POLITECNICO DI TORINO

Master's Degree in Mechatronic engineering

A.Y. 2022/2023

Retrofit Kit: applications of CANbus to Virtual - Semi Real - Real Prototyping

Supervisor:

Prof. Stefano Carabelli

Tutor:

Ing. Mohammad Taffal

Candidate:

Stefano Delle Donne

Index

List of Figures

Abstract

Electric mobility has been identified as a promising solution for reducing CO2 emissions in the transport sector. However, the high cost of purchasing a new electric vehicle (EV) may be prohibitive for many people. Retrofitting a traditional vehicle with an electric conversion kit can be a more affordable way to transition to electric mobility. A retrofit kit typically includes an electric motor, a battery pack, a controller, a charger, and other necessary components. By replacing the internal combustion engine with an electric motor, it allows the vehicle to run on electricity, reducing greenhouse gas emissions and air pollution. Moreover, retrofitting an existing vehicle is a sustainable solution that can extend the life of the vehicle and save resources and energy.

The integration of the retrofit kit into the vehicle's systems must be carefully planned and executed to ensure safety, reliability, and optimal performance: software development methodologies, such as the extended V-cycle (Figure 1) and CAN network integration, can provide a structured approach to developing and integrating software components into the retrofit kit and the vehicle's existing systems.

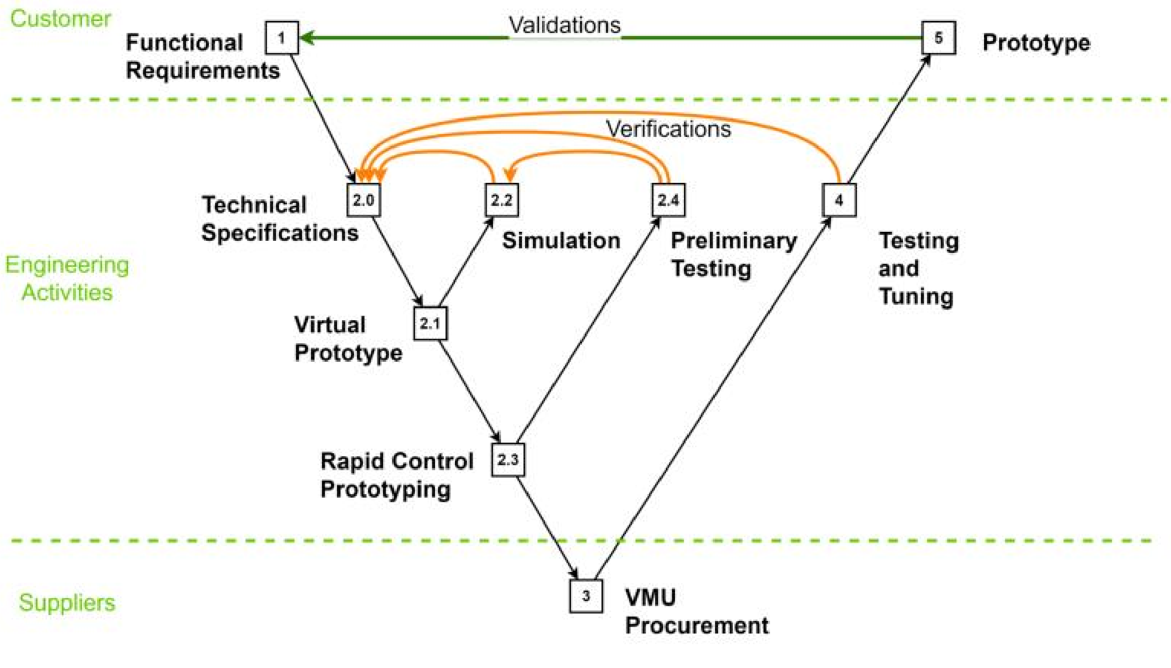

Figure 1. Extended V-Cycle

The thesis project therefore aims at the use of the Extendend V-Cycle and the CAN network, applied to the three main phases of the project: Virtual, Semi-Real and Real Prototype. The Extended V-Cycle and the CAN network can be used together to improve the vehicle software development process.

The creation of the Virtual Prototype requires the creation of a Control Logic module that communicates with the Environment, the Plant, the HMI and the User. This is a purely simulation phase, since it replicates reality numerically, through a movement of states. The CAN network can be used to allow communication between these components, ensuring greater integration and synchronization between the various systems.

The Semi-Real Prototype, i.e. the bench, has a dual function: it ensures that the results of the virtual model are compliant and at the same time emulates the real model, i.e. it replicates the weight of the vehicle and adds elements in power. The CAN network therefore becomes a vitally important tool, as it allows detecting any problems and at the same time being able to carry out analyzes on the results obtained.

Finally, in the Real Prototype, the CAN network can be used for the transmission of road test data, allowing technicians to analyze and interpret the data to evaluate the vehicle's performance in real conditions. Through suitable CANanalyzers it is also possible to monitor in real time critical parameters of a retrofit kit, such as the battery temperature.

In summary, the use of the CAN network can improve the Extended V-Cycle through reliable communication between various vehicle components, diagnostics and troubleshooting during testing, and transmission of road test data for vehicle performance analysis. This allows for greater integration and synchronization between the various systems, increasing the quality and predictability of the software development process.

1. Digital Triplet Overview

A "Digital Twin" can be defined as "the creation of a virtual equivalent of a physical product".

The necessary conditions for the creation of a Digital Twin are the existence of:

- A physical product in real space
- A virtual product in the virtual space
- A connection system of data and informations flows that joins the physical space to the virtual one and to the virtual

Speaking of Digital Twin, therefore, we should refer to the Real Prototype and the Virtual Prototype.

The transition from Digital Twin to Digital Triplet takes place thanks to the presence of a third twin: the Semi-real Prototype, i.e. the whole electric powertrain on the Test Bench.

The ultimate goal is to make sure that these three prototypes are as faithful as possible to each other.

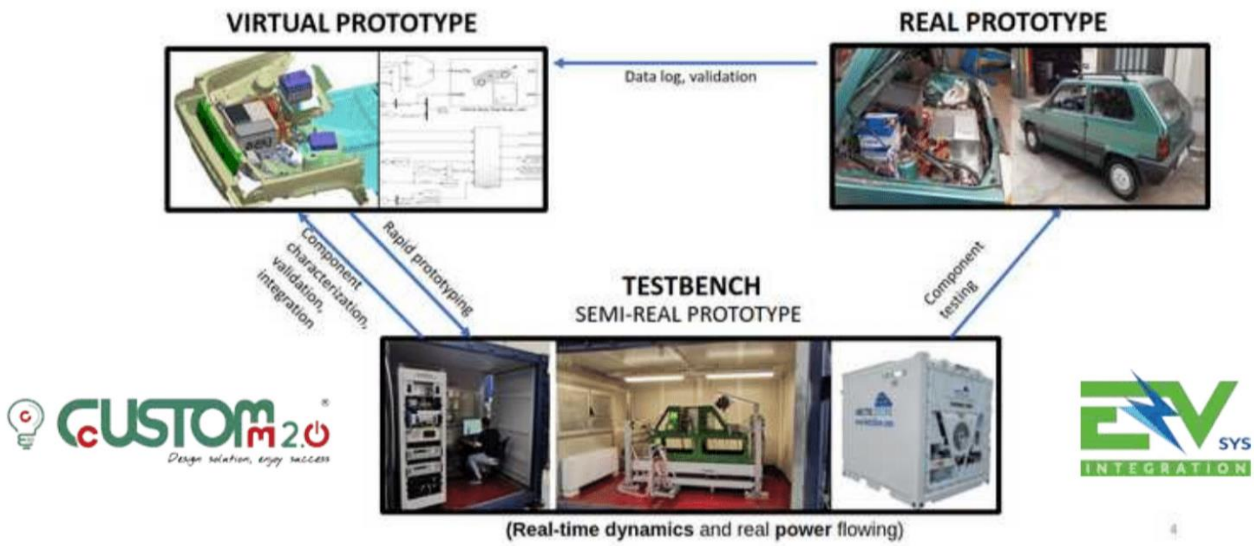

Figure 2. Extended V-Cycle with focus on Prototypes.

The whole theoretical and applicational process is represented in Figure 2. The directions of the arrows between the three prototypes assume fundamental importance.

It is therefore important to analyze the dynamics that exist between the three prototypes. Referring to the figure, it can be established that:

- The Virtual Prototype is characterized through the behavior of the Semi-Real prototype. By performing bench tests, the virtual model is modeled on the basis of the results obtained.
- With the term "Data log, validation" we mean the transition from the Real Prototype to the Virtual Prototype, giving an estimate of how effectively they can be defined as "twin".

2. Controller Area Network Overview

To better understand some details of the project, it is first necessary to explain some basic concepts of CAN communication. It is important to divide the context into three main groups:

2.1 Physical Layer

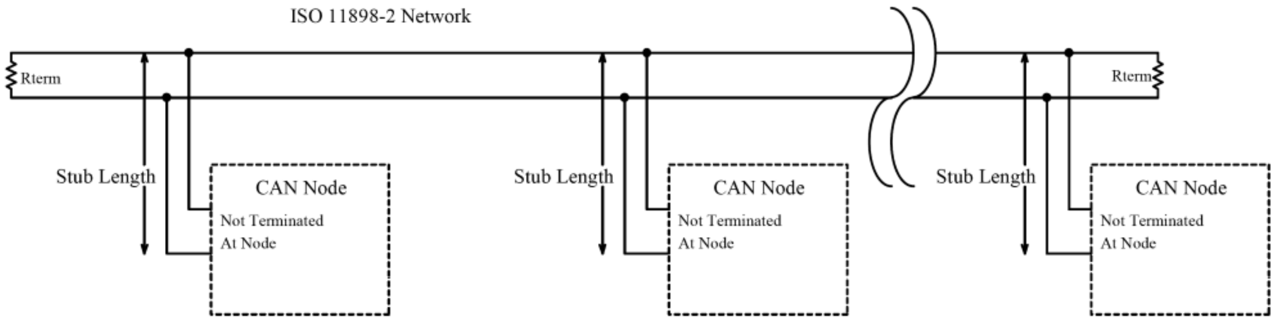

Figure 3. Can Network Physical Layer

Figure 3 shows the generic diagram of the CAN physical network. The main elements are:

- Nodes of the network, therefore all the components capable of receiving/transmitting messages
- Two wires, representing the CanH (High Voltage) and CanL (Low Voltage).
- 120 Ohm resistor to terminate the line.

It is necessary to focus now on the aspect regarding CanH and CanL

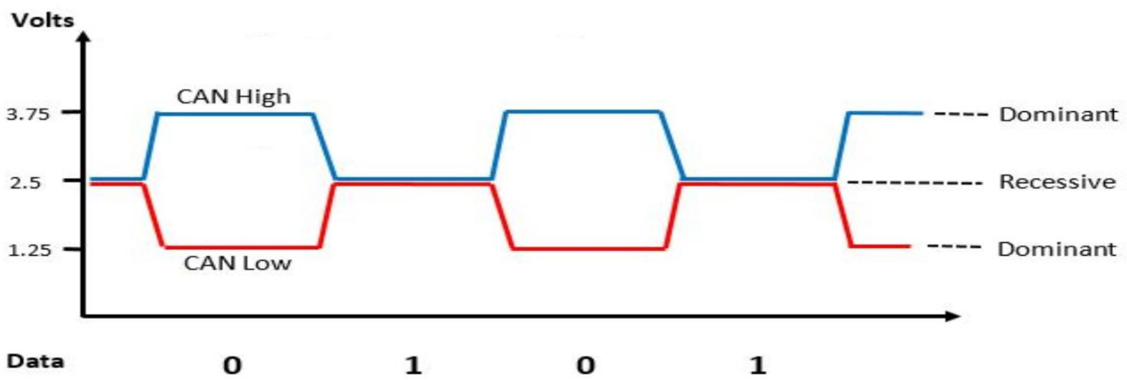

Figure 4. Can High and Low voltage levels.

In Figure 4 the wire called "Can High Voltage" is shown in blue line, the "Can Low Voltage" in red.

Their primary function is to create a differential system, based on the voltage difference between the two, regardless of the system's ground connection. In this way, in the event of systems highly affected by electrical interference, the communication network will not be affected by them, ensuring greater stability and reliability in terms of performance and results.

Furthermore, through this differential system, the Bus can have two states, Dominant or Recessive, which will correspond to two logical levels, respectively 0 and 1.

CAN High Value	Can Low Value	Bus State	Driver Logic (Data)
3.75 Volt	1.25 Volt	Dominant	0
2.5 Volt	2.5 Volt	Recessive	1

Table 1

The CAN network is made up of a set of nodes. It suffices that only one node is in the dominant level, for the Bus (the set of all nodes) to be in the dominant level.

Node 1	Node 2	Node 3	Bus
D	D	D	D
D	D	r	D
D	r	D	D
D	r	r	D
r	D	D	D
r	D	r	D
r	r	D	D
r	r	r	r

Table 2

- D = Dominant Level. CanH and CanL different
- r = Recessive Level. CanH = CanL

2.2 Data Link Layer

Figure 5 shows all the blocks constituting a message. It is important to pay attention to the blocks circled in red, in particular to the Identifier block.

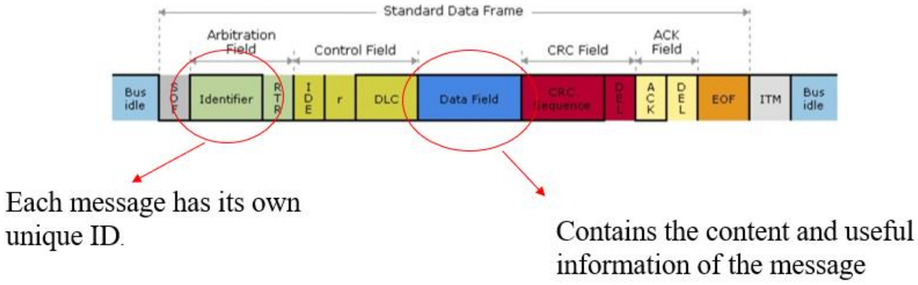

Figure 5. Data Link Layer

The arbitration system can be divided into two cases:

- Carrier Sens Multiple Access (Figure 6)

It is based on the recessive or dominant state of the nodes (described in the Physical Layer). When there are multiple nodes that simultaneously want to send messages,the node which will win the "competition" and which will therefore have the priority to write messages on the Bus, is the one with dominant states even when all the other states are recessive.

Here is an example of a possible arbitration on the Bus, applied to our project.

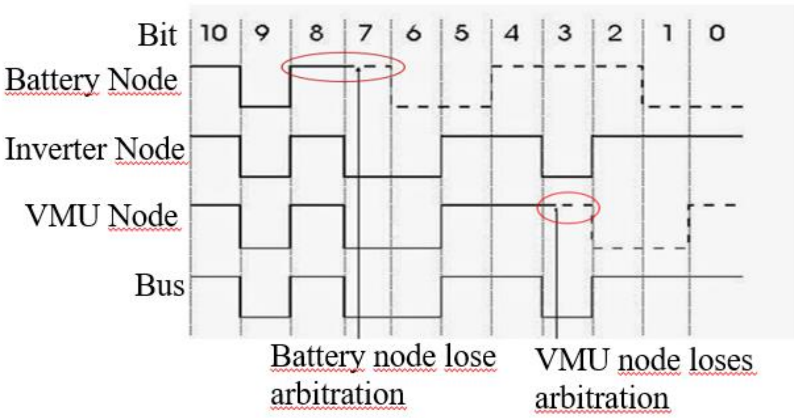

Figure 6. Example of Arbitration using CSMA

These three nodes represent the nodes belonging to all three prototypes.

- AMP (Arbitration by Message Priority)

Once a node has won the competition and is therefore allowed to publish messages on the network, the most important message to publish is decided based on the ID. The one showing the lowest ID will take precedence.

Here is an example of the practical results obtained, through the logging of the Virtual Prototype and the use of CANexplorer.

Message Table

Time	ID	Message	Length	Data
0.179530	180	SME_Slave_TPDO_1	8	06 00 C0 03 A6 02 78 00
0.172389	480	SME_Slave_TPDO_4	8	17 11 7C 0F 8D 6E 01 00
0.167287	6B1	VMU_TPDO_4	8	00 09 04 59 02 01 00 00
0.157273	200	VMU_TPDO_1	8	08 00 00 00 00 00 00 00
0.149171	300	VMU_TPDO_2	8	20 1C E0 24 0D 0E 00 00
0.136448	18F124F4x	BMS_VCU1	8	00 00 5D 00 3F 7D DE 39
0.086180	18F124F4x	BMS_VCU1	8	00 00 5D 00 3D 7D DE 39
0.080840	6A0	SME_Slave_TPDO_6	8	00 00 00 00 00 00 14 00
0.079309	180	SME_Slave_TPDO_1	8	06 00 C0 03 65 02 78 00
0.073471	480	SME_Slave_TPDO_4	8	15 11 4F 0E 8D 6E 01 00
0.067614	6B1	VMU_TPDO_4	8	00 09 04 59 02 01 00 00
0.055949	200	VMU_TPDO_1	8	08 00 00 00 00 00 00 00
0.049406	300	VMU_TPDO_2	8	20 1C 21 23 0D 0E 00 00
0.043032	180	SME_Slave_TPDO_1	8	41 06 0B 00 09 00 43 03
0.036566	18F124F4x	BMS_VCU1	8	00 00 5D 00 3D 7D DE 39

Figure 7. CANExplorer

Message	Time (sec)	Delta Time from Previous Message (sec)
BMS_VCU1	0.036566	
SME_Slave_TPDO_1	0.043032	0.0065
VMU_TPDO_2	0.049406	0.0064
VMU_TPDO_1	0.055949	0.0066
VMU_TPDO_4	0.067614	0.012
SME_Slave_TPDO_4	0.073471	0.0058
SME_Slave_TPDO_1	0.079309	0.0058
SME_Slave_TPDO_6	0.080840	**0.0015**

Table 3

In all cases, the components that are sending information are those who won the arbitration described in Figure 6.

The most interesting datum, however, is represented by the 0.0015, in bold, of Delta Time between the SME_Slave_TPDO_1 and SME_Slave_TPDO_6 message.

Since this dela time is much smaller than the others, it describes the concept of AMP well. When two messages belonging to the same node in possession of arbitration (therefore, SME_Slave_TPDO_ node, Inverter) the message with the lowest ID (SME_Slave_TPDO_1) is authorized to send messages on the bus.

2.3 Signals values extraction

Once received the adequate knowledge about the CAN network, it is possible to take one more step towards the final values that we want to be able to read from the network.

The information published on the network cannot be interpreted by the human eye, and are in fact defined as "raw values". Figure 6 shows the appearance of this data, in "Data" columns. We therefore need a method to be able to filter this information and make it understandable to us.

Figure 8 shows the basic scheme of the conversion process.

Logger → Raw data → DBC file → Physical Value

Figure 8. Conversion process.

Logging alone is not enough to be able to extract and understand information from a CAN network. It is necessary to use the so-called dbc files, a text file that contains information for decoding raw CAN bus data to 'physical values', intended as interpretable, through conversion rules.

The formula applied by these files corresponds to:

$$"Physical\ Value" = "Raw\ Value" * Factor + Offset$$

With:

- “Physical Value”, the final expected, interpretable real value
- "Raw Value”, the raw value coming from the logging, which cannot be interpreted, in Decimal
- Factor, a number present in the dbc file
- Offset, a number present in the dbc file

Example:

Factor A = 0.25

Offset B = 0

“Raw” Value = 0xFFFF (HEX) = 65535 (DEC)

“Physical” Value = (0.25*65535 + 0) = 16383.75

As for our project, a single dbc file was created, applicable to all three prototypes, corresponding to the link https://www.dropbox.com/s/7czi593kx5tf8ws/E_Fit_Powertrain_Kit_CAN_Database.dbc?dl=0.

3. Logging Overview

It is evident that, in order to be able to pass easily from one prototype to another, and to verify that these prototypes are as similar as possible, it is necessary to be able to extract information from each of them.

For our purposes, we have decided to adopt the use of two loggers: CANedge2 and PCAN-USB.

3.1. CANedge2/CANmod.GPS

Figure 9. CSS Electronics: CANedge2 and CANmod.gps

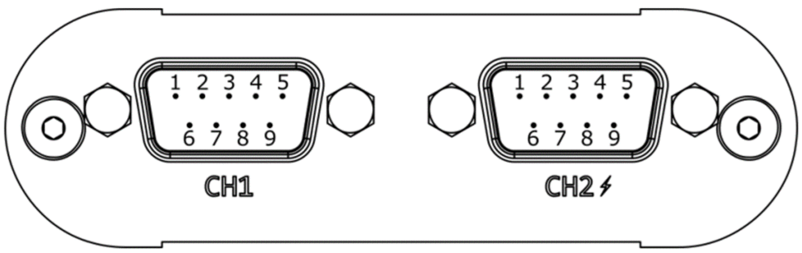

Figure 10. CANedge2 Pin Visualization

Pin #	Channel 1	Channel 2
1	5V	5V supply output
2	CAN 1 L	CAN 2 L
3	GND	GND
4	LIN Data 1	LIN Data 2
5	NC	NC

6	GND (optional)	GND (optional)
7	CAN 1 H	CAN 2 H
8	NC	NC
9	Supply & LIN1 VBAT	LIN2 VBAT

Table 4

- Advantages:

- It doesn't need any user input to start logging. Once connected to the CAN network and powered with the 12V line, logging takes place automatically. When the transmission of messages on the network is interrupted (therefore when the test is finished) the data logging automatically ends.

  When a new test is run, it is saved in a new .mf4 file, inside a new folder, itself inside the test folder.

- The possibility to use another CSS Electronics product, the CANmod.gps shown in Figure 9, to receive GPS information from the CAN network. This product, used in combination with the CANedge2, offers the possibility to see relevant information such as the distance traveled and the inclination of the road.

- The ability to use both SD card and Cloud function. Using the cloud, it is possible to upload data directly to a cloud, thus speeding up its analysis. Within this project, only the SD card has been used.

- Disadvantages:

- It does not offer its own dedicated software. It is therefore not possible to provide a real-time dashboard

3.2. PCAN-USB

Figure 11. PEAK-System: PCAN-USB

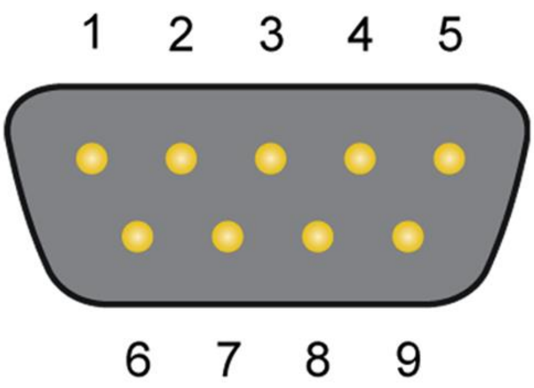

Figure 12. PCAN-USB Pin Visualization

Pin #	Assignment
1,9	CAN_V+ (optional)
2	CAN_L
7	CAN_H
3,6	CAN_GND
4,5,8	Not connected

Table 5

- Advantages:

- Through its own dedicated software, PCAN-explorer, it is possible to implement dashboards with panels in real time. Once the peak is connected to the CAN network, it will be enough to start the test to see the signals in the dashboards.

- Disadvantages:

- It does not offer gps functions and logging must be done through the software.

The table summarizes the features described so far, and also indicates where and how these analyzers can be used.

	CANedge2/CANmod.gps	**PCAN-USB**
Automatic logging	Yes	No
Gps informations	Yes	No
Real-Time dashboards	No	Yes
Output File Format	.mf4	.csv (comma separated value)

Table 6

4. CANbus and Virtual Prototype

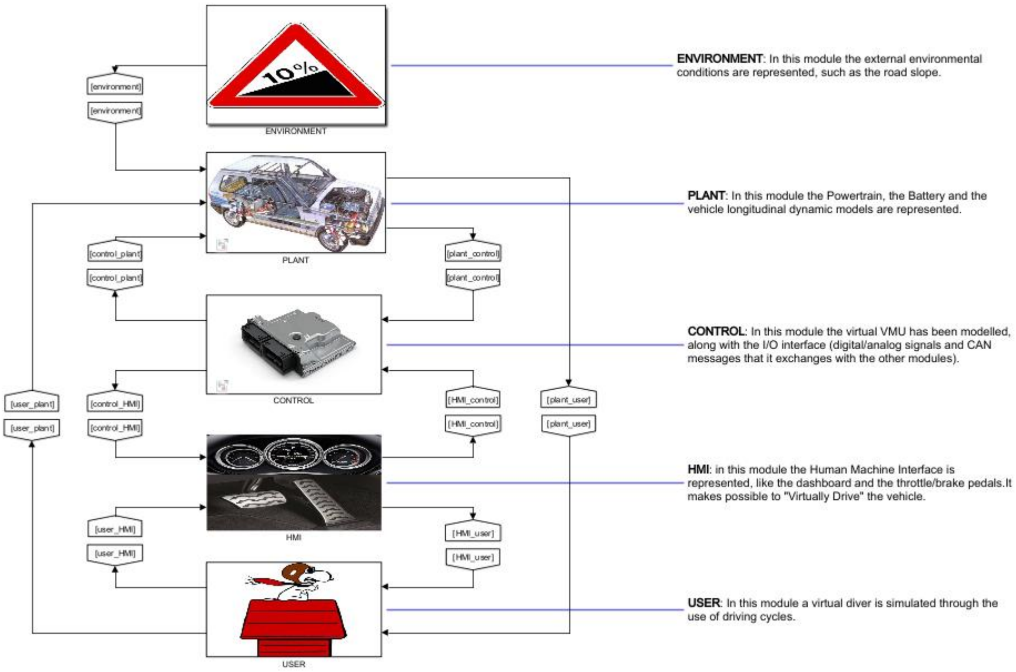

Figure 13. Modular Technical Model

Figure 13 shows the Modular Technical Model. This model consists of the following blocks:

- Environment
- Plant
- Control Logic
- HMI
- User

The MTM tries to virtually simulate the behavior of the real prototype. The final function of the model is to perform an automatic generation of code, with the aim of being able to then insert these codes inside the VMU. Once the VMU will be based on the codes created by the MTM, we can talk about Virtual Prototype.

It was implemented in Simulink environment, creating a message exchange network (therefore formed by receivers and transmitters). The network is completely simulated, but through a specific device, the Kvaser Leaf Light v2, it is possible to log this network as if it were a physical network. Through the union of this device with the mentioned loggers present in Table 6, it is possible to consider the virtual line identical to a physical network. Within the Virtual Prototype, the CAN network is the only element capable of carrying out this virtual/real transformation, underlining its importance and adaptability. Furthermore, it also allows you to check if all the components of the Modular Technical Model (MTM), the model on which the final creation of the Virtual Prototype is based, are actually communicating correctly with each other. Through the IDs of the messages, in fact, it is very easy to trace the sender of the message, thus being able to trace the components that have errors or modeling or communication errors.

The double function of the line is therefore underlined: an "internal" function for the communication of the various components with each other, and an "external" communication for being able to keep track of how these components are communicating.

Figure 14 shows, by way of example, the CAN network inside the battery.

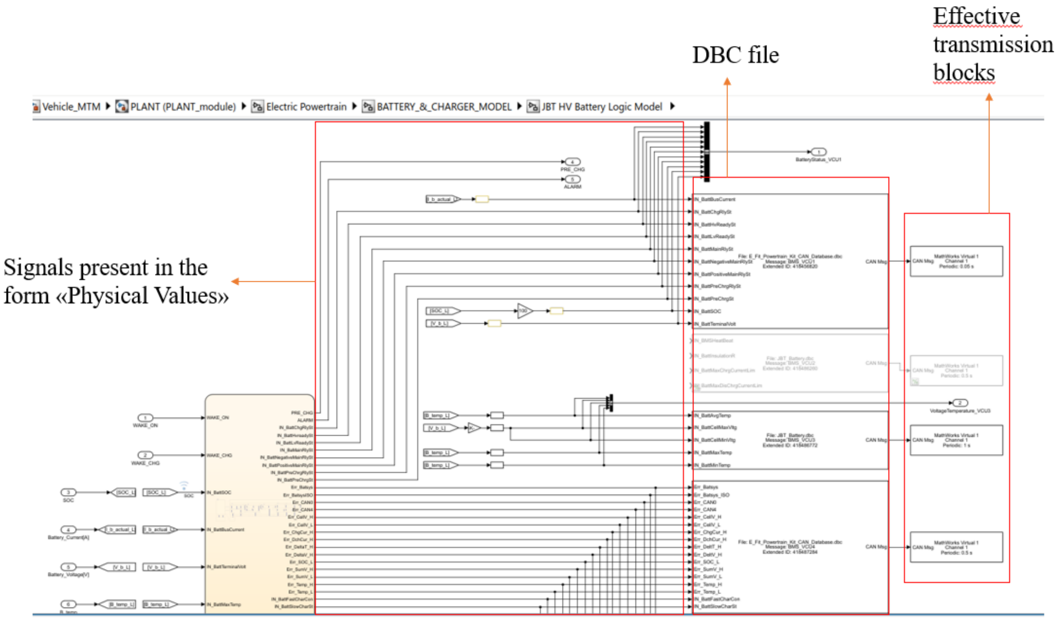

Figure 14. Battery CAN Network

- The MTM needs some modifications to perform a logging. They have been analyzed and collected within https://www.dropbox.com/scl/fi/j4ap6jfhhoxem0h2be2ny/Logging-Real-Time-vs.-batch-vs.-Virtual_Semi-real_Real-Prototype-_V2.paper?dl=0&rlkey=t4jc98gmakxunlzd5nbrdk25k#:h2=1.-MTM's-INITIAL-CONFIGURATION

- Regarding logging and data analysis via CANedge2, refer to

  https://www.dropbox.com/scl/fi/j4ap6jfhhoxem0h2be2ny/Logging-Real-Time-vs.-batch-vs.-Virtual_Semi-real_Real-Prototype-_V2.paper?dl=0&rlkey=t4jc98gmakxunlzd5nbrdk25k#:uid=536683309067400187494896&h2=2.-VIRTUAL-PROTOTYPE:-From-Vir

- Regarding logging and data analysis via PCAN-USB, refer to https://www.dropbox.com/scl/fi/j4ap6jfhhoxem0h2be2ny/Logging-Real-Time-vs.-batch-vs.-Virtual_Semi-real_Real-Prototype-_V2.paper?dl=0&rlkey=t4jc98gmakxunlzd5nbrdk25k#:h2=3.-VIRTUAL-PROTOTYPE:-From-Vir

5. CANbus and Semi-Real Prototype

Figure 15. Semi-Real Prototype

Bench tests are performed in order to model the MTM. The bench, in fact, offers the possibility of adding elements in power. It supplies the components of the electric propulsion with loads as if they were installed on the vehicle, with the possibility, however, of being able to condition the environment, so as to be able to carry out a data analysis in any desired context.

Semi-Real Prototype logging is simpler than Virtual Prototype, as the network is already physical and there is no need for any special precautions.

The data analysis of the Semi-Real model was mainly performed through the use of the PCAN-USB, but it is still possible to do it through Logging via CANedge2.

- Regarding logging and data analysis via CANedge2, refer to https://www.dropbox.com/scl/fi/j4ap6jfhhoxem0h2be2ny/Logging-Real-Time-vs.-batch-vs.-Virtual_Semi-real_Real-Prototype-_V2.paper?dl=0&rlkey=t4jc98gmakxunlzd5nbrdk25k#:h2=4.-SEMI-REAL-PROTOTYPE:-CANedg

- Regarding logging and data analysis via PCAN-USB, refer tohttps://www.dropbox.com/scl/fi/j4ap6jfhhoxem0h2be2ny/Logging-Real-Time-vs.-batch-vs.-Virtual_Semi-real_Real-Prototype-_V2.paper?dl=0&rlkey=t4jc98gmakxunlzd5nbrdk25k#:h2=5.-SEMI-REAL-PROTOTYPE:-PCAN-U

6. CANbus and Real Prototype

As far as the Real Prototype is concerned, the CAN network finds its maximum expression.

In fact, in the other two models, the CAN network has proved to be an excellent tool as regards subsequent data analysis and therefore modeling.

In general, however, through the CAN network it is possible to carry out analyzes in real time, a fundamental aspect in Real Prototype.

Through the logging of the PCAN-USB (the CANedge 2 does not offer Real Monitoring possibilities) it is possible to create an interface between the network and a data analysis software in real time; some critical parameters, such as battery temperature, inverter temperature, motor temperature, can therefore be monitored in real time, virtually recreating a cluster. The advantage of this virtual cluster is that it can be modeled at will, being able to view any signal present on the network (third column of the table present in https://www.dropbox.com/scl/fi/jqqze17msjciybrnffuch/Communication-inside-Virtual-Semi-Real-Real-Prototype.paper?dl=0&rlkey=aong8nwwf6vsi04fh94sk99vv)

The panels dedicated to battery monitoring are shown here as an example.

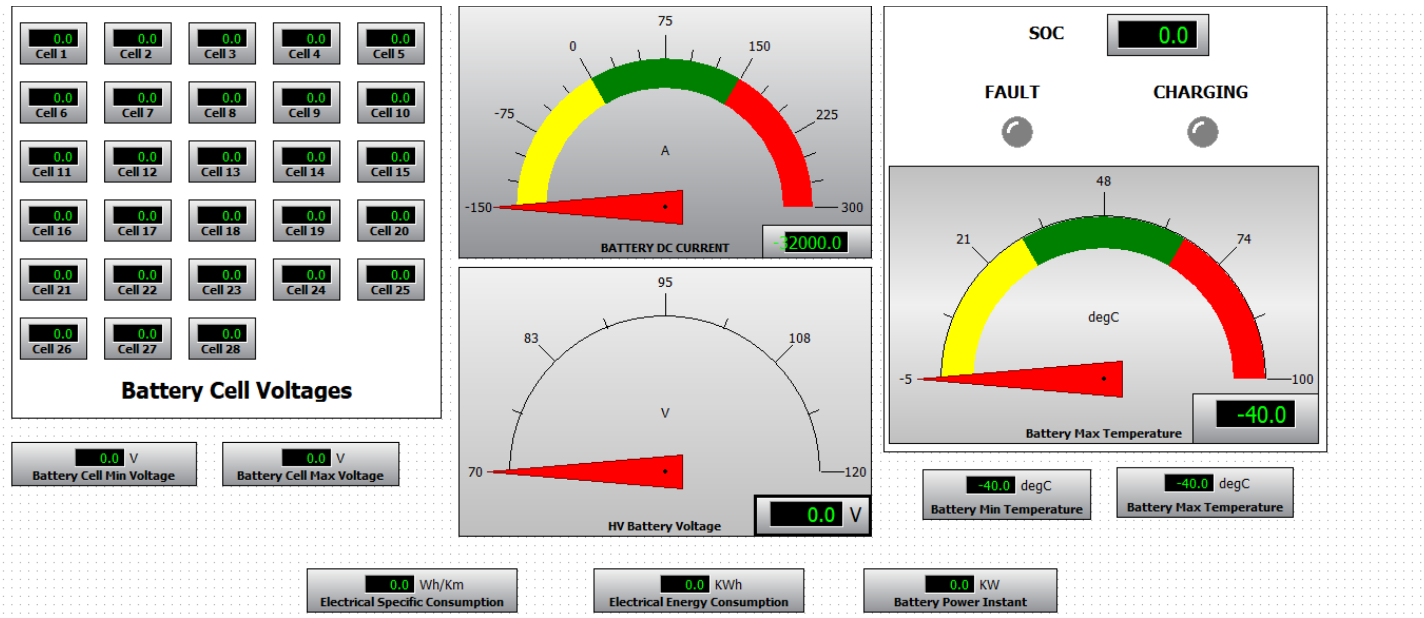

Figure 16. Battery panels on PCAN-Explorer

For a correct use of the software and for further information regarding the Real-Monitoring applied to our project, refer to https://www.dropbox.com/scl/fi/mctib8a9siql0plc66krf/P-CAN-Explorer-Projects.paper?dl=0&rlkey=ho5nn0td72nla3xbkh0m37e67

- Regarding logging and data analysis via CANedge2, refer to https://www.dropbox.com/scl/fi/j4ap6jfhhoxem0h2be2ny/Logging-Real-Time-vs.-batch-vs.-Virtual_Semi-real_Real-Prototype-_V2.paper?dl=0&rlkey=t4jc98gmakxunlzd5nbrdk25k#:h2=6.REAL-PROTOTYPE:-CANedge2-Log

- Regarding logging and data analysis via PCAN-USB, refer to

  https://www.dropbox.com/scl/fi/j4ap6jfhhoxem0h2be2ny/Logging-Real-Time-vs.-batch-vs.-Virtual_Semi-real_Real-Prototype-_V2.paper?dl=0&rlkey=t4jc98gmakxunlzd5nbrdk25k#:h2=7.-REAL-PROTOTYPE:-PCAN-USB-Lo

7. Results and Conclusion

The results obtained so far are in line with what has been described at a theoretical level and with the objectives set at the beginning of the project. In particular, for each prototype, through logging and scripts, it is possible to obtain a .mat file containing all the information of the network and a quick view of the results via pdf file to decide whether the test carried out can be useful or not.

The next step involves the Real Prototype test activity, to be virtually replicated (through the .mat file coming from the logging, then used in the Virtual Prototype). The analysis via MATLAB of the future comparison is already underway, but the tests on the Real Prototype are not yet available.

A Retrofit Kit activity must be carefully studied in order to lead to consistent results. The use of the CAN network finds its maximum expression in the validation phase of the Real Prototype. Referring to Figure 1, the activity that leads from point 5 (Real final Prototype) to point 1 (functional requirements, the first step of the project) is entirely based on CAN communication.

It is necessary to make one last consideration, partly already seen in chapter 2. The focus of this activity centered around the creation of the MTM. Within it, the only virtual element, capable of transforming itself into real, physical, is the CAN network. It must be emphasized that this created information comes from models that try to reflect reality, but still virtual models. The information they create, virtual in turn, however, has the possibility of being treated as if it were real.