struct.days++;
    }
}
```

Figure 3.28: *update_timer()* routine to increment the variables of *Timer_struct* structure periodically each 1ms.

There is also another routine which is used to clear the variables inside the *Timer_struct* structure by setting all to zero. This routine is called *void clear_timer()* and its declaration is shown in Figure 3.29.

```
void clear_timer()
{
    Timer_struct.milliseconds = 0;
    Timer_struct.seconds = 0;
    Timer_struct.minutes = 0;
    Timer_struct.hours = 0;
    Timer_struct.days = 0;
}
```

Figure 3.29: *clear_timer()* routine to reset the variables of *Timer_struct* by setting all to 0.

Thus, in each millisecond the timer structure is incremented, working as an internal clock of the system and providing a time base to the FSM and its implemented functionalities. Besides that, the timer structure is reset each time a transition happens in the flow of the FSM, in order to reset time counting to be considered in the next transition.

3.5.3 Supply Equipment Charge Controller (SECC)

The FSM diagram, as well as the actions executed by the SECC according to its states, is presented in Figures 3.30 and 3.31.

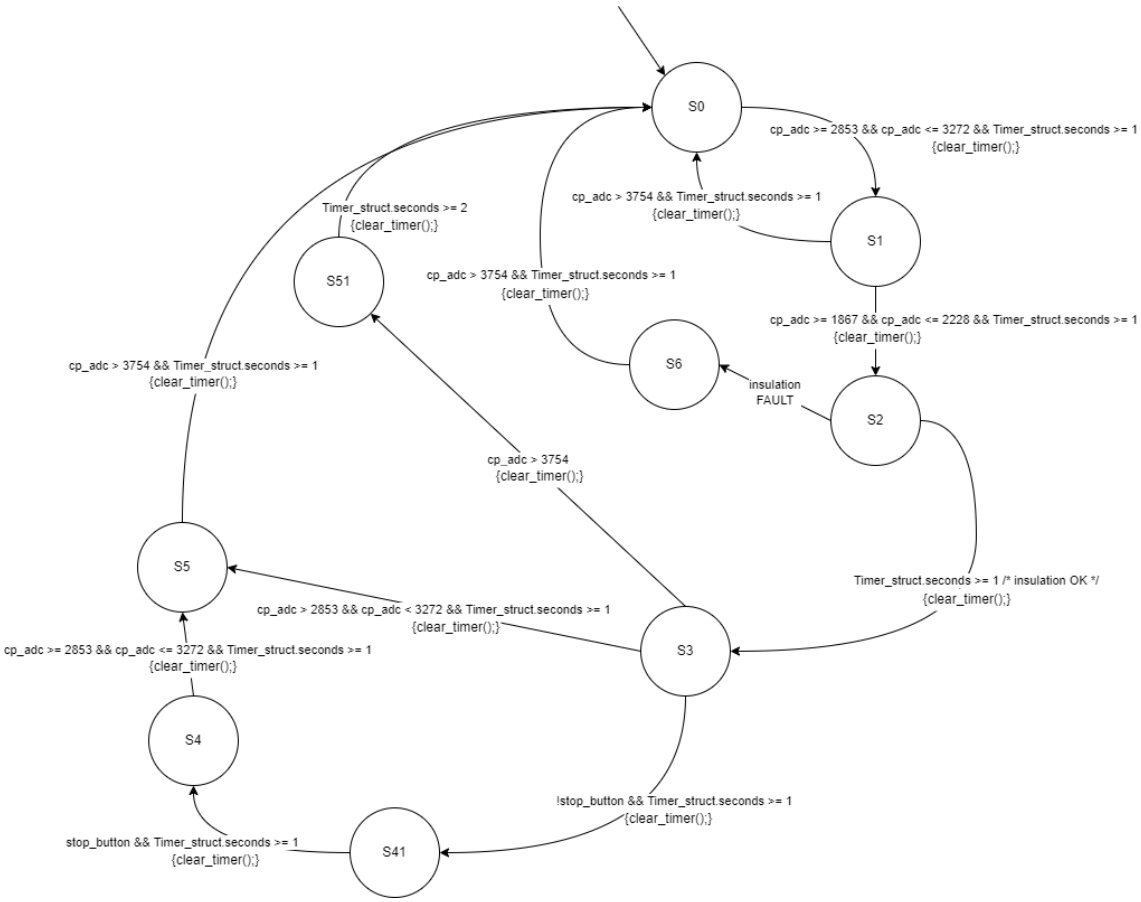


Figure 3.30: Finite State Machine for SECC Firmware.

S0, S1, S2, S4, S5	S3
DIGITAL_IO_SetOutputLow(&RELAY_DRIVER); CP_PWM_duty_cycle = 10000;	DIGITAL_IO_SetOutputHigh(&RELAY_DRIVER); CP_PWM_duty_cycle = duty_cycle_max_current;
S41, S51	**S6**
DIGITAL_IO_SetOutputHigh(&RELAY_DRIVER); CP_PWM_duty_cycle = 10000;	DIGITAL_IO_SetOutputLow(&RELAY_DRIVER); CP_PWM_duty_cycle = 0700;

Figure 3.31: Actions executed according to the states in the Finite State Machine of SECC Firmware.

When the SECC is powered on, it remains on state S0 while waiting for a change in the Control Pilot (PP) voltage, which happens at the moment the Type 2 plug is connected to the charging socket inlet. At this state the SECC outputs a +/-12V 1kHz PWM signal with a duty cycle of 100%, which consists in a $12V_{DC}$ signal. When the plug is connected, the SECC detects a voltage drop

from approximately +12V to approximately +9V, waits 1 second and then moves on to state S1. At the state S1, the SECC is waiting for the EVCC to send a *ready to charge* information, which is detected by a drop on the CP voltage from approximately +9V to approximately +6V, since no ventilation requirements were implemented. When the voltage drop is detected, the SECC moves on to state S2, which is the state for checking the insulation resistance. As the insulation meter was not added to the project, this state is just a *hold state* in which the SECC waits 1 second and then moves on to state S3. The state S3 is the *charging state*, when the SECC outputs the signal to close the relay and the contactor and sends a PWM signal with variable duty cycle at the CP line. The duty cycle of this signal is modeled according to the maximum available current from the EVSE three-phase source, as it is stated in SAE J1772 charging protocol, which is shown in Equation 3.12.

$$\text{duty cycle } (\%) = \begin{cases} 10\,\frac{\text{max_current}}{6} & \text{if max_current} \leq 51 \\ \frac{4\,\text{max_current}+640}{10} & \text{if max_current} > 51 \end{cases} \tag{3.12}$$

For that, the routine

$$\text{uint32_t max_current_to_duty_cycle(uint32_t max_current)}$$

was implemented. Its declaration is presented in Figure 3.32.

```
uint32_t max_current_to_duty_cycle(uint32_t max_current)
{
    if(max_current <= 51)
        return 10*(100*max_current/6);
    else
        return 10*(4*max_current + 640);
}
```

Figure 3.32: *max_current_to_duty_cycle()* routine that converts a current value in Ampère into a duty cycle value in percentage, according to SAE J1772.

As shown, the routine models the current value into its correspondent duty cycle percentage value, according to SAE J1772 and with the duty cycle percentage value in the right format to be used for configuration in the PWM Generation App in DAVE™ IDE. However, the current value is not always constant. Instead, it is configurable by the user through an auxiliar state machine, which is the Toggle Current State Machine, shown in Figure 3.33.

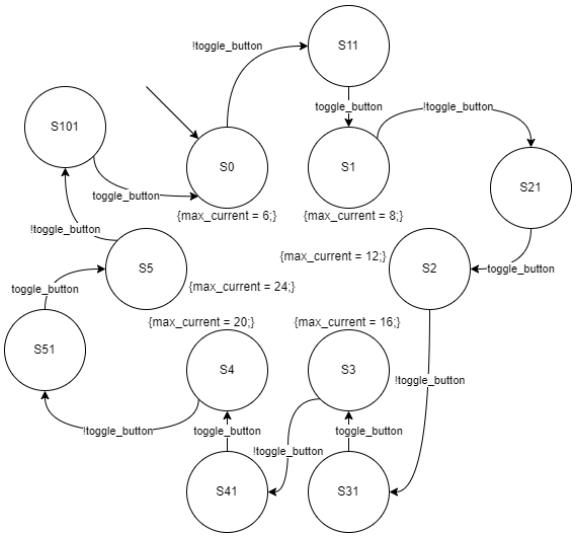


Figure 3.33: Toggle Current auxiliary Finite State Machine for SECC Firmware to set the maximum available output current.

As presented, the state machine consists in a small toggle function that changes the value of the maximum available current, which is the input to the routine shown in Figure 3.32. It is to be noted that this routine is invoked in the beginning of every cycle of the program, in order to have the actual value for the maximum available current. The button used for this purpose is embedded into the XMC 4500 Relax Kit development board.

By the moment the SECC is charging a load, it can be interrupted either by the EVCC or by the SECC itself. The EVCC interrupts the charging when it receives a message containing the information of a 95% or greater State of Charge (SoC). For the charging to be interrupted by the SECC, it can end by pressing a stop button or by pulling out the Type 2 connector. For the first one, it is just needed to press another button which is also embedded into the XMC 4500 Relax Kit development board and it moves on the SECC state to S41 at the falling edge and next to state S41 at the rising edge, noting that the input logic of the button is with a Pull-up Resistor. When the charging connector is pulled out, it is sensed by the SECC a change in the CP voltage from approximately +6V to approximately +12V, which moves the state from S3 to S51, waits 2 seconds for the load to stops drawing current and then moves back to state S0, ready to start a new charging session. The charging interrupted by the EVCC is detected by the voltage change on CP line from +6V to +9V, which moves SECC's state from S3 to S5. In state S5, the SECC is just waiting for the Type 2 connector to be pulled out and then changing the CP voltage from approximately +9V to approximately +12V.

3.5.3.1 Control Pilot PWM Signal Generation

The most rich information that flows from the SECC to the EVCC is the +/-12V PWM signal through the Control Pilot terminal. For that, the SECC must generate a 1kHz PWM signal with controlled duty cycle. As the XMC 4500 Relax Kit reference voltage is 3.3V, the output of the

microcontroller is a PWM signal with a minimum voltage of 0V and a maximum voltage of 3.3V. This signal is an input to the Operational Amplifier in a Comparator configuration that is presented in Section 3.4.1.4. The output of this comparator is the desired PWM signal from -12V to +12V.

For this feature, its implementation just needed the addition of a PWM App instance in the project. Then, this instance was configured according to Figure 3.34. It is to be noted that, with the exception of *Frequency [Hz]* and the unchecked boxes on *Start after initialization* and *Enable single shot mode*, the values set in **PWM Settings** doesn't matter since the duty cycle of this signal is set by code in the FSM and this signal runs continuously.

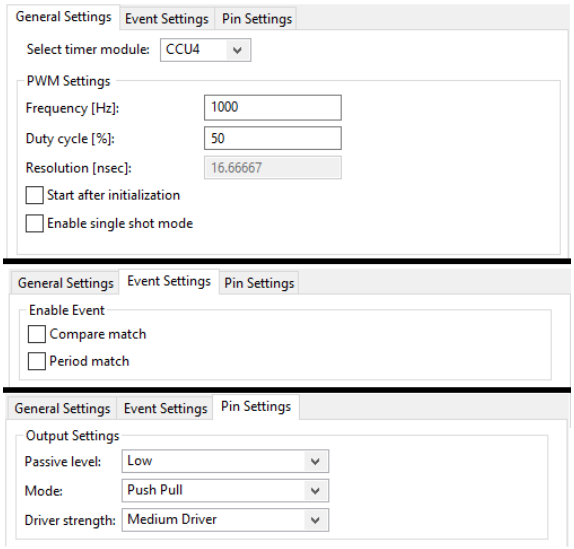


Figure 3.34: *PWM Generation* App configuration windows in DAVE™.

Its setup and initialization through code was done with the calling of the following functions as is presented in Figure 3.35.

```
CP_PWM_duty_cycle = 10000;

PWM_SetDutyCycle(&CP_PWM, CP_PWM_duty_cycle);

if (!PWM_GetTimerStatus(&CP_PWM))
{
    PWM_Start(&CP_PWM);
}
```

Figure 3.35: Duty cycle setup and PWM Generation start command.

Those functions are already implemented in the PWM API that was configured in the PWM App.

3.5.3.2 Control Pilot Voltage Sensing

In the Control Pilot line there is a two-way communication established. Then, besides the generation of a PWM signal by the SECC, there is also a voltage sensing on that line, since

there is a voltage divider when the SECC and EVCC are wired together by CP when plugged. The objective is to sense the voltage level on the line, which is imposed by the voltage divider. However, since on that line there is a PWM signal, the measurement should be precised in timing in order to always sample only the maximum voltage window instead of the 0V window, when the duty cycle of the signal is not 100%.

Therefore, the found solution consists in the generation of an auxiliary internal PWM signal, synchronized in time with the Control Pilot PWM presented in Section 3.5.3.1 and with the same frequency, but always with a half of its duty cycle. Then, through the App of this auxiliary signal it was set an event to be generated in the *Compare Match*.

In most of the microcontrollers, timers and signal generation are implemented in hardware through counters. In the XMC 4500 it is no different, there are 3 registers which are responsible for this. These are shown in Figure 3.36.

Figure 3.36: Principal compare blocks for PWM generation in XMC 4500 [17].

By standard, the counting is configured in *up-count* mode. The up-count mode of the compare rule is simple: as long as the value of the timer register (TR) is equal to or greater than the value of the compare register (CR), the status bit is set to one. Otherwise, it is set to zero. The PWM duty cycle dynamic range can be set to any value from 0% to 100%. The illustration of this working principle is shown in Figure 3.37, as it is presented in XMC 4500 datasheet [17].

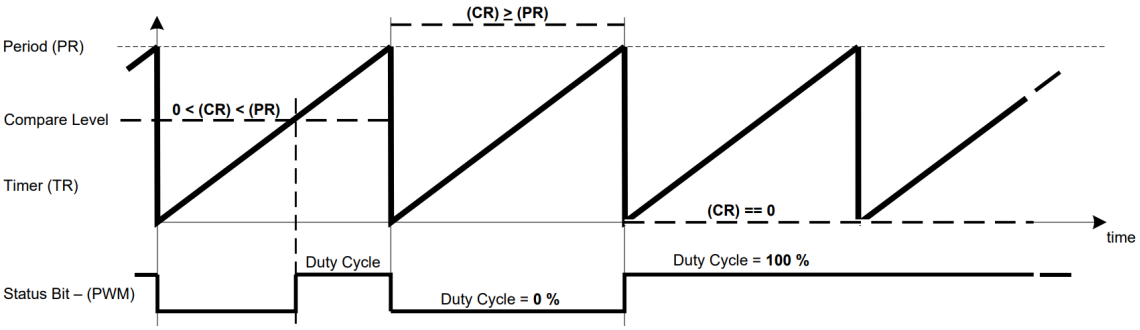


Figure 3.37: PWM generation in *up-count* mode [17].

In addition to generating the PWM signal itself, the PWM App is also capable of configuring events that can trigger actions or interruptions external to the App.

It was with purpose that a new PWM App called **ADC_PWM_aux** was implemented, using the same clock base and with the same frequency (1kHz) as the Control Pilot PWM presented

previously, meaning that both are synchronized in time. However, in this other App, in a configuration window similar to the one shown in Figure 3.34, the Compare Match check box field was checked and therefore a Compare Match Event was enabled in this App instance, keeping all other fields just like in Figure 3.34. It is to be noted that this PWM signal is internal and then it is not available as an output of the microcontroller. Its purpose is just to generate the Compare Match Event. An illustration of this PWM signal generation, together with the signal presented in Section 3.5.3.1, is shown in Figure 3.38.

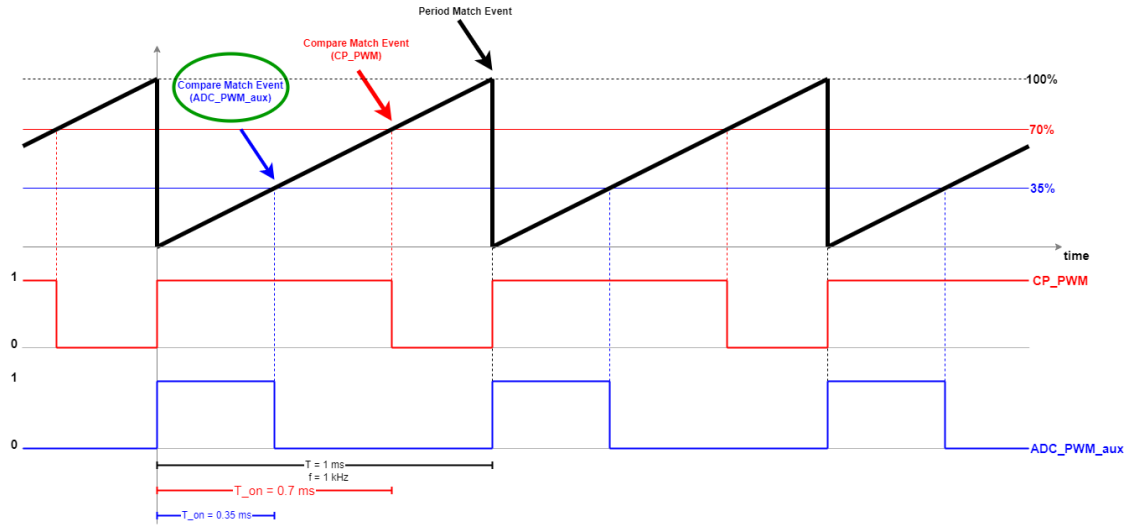


Figure 3.38: Synchronized PWM generation to generate Compare Match Event.

Accordingly, an Analog to Digital Converter (ADC) App was configured. However, differently from the PWM signal, the conversion does not run continuously by itself. The ADC is triggered by the Compare Match Event previously mentioned. Therefore, this approach can guarantee that the voltage sensing is always done at the "high level" of the PWM signal, since it is done always on the half of the "t_on window". The timing of those measurements goes as is presented in Figure 3.39.

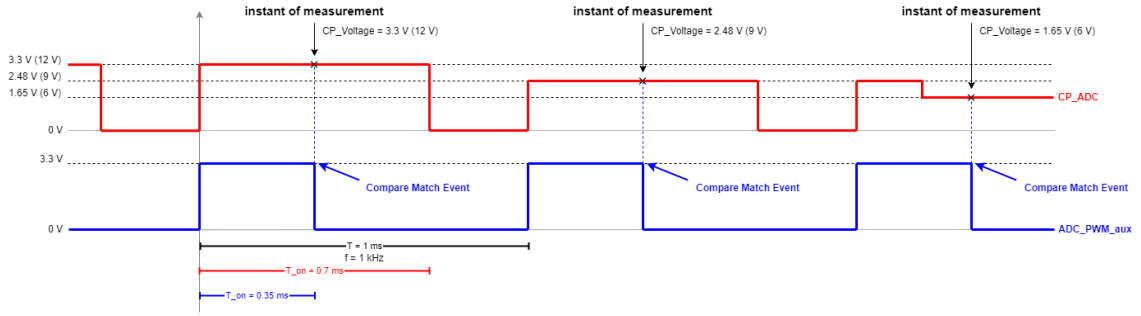


Figure 3.39: Synchronized ADC triggering for measurements of the high level voltage of the PWM signal on Control Pilot line.

It is to be noted that the measured voltage at the Control Pilot line is scaled in the range

from 0V to 3.3V, in order to be compatible with the microcontroller's GPIOs. This is done by the rectification of the +/-12V, 1kHz PWM signal through a diode, and having this signal in the voltage divider that scales the voltage range from [0; 12] V back to [0; 3.3] V. This signal is then buffered by an Operational Amplifier and its output is the input to the microcontroller, as is presented in Section 3.4.1.4.

3.5.3.3 Relay Driver

The main three-phase supply line that wires both the entities (EV and EVSE) of the charging case is connecting them through a three-phase contactor, following the diagram presented in Figure 3.7. According to this diagram, this contactor is commanded by a relay, which is driven by a MOSFET, at the *high level* of the Vdrive signal, coming from a GPIO of the microcontroller.

In software, this signal is a Digital Output and have its values depending on the state of the FSM, being at *high level* only on the charging state, when the power contacts are expected to be closed, being open otherwise.

3.5.3.4 Stop Button

As the vehicle is charging, it can end by three ways, by fully charging the vehicle, by unplugging the cable or by an interruption. For the first one to be achieved, it is only necessary for the EVCC to receive information that the battery state of charge is 100%, as is shown further. For the second one, this can be safelly achieved since in the male IEC 62196 Type 2 inlet of the vehicle the CP pin has a considerably smaller length than all other pins. Therefore, the SECC can detect a partial unplug with the vehicle still charging and then it ends the charging procedure.

In alternative, for any reason the user can interrupt the charging by pressing a **stop** button. This button is a standard GPIO and for that it was used one of the embedded buttons on the XMC 4500 Relax Kit board. In every iteration, the digital value of the pin connected to the button is read and stored in the variable **uint8_t stop_button**. Then, when the system is at the charging state, if the button is pressed, the FSM goes to the *charging finished* state and then start to present 12 V_{DC} at the Control Pilot line, indicating to the EVCC that the charging is finished. The input is implemented with a Pull-up Resistor, meaning that when the button is pressed, the input is in low state, being high otherwise.

3.5.3.5 Toggle Current Button

In order to command the transitions in the Toggle Current State Machine and then to change the value of the maximum available output current, a *toggle* button was implemented. Just like the *stop* button, this button is a standard GPIO and also is implemented using one of the embedded buttons on the XMC 4500 Relax Kit board. Then, the rising and falling edges of the button are monitored in order to change the value of the maximum available output current on each rising edge of the input, according to the Toggle Current State Machine in Figure 3.33. The used buttons XMC 4500 Relax Kit board are highlited in Figure 3.40.

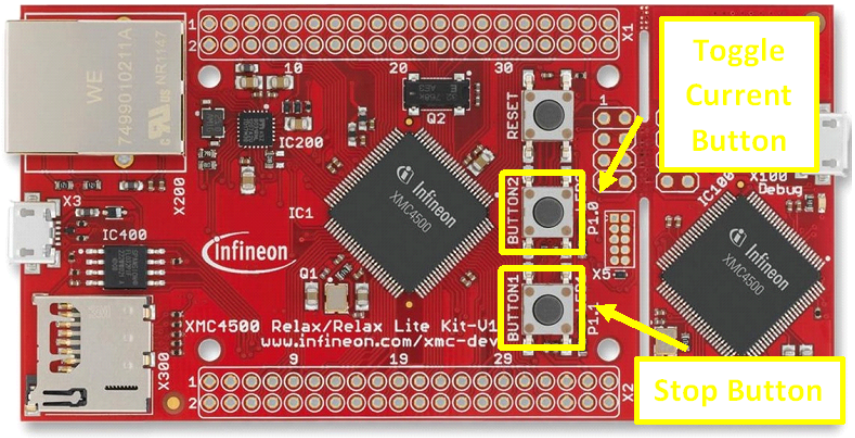


Figure 3.40: Indication of the used buttons in the XMC 4500 Relax kit board for the functionalities of Stop and Toggle Current (adapted from [15]).

3.5.3.6 App Dependency

Therefore, with all the functionalities implemented, the final aspect of the App Dependency Tree in DAVE™ IDE for the SECC is shown in Figure 3.41.

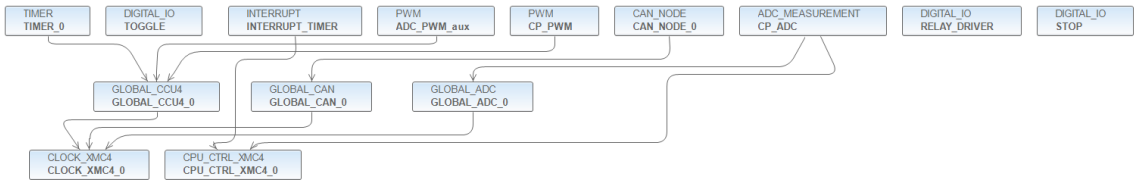


Figure 3.41: App Dependency Tree in DAVE™ IDE for the SECC.

3.5.4 Electric Vehicle Charge Controller (EVCC)

The FSM diagram, as well as the actions executed by the EVCC according to its states, is presented in Figures 3.42 and 3.43.

When powered on, the EVCC remains in state S0 while waiting for a change in the Proximity Pilot (PP) voltage, which happens when the Type 2 plug is connected to the charging socket inlet. At this state, the EVCC keeps the Control Pilot (CP) switch open, and sets off the flags that indicate to send available current information or to calculate PP resistance. When the plug is connected, the EVCC waits 1 second at the hold state S11 and then moves on to the state S1. At the state S1, the EVCC sets the flag to calculate the proximity pilot resistance of the plug that has just been connected. Still, the EVCC waits for the SECC to send a 1kHz PWM with variable duty cycle between 9% and 96% at CP line and when it happens the EVCC moves on to state S2. The state S2 is the charging state, it is when the EVCC turns on the switch at the CP voltage divider and send the messages to turn on AddVolt's unit and to send the messages with the value of maximum available output charging current, after receiving it through the duty cycle of the CP PWM signal which

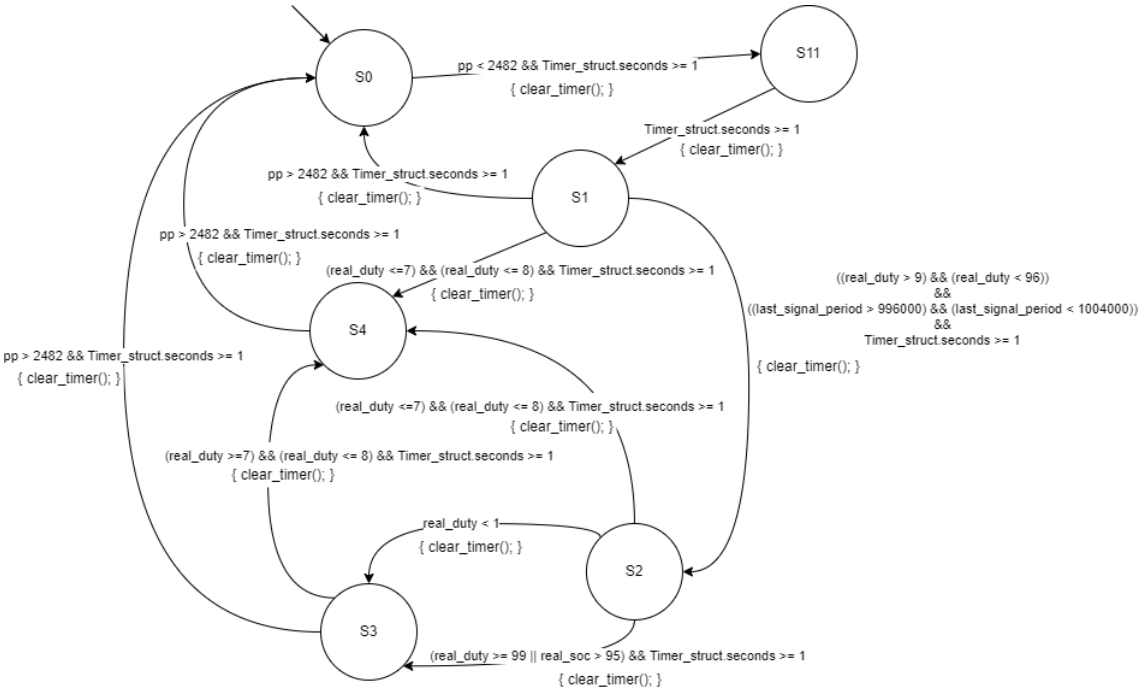


Figure 3.42: Finite State Machine diagram for EVCC Firmware.

was generated by the SECC. The conversion of values from duty cycle percentage to Ampères (AC RMS) is done in code following Equation 3.13.

$$\textit{max current available}\ (AC) = \begin{cases} 0 & \text{if duty cycle} < 10\% \\ (\text{duty cycle}\ (\%)) * 0.6 & \text{if } 10\% \leq \text{duty cycle} < 85\% \\ ((\text{duty cycle}\ (\%)) * 10 - 640) * 0.25 & \text{if } 85\% \leq \text{duty cycle} < 96\% \\ 0 & \text{if duty cycle} \geq 96\% \end{cases} \tag{3.13}$$

As this value is the AC available maximum current and the sent current reference set-point to AddVolt's unit is the DC current directly to the batteries, the value must be converted before sent. A simplified diagram of the current path is shown in Fgure 3.44.

S0

```
DIGITAL_IO_SetOutputLow(&CP_SWITCH);
data_sent_100ms[0] = 0x02;
send_current_flag = 0;
calculate_resistance_flag = 0;
```

S11

```
DIGITAL_IO_SetOutputLow(&CP_SWITCH);
data_sent_100ms[0] = 0x02;
send_current_flag = 0;
calculate_resistance_flag = 1;
```

S1

```
DIGITAL_IO_SetOutputLow(&CP_SWITCH);
data_sent_100ms[0] = 0x02;
send_current_flag = 0;
calculate_resistance_flag = 1;
```

S3

```
DIGITAL_IO_SetOutputLow(&CP_SWITCH);
data_sent_100ms[0] = 0x02;
send_current_flag = 0;
calculate_resistance_flag = 1;
```

S4

```
DIGITAL_IO_SetOutputLow(&CP_SWITCH);
data_sent_100ms[0] = 0x02;
send_current_flag = 0;
calculate_resistance_flag = 1;
```

S2

```
DIGITAL_IO_SetOutputHigh(&CP_SWITCH);
data_sent_100ms[0] = 0x01;

if(real_duty < 10)
max_current_available = 0;
else if(real_duty < 85)
max_current_available = real_duty*10*6/100;
else if(real_duty < 96)
max_current_available = (real_duty*10 - 640)*25/100;
else
max_current_available = 0;

max_current_available = max_current_available*(1*1.73205080757*400)/real_voltage;

data_sent_500ms[0] = 0x0C;
data_sent_500ms[1] = max_current_available;
data_sent_500ms[2] = 0;
data_sent_500ms[3] = 0;
data_sent_500ms[4] = 0;
data_sent_500ms[5] = 0;
data_sent_500ms[6] = 0;
data_sent_500ms[7] = 0;

send_current_flag = 1;
calculate_resistance_flag = 1;
```

Figure 3.43: Actions executed according to the states in the Finite State Machine of EVCC Firmware.

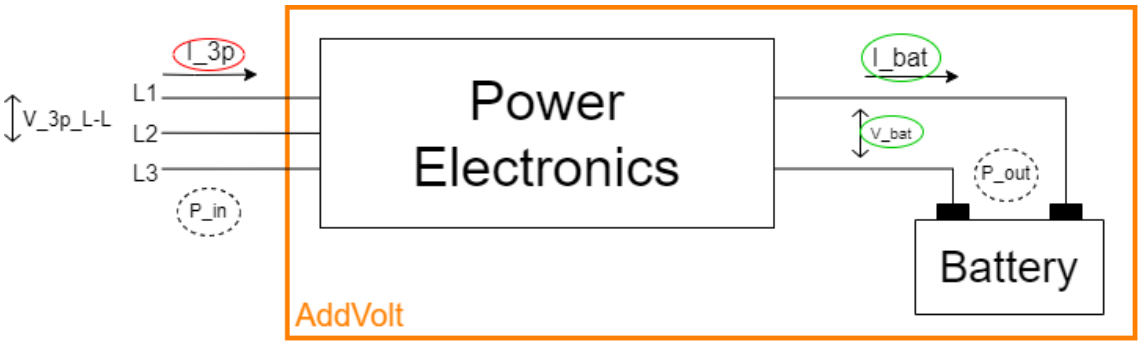


Figure 3.44: Simplified diagram of the charging current path from the grid to the battery pack to be charged.

The power efficiency of AddVolt's Power Electronics block is η. Then, the following can be deducted:

$$P_{out} = \eta P_{in} \tag{3.14}$$

and, as the equation for three-phase power is:

$$P_{in} = 3 V_{3p_{L-N}} I_{3p} = 3 \frac{V_{3p_{L-L}}}{\sqrt{3}} I_{3p} = \sqrt{3} V_{3p_{L-L}} I_{3p} \tag{3.15}$$

and the power delivered to the battery pack is:

$$P_{out} = V_{bat} I_{bat} \tag{3.16}$$

It results

$$V_{bat} I_{bat} = \eta \sqrt{3} V_{3p_{L-L}} I_{3p} \tag{3.17}$$

Thus, considering the efficiency of the Power Electronics block as being $\eta = 1$, that is, 100%, in order to simplify calculations, the current sent correspond to:

$$I_{bat} = \frac{\sqrt{3} V_{3p_{L-L}} I_{3p}}{V_{bat}} \tag{3.18}$$

and the estimated line current drawn from the three-phase grid is:

$$I_{3p} = \frac{V_{bat} I_{bat}}{\sqrt{3} V_{3p_{L-L}}} \tag{3.19}$$

with the value of V_{bat} received from AddVolt's CAN bus and I_{bat} set by the EVCC. As the value of $V_{3p_{L-L}}$ is known and corresponds to line-to-line voltage of the grid, it is equal to 400V. Therefore,

$$I_{bat} = 400 \sqrt{3} \frac{I_{3p}}{V_{bat}} \tag{3.20}$$

with $\sqrt{3}$ being rounded to $\sqrt{3} \approx 1.73205080757$ in code, in order to avoid including math libraries into EVCC's Firmware. It is to be noted that the values are approximated since the power efficiency is simplified and considered 100%.

Charging can be terminated by the SECC or by the EVCC. The SECC terminates charging either by unplugging or by pressing the Stop Button, the first alternative incurs in a duty cycle of 0% and the second alternative incurs in a duty cycle of 100% of the PWM signal in Control Pilot line. For the charging to be terminated by EVCC, it needs to get a value of battery State of Charge (SoC) equal or grater than 95%, this value is get from AddVolt's unit through CAN bus. Both of the listed conditions move EVCC's state to S3, which corresponds to a charging finished state, when the EVCC is waiting for the plug to be pulled out and then moving the state back to S0. It is to be noted that, at most of the states, the charging sequence may incur in a fault state, which is defined by a duty cycle between 7% and 8% generated by the SECC at Control Pilot line. The fault state can be detected at the states when the plug is connected and it is acknowledged by the EVCC by pulling out of the Type 2 connector, which corresponds to a voltage change at PP line.

3.5.4.1 Proximity Pilot Voltage Sensing

In all states of the FSM the vehicle is always sensing the voltage at the Proximity Pilot line. As shown in Section 3.4.2.3, the circuit is simple and the output of the operational amplifier in Buffer configuration is sensed by the microcontroller.

As there is a voltage sensing on this line, the DAVE App that is used for it is the Analog to Digital Converter, or ADC_MEASUREMENT, as it is called in the IDE. In order to achieve the functionality presented in Section 3.5.4.3, this ADC is triggered by an event generated by an implemented TIMER App, each 10μs.

As the operational amplifier saturates at 3.3V, it is the value read by it when the EVCC is disconnected from the SECC. Then, the ADC senses the maximum integer value of the measurement, which corresponds to 4095 as the conversion is done with 12 bits and $2^{12} = 4095$, with 4095 corresponding to 3.3V at the ADC input.

When the EVCC connects to SECC, a resistance is added in parallel to the voltage divider present at the Proximity Pilot line shown in Section 3.4.2.3. It lowers the voltage on the node connected to the microcontroller from 3.3V to approximately 1.91V, which corresponds to 2370 in the ADC. This event of voltage changing is what triggers the transition from the initial state to the second and backwards.

3.5.4.1.1 Proximity Pilot Resistance Estimation

If the value of the PP resistance of the *Mennekes* plug indicatinng cable capacity is unkonown, it can be estimated by the voltage value that is read from the ADC. As it was presented and following the schematic that is shown in Figure 3.20, the voltage v' on Control Pilot terminal is:

$$v' = v_{ADC}\frac{3.3}{4095} \tag{3.21}$$

with v_{ADC} being the value that is read from the ADC. Besides that, the equation for the voltage divider is:

$$v' = \frac{2700||R_{PP}}{330+2700||R_{PP}}5 \tag{3.22}$$

with R_{PP} being the Proximity Pilot Resistance to estimate. Then, rearranging Equation 3.22, it gives:

$$v'330 + v'2700||R_{PP} = (2700||R_{PP})5 \Leftrightarrow 2700||R_{PP} = \frac{v'330}{5-v'} \tag{3.23}$$

and, by the association of resistances 2700 and R_{PP} in parallel,

$$2700||R_{PP} = \frac{2700R_{PP}}{2700+R_{PP}} \tag{3.24}$$

$$2700R_{PP} = (2700||R_{PP})(2700+R_{PP}) \tag{3.25}$$

$$R_{PP} = \frac{(2700||R_{PP})2700}{2700-2700||R_{PP}} \tag{3.26}$$

Thus, by joining Equations 3.22 and 3.26, substituting $2700||R_{PP}$ it gives:

$$R_{PP} = \frac{\frac{v'330}{5-v'}2700}{2700-\frac{v'330}{5-v'}} \tag{3.27}$$

and by substituting Equation 3.21 in Equation 3.27, it results:

$$R_{PP} = \frac{\frac{v_{ADC}\frac{3.3}{4095}330}{5-v_{ADC}\frac{3.3}{4095}}2700}{2700-\frac{v_{ADC}\frac{3.3}{4095}330}{5-v_{ADC}\frac{3.3}{4095}}} \tag{3.28}$$

that gives R_{PP} in function of v_{ADC}, which is read by the ADC and then can be used to estimate the Proximity Pilot Resistance of the plug connected to the EVCC, which indicates the current capability of the cable. When the cable is connected, the flag **calculate_resistance_flag** is set to "1" and the estimated value of the resistance is calculated according to Equation 3.28. When the cable is disconnected, the flag is set back to "0" and the estimation is not calculated. The implementation of this condition is shown in Figure 3.45.

```
/* CALCULATE PROXIMITY PILOT RESISTANCE */
if(calculate_resistance_flag)
{
    Rpp = (((330*(pp*3.3/4095))/(5 - (pp*3.3/4095)))*2700)/(2700 - (330*(pp*3.3/4095))/(5 - (pp*3.3/4095)));
}
else
    Rpp = 0;
```

Figure 3.45: Estimation of Proximity Pilot Resistance in code.

3.5.4.2 Control Pilot Voltage Divider

Also on the Control Pilot line, there is a "response" from the EV to the EVSE, according to SAE J1772. This response is given by acting on a switch, which in turn adds a resistance in parallel to the voltage divider present at the Control Pilot line. This is presented in Section 3.4.2.4.

The switch is a MOSFET with two resistances to drive it. The input to the gate resistance is a digital output coming from the microcontroller, which is at *high level* when the EV is ready to charge, being at *low level* otherwise.

The addition of the resistance to the voltage divider changes the voltage from 9V to 6V (approximately), as there is no ventilation requirements implemented. If the EV in question requires ventilation in the charging procedure, the value of the resistance would be different and the voltage at Control Pilot line would be close to 3V.

Therefore, the DAVE™ App used to implement this feature was the DIGITAL_IO App.

3.5.4.3 Control Pilot PWM Signal Sensing

Complementary to the PWM signal generation that is done at the SECC side, at the EVCC that signal is sensed, reading its frequency and duty cycle according to what is established in the SAE J1772 protocol for low level communication.

In software, this is another feature that can take apart of the Apps in DAVE IDE, in order to simplify the code to be developed for the microcontroller XMC 4500. The feature in question uses the CAPTURE App existent in DAVE. The app provides an API with methods that can read the

period and duty cycle of a given signal. The App works by counting the time between the edges of the input signal, those can be rising or falling edges.

The configuration of CAPTURE App is similar to the ADC_MEASUREMENT App, also having an interrupt routine to handle measurement results. For the CAPTURE App, the routine was called **void CaptureIRQHandler()** and it is presented in Figure 3.46.

```
void CaptureIRQHandler()
{
    if(CAPTURE_GetPeriodInNanoSecDutyCycleInPercentage(&CP_PWM_INPUT, &signal_period, &signal_duty) == CAPTURE_STATUS_SUCCESS)
    {
        last_signal_period = signal_period;
        last_signal_duty = signal_duty;
    }
    else
    {
        XMC_DEBUG("Failed to capture\n");
    }
}
```

Figure 3.46: *void CaptureIRQHandler()* interrupt routine to handle the results from the measurements done in CAPTURE App.

Inside the interrupt routine is called a function that reads the period and duty cycle of the signal being sensed by the assigned pin in CAPTURE App and assigns it to the variables **uint32_t signal_period** and **uint32_t signal_duty**. If this capture is successfully done, the values of those variables are assigned to **uint32_t last_signal_period** and **uint32_t last_signal_duty**, respectively. Therefore, after the assign, those variables contain the values of duty cycle in percentage and of period in nanoseconds.

However, as the CAPTURE App only works by counting time between rising or falling edges, it has a limitation on sensing duty cycles of 0% and 100%, since the signals with those duty cycles do not present rising or falling edges.

The solution found to overcome this limitation consists in directly counting high and low levels when the CP signal has a duty cycle of 100% or 0%. It was done by adding a channel in the ADC_MEASUREMENT App instance already implemented for sensing the voltage on PP. This channel is assigned to the pin that is connected to the same terminal as the pin assigned to the CAPTURE App. That is, the channel measures the voltage at the output of the MOSFET driver connected to the output of the operational amplifier in Buffer configuration, on the Control Pilot line.

Then, the interrupt routine that is executed in the end of the measurements is presented in Figure 3.47.

```
void Adc_Measurement_Handler()
{
    pp = ADC_MEASUREMENT_GetResult(&ADC_MEASUREMENT_PP_Voltage);
    cp = ADC_MEASUREMENT_GetResult(&ADC_MEASUREMENT_CP_Voltage);

    if(cp >= 2048)
        high++;
    else
        low++;

    adc_count++;

    if(adc_count >= 10000)
    {
        last_high = high;
        high = 0;
        last_low = low;
        low = 0;
        adc_count = 0;
    }
}
```

Figure 3.47: Interrupt routine to handle values after ADC measurements are done.

As it is shown, it is verified if the voltage on CP line is greater or less than the middle value, 2048, which corresponds to 1.65V. If the voltage is greater, the variable **uint16_t high** is incremented. Otherwise, the variable **uint16_t low** is incremented. After the verification, the variable **uint16_t adc_count** is incremented and this repeats for each 10μs until **adc_count** reaches the value 10000. When it happens, the values stored in **high** and **low** are assigned to the variables **uint16_t last_high** and **uint16_t last_low** and **adc_count** starts to be incremented again each 10μs. An illustration of this measurement is presented in Figure 3.48.

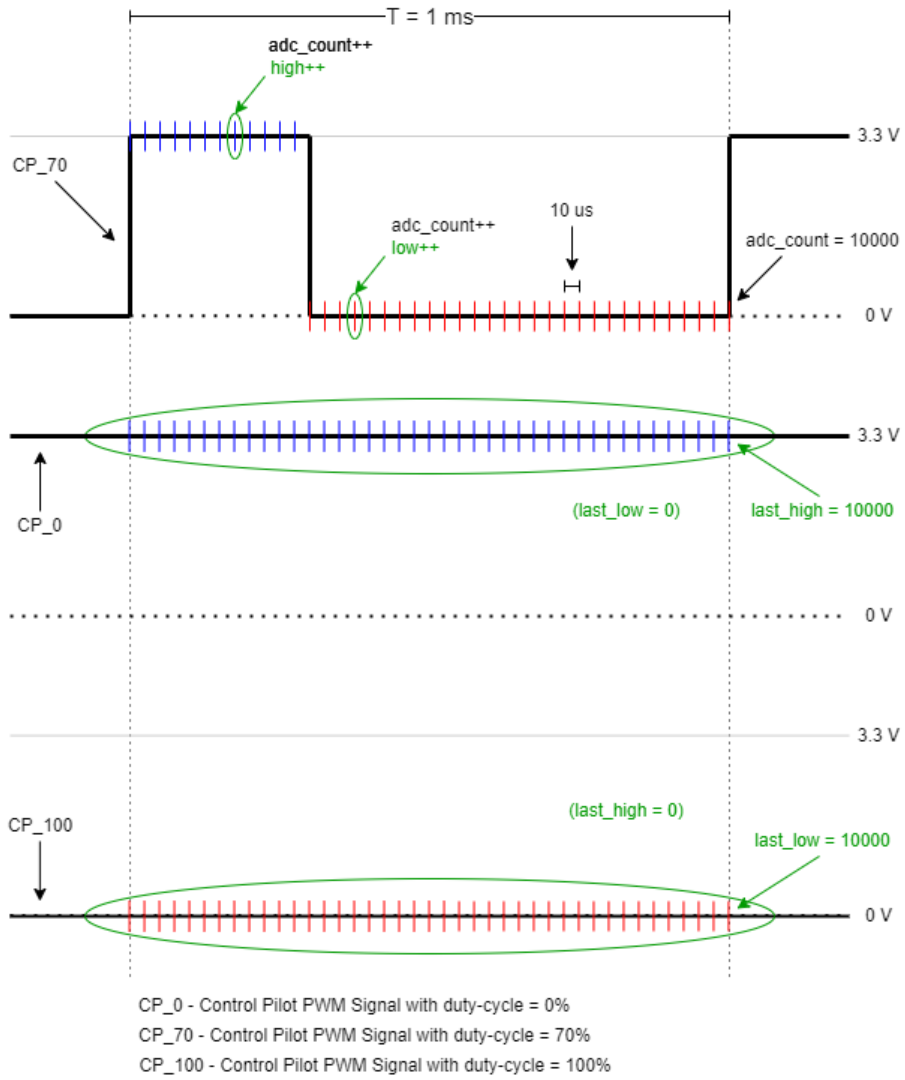


Figure 3.48: Example on ADC measurement for level counting to determine duty cycle of CP signal.

After having the values from CAPTURE App and from periodic counting, those values are processed. For that, a small function called **uint32_t get_duty_cycle()** was implemented and it is presented in Figure 3.49.

```
uint32_t get_duty_cycle()
{
    if(last_high >= 9900)
        return last_low/100;
    else if(last_low >= 9900)
        return last_low/100;
    else
        return 100 - last_signal_duty;
}
```

Figure 3.49: *get_duty_cycle()* function to determine the duty cycle of the PWM signal on CP line.

This routine is called in the beginning of every cycle of the EVCC to determine the current duty cycle of the signal on CP line and stores the value in the variable **uint32_t real_duty**. It checks if the duty cycle is between 0% and 1% or between 99% and 100% by checking the values of **last_high** and **last_low**, respectively. It is to be noted that as the input to the pin assigned to CAPTURE App and to the pin assigned to the ADC responsible for the periodic counting is the output of the MOSFET drive presented in Section 3.4.1.4, that means the duty cycle of the PWM signal at the inputs of the microcontroller is reversed when compared to the signal that was generated by the SECC. In order to overcome that, the variable taken into consideration for the determination of duty cycle is **last_low** divided by 100 instead of **last_high** divided by 100, resulting in the percentage value of the duty cycle. On the other hand, if the duty cycle is not in any of the mentioned intervals, it is used the value from the CAPTURE App due to its good precision. In the same way as the periodic counting, the duty cycle has to be reversed back to its original value. As the value from the CAPTURE App is already in percentage, it is only necessary to subtract this value from 100 and it results in the right value for the Control Pilot PWM duty cycle.

3.5.4.4 CAN Interface

Just like the other features, the CAN Interface is not different and can be implemented through an App, which in DAVE™ is called CAN_NODE. However, when processing the messages it was required a bit more of low level coding.

As the CAN communication was done through AddVolt's CAN bus Protocol the message structure and timing were needed to be in accordance with what was already implemented. Then, the CAN_NODE App was configured with 2 Logical Message Objects (LMO) and a baud rate of 500Kbps, being the message objects one of receive type and the other of transmit type and the baud rate being the same as the used in AddVolt's CAN bus Protocol. The configuration of the LMOs in the App instance is shown in Figure 3.50.

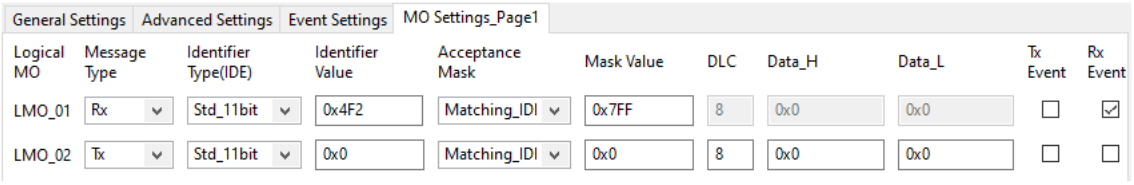


Figure 3.50: Configuration of Logical Message Objects (LMOs) in CAN_NODE App instance.

It is to be noted that the *Mask Value* is a software filter which is configured to only receive messages with the same *Identifier Value* as the configured in the App. Accordingly, whenever a message with the specified identifier is received, an *Rx Event* is generated, which is used to trigger an interrupt, shown further. Besides that, no value is configured in the parameters of the Tx LMO, since the transmission of messages is configured using an implemented routine, presented further.

The structure of the exchanged messages as defined in the protocol is presented in Tables 3.1, 3.2 and 3.3.

Table 3.1: Overview of the *start/stop* message.

Name	**Periodicity (ms)**	**Notes**
start/stop message	100	*Command to start/stop the unit*
Data Bytes		
Byte 0	Start = 0x01; Stop = 0x02	
Byte 1	0x01	
Byte 2	0x00	
Byte 3	0x02	
Byte 4	0x00	
Byte 5	0x00	
Byte 6	0x00	
Byte 7	0x00	

Table 3.2: Overview of the *set-point* message.

Name	**Periodicity (ms)**	**Notes**		
set-point message	500	*Command references/set-poits for variables of the unit*		
Variable/command list		**Data Bytes**		**Range**
1 (0x01)	...	**Byte 0**	Variable/command identifier	1 - 29 (0x01 - 0x1D)
...	...	**Byte 1**	Value of the variable (unsigned int)	0 - 255
12 (0x0C)	BATTERY_CHARGE_CURRENT	**Byte 2**	0x00	-
...	...	**Byte 3**	0x00	-
29 (0x1D)	...	**Byte 4**	0x00	-
		Byte 5	0x00	-
		Byte 6	0x00	-
		Byte 7	0x00	-

Table 3.3: Overview of the *variables* message.

Name	**Periodicity (ms)**	**Notes**		
variables message	1500	*Contain values of variables of the unit*		
Variable/command list		**Data Bytes**		**Range**
...	...	**Byte 0**	Variable/command identifier	1 - 190 (0x01 - 0xBE)
87 (0x57)	BATTERY_SOC	**Byte 1**	0x00	-
88 (0x58)	BATTERY_VOLTAGE	**Byte 2**	0x00	-
89 (0x59)	BATTERY_CURRENT	**Byte 3**	0x00	-
...	...	**Byte 4**	Float MSB	IEEE 754 FLOAT REPRESENTATION
		Byte 5	Middle Byte	
		Byte 6	Middle Byte	
		Byte 7	Float LSB	

Then, all the communication and variable exchange with AddVolt's power system was done following the existing CAN protocol. The EVCC sends the messages to command the start and stop of the power system through the *start/stop* message, as well as the message containing the maximum available output current by the charging station, which was sent by the SECC through the duty cycle of the PWM signal at the Control Pilot line, converted into an integer value representing the available DC current and sent as the *set-point* message. On the other hand, the EVCC receives *variables* messages, which can represent many variables of the system. As presented in the protocol, the first data byte identifies which variable is being received and the last 4 bytes correspond to the floating point variable representation, as stated in the standard IEEE 754. It is to be noted that as the variables of interest are only *State of Charge (SoC)*, *actual battery voltage* and *actual charging current*, the EVCC only stores values of messages with their first data byte being **0x57**, **0x58** or **0x59**.

As it can be seen in Table 3.3, the representation of floating point variables is done as stated in IEEE 754 standard and it is an automatic procedure of microcontrollers. Then, in order to access each byte of the float variable representation, the solution found is to take advantage of **typedef union** structure in C, which consists in sharing the same memory space to the variables inside it. Its implementation in code follows in Figure 3.51 with the declaration of the **flint** structure.

```
typedef union {
    float f;
    uint8_t arr[4];
} flint;

flint battery_current, battery_voltage, battery_soc;
```

Figure 3.51: Declaration of structure *flint* and of the necessary variables.

Therefore, both variables **float f** and **uint8_t arr[4]** are sharing the same memory space and because in C a float takes 4 bytes and the declared array consists in 4 elements of 1 byte each, both represent the same information, the float variable being the number itself and the array of bytes being its representation in IEEE 754. Then, whenever is assigned a value to **float f**, the array **uint8_t arr[4]** will contain its representation in IEEE 754 and vice versa.

Thus, besides the configuration of the App CAN_NODE, there were also some coding needed for the proper work. Those additions were the developing of a routine

$$\text{void can_transmit(uint16_t id, uint8_t *data, uint8_t length)},$$

which is called to send a CAN data frame with desired id and size, and the interrupt routine

$$\text{void CAN_RX_LMO_01()},$$

which is called whenever a message is received and actually receives and stores the desired message, in order to further use the value contained on it. The declaration of both routines is presented in Figures 3.52 and 3.53.

```
void can_transmit(uint16_t id, uint8_t *data, uint8_t length)
{
    const CAN_NODE_t *HandlePtr1 = &CAN_NODE_0;
    XMC_CAN_MO_t *MO_Ptr;
    MO_Ptr = HandlePtr1->lmobj_ptr[1]->mo_ptr;

    MO_Ptr->can_mo_type = XMC_CAN_MO_TYPE_TRANSMSGOBJ;  //Configure message object type as transmit type
    MO_Ptr->can_id_mode = XMC_CAN_FRAME_TYPE_STANDARD_11BITS; //configure MO identifier type
    MO_Ptr->can_identifier = id; // Configure id of the message
    MO_Ptr->can_data_length = length; // Configure CAN transmit MO data length field (set as 8)
    MO_Ptr->can_data[0] = data[0] + (data[1]<<8) + (data[2]<<16) + (data[3]<<24);// Configure Higher 4 bytes of Data
    MO_Ptr->can_data[1] = data[4] + (data[5]<<8) + (data[6]<<16) + (data[7]<<24);  // Configure Lower 4 bytes of Data

    // Runtime change the MO configuration
    CAN_NODE_MO_Init(HandlePtr1->lmobj_ptr[1]);

    CAN_NODE_STATUS_t mo_tranmit_status;
    CAN_NODE_STATUS_t can_status;

    mo_tranmit_status = CAN_NODE_MO_Transmit(HandlePtr1->lmobj_ptr[1]);
    if (mo_tranmit_status == CAN_NODE_STATUS_SUCCESS)
    {
          //message object transmission success.
          // read the TXOK status bit
          can_status = CAN_NODE_MO_GetStatus(HandlePtr1->lmobj_ptr[1]);

          if (can_status &  XMC_CAN_MO_STATUS_TX_PENDING)
          {
              //Clear the transmit OK flag
              XMC_CAN_MO_ResetStatus(MO_Ptr,XMC_CAN_MO_RESET_STATUS_TX_PENDING);
          }
    }
    else
    {
          // message object failed to transmit.
    }

}
```

Figure 3.52: Declaration of the routine *can_transmit()*.

The Figure 3.52 shows the *low level* coding that is needed to setup the registers of XMC 4500 to have a routine which can be directly invoked in code to transmit a CAN message. It basically consists of assigning to a pointer the address of the register corresponding to the configured LMO and then assigning the values from the parameters of the routine to configure a message with desired **id**, array of **data** bytes and **length**. The message is sent when the routine from CAN_NODE App

CAN_NODE_STATUS_t CAN_NODE_MO_Transmit(const CAN_NODE_LMO_t *lmo_ptr),

is called and it returns a successful status value.

```
void CAN_RX_LMO_01()
{

    CAN_NODE_STATUS_t receive_status;
    CAN_NODE_STATUS_t status;
    XMC_CAN_MO_t *MO_Ptr;
    const CAN_NODE_t *HandlePtr1 = &CAN_NODE_0;
    MO_Ptr = HandlePtr1->lmobj_ptr[0]->mo_ptr;

    status = CAN_NODE_MO_GetStatus(HandlePtr1->lmobj_ptr[0]);
    //Check receive pending status
    if ( status & XMC_CAN_MO_STATUS_RX_PENDING)
    {
      // Clear the flag
      XMC_CAN_MO_ResetStatus(MO_Ptr,XMC_CAN_MO_RESET_STATUS_RX_PENDING);
      // Read the received Message object
      receive_status = CAN_NODE_MO_Receive(HandlePtr1->lmobj_ptr[0]);
      if (receive_status == CAN_NODE_STATUS_SUCCESS)
      {
          // Clear the reception buffer bytes
          data_rx[0] = 0;
          data_rx[1] = 0;
          data_rx[2] = 0;
          data_rx[3] = 0;
          data_rx[4] = 0;
          data_rx[5] = 0;
          data_rx[6] = 0;
          data_rx[7] = 0;
          memcpy(data_rx, &MO_Ptr->can_data[0], sizeof(MO_Ptr->can_data[0]));
          memcpy(data_rx + sizeof(MO_Ptr->can_data[0]), &MO_Ptr->can_data[1], sizeof(MO_Ptr->can_data[1]));

          process_CAN_RX();
          // message object receive success.

      }
      else
      {
        // message object failed to receive.
      }
    }
}
```

Figure 3.53: Declaration of the interrupt routine *CAN_RX_LMO_01()*.

In Figure 3.53 there is presented the interrupt routine that is called when an *Rx Event* is generated and triggers the INTERRUPT App instance that was implemented for it. Similarly to Figure 3.52, it also assigns addresses to pointers, but in this routine its purpose is to access the data from the LMO instead of writing on it. The received value from the CAN bus is stored in the structure of the LMO by calling the routine

$$\text{CAN_NODE_STATUS_t CAN_NODE_MO_Receive(CAN_NODE_LMO_t *lmo_ptr)}$$

from CAN_NODE App. Then, if the reception is successful, the value in the data bytes is copied to a buffer which is an array of 8 bytes, **uint8_t data_rx[8]**. This data is then forwarded to global variables by the call of the routine

$$\text{void process_CAN_RX()}$$

which its declaration is shown in Figure 3.54.

```
void process_CAN_RX()
{
    if(data_rx[0] == 0x57) // Battery SoC
    {
        battery_soc.arr[3] = data_rx[4];
        battery_soc.arr[2] = data_rx[5];
        battery_soc.arr[1] = data_rx[6];
        battery_soc.arr[0] = data_rx[7];
        real_soc = battery_soc.f;
    }

    else if(data_rx[0] == 0x58) // Battery Voltage
    {
        battery_voltage.arr[3] = data_rx[4];
        battery_voltage.arr[2] = data_rx[5];
        battery_voltage.arr[1] = data_rx[6];
        battery_voltage.arr[0] = data_rx[7];
        real_voltage = battery_voltage.f;
    }

    else if(data_rx[0] == 0x59) // Battery Current
    {
        battery_current.arr[3] = data_rx[4];
        battery_current.arr[2] = data_rx[5];
        battery_current.arr[1] = data_rx[6];
        battery_current.arr[0] = data_rx[7];
        real_current = battery_current.f;
    }
}
```

Figure 3.54: Declaration of the routine *process_CAN_RX()*.

First, the routine checks the first data byte received, in order to define what is the data being received. As the value of the variables in the *variables* message are in IEE 754 representation, the bytes that represent it are individually assigned to the arrays **uint8_t arr[4]** in the declared **flint** structures presented in Figure 3.51. Therefore, the **float f** variables of the structure now contain the float value itself and those values are then assigned to the global **float** variables **real_soc**, **real_voltage** and **real_current**.

To send messages periodically, it was needed to update the **void update_timer()** routine that was presented in Section 3.5.2 with the call to the routine that sends the CAN messages. The updated routine is presented in Figure 3.55.

```
void update_timer()
{
    Timer_struct.milliseconds++;
    count_100ms++;
    count_500ms++;
    if(count_100ms >= 100)
    {
        //  send periodic 100ms messages
        can_transmit(0x2F0,data_sent_100ms,8);
        count_100ms = 0;
    }
    if(count_500ms >= 500)
    {
        //  send periodic 500ms messages
        if(send_current_flag)
            can_transmit(0x3F0,data_sent_500ms,8);
        count_500ms = 0;
    }
    if(Timer_struct.milliseconds >= 1000)
    {
        Timer_struct.milliseconds = 0;
        Timer_struct.seconds++;
    }
    if(Timer_struct.seconds >= 60)
    {
        Timer_struct.seconds = 0;
        Timer_struct.minutes++;
    }
    if(Timer_struct.minutes >= 60)
    {
        Timer_struct.minutes = 0;
        Timer_struct.hours++;
    }
    if(Timer_struct.hours >= 24)
    {
        Timer_struct.hours = 0;
        Timer_struct.days++;
    }
}
```

Figure 3.55: Updated *update_timer()* routine to increment the variables of *Timer_struct* structure periodically each 1ms and to send periodic CAN messages.

As it is shown, the update routine sends periodic messages with periods of 100ms and 500ms. Each 100ms it is sent the data contained in **uint8_t data_sent_100ms[8]**, which corresponds to the *start/stop* message with the instruction to start or stop being dependent to its first data byte, which is dependent to EVCC's state. Besides that, each 500ms is checked whether the flag **send_current_flag** is active, in order to send or not to the CAN bus the data contained in **uint8_t data_sent_500ms[8]**, which corresponds to the *set-point* message. The flag is set to 1 or to 0 depending on EVCC's state.

3.5.4.5 App Dependency

Therefore, with all the functionalities implemented, the final aspect of the App Dependency Tree in DAVE™ IDE for the EVCC is shown in Figure 3.56.

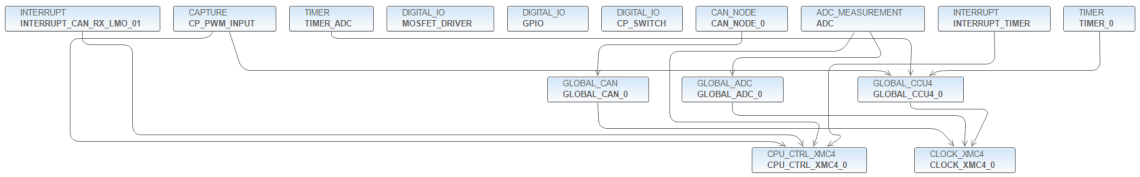


Figure 3.56: App Dependency Tree in DAVE^{TM} IDE for the EVCC.

In its final aspect, two more Apps instances were configures but not used, in order to forsee future updates in the EVCC. Those are instances of DIGITAL_IO App, named GPIO and MOSFET_DRIVER.

Chapter 4

Tests and Results

To prove the proposed concept, as well as to verify the good operation of the developed system, tests were carried out, together with the monitoring and analysis of relevant system variables.

It was for this purpose that the Micrium tool was used. Micrium is data monitoring and visualization software to track and, if necessary, even modify data in real time on a target Infineon XMC microcontroller. It works by tracking global variables in code and it can provide a Graphical User Interface (GUI), that can be designed by the user [48].

Therefore, different assemblies were made in order to test the different Use Cases desired for the system developed in a final implementation.

4.1 Physical Assembly

The tests assembly follow some of the desired the Use Cases for the designed system.

4.1.1 Test 1 - Charging the unit from the grid

The purpose of this assembly is to prove the general concept of AC charging according to SAE J1772 and the designed system itself, in order to have it validated to have the AddVolt unit being charged by an external AC charger. A schematic and a photograph of the assembly in question are shown in Figures 4.1 and 4.2.

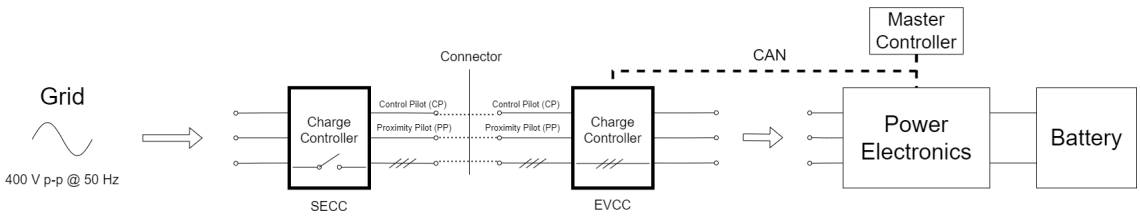


Figure 4.1: Schematic of the assembly made for Test 1.

Figure 4.2: Physical assembly made for Test 1.

4.1.2 Test 2 - Charging the unit from the grid using external charger

This assembly is the most important and proves the concept of having the AddVolt unit to be charged by an external charging station. For this, the Khons Mode 2 EV Portable Charger was used in order to simulate a charging station, to be connected to the EVCC, which in turn is connected to the AddVolt unit [49]. A schematic and a photograph of the assembly carried out in question are shown in Figures 4.3 and 4.4.

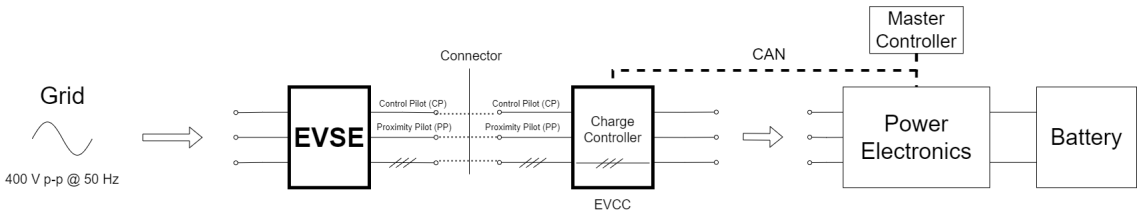


Figure 4.3: Schematic of the assembly made for Test 2.

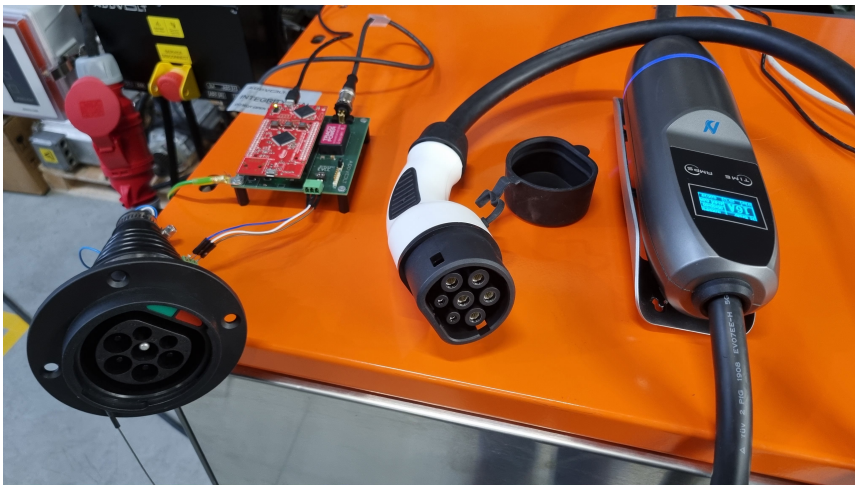


Figure 4.4: Physical assembly made for Test 2.

4.1.3 Test 3 - Charging the unit from another unit

This assembly is interesting because it allows an owner of more than one AddVolt unit to transfer energy from one to the other. For this, it is enough to use the SECC and EVCC. With the SECC connected to the three-phase output of one unit and the EVCC connected to the input of the other unit, as in the previous assemblies. A schematic and a photograph of the assembly carried out in question are shown in Figures 4.5 and 4.6.

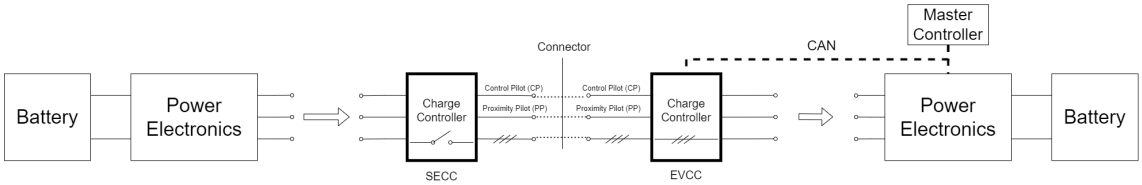


Figure 4.5: Schematic of the assembly made for Test 3.

Figure 4.6: Physical assembly made for Test 3.

4.2 Test of Use Cases

The tests carried out consisted of observing transitions at the beginning and the end of charging cycles, as well as counting the time taken to charge a certain State of Charge percentage value. For all tests, a GUI was developed in Micrium for monitoring and debugging system variables, in order to have a better observation of its operation. Two GUIs were created, one to monitor the SECC and the other to monitor the EVCC. In Figures 4.7 and 4.8, the GUIs to monitor the SECC and the EVCC are shown.

Figure 4.7: Graphical User Interface designed in Micrium to monitor the operation of the SECC.

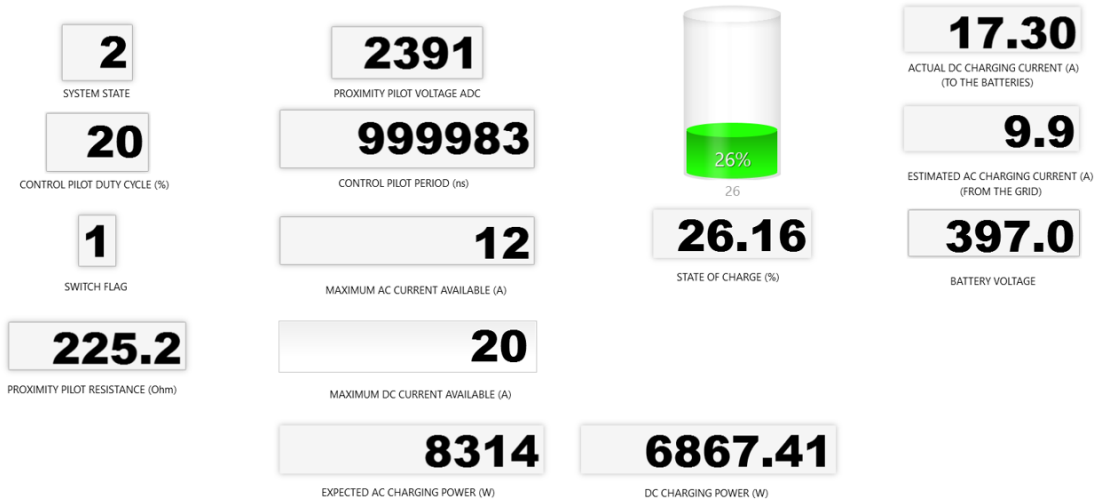


Figure 4.8: Graphical User Interface designed in Micrium to monitor the operation of the EVCC.

Besides that, in another section of the GUI, it was possible to trace and plot the evolution in time of desired variables of the Firmware. Those plots were used to prove the concept of the designed system.

An issue to be taken into account in the tests that were carried out is the fact that the Firmware version used to control the AddVolt units used in the tests was an experimental version for testing, which contained a bug. This bug consisted of an offset of approximately -2A in the value of the DC charging current that was sent by the AddVolt unit to the EVCC via CAN when in charging mode, thus, the charging current values received by the EVCC were always around 2A less than

the actual value the system was being charged with. Therefore, it is a bug that does not affect the operation of the system, but only the data that was exposed in the test.

4.2.1 Test 1 - Charging the unit from the grid

The test performed consisted of a connection and removal of the connector, observing in Micrium the state of the most relevant variables for the operation of the system. Since both SECC and EVCC were used, it was possible, in separate tests, to analyze the operation of each controller individually.

4.2.1.1 SECC

At the beginning, the SECC keeps in state 0 until the connection of the plug is sensed at the Control Pilot terminal. Besides that, as the SECC initiates with the maximum current available of 6A, the Toggle Current Button was used to change the maxium current to 16A. The evolution in time of the variable **max_current** is shown in Figure 4.9.

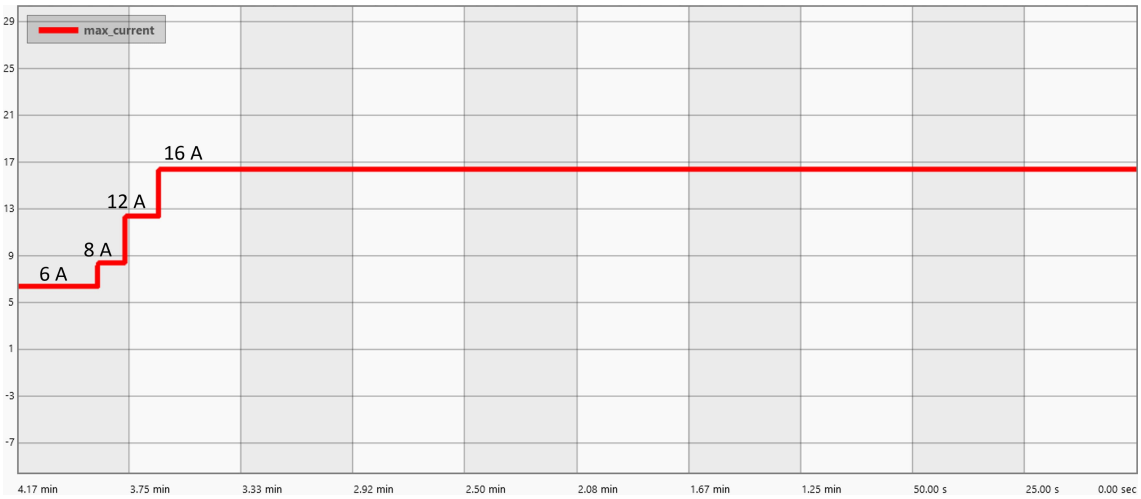


Figure 4.9: Evolution in time of the changes in the maximum available output current of the charger.

After the selection of the desired value for maximum output current, the charging plug was connected and, after the EVCC sensed the connection through Proximity Pilot and closed the switch on the Control Pilot voltage divider, the SECC detected the *ready to charge* state from the EVCC. This flow of states was detected through the voltage sensing on Control Pilot line, which is presented by the tracking of the variable **cp_adc** in Figure 4.10. The flow of states of the SECC monitored by tracking the variable **global_state** is presented in Figure 4.11.

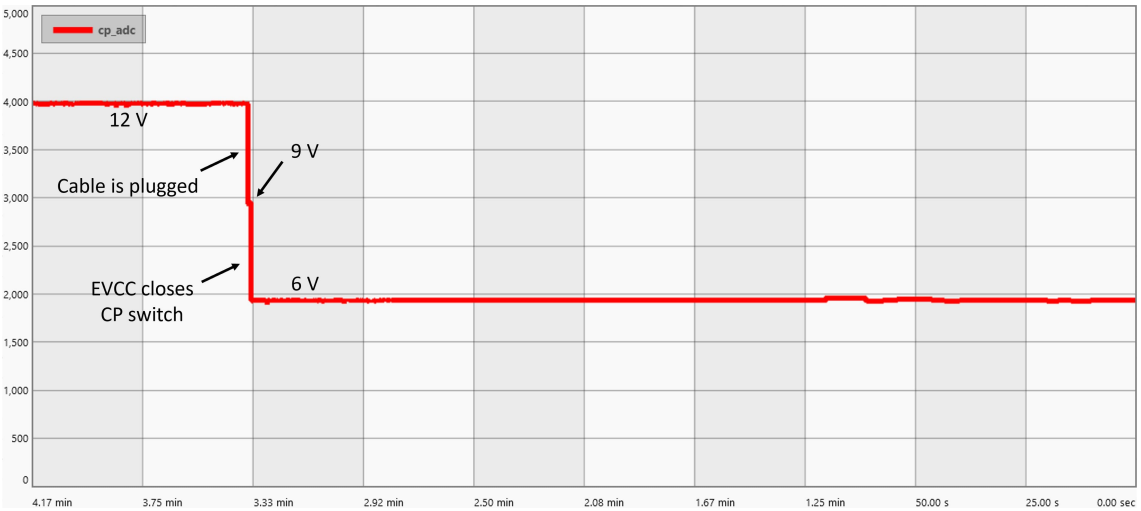


Figure 4.10: Evolution in time of the voltage on Control Pilot line, sensed by the SECC.

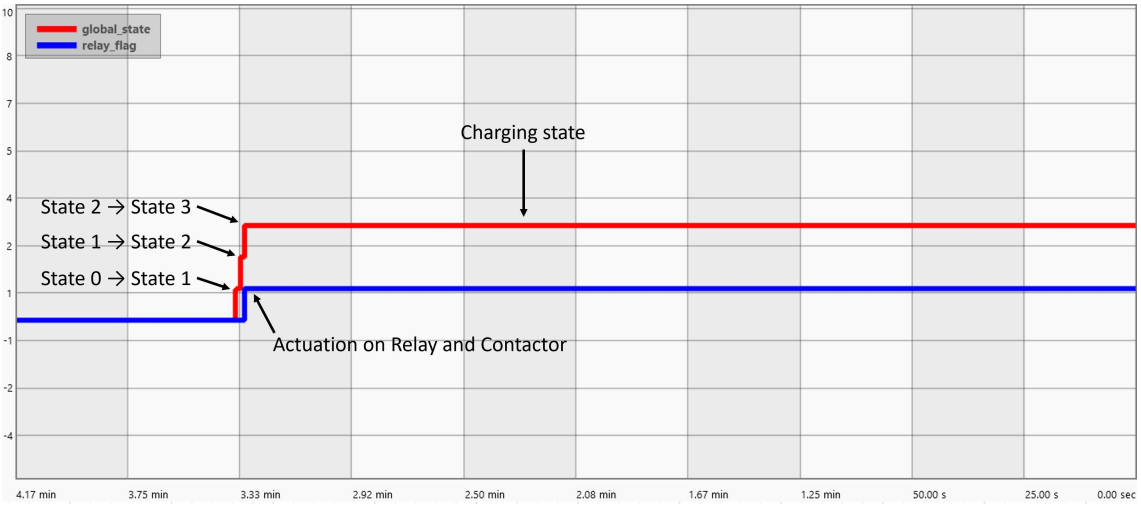


Figure 4.11: Evolution in time of the state of the SECC and the state of the relay and contactor.

An observation to be made is in the voltage trace on the Control Pilot line, **cp_adc**, in Figure 4.10. The values shown refer to periodic measurements that capture only the high level of the PWM signal on the Control Pilot line, as presented in the Section 3.5.3.2. Because of this, the values are constant and not a PWM signal as might be expected.

The proper operation of the PWM signal generation was verified using an Oscilloscope, analyzing the signal present on Control Pilot line, when the plug connection is made. The waveforms of two PWM signals, with duty cycles of 10% and 20%, respectively, are shown in Figures 4.12 and 4.13.

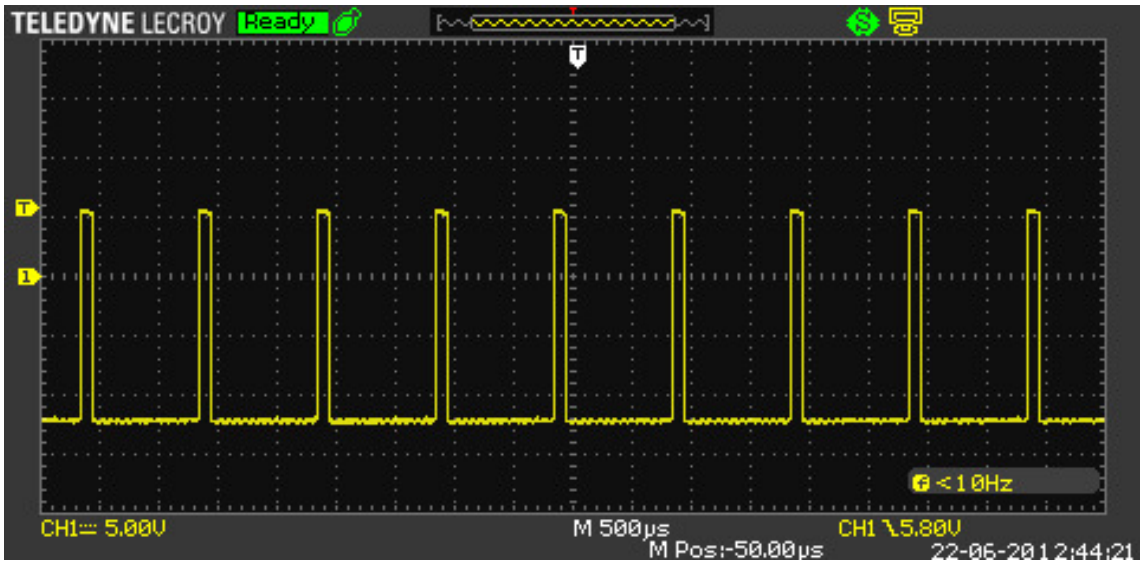


Figure 4.12: Control Pilot PWM signal with 10% duty cycle when the plug is connected and the switch on EVCC is closed, causing the high level of the signal to be at 6V and the low level remaining at -12V.

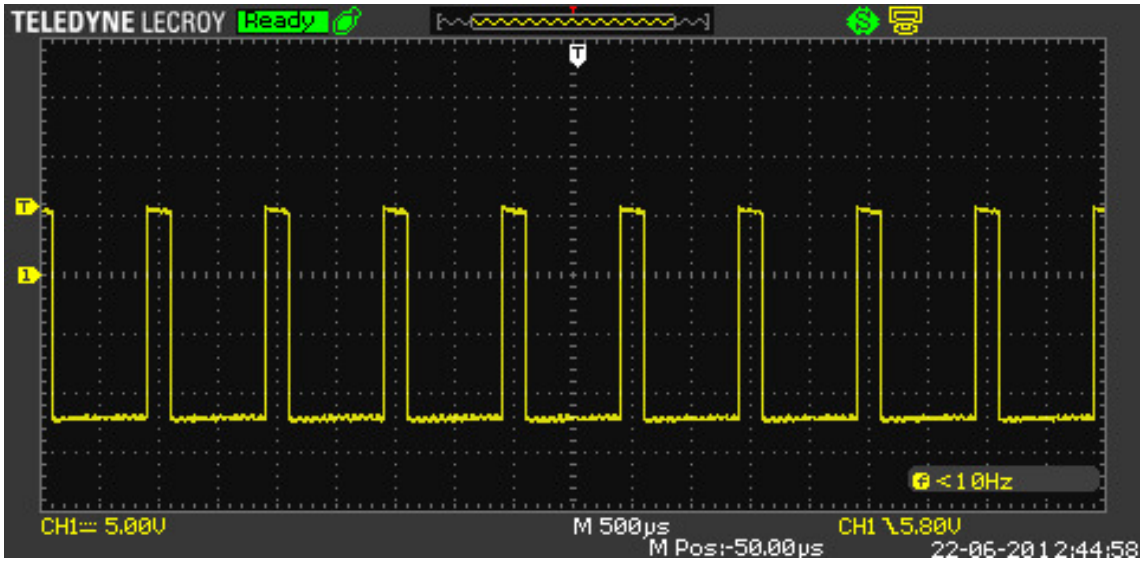


Figure 4.13: Control Pilot PWM signal with 20% duty cycle when the plug is connected and the switch on EVCC is closed, causing the high level of the signal to be at 6V and the low level remaining at -12V.

It is to be noted in Figure 4.11 that the actuation on the relay and contactor is done only at the *charging state*, which is the state 3 in SECC's FSM.

Before the charging cable is connected, the SECC outputs a $+12V_{DC}$ signal on Control Pilot line. When the charging cable is connected, it is sensed by the SECC as a drop on the voltage as shown in Figure 4.10 and then the SECC starts to output a PWM +/-12V with variable duty cycle on the Control Pilot line when the 6V voltage level is detected. As the maximum available current

selected was of 16A, the corresponding duty cycle of the signal is 27%, according to Equation 3.12. The evolution of the duty cycle of the PWM signal on Control Pilot line is shown in Figure 4.14.

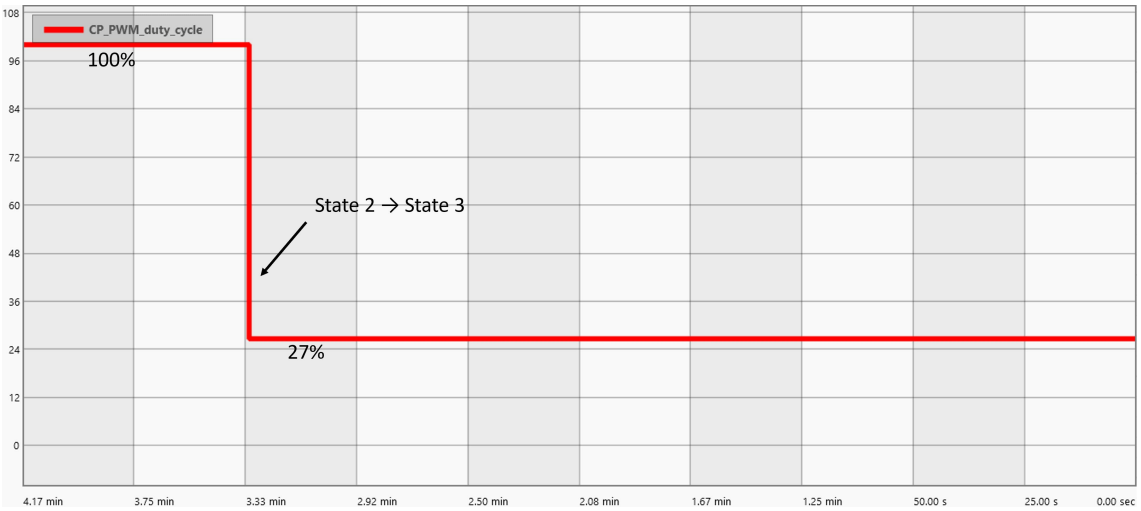


Figure 4.14: Evolution in time of the duty cycle of the PWM signal on Control Pilot line.

After some time charging, it can be terminated by pulling out the plug, by reaching a 95% of SoC or by pressing the Stop Button. In this test, charging was interrupted by pressing the Stop Button, before pulling out the charging plug. The behavior of SECC's state and of duty cycle and voltage level on Control Pilot line can be seen in Figures 4.15, 4.16 and 4.17, respectively.

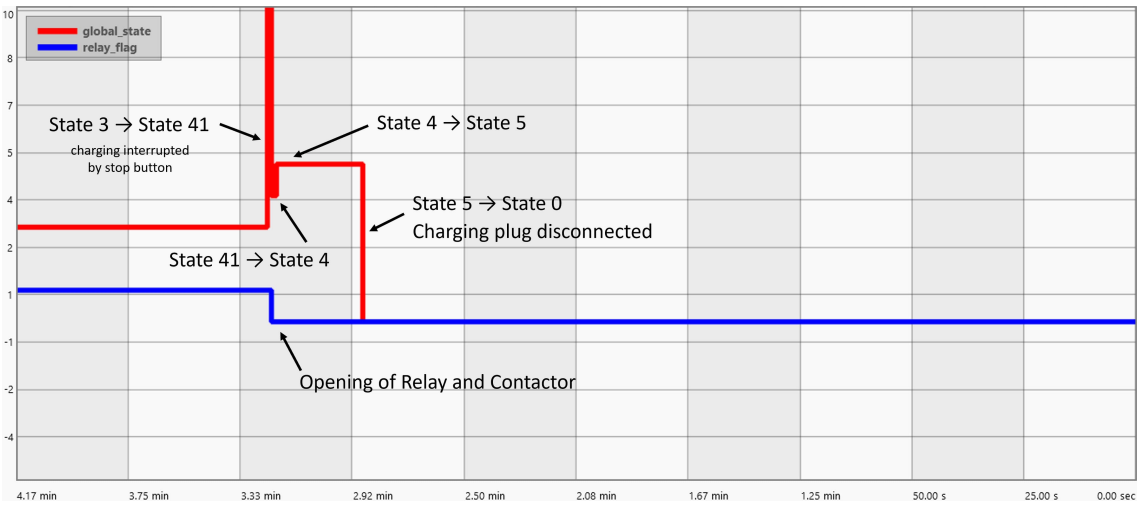


Figure 4.15: Evolution in time of SECC's state in its FSM when charging is interrupted.

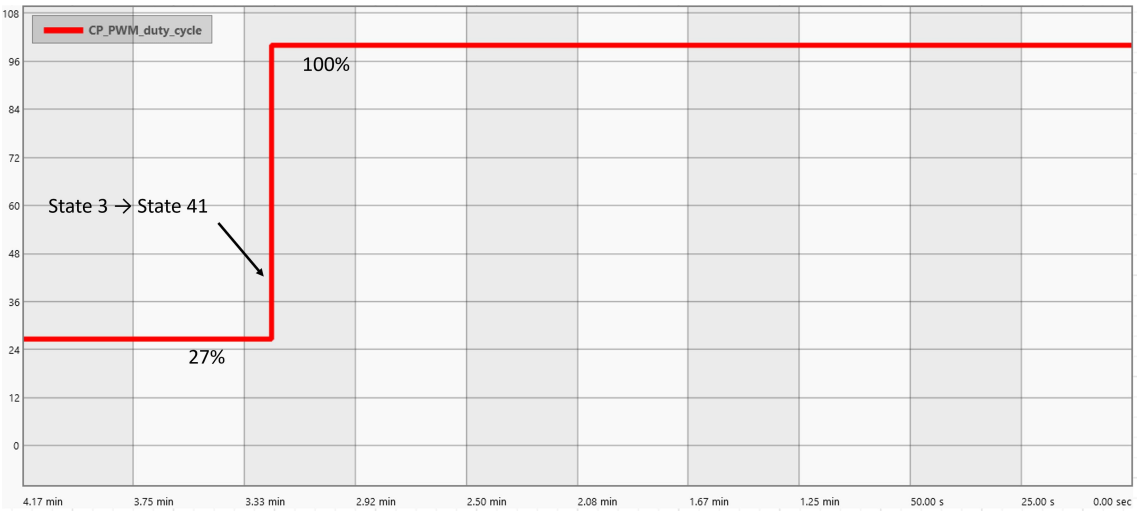


Figure 4.16: Evolution in time of Control Pilot PWM signal duty cycle when charging is interrupted.

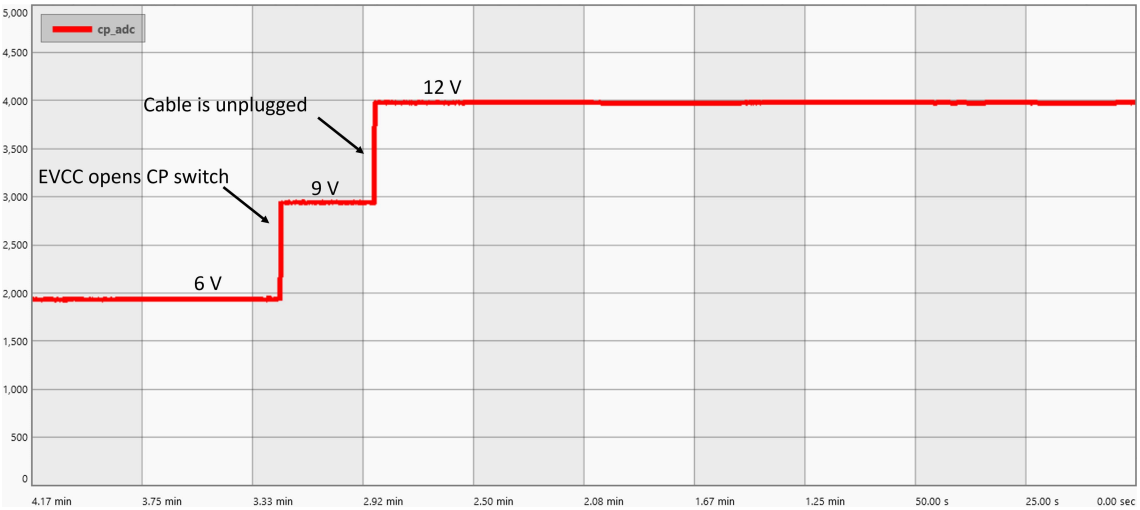


Figure 4.17: Evolution in time of Control Pilot voltage when charging is interrupted.

4.2.1.2 EVCC

The same way as the SECC, the EVCC also keeps in the state 0 of its FSM until a plug is connected to the Type 2 socket. When the plug is connected, it can be detected by the EVCC through voltage sensing on Proximity Pilot terminal. The evolution in time of the Proximity Pilot voltage that is sensed by the EVCC through ADC is shown in Figure 4.18 when the plug is connected.

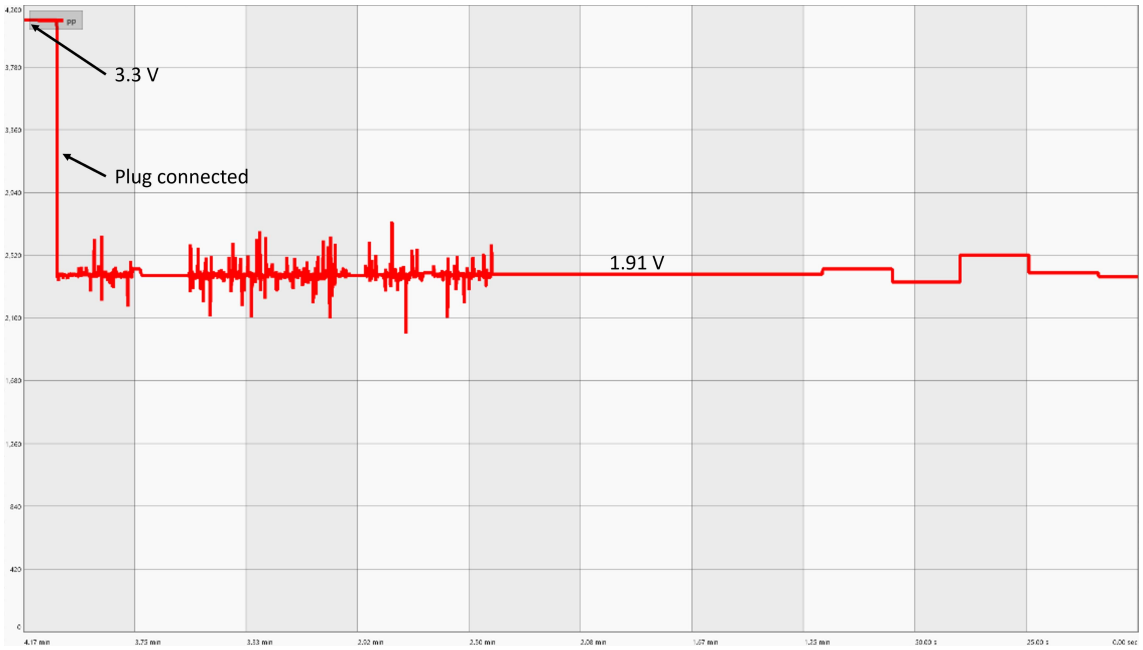


Figure 4.18: Evolution in time of Proximity Pilot voltage when Type 2 plug is connected.

The connection of the plug is also detected by sensing the duty cycle of the signal on Control Pilot line. When the EVCC is waiting for a plug to be connected, it senses a duty cycle of 0%, if a plug is connected, it senses a duty cycle of 100% (corresponding to $+12V_{DC}$) for an instant and then it senses a variable duty cycle according to the maximum available output current when it is at *charging state*. In this test, the maximum available current selected in the SECC was 12A. Then, the duty cycle to be sensed corresponds to 20% according to Equation 3.13. The described behavior can be seen in Figure 4.19.

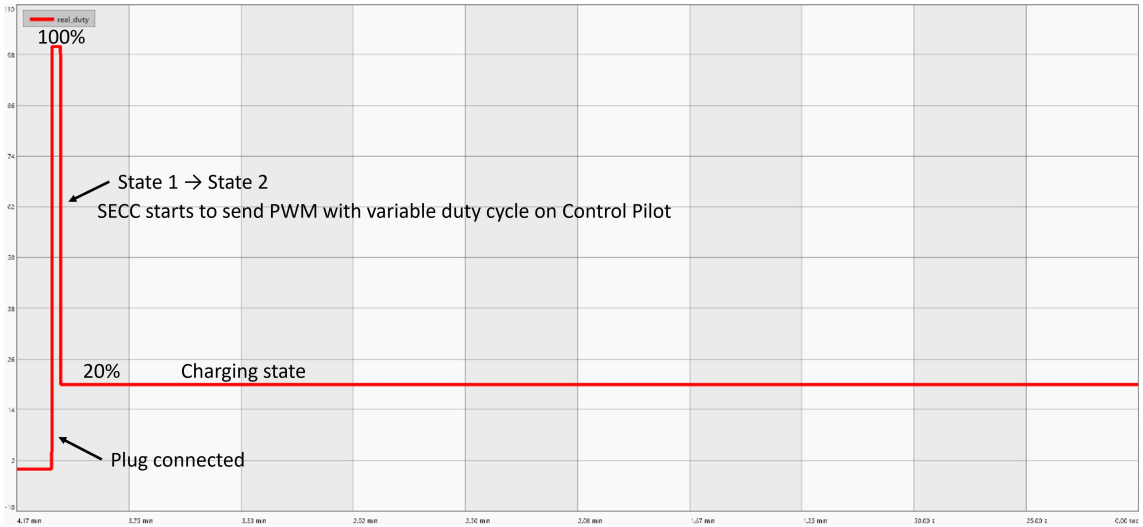


Figure 4.19: Evolution in time of Control Pilot duty cycle when Type 2 plug is connected.

As the duty cycle of the signal on Control Pilot line is the main variable that dictates the flow

of states in EVCC's FSM, the tracking of **global_state** variable follows the tracking of the duty cycle and can be seen in Figure 4.20.

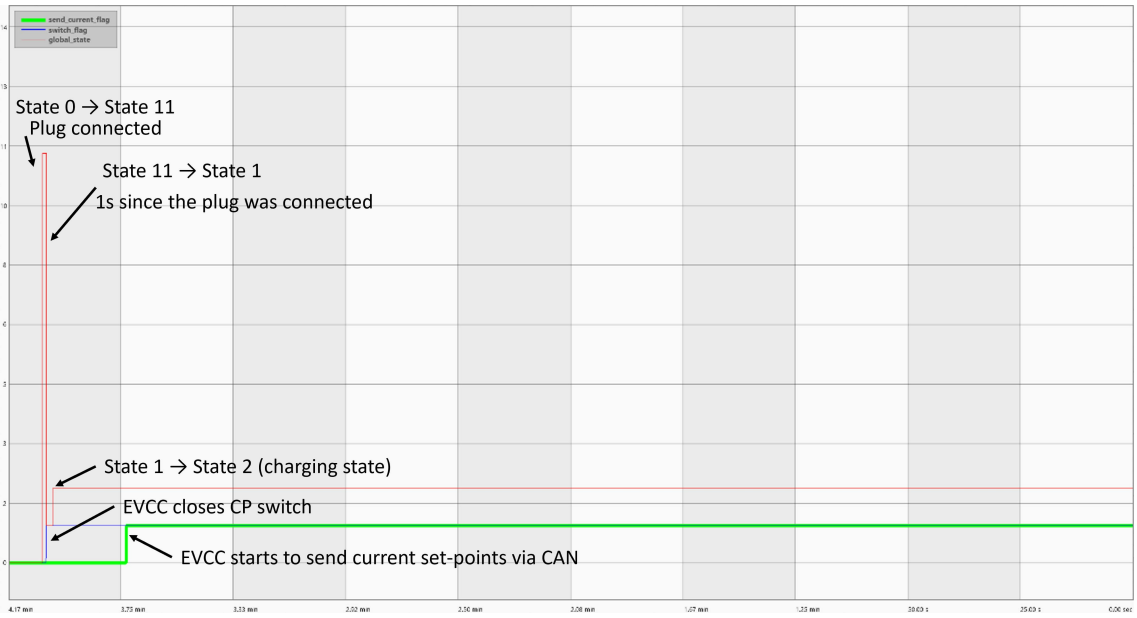


Figure 4.20: Evolution in time of EVCC's state when Type 2 plug is connected.

When the EVCC reaches *charging state*, it starts to send by CAN the DC current set-points to AddVolt's unit, after the conversion presented in Equation 3.18 and with the message structure presented in Table 3.2, and also sends periodically the *start* message to turn on the unit, by changing the first byte of the message with the structure presented in Table 3.1. Then, when AddVolt's unit start to respond to the CAN inputs, the behavior of the charging current and power can be seen in Figures 4.21 and 4.22, respectively.

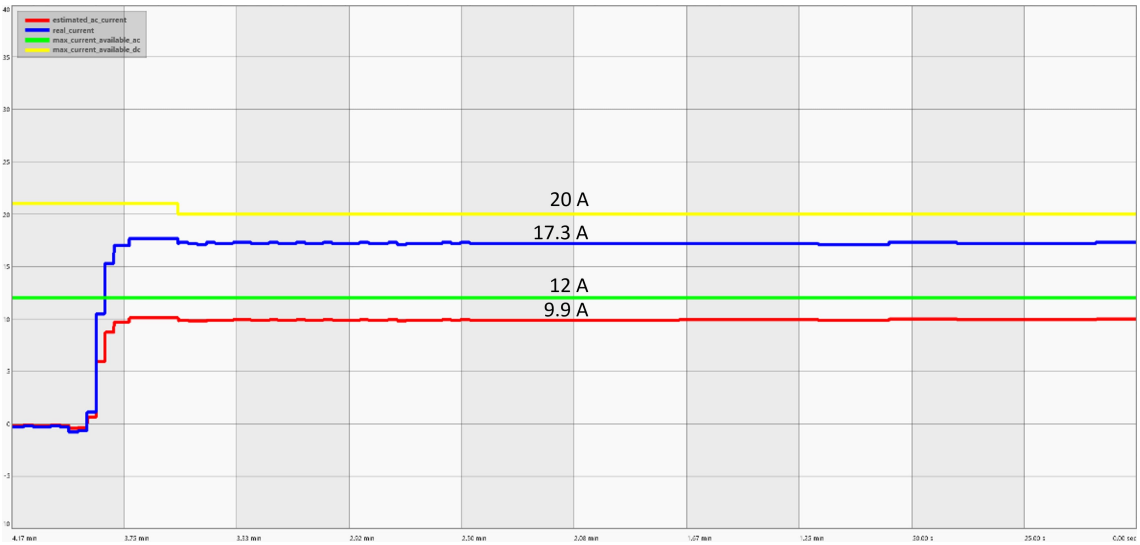


Figure 4.21: Evolution in time of the charging current in the beginning of the charging procedure, when the Type 2 plug is connected.

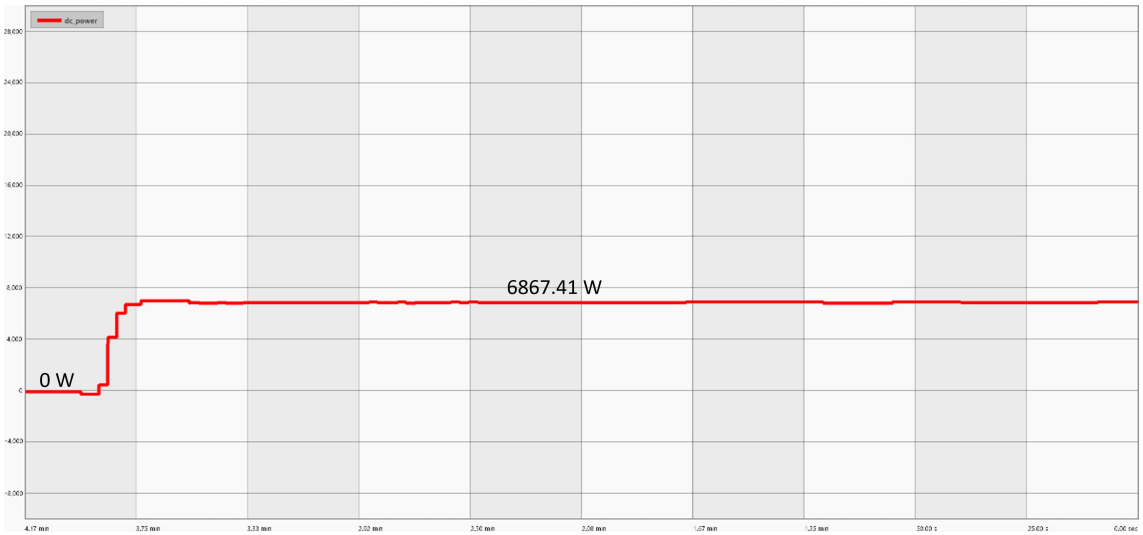


Figure 4.22: Evolution in time of the charging power in the beginning of the charging procedure, when the Type 2 plug is connected.

The other tracked variable throughout the whole charging procedure is the SoC, which its evolution in time at the beginning of the charge is shown in Figure 4.23.

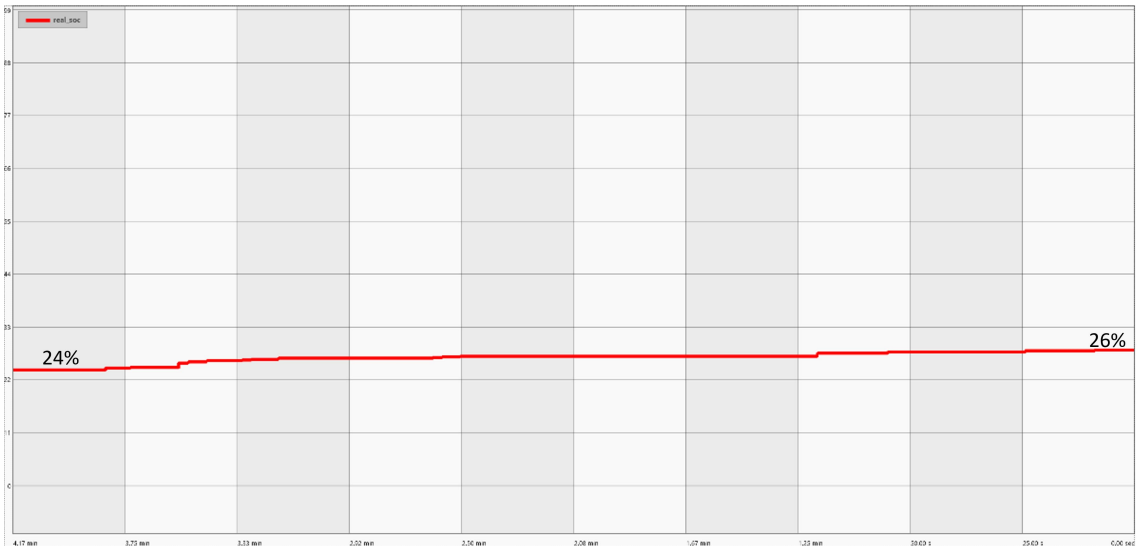


Figure 4.23: Evolution in time of the *State of Charge (SoC)* in the beginning of the charging procedure, when the Type 2 plug is connected.

During the charging procedure, the user can change the maximum available current by pressing the Toggle Button on the SECC. In a separate test of this Use Case, this feature was tested by changing the maximum available AC output current during the charging, causing the duty cycle of the PWM signal on Control Pilot line to change and then changing the current set-point to be sent by the EVCC to AddVolt's unit. This behavior was detected by observing Control Pilot PWM duty cycle, charging current and charging power and their evolution in time is presented in Figures 4.24, 4.25 and 4.26, respectively.

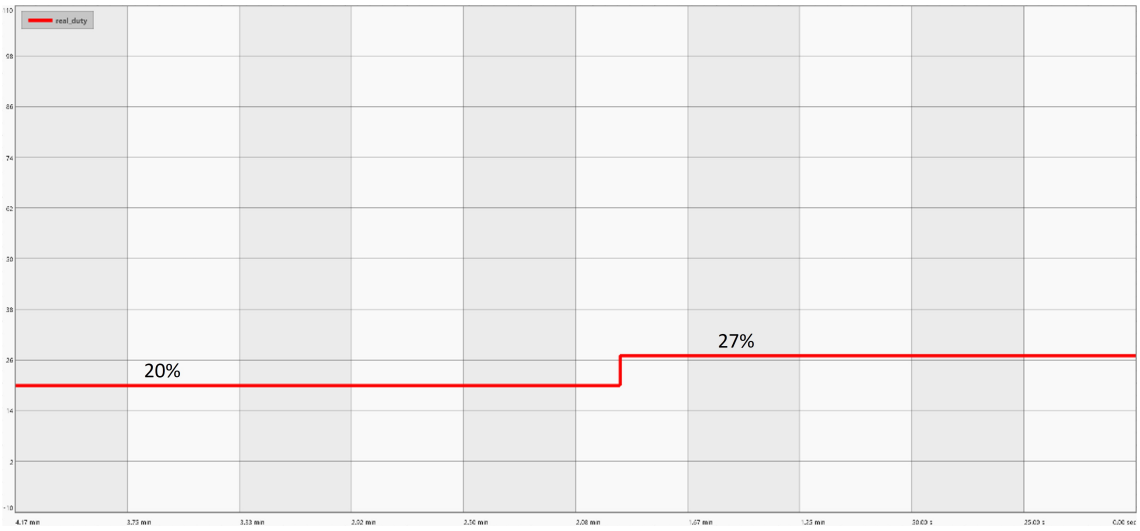


Figure 4.24: Control Pilot PWM signal duty cycle change in response to changing the maximum available AC current from the SECC.

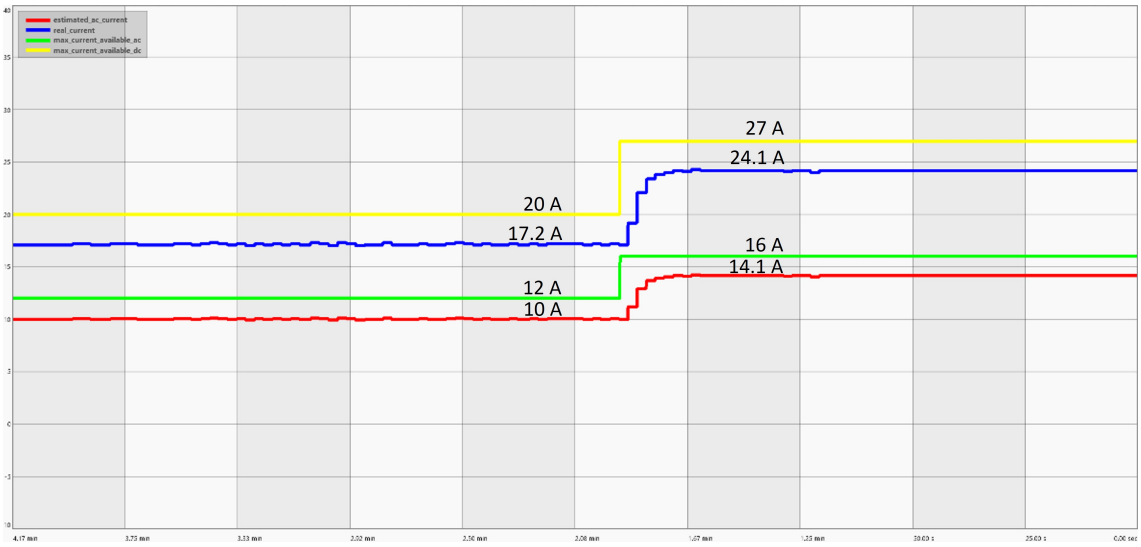


Figure 4.25: Charging current response to the change on maximum available AC current.

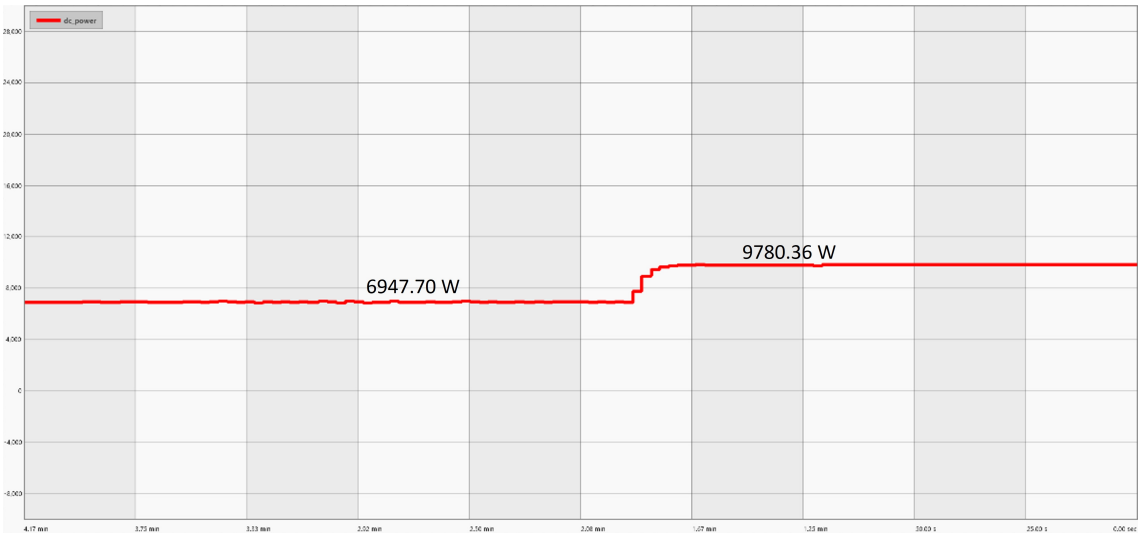


Figure 4.26: Charging power response to the change on maximum available AC current.

After some time charging, in this test it was interrupted by pressing the stop button on the SECC. By doing that, the SECC starts to output +12V_{DC} again on Control Pilot line, causing the EVCC to detect a duty cycle of 100% at this same terminal until the charging plug is disconnected. At this instant, the EVCC sends to AddVolt's unit via CAN a current set-point of 0A, in order to stop charging, and the EVCC also starts to send periodically each 100ms a *stop* message to turn off AddVolt's unit, by changing the first byte of the message with the structure presented in Table 3.1. When the plug is pulled out, the EVCC starts to sense a 0% duty cycle on Control Pilot line and the voltage on Proximity Pilot comes back to its initial value. This behavior and the system state, as well as the value of Control Pilot PWM duty cycle, Proximity Pilot voltage, charging current and charging power are shown in Figures 4.27, 4.28, 4.29, 4.30 and 4.31, respectively.

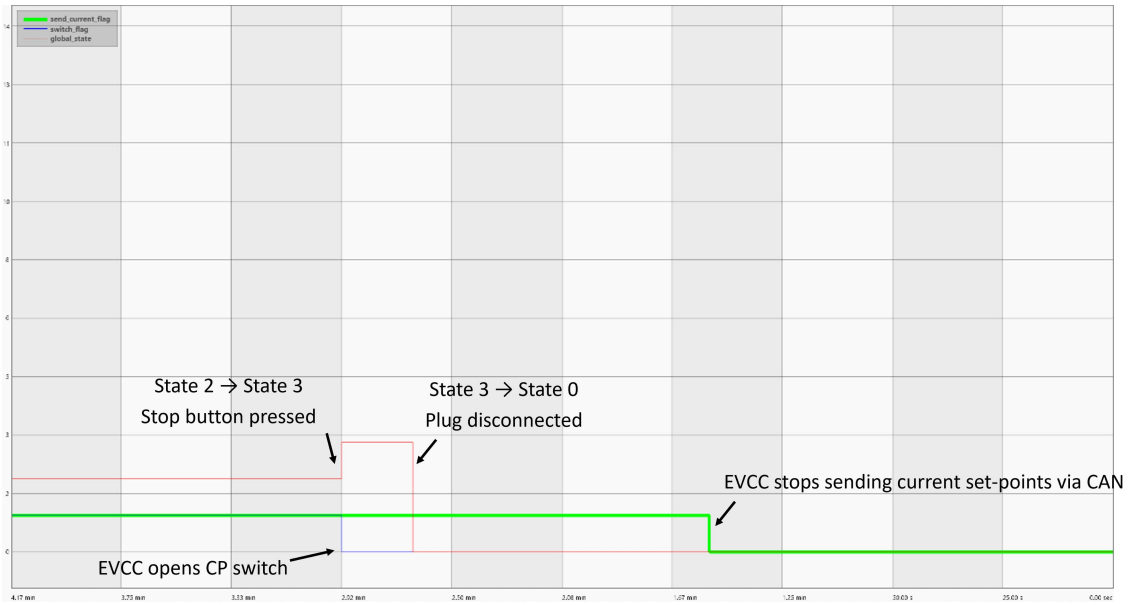


Figure 4.27: Evolution in time of EVCC's state in the end of the charging procedure, when the stop button of the SECC is pressed and the Type 2 plug is disconnected.

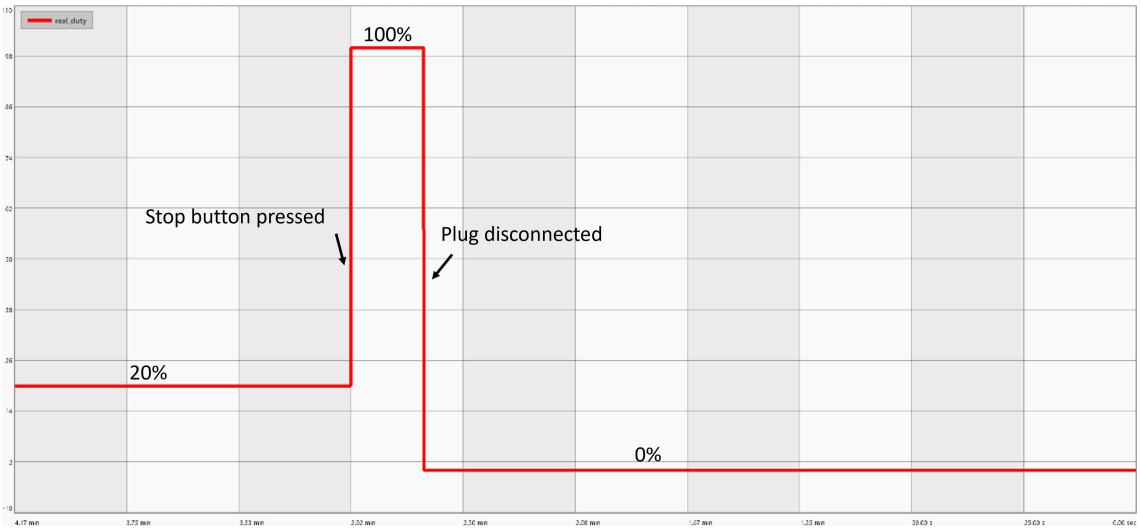


Figure 4.28: Evolution in time of Control Pilot PWM duty cycle in the end of the charging procedure, when the stop button of the SECC is pressed and the Type 2 plug is disconnected.

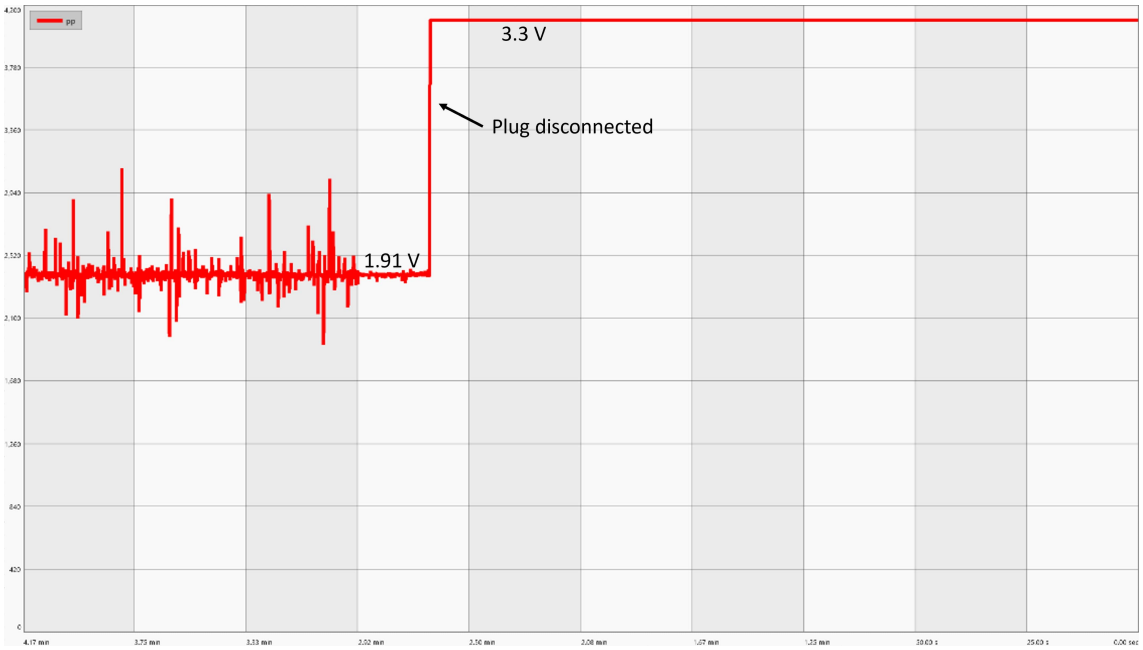


Figure 4.29: Evolution in time of Proximity Pilot voltage in the end of the charging procedure, when the stop button of the SECC is pressed and the Type 2 plug is disconnected.

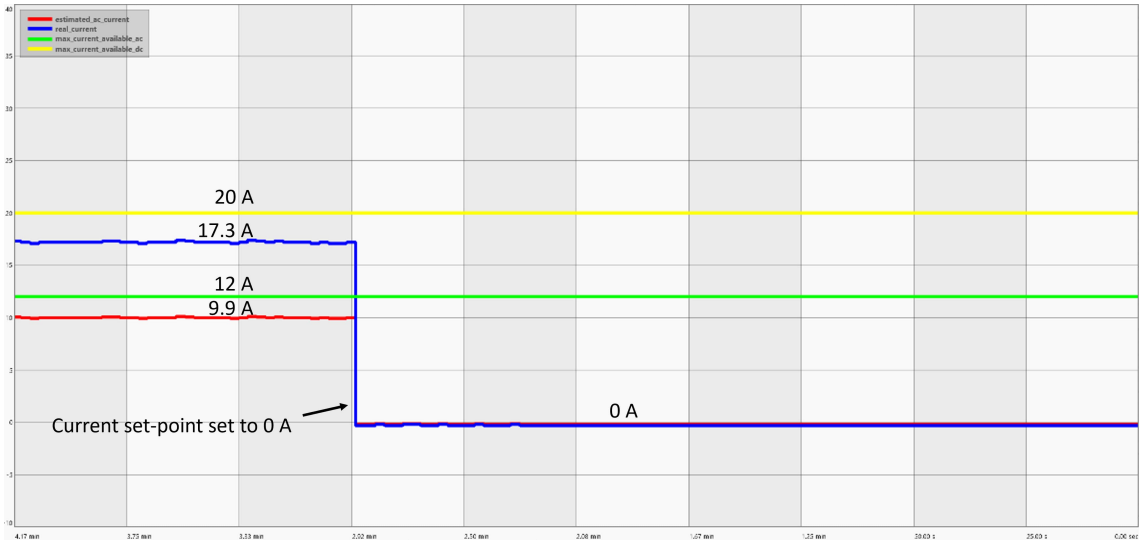


Figure 4.30: Evolution in time of the charging current in the end of the charging procedure, when the stop button of the SECC is pressed and the Type 2 plug is disconnected.

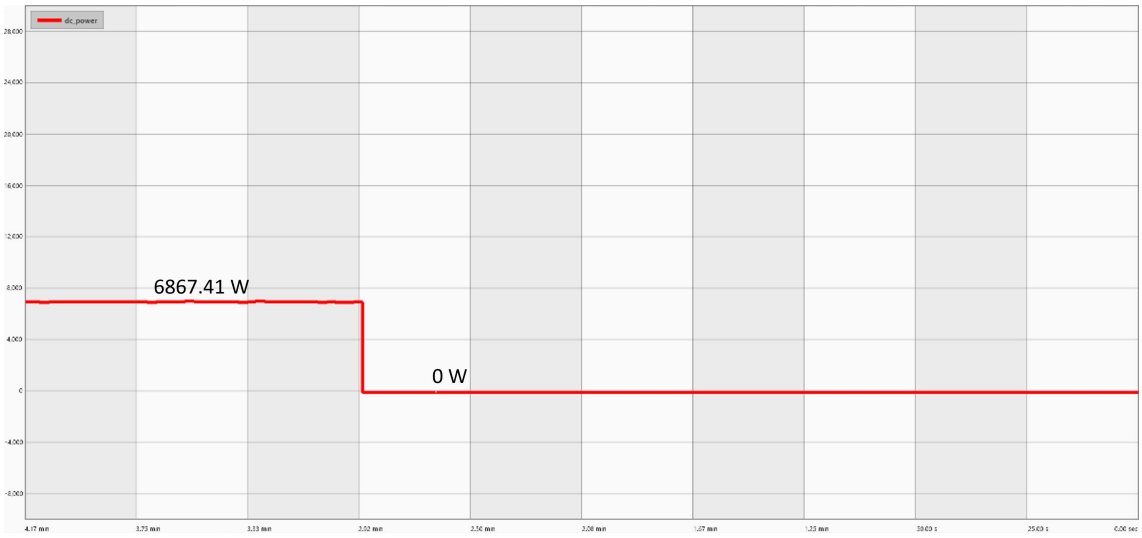


Figure 4.31: Evolution in time of the charging power in the end of the charging procedure, when the stop button of the SECC is pressed and the Type 2 plug is disconnected.

It is to be noted that in Figures 4.30 and 4.31 can be seen the fast response of the system to interrupt charging. Furthermore, it can be seen by observing the moment the plug is disconnected in Figure 4.29 that the charging is already interrupted for some seconds, with a current of 0A and the contactors in open state at the SECC side. This can ensure a safe disconnection of the charging plug. That is the reason the EVCC keeps sending current set-points via CAN for some time after the stop button was pressed, as shown in Figure 4.27.

4.2.2 Test 2 - Charging the unit from the grid using external charger

As an external EVSE was used in this assembly, the SECC was not used and because of that, there is only analysis to be done in the EVCC.

4.2.2.1 EVCC

Following the same procedure as the previous test, the EVCC keeps in state 0, the initial state, until the charging plug is connected. When connected, the plug presence is sensed by the EVCC the same way. The plug presence detection through Proximity Pilot terminal is presented in Figure 4.32.

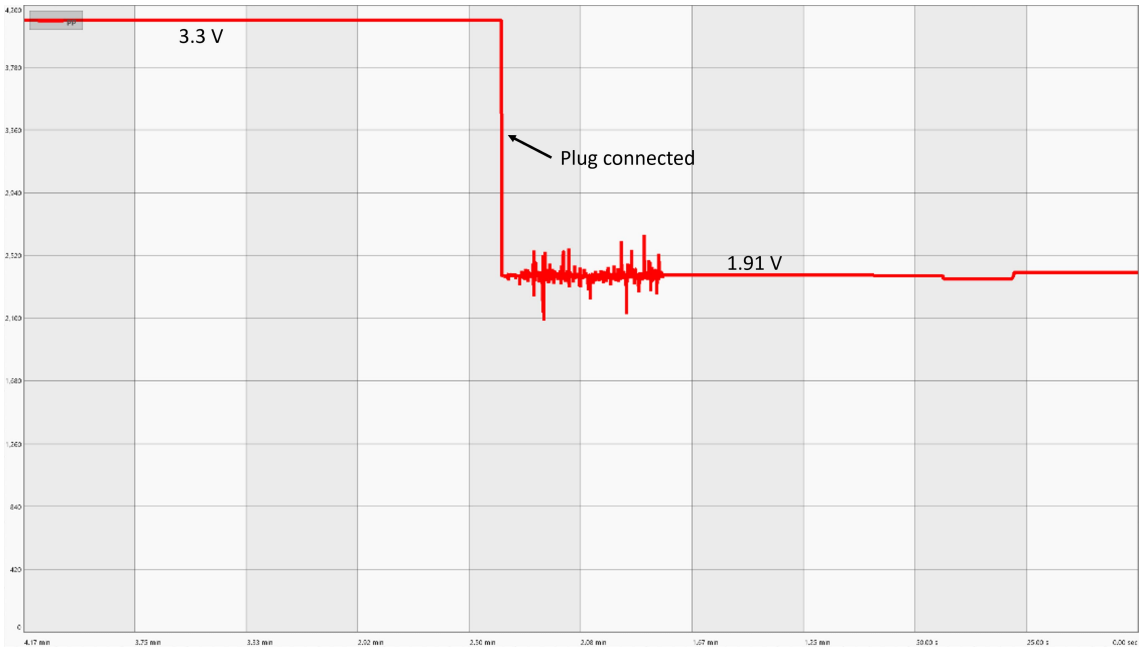


Figure 4.32: Evolution in time of Proximity Pilot voltage when Type 2 plug of the external EVSE is connected.

As in previous test, the plug presence can also be detected by sensing the duty cycle of the signal in Control Pilot line. Repeating the same behavior as in Figure 4.19 in previous test, the behavior of the Control Pilot PWM duty cycle in the test with the external EVSE is presented in Figure 4.33.

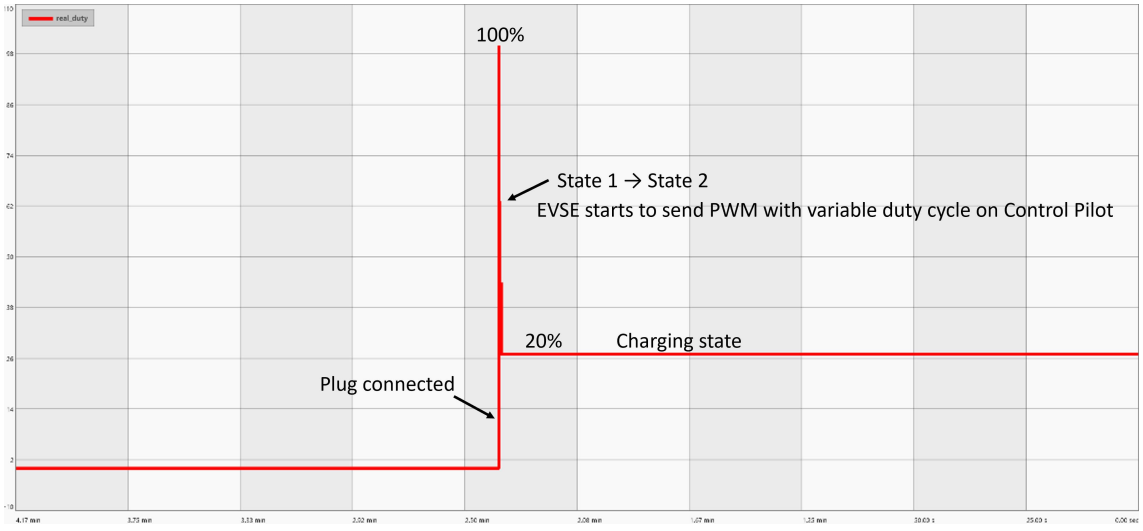


Figure 4.33: Evolution in time of Control Pilot PWM duty cycle in the beginning of the charging procedure, when Type 2 plug of the external EVSE is connected.

It is to be noted by comparison of Figures 4.19 and 4.33 that the EVSE starts to output a signal with variable duty cycle sooner than the designed SECC. This can be due to the fact that, according to IEC 61851-1 [7], the station-side charging controller can output a variable duty cycle

at any moment since the plug is connected, being capable of generate on its output a variable duty cycle PWM even right at the moment the plug is connected, whereas the designed SECC only does it when the EVCC closes the CP switch, corresponding to a *ready to charge* state.

As the duty cycle of the PWM signal at Control Pilot line dictates the flow of states in EVCC's FSM, it guides the behavior of EVCC's state, that is presented in Figure 4.34.

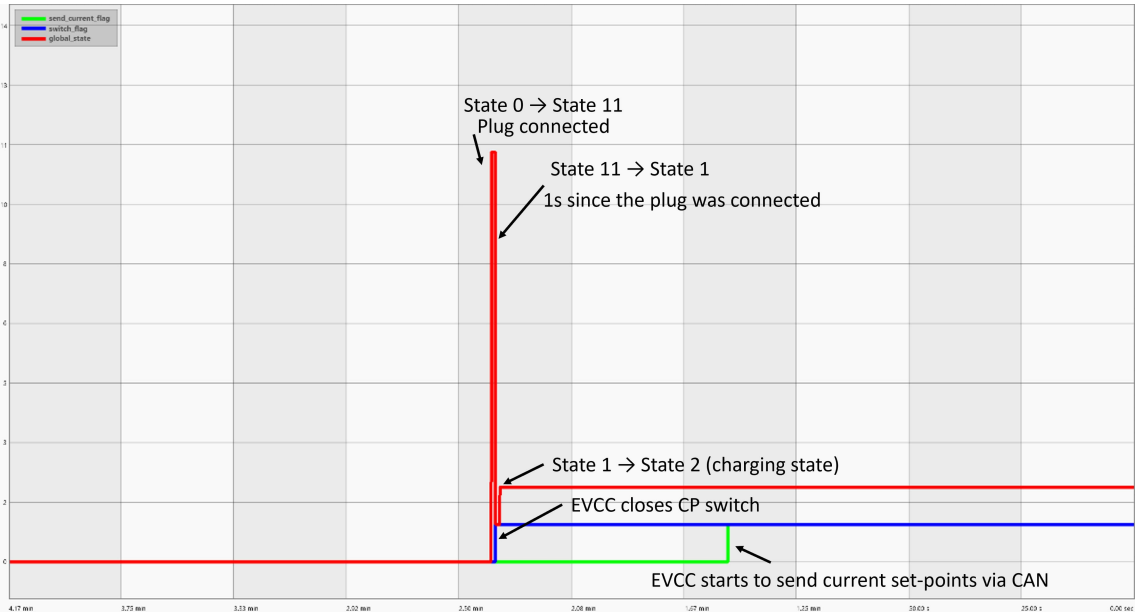


Figure 4.34: Evolution in time of EVCC's state in the beginning of the charging procedure, when Type 2 plug of the external EVSE is connected.

At the *charging state*, when **send_current_flag** is "on" the EVCC starts to send by CAN the *start* message and the DC current set-points to AddVolt's unit, after the conversion and with the message structure presented in the previous test. The behavior of the charging current and power can be seen in Figures 4.35 and 4.36, respectively.

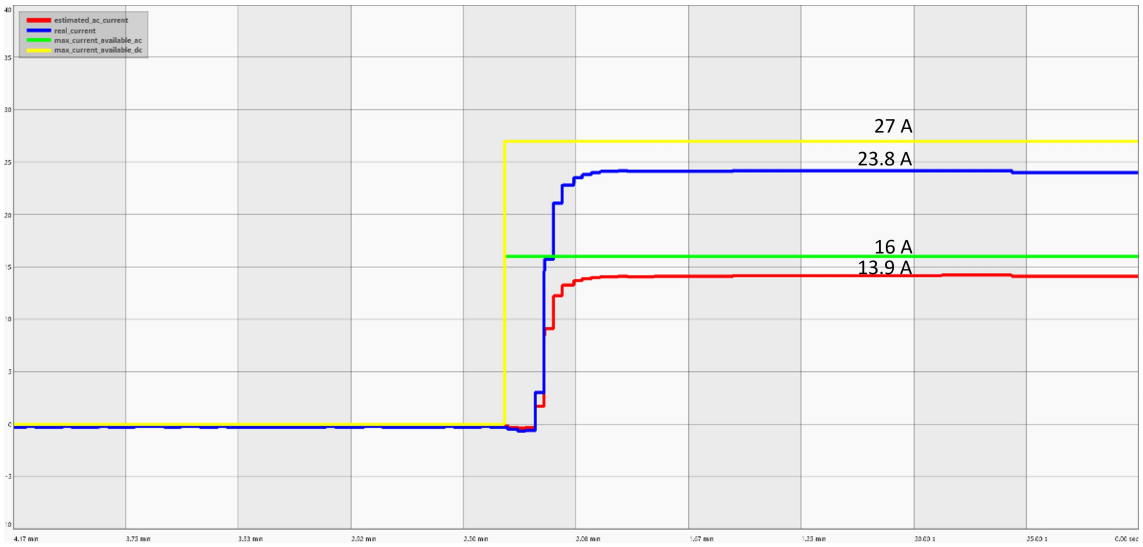


Figure 4.35: Evolution in time of the charging current in the beginning of the charging procedure, when Type 2 plug of the external EVSE is connected.

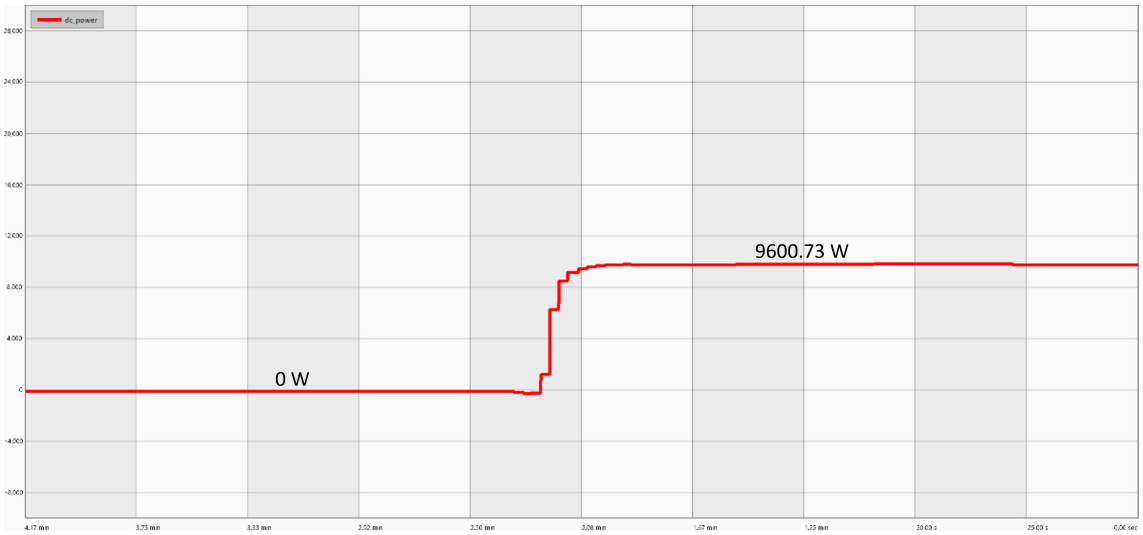


Figure 4.36: Evolution in time of the charging power in the beginning of the charging procedure, when Type 2 plug of the external EVSE is connected.

As in the previous test, the battery SoC can also be tracked throughout the whole charging procedure. The behavior of the battery SoC in the beginning of the charge can be seen in Figure 4.37.

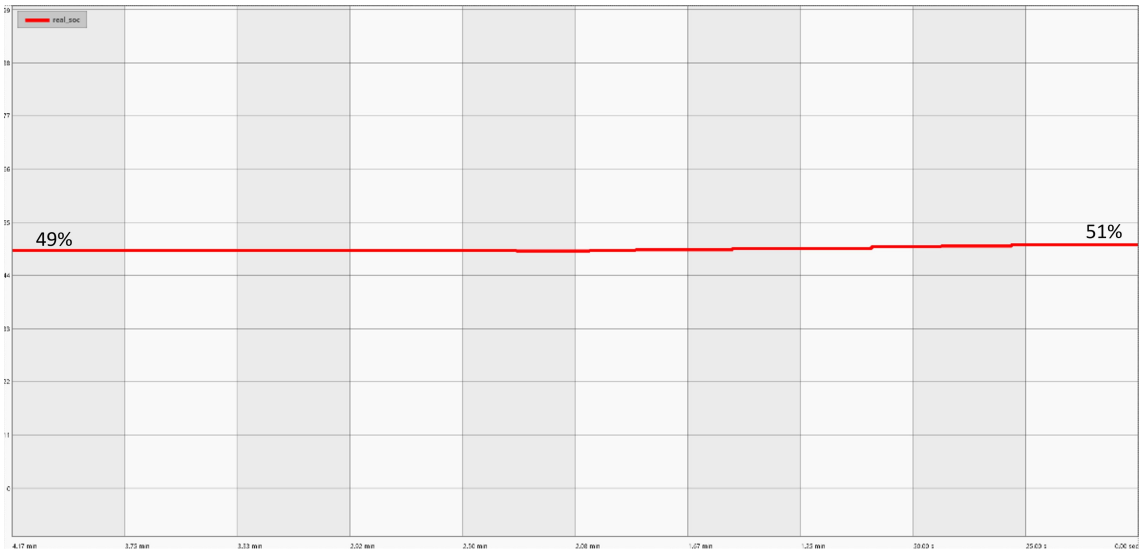


Figure 4.37: Evolution in time of the battery *State of Charge (SoC)* in the beginning of the charging procedure, when the Type 2 plug from the EVSE is connected.

In this test, differently from the tests with the SECC, the external EVSE used (as well as most of the EVSEs) does not have a stop button incorporated. Thus, charging can be interrupted only by reaching a SoC of 95% or by pulling out the charging plug. Charging interruption through pulling out the plug is possible due to the fact that, on any Type 2 inlet, not different in EVCC's Type 2 inlet, the pin for the Control Pilot line is shorter than all the other pins, including the Proximity Pilot pin. This implies that when the charging plug is partially removed, both the SECC and EVCC can detect the absence of contact on the Control Pilot pin, even with all other pins still connected. With this being considered in the FSM, when the charging plug is partially disconnected, the EVCC changes its state to 3, until the complete disconnection of the plug, when the EVCC passes back to state 0. With this behavior, the EVCC does not sense a duty cycle of 100% on Control Pilot PWM signal, instead, the sensed duty cycle changes from the variable value, corresponding to the maximum available current, straight to 0%, corresponding to an open contact state on the Control Pilot. The mentioned behavior was observed in the test and it is presented in Figures 4.38, 4.39, 4.40, 4.41 and 4.42 by the evolution in time of the EVCC's state, Control Pilot PWM duty cycle, Proximity Pilot voltage, charging current and charging power, respectively.

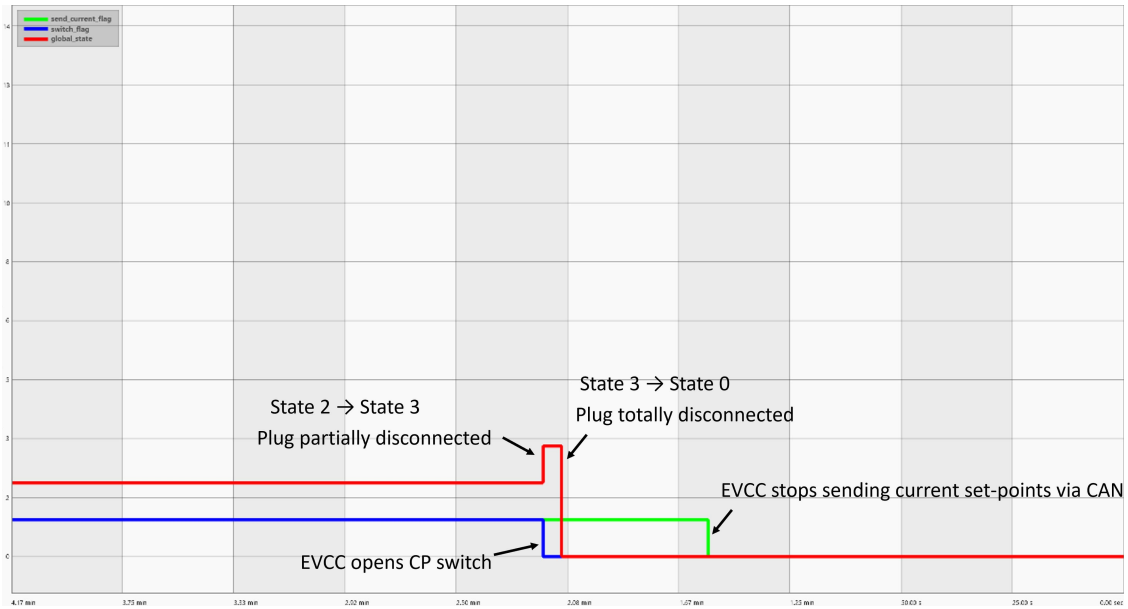


Figure 4.38: Evolution in time of EVCC's state in the end of the charging procedure, when the Type 2 plug is partially and fully disconnected.

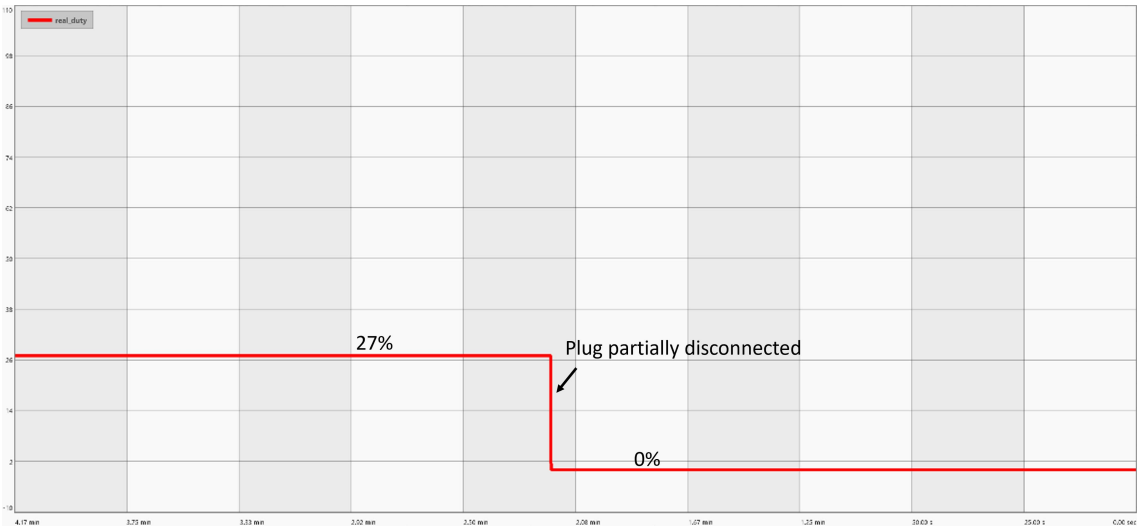


Figure 4.39: Evolution in time of Control Pilot PWM duty cycle in the end of the charging procedure, when the Type 2 plug is partially and fully disconnected.

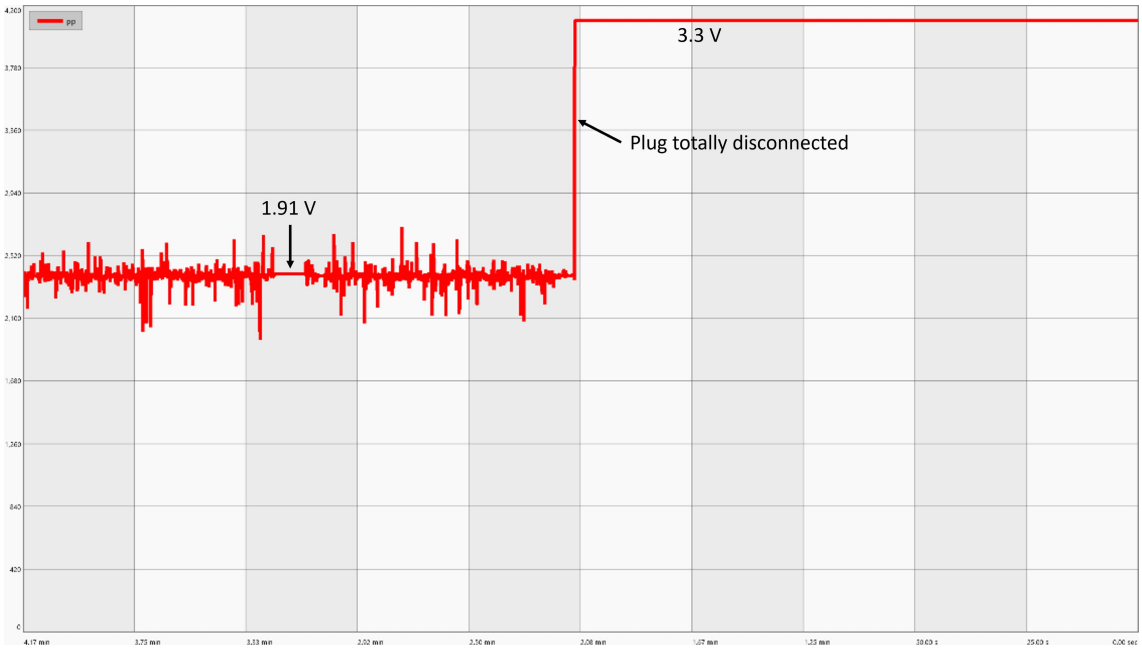


Figure 4.40: Evolution in time of Proximity Pilot voltage in the end of the charging procedure, when the Type 2 plug is partially and fully disconnected.

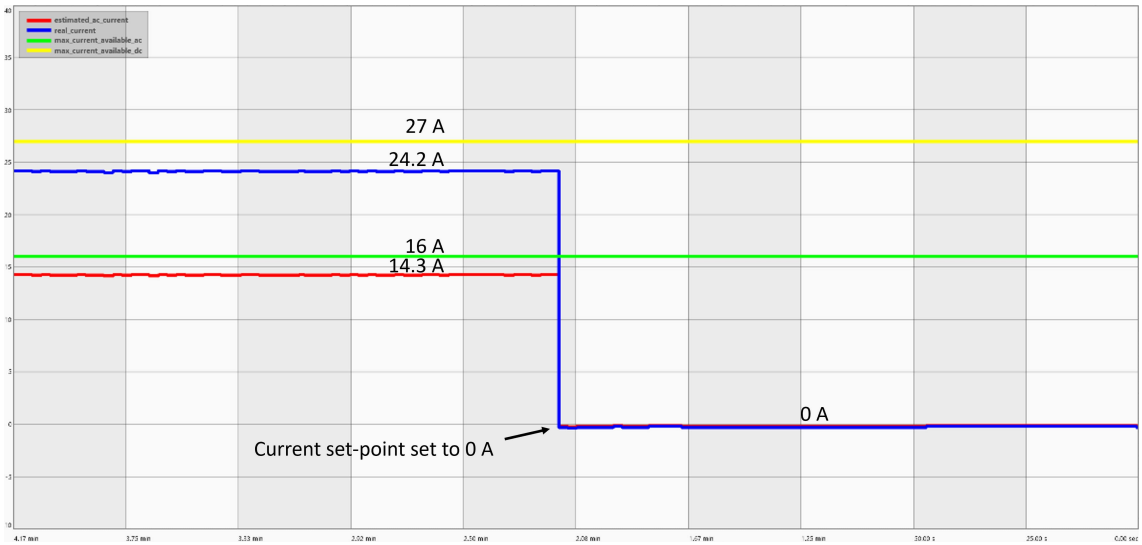


Figure 4.41: Evolution in time of the charging current in the end of the charging procedure, when the Type 2 plug is partially and fully disconnected.

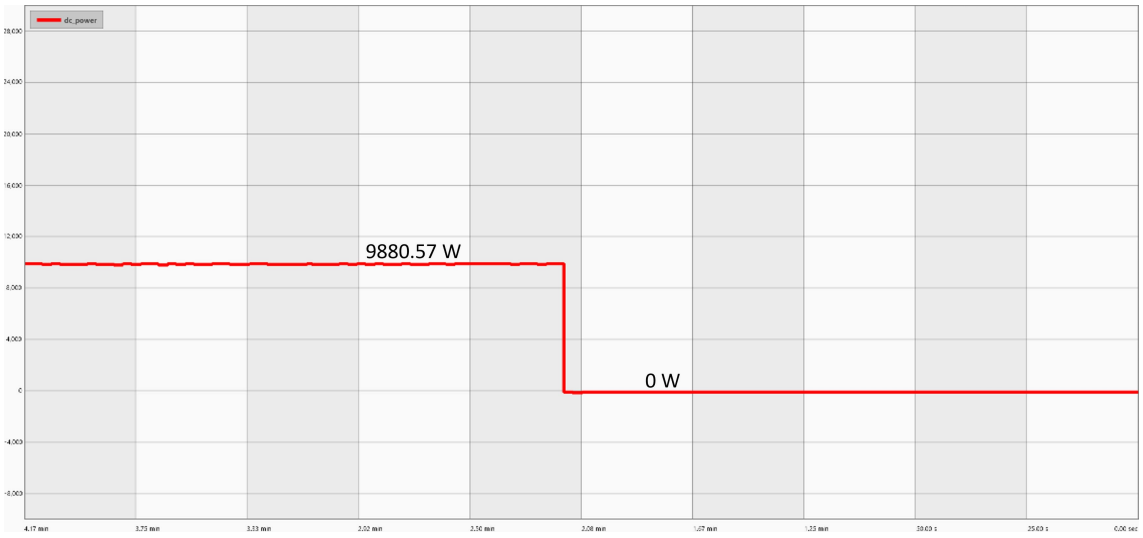


Figure 4.42: Evolution in time of the charging power in the end of the charging procedure, when the Type 2 plug is partially and fully disconnected.

As shown, the partially and fully disconnection of the plug is perfectly detectable by the EVCC. It can be noted by comparing Figures 4.39 and 4.40 and observing the time instants of the transitions of Control Pilot PWM duty cycle and Proximity Pilot voltage. When analyzing those instants in Figure 4.38, it can be seen when the plug is partially and fully disconnected, with the **send_current_flag** still "on" for some time while sending a 0 A current set-point. Besides that, when the plug is still partially disconnected, in Figures 4.41 and 4.42, it is evident the AddVolt's unit fast response to the 0A current set-point, ensuring a safe disconnection. It is due to the *stop* message being sent periodically each 100ms.

4.2.3 Test 3 - Charging the unit from another unit

Since the concept of the operation of the system is already proven by carrying out the previous tests, the purpose of this test was only to validate the Use Case of having an AddVolt unit charging another, using the proposed system. For that, the SECC and EVCC were used and their operation was monitored using the designed GUIs, the same way as the previous tests, observing the transitions in the beginning and in the end of the charging sequence. The unit connected to the SECC works as a three-phase electrical grid, while the unit connected to the EVCC is the load to be charged, in the same manner as the previous tests. As the results and the temporal tracings obtained were exactly the same as those of Test 1 and showed the same behavior, these figures were not added to the document.

4.2.3.1 SECC

In the beginning, the SECC keeps in state 0 until the Type 2 plug is connected to the inlet likewise Test 1.

When the charging plug was connected and the EVCC sensed the connection through Proximity Pilot and closed the switch on the Control Pilot voltage divider, the SECC detected the *ready to charge* state from the EVCC. This flow of states is detected through the voltage sensing on Control Pilot line, which triggers the transitions in SECC's FSM and presents the same behavior as in Test 1.

The same way as in Test 1, the actuation on the relay and contactor is done only at the *charging state*, which is the state 3 in SECC's FSM.

While the EVCC waits for a charging cable to be connected, it maintains a $+12V_{DC}$, which is equivalent to a 100% duty cycle, output on Control Pilot line. When the cable is connected, the SECC senses a voltage drop on Control Pilot line as it is shown and, when the EVCC presents a *ready to charge* state, it is sensed by the SECC as another voltage drop on the same line and then the SECC starts to output a PWM signal +/-12V with variable duty cycle on Control Pilot line.

While charging, the charging can be interrupted by either pressing the Stop Button or pulling out the Type 2 plug. The behavior of SECC is just like it was presented in Test 1.

4.2.3.2 EVCC

As in the previous tests, the EVCC remains in the state 0 of its FSM until a plug is connected to the Type 2 socket. When the plug is connected, the EVCC detects this connection by measuring the voltage at the Proximity Pilot terminal.

The presence of the plug is also detected and monitored by measuring the duty cycle of the signal on the Control Pilot line. While the EVCC is waiting for a plug to be connected, it senses a duty cycle of 0% on the Control Pilot line. When a connector is plugged in, the EVCC starts sensing a duty cycle of 100% (corresponding to $+12V_{DC}$) for a moment, and when the EVCC closes the CP_SWITCH, attesting to a *ready to charge* state, it senses a variable duty cycle corresponding to the maximum available AC output current, according to the Equation 3.13.

From the moment the EVCC reaches the *charging state*, it starts to send the *start* message and the DC current set-points via CAN to the AddVolt unit, after the conversion shown in Equation 3.18 and with the message structure shown in Table 3.2.

The same way as in previous tests, the charging procedure can be terminated by either pulling out the Type 2 plug or pressing the Stop Button. The first event causes a variation on Control Pilot PWM duty cycle from the variable value to 0% and the second event causes the variation to 100% duty cycle. Charging is interrupted by the EVCC when sending periodically a current set-point of 0A to the AddVolt unit via CAN, and a *stop* message, giving an indication to stop charging. When the plug is fully disconnected, the EVCC starts senses a duty cycle of 0% on the Control Pilot line and the Proximity Pilot voltage returns to its initial value.

4.3 Analysis of the Results

With the aforementioned tests, it was possible to validate the good functioning of the proposed system, being in accordance with the provisions of the SAE J1772 standard and with its concept

validated through the test carried out with an external EVSE. The system behaved well and the response times were within the acceptable range provided for by the standard in question [5].

A detail to be mentioned is that, in the measurement of the Proximity Pilot voltage, a lot of noise was obtained due to external interference, which may affect the threshold values to be determined in the transitions of states of the EVCC's FSM.

Finally, in parallel with the tests to validate the transitions in the charging sequence, charging was also carried out at short intervals, in order to calculate the charging power and verify that the current set-points followed were in accordance with the appropriate power. For this, 5 short charging intervals were used, measuring the variation of the SoC in a given time interval and, with the voltage and current data, the expected charging power was calculated and compared with the actual charging power that was accomplished. The expression used to calculate the power is as follows:

$$Calculated\ DC\ Charging\ Power\ (W) = \frac{\Delta Energy}{time} = \frac{\frac{\Delta SoC(\%)}{100} E}{\frac{m}{60}} \tag{4.1}$$

with $\Delta SoC(\%)$ being the variation in the SoC of the battery pack in the referred interval, E being the capacity of the battery pack, in this case, approximately 19kWh, and m being the charging duration in minutes, since no charging interval took more than 23 minutes.

Therefore, after the tests, the calculated power of the loads, compared with the expected power, is summarized and displayed in the Table 4.1.

Table 4.1: Calculated DC Charging Power for different charging intervals.

Initial SoC (%)	**Final SoC (%)**	**ΔSoC (%)**	**Charging Duration (minutes)**	**DC Current (A)**	**DC Voltage (V)**	**DC Power (W)**	**Calculated DC Power (W)**
24.08	34.04	9.96	17.5	17.3	391.1	6867	6501
34.35	39.29	4.94	8	17.2	403.9	6948	7040
39.29	44.59	5.30	7	24.1	405.8	9780	8631
44.59	49.22	4.63	5.5	24.1	404.4	9745	9597
49.14	59.18	10.04	12	23.8	397.7	9600	9538

Chapter 5

Conclusion

5.1 Achievement of Objectives

The work developed within the scope of this dissertation was based on the study and analysis of concepts related to EV Charging, namely the main standards and definitions that govern this domain.

As a result of this analysis, it was established as an objective in carrying out this work the development of a Charge Controller that would allow the AddVolt unit to be recharged in any compatible EVSE. The first and main objective established in carrying out the work was that the developed system would provide AC Charging, following the indications presented in the SAE J1772 standard for charging of Electric Vehicles due to its interoperability with most devices on the market.

It was also established that both sides of the charging procedure should be developed. For this, the SECC and the EVCC were proposed, each one having its PCB with its subsystems commanded by what was established in the Firmware of each one.

Based on the results of the tests carried out, the concept of the operation of the Charge Controllers was proved by being possible to charge the AddVolt unit through an external EVSE, following the SAE J1772 standard.

5.2 Future Work

For the work to be carried out afterwards, it can be highlited:

- Update/evolve the implementation into Charging Controllers intended for DC Charging;
- Update the Proximity Pilot Interface, with a solution that scales the voltage value in the 0V to 5V range to the 0V to 3.3V range, without having to resort to the saturation of the Operational Amplifier in the buffer configuration, since the implementation performed only allows the EVCC to connect to cables with a maximum current of 32A;

- Implement charging interruption when the charging current exceeds cable current capacity indicated by PP resistance;
- Change PCB connectors with more robust options with better fixation;
- Improve the immunity of systems against noise, wrapping them in metal boxes and using shielded cables;
- Add filters to the inputs of the ADCs, analog or digital, in order to have a "cleaner" measurement and suppress the noise that may appear.
- Implement the functionality of Insulation Measurement, already planned to be added, in both hardware and software;
- Add an Interlock System to the EVCC, in order to improve the security and reliability of the system;
- Run tests and prove the concept by charging an EV;
- Update the Power Supply Block, replacing the oversized converters with lower consumption converters, in order to improve the efficiency of the systems.
- Adapt the solution to be controlled by a simpler microcontroller, but with similar functionalities, in order to reduce costs.

Appendix A

Controller Area Network (CAN)

CAN is a message-based communication protocol. The initial proposal was to allow a vehicle's devices, such as the Electronic Control Unit (ECU), to communicate with each other in a robust and reliable way, with priority assignments. The main advantage is the reduction of cabling in the connections, since all the devices in the network are connected to the same bus. These devices, working as *nodes* of the network, receive messages or *frames* from all devices connected to this network, which in turn do not need a local host computer. A schematic of a node connected to a CAN bus is shown in the Figure A.1 [18].

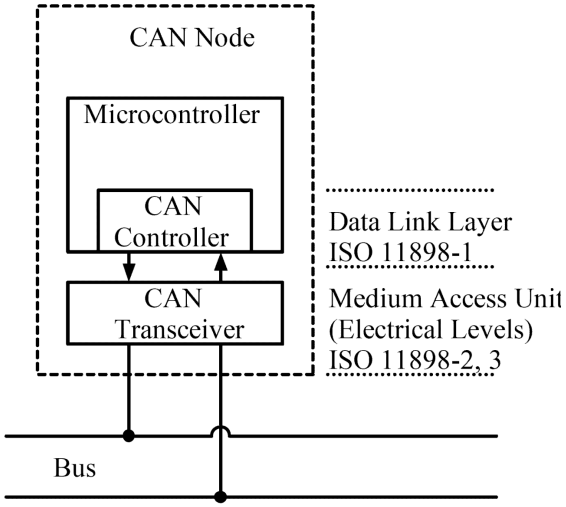


Figure A.1: CAN node connection to CAN bus network schematic.

Each CAN node in the network is composed of a Central Processing Unit (CPU), a CAN Controller and a Transceiver, which in turn adapts the signal levels of both data sent and received by the node. All nodes belonging to the network can send and receive data, however, not at the same time [18].

Nodes belonging to the network cannot send data directly to another node on the network. Instead, the nodes broadcast the data throughout the network, making the data available to any node on the network. However, only the nodes to which the data has been addressed will receive

and process this data. The CAN protocol is error-proof by taking advantage of a bitwise arbitration method to resolve contention on the bus. The nodes are all synchronized, so they all sample the data on the network with the same frequency. However, the system is not clock synchronized, that is, although these samplings occur at the same frequency, they do not occur simultaneously [18].

In CAN protocol, there are four types of frames sent to the bus:

- **Data frames** to transfer data to one or many receiver nodes;
- **Remote frames** to ask data from other nodes;
- **Error frames** to report errors;
- **Overload frames** to report overload conditions.

CAN messages can be sent with two possible lengths, those are standard and extended. The big difference is found in the size of the arbitration field, having in the extended length 18 additional bits for attributing identifiers, which increases the number of possible combinations of messages to be sent on the bus. An illustration of data frames is shown in the Figure A.2 and the architecture of the message structure is presented in Table A.1 [18].

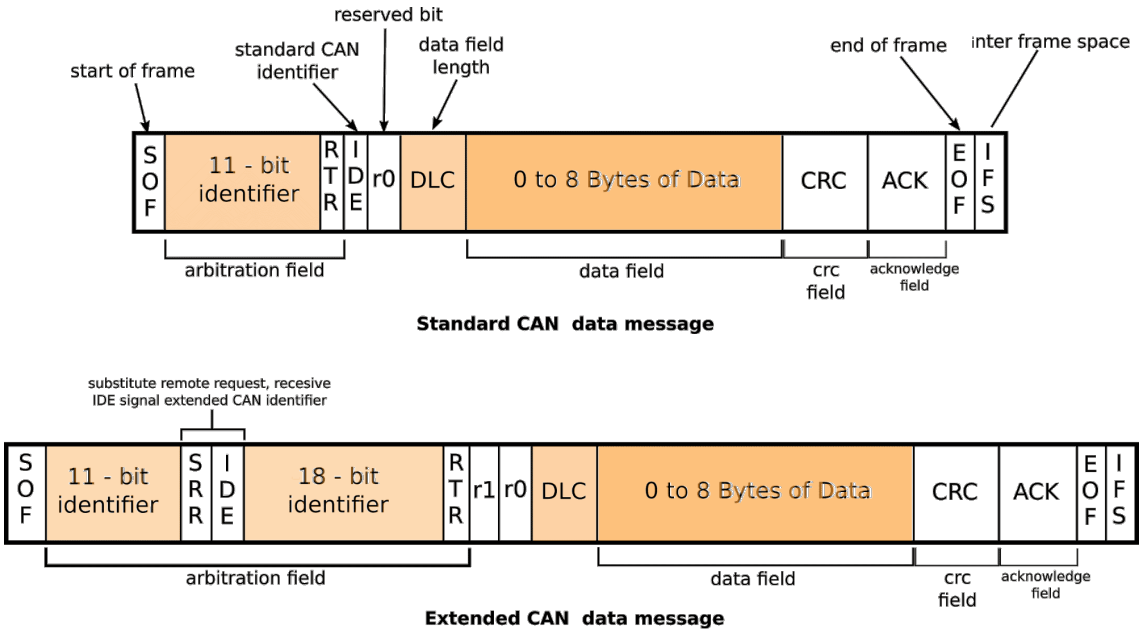


Figure A.2: CAN data message structure for standard and extended frame [18].

Table A.1: Bit fields of CAN data transmit messages [18].

Field	**Bits**	**Description**
SOF	1	*Start Of Frame bit*
Identifier	11	*Identifier data field*
RTR	1	*Remote Transmission Request*
IDE	1	*Identifier Extension bit*
R0	1	*Reserved for future use*
DLC	4	*Data Length Code*
Data	0 - 64	*Data Bytes*
CRC	16	*Cyclic Redundancy Check*
ACK	2	*Acknowledge bit*
EOF	7	*End Of Frame*
IFS	3+	*Inter Frame Space*

The arbitration field consists of the message identifier and the Remote Transmission Request (RTR) bit. The greater the importance of the message, the smaller its identifier, in order to guarantee the required priority in sending and receiving it [18].

If more than one node tries to transmit a message simultaneously, an arbitration is initiated. The message with the lowest identifier number grants its referred node priority. Dominant bits overwrite recessive bits on the CAN bus [18].

The CAN message identifier can be 11 bits long, representing Standard CAN, with 2048 possible message identifiers, and also 29 bits long, representing Extended CAN, with approximately 537 million possible message identifiers. The RTR bit is dominant and indicates that data is being transmitted [18].

In most digital systems, a logical 1 represents a high level and a logical 0 represents a low level. However, in CAN it is exactly the opposite, as CAN transceivers generally use a pull-up on their transmit and receive interfaces, thus having the devices configured for a recessive state of the bus. The CAN data transmission signaling is presented in Figure A.3 [18].

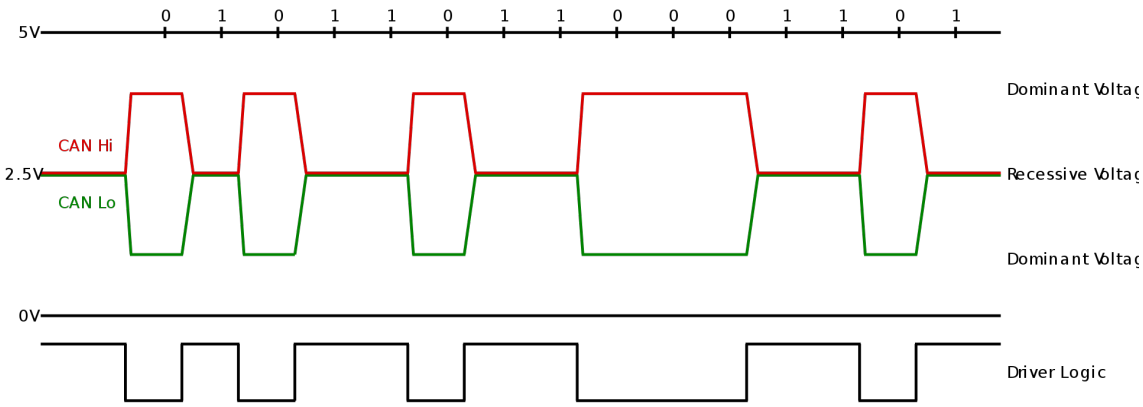


Figure A.3: CAN bus signaling logic [19].

All nodes are connected to each other through a two-wire bus and these wires have terminating resistances of 120 Ohms at their ends [19].

Since the bus has a differential data transfer (CAN_H & CAN_L), communication via CAN becomes noise-immune, since an eventual presence of electromagnetic noise is also present on both wires and, therefore, it is subtracted and in turn suppressed. The CAN connection topology is presented in Figure A.4 [19].

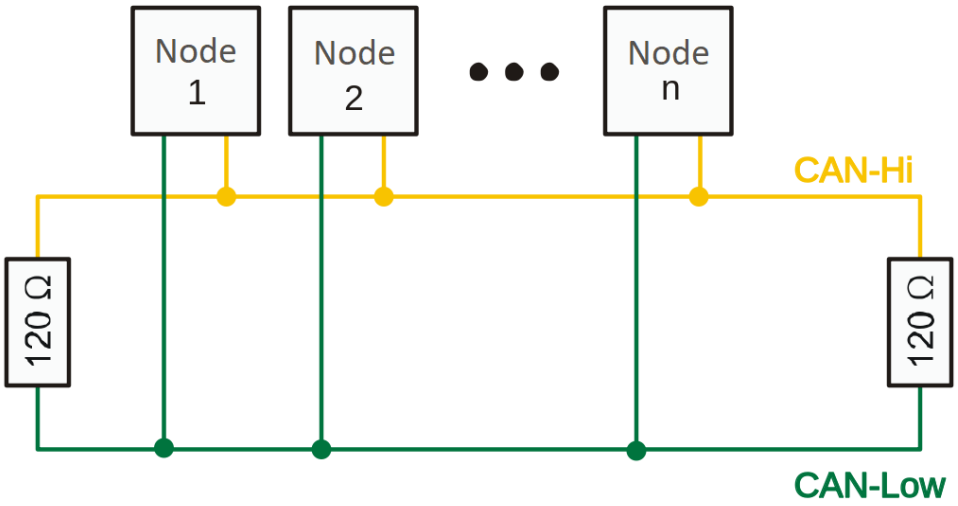


Figure A.4: CAN bus electrical topology [19].

Appendix B

IEEE 754

The IEEE Standard for Floating-Point Arithmetic (IEEE 754) was proposed and established in 1985 by the Institute of Electrical and Electronic Engineers (IEEE). The idea behind this standard was to provide an accurate representation of floating point data on machines. The IEEE 754 format consists of a set of representations of numerical values and symbols [50]. The IEEE 754 standard is the most common representation for real numbers in computers and microcontrollers of different architectures [20].

The IEEE 754 format has three basic components:

- **Sign of number:** Identification bit which describes if the number is positive or negative;
- **Exponent with bias:** A biased binary number to represent the exponent of the number being represented;
- **Mantissa:** The represented number in scientific notation, with the significant digits.

It is to be noted that as the representation is in computer logic, all the digits are bits, that is, binary digits that could be only "0" or "1". The numbers in IEEE 754 representation are divided in two possible representations: Single Precision or Double Precision. Illustrations of those precision options are presented in Figures B.1 and B.2, respectively [17][20].

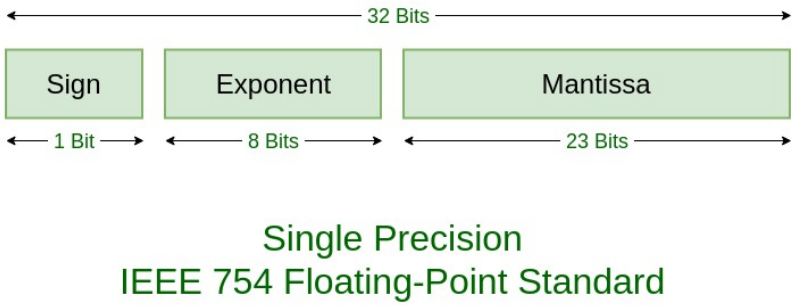


Figure B.1: Representation of floating-point numbers in IEEE 754 standard for Single Precision [20].

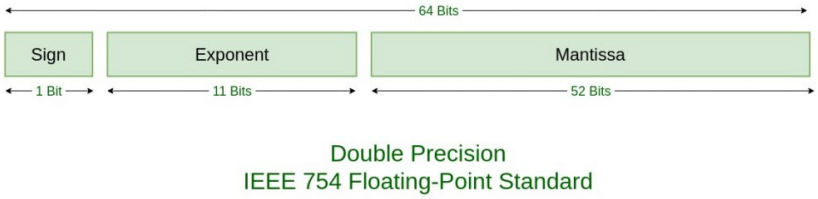


Figure B.2: Representation of floating-point numbers in IEEE 754 standard for Double Precision [20].

As shown, the difference is existence of extra bits for the representation of the Mantissa and the Exponent. An example of the IEEE 754 Single Precision representation for the number 257.75 is shown in Table B.1.

Table B.1: Representation of the number 257.75 in IEEE 754 standard for Single Precision.

MSB																												**LSB**
Sign bit	**Exponent (8 bits)**								**Mantissa (23 bits)**																			
0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0 = Positive 1 = Negative	Decimal value: 257.75								Hexadecimal value: 4380E000																			

Appendix C

Circuit Schematics

C.1 SECC Circuit CAD Schematic

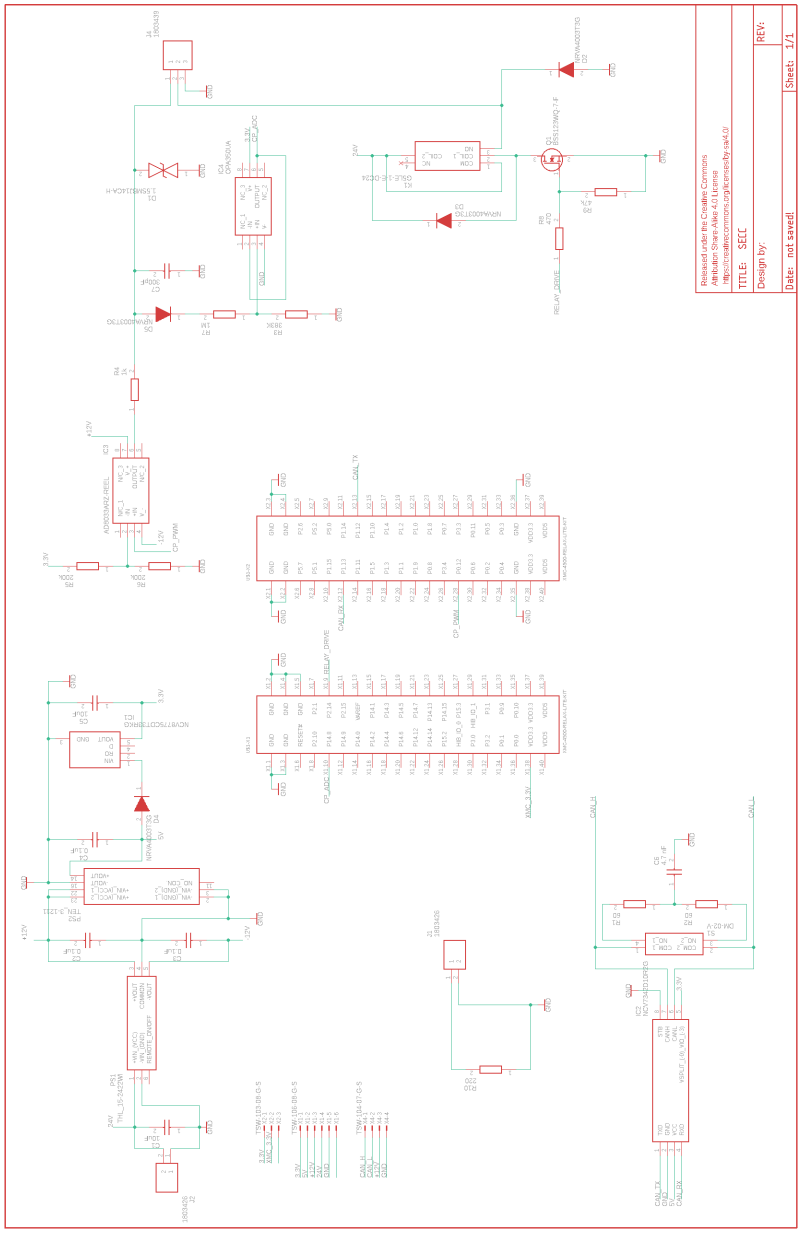


Figure C.1: Final circuit schematic design of SECC in CAD software.

C.2 EVCC Circuit CAD Schematic

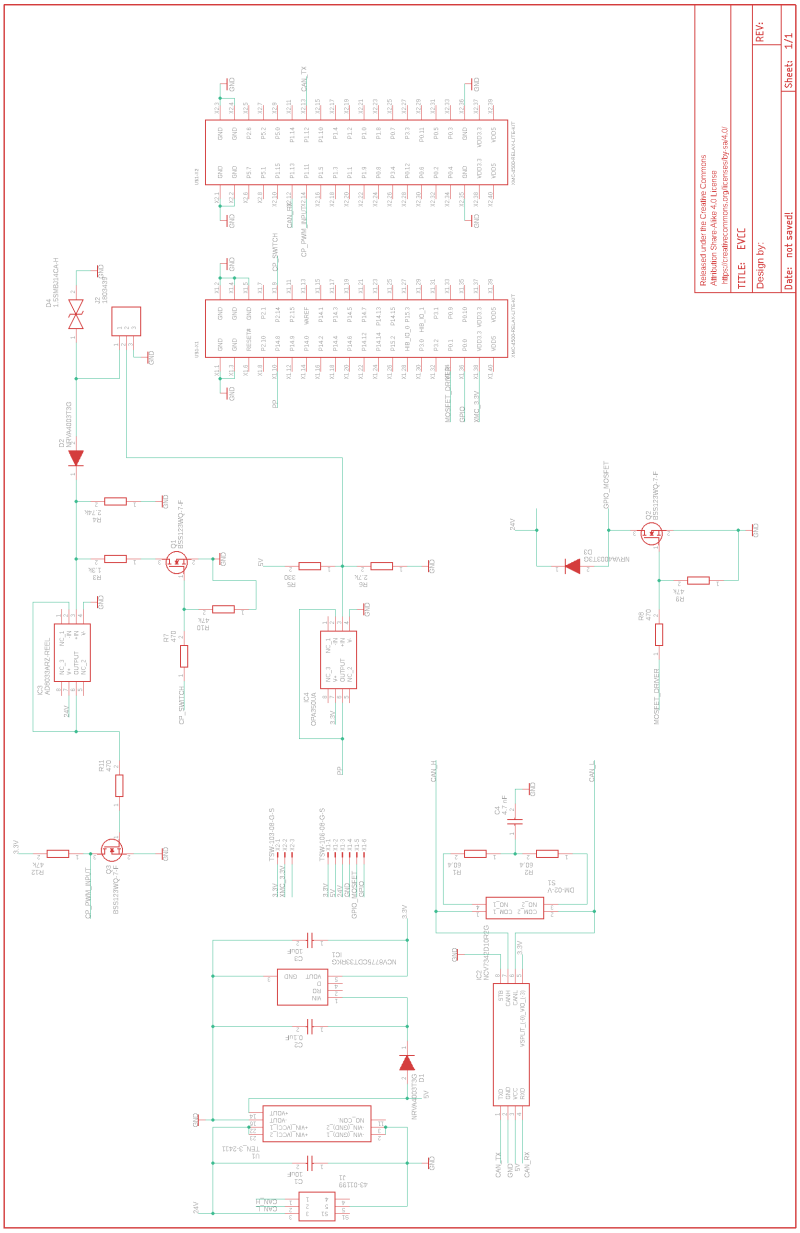


Figure C.2: Final circuit schematic design of EVCC in CAD software.

References

[1] Pavol Bauer Gautham Mouli Auke Hoekstra, Marnix Wagemaker. DelftX eCARS2 - Lecture Notes - Electric Cars: Tecnology, February 2022.

[2] Ali Bahrami. *EV Charging Definitions, Modes, Levels, Communication Protocols and Applied Standards*, 01 2020. doi:10.13140/RG.2.2.15844.53123/11.

[3] CharIN. Design Guide for Combined Charging System V7. CharIN, available in https://www.charin.global, Accessed: 2022-02-20.

[4] BlueSky Electric Company. AC Level 2 Charging Station. BlueSky Electric Company, available in https://www.blueskyelectricco.com/renewable-and-alternative-energy-1, Accessed: 2022-02-20.

[5] Hybrid EV Committee. *SAE J1772 - SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler*, oct 2017. URL: https://doi.org/10.4271/J1772_201710, doi:https://doi.org/10.4271/J1772_201710.

[6] Best EV Chargers - Overview of EV Charging Modes. https://bestchargers.eu/blog/overview-of-ev-charging-modes. Accessed: 2022-02-20.

[7] International Electrotechnical Commission. *IEC 61851-1 - Electric vehicle conductive charging system - Part 1: General requirements*, feb 2017.

[8] Wikipedia - SAE J1772. https://en.wikipedia.org/wiki/SAE_J1772. Accessed: 2022-02-20.

[9] Wikipedia - CHAdeMO. https://en.wikipedia.org/wiki/CHAdeMO. Accessed: 2022-02-20.

[10] CHAdeMO Association - EV Fast Charging Organisation. https://chademo.com/. Accessed: 2022-02-20.

[11] Takafumi Anegawa. Safety design of chademo quick charging system. *World Electric Vehicle Journal*, 4:855–859, 01 2011. doi:10.3390/wevj4040855.

[12] Chinese Standards. *GB/T 18487-1 - Electric vehicle conductive charging system – Part 1: General requirements*, dec 2015.

[13] Wu Wu Tang, Yu Ming Wu, and Jian Qin. Comparative study on electric vehicle charging standards at home and abroad. In *Progress in Renewable and Sustainable Energy*, volume 608 of *Advanced Materials Research*, pages 1553–1559. Trans Tech Publications Ltd, 2 2013. doi:10.4028/www.scientific.net/AMR.608-609.1553.

[14] Interchargers - CCS Charging Plugs. https://interchargers.com/plug/type-2. Accessed: 2022-02-20.

[15] Infineon Technologies. Kit XMC4500 Relax V1. Infineon Technologies, available in https://www.infineon.com/cms/en/product/evaluation-boards/kit_xmc45_relax_v1, Accessed: 2022-02-20.

[16] ON Semiconductor. *NCV7342D13R2G High Speed Low Power CAN Transceiver*, 6 2016. Rev. 4.

[17] Infineon Technologies. *Capture Compare Unit 4 (CCU4) - XMC1000, XMC4000 32-bit microcontroller series for industrial applications*, February 2016. Rev. 1.1.

[18] Grant Maloy Smith. What Is CAN Bus (Controller Area Network) and How It Compares to Other Vehicle Bus Networks, Feb 2021. DEWESoft, accessed: Jun-2022. URL: https://dewesoft.com/daq/what-is-can-bus.

[19] Wikipedia. CAN bus — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=CAN%20bus&oldid=1084382983, 2022. [Online; accessed 15-June-2022].

[20] IEEE standard 754 Floating point numbers, Mar 2020. URL: https://www.geeksforgeeks.org/ieee-standard-754-floating-point-numbers/.

[21] International Electrotechnical Commission. *IEC 62196-1 - Plugs, socket-outlets, vehicle connectors and vehicle inlets - Conductive charging of electric vehicles - Part 1: General requirements*, jun 2014.

[22] International Organization for Standardization. *ISO 15118-1 - Road vehicles — Vehicle to grid communication interface — Part 1: General information and use-case definition*, apr 2019.

[23] International Electrotechnical Commission. *IEC 62196-2 - Plugs, socket-outlets, vehicle connectors and vehicle inlets - Conductive charging of electric vehicles - Part 2: Dimensional compatibility and interchangeability requirements for a.c. pin and contact-tube accessories*, feb 2016.

[24] International Electrotechnical Commission. *IEC 62196-3 - Plugs, socket-outlets, vehicle connectors and vehicle inlets - Conductive charging of electric vehicles - Part 3: Dimensional compatibility and interchangeability requirements for d.c. and a.c./d.c. pin and contact-tube vehicle couplers*, jun 2014.

[25] International Electrotechnical Commission. *IEC 61851-21-1 - Electric vehicle conductive charging system - Part 21-1 Electric vehicle on-board charger EMC requirements for conductive connection to AC/DC supply*, jun 2017.

[26] International Electrotechnical Commission. *IEC 61851-21-2 - Electric vehicle conductive charging system - Part 21-2: Electric vehicle requirements for conductive connection to an AC/DC supply - EMC requirements for off board electric vehicle charging systems*, apr 2018.

[27] International Electrotechnical Commission. *IEC 61851-23 - Electric vehicle conductive charging system - Part 23: DC electric vehicle charging station*, mar 2014.

[28] International Organization for Standardization. *ISO 15118-2 - Road vehicles — Vehicle-to-Grid Communication Interface — Part 2: Network and application protocol requirements*, apr 2014.

[29] International Organization for Standardization. *ISO 15118-2 - Road vehicles — Vehicle to grid communication interface — Part 3: Physical and data link layer requirements*, may 2015.

[30] European Standards. *DIN SPEC 70121 - Electromobility - Digital communication between a d.c. EV charging station and an electric vehicle for control of d.c. charging in the Combined Charging System*, dec 2014.

[31] Wikipedia - Combined Charging System. https://en.wikipedia.org/wiki/Combined_Charging_System. Accessed: 2022-02-20.

[32] Jason D. Harper. Development and implementation of sae dc charging digital communication for plug-in electric vehicle dc charging. In *SAE 2013 World Congress Exhibition*. SAE International, apr 2013. URL: https://doi.org/10.4271/2013-01-1188, doi:https://doi.org/10.4271/2013-01-1188.

[33] Wikipedia. Tesla Supercharger. Wikipedia, available in https://en.wikipedia.org/wiki/Tesla_Supercharger, Accessed: 2022-02-20.

[34] Mean Well. *WDR-60-24 60W Ultra Wide Input Industrial DIN Rail Power Supply*, 8 2019. Rev. 1.

[35] Bender. *iso165C Insulation Monitoring Device (IMD) for unearthed DC drive systems (IT systems) in electric vehicles*, 10 2019. Rev. 1.

[36] TRACO Power. *THL 15-2422WI DC/DC Converter THL 15WI Series, 15 Watt*, 12 2021. Rev. 1.

[37] TRACO Power. *TEN 3-1211 DC/DC Converter TEN 3 Series, 3 Watt*.

[38] ON Semiconductor. *NCV8775CDT33RKG Ultra Low Iq 350 mA LDO Regulator with Reset*, 12 2019. Rev. 1.

[39] Schneider Electric. *LC1D32BL TeSys D contactor - 3P(3 NO) - AC-3 - <= 440 V 32 A - 24 V DC coil*, 1 2021. Rev. 1.

[40] Omron Electronics. *G5LE-1-E-DC24 Cubic, Single-pole 10A Power Relay*.

[41] Diodes Incorporated. *BSS123WQ-7-F N-CHANNEL ENHANCEMENT MODE MOSFET*, 9 2014. Rev. 2.

[42] onsemi. *Surface Mount Standard Recovery Power Rectifier SMA Power Surface Mount Package MRA4003T3G Series, NRVA4003T3G Series*, 8 2021. Rev. 12.

[43] Analog Devices. *AD8033ARZ-REEL Low Cost, 80 MHz FastFET Op Amps*.

[44] Texas Instruments. *OPAx350 High-Speed, Single-Supply, Rail-to-Rail Operational Amplifiers MicroAmplifier Series*, 9 2000. Rev. 12-2015.

[45] Bourns. *1.5SMBJ Transient Voltage Suppressor Diode Series*.

[46] TRACO Power. *TEN 3-2411 DC/DC Converter TEN 3 Series, 3 Watt*.

[47] Wikipedia. DAVE (Infineon). Wikipedia, available in https://en.wikipedia.org/wiki/DAvE_(Infineon), Accessed: 2022-06-06.

[48] Infineon μC/Probe by Micrium. https://infineoncommunity.com/uc-probe-xmc-software-download_id712. Accessed: 2022-06-27.

[49] Mode 2 portable EV Charger, Nov 2018. URL: https://www.khonsevse.com/mode-2-portable-ev-charger/.

[50] IEEE Standard for Floating-Point Arithmetic. *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pages 1–84, 2019. doi:10.1109/IEEESTD.2019.8766229.