



Politecnico di Torino

Master's degree in Mechatronic Engineering

A.a.2021/2022

Model based design of electric/hybrid powertrain with Extended V-Cycle development

Supervisor

Massimo Violante

Stefano Carabelli

Candidate

Andrea Lamanuzzi

Index

Abstract	2
Extended V-Cycle.....	3
V-model Tutorial: Digital Filter	6
Application: Panda 4WD-H	13
Future works	20

Abstract

The automotive world over the past 20 years has undergone unprecedented development, just think of the introduction of all the various emergency systems (ABS, TCR, etc), the electronic management of the combustion engine, and the development of advanced driver assistance systems.

With the introduction of electronics in vehicles, and the increasing complexity of these and the tightening of safety standards, it became necessary to introduce a systematic procedure for approaching the project.

This working methodology is based on a [Model-Based](#) approach.

The project phases can essentially be divided into four stages:

1. Requirements definition and feasibility study phase;
2. Model design phase;
3. Model-in-the-loop & Software-in-the-loop;
4. Hardware-in-the-loop (RCP, VMU).

In particular, it must be a systematic step-by-step procedure to achieve the functional safety standards that have been defined by [ISO 26262](#).

Once this methodology was defined, the digital filter tutorial was developed, which goes through all the stages of the V-Cycle.

There is then a real application of the V-Cycle on a hybrid powertrain.

Chapter 1

Extended V-Cycle

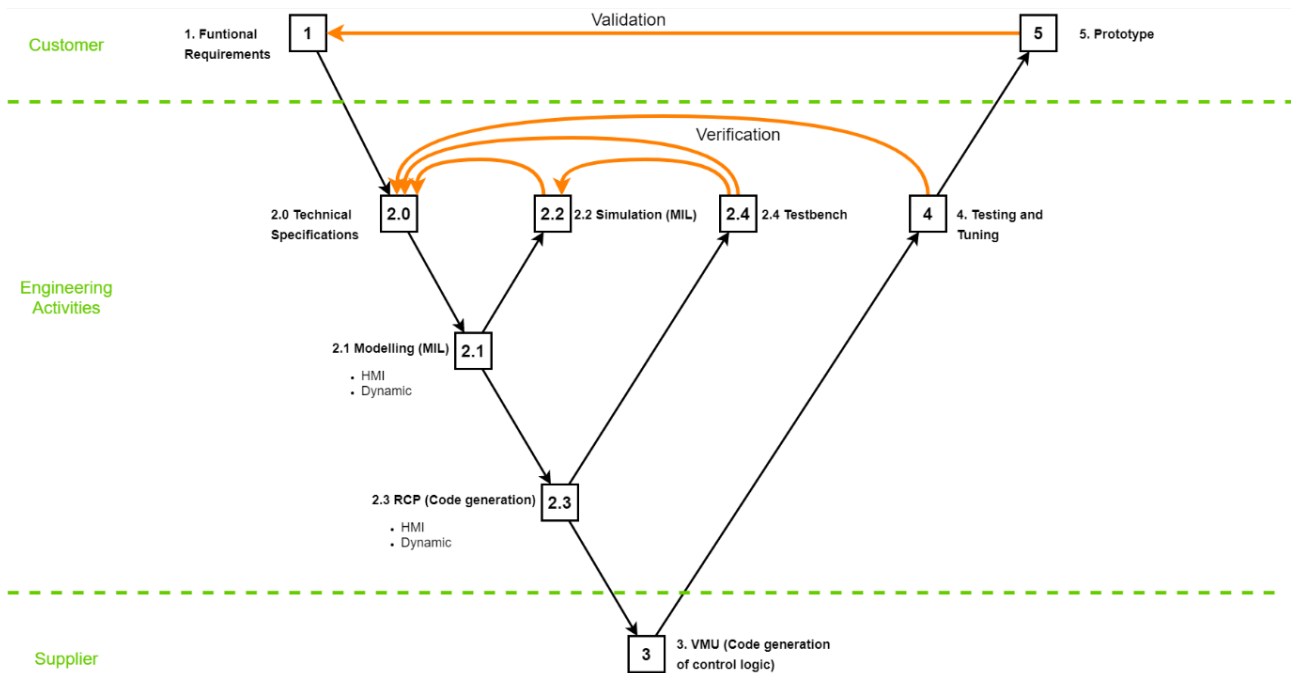


Figure 1 Extended V-Model Scheme

The Extended V-Model Scheme is shown in the figure above, this scheme allows us to understand the project phases and roles very effectively. It can also be seen that there are verification and validation phases.

Verification: in this phase it is ascertained that the technical specifications have been met during the simulation, RCP and final VMU phases respectively.

Validation: phase in which it is ascertained that the prototype meets all the functional requirements requested by the customer.

1.0 Functional Requirements: this is the phase in which the general requirements of the project are drawn up, often this phase is carried out by the customer.

2.0 Technical Specifications: In this phase, a preliminary study is made and the technical specifications of the project are produced.

2.1 Modelling: construction of the model on the Development Workstation.

2.2 Simulation: In this phase, simulations are carried out to verify that the model behaves correctly. This phase is very important since it is possible to make a complete simulation of our vehicle, we can test and eliminate bugs from our logic and verify that the plant behaves properly.

As we can see from the diagram, in this phase there is a verification of the technical specifications.

2.3 RCP: [Rapid Control Prototyping](#) makes it possible to quickly verify control logic in a real environment. These systems are often integrated in MATLAB/Simulink and allow code to be generated in automatic mode via Embedded Coder.

An example of such systems is dSpace, which provides both the hardware ([MicroAutoBox II](#)) and the software ([ControlDesk](#)).

2.4 Testbench: Bench test of the Rapid Prototyping Phase (HIL), here it is tested that the code produced and integrated on the hardware works, also in this phase there is a verification phase of the technical requirements.

3 VMU (Code Generation): Once all technical requirements have been verified in RCP and the hardware requirements of the VMU have been

defined, we move on to the selection of what will be the final VMU for the project.

This is the phase in which the code is generated and uploaded to the VMU.

4 Testing and tuning: Testing of the VMU on a test bench, the technical requirements are checked and parameters are tuned.

5 Prototype: Construction of the prototype and validation of the Functional requirements.

Chapter 2

V-model Tutorial: Digital Filter

To better understand this methodology, a simple tutorial was created in which a multi-task digital filter was implemented.

In the first step, the Functional requirements were drawn up:

- Cut-off frequency of 10 rad/s
- Double filter with different sampling times (20ms & 50ms)
- ON/OFF toggle switch
- Emergency momentary button
- Status LEDs

Subsequently, the technical specifications were developed, in particular, states, transitions, hardware and software requirements were defined.

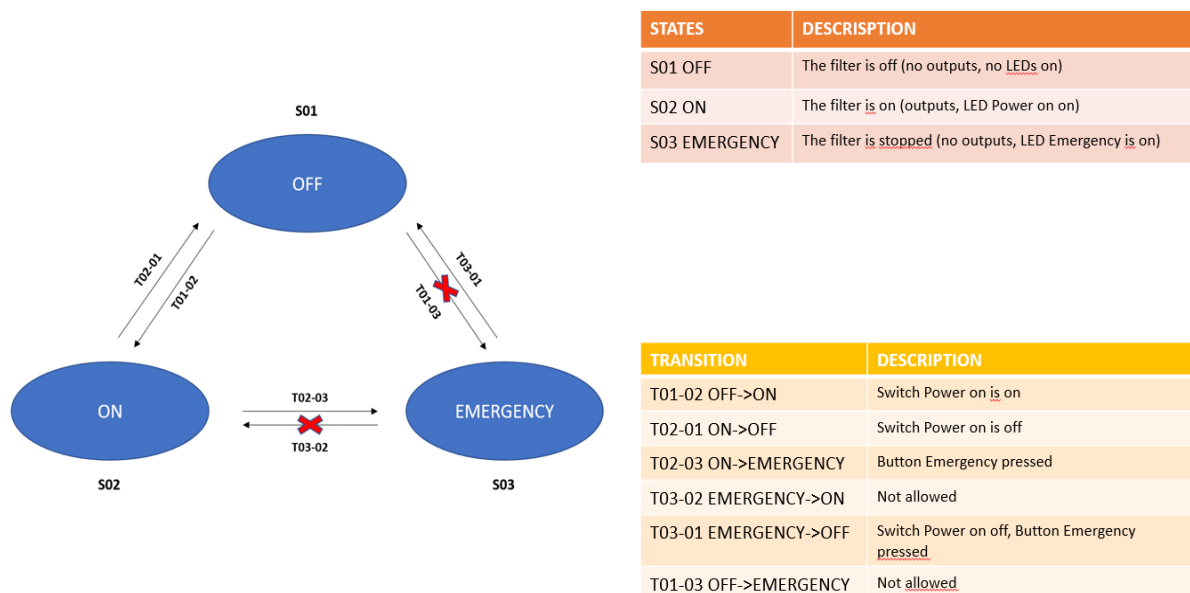


Figure 2 States, transitions and requirements

From the figure above, the transitions that are not allowed are also highlighted.

Once the technical specifications were completed, we moved on to the system design phase.

The [MTM template](#) was used for the construction of the Simulink model.

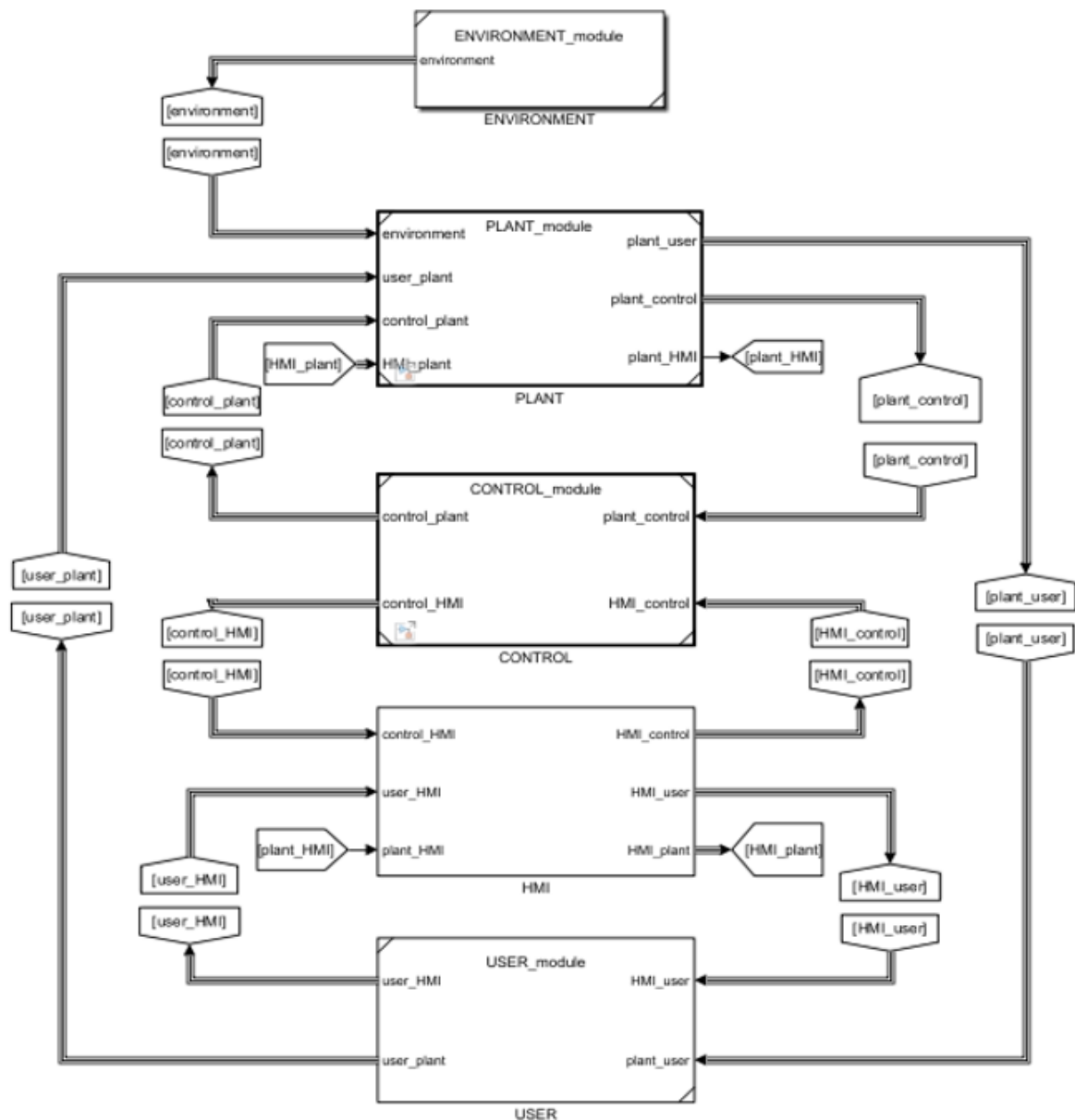


Figure 3 MTM

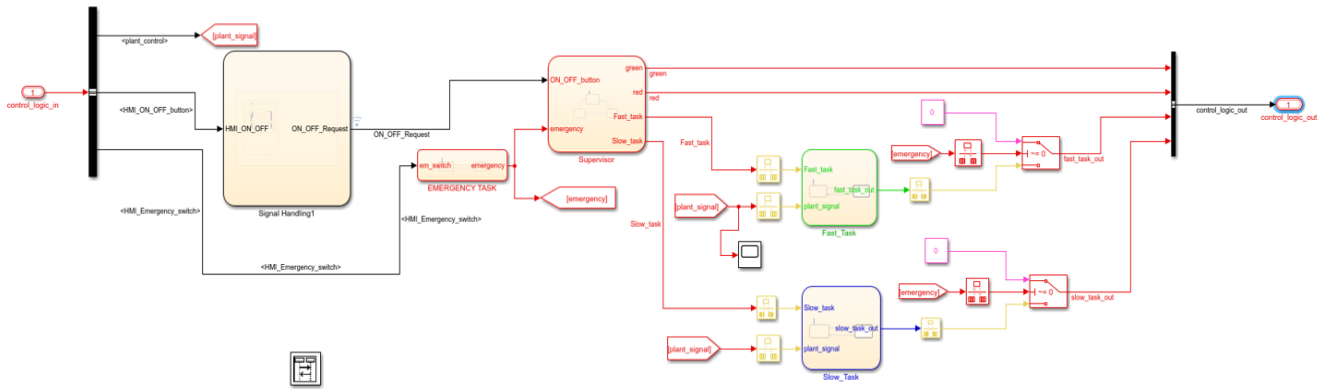


Figure 4 Control Logic

In the HMI, the signal generator controls were implemented, as well as the oscilloscope interface and the filter's buttons and status LEDs.

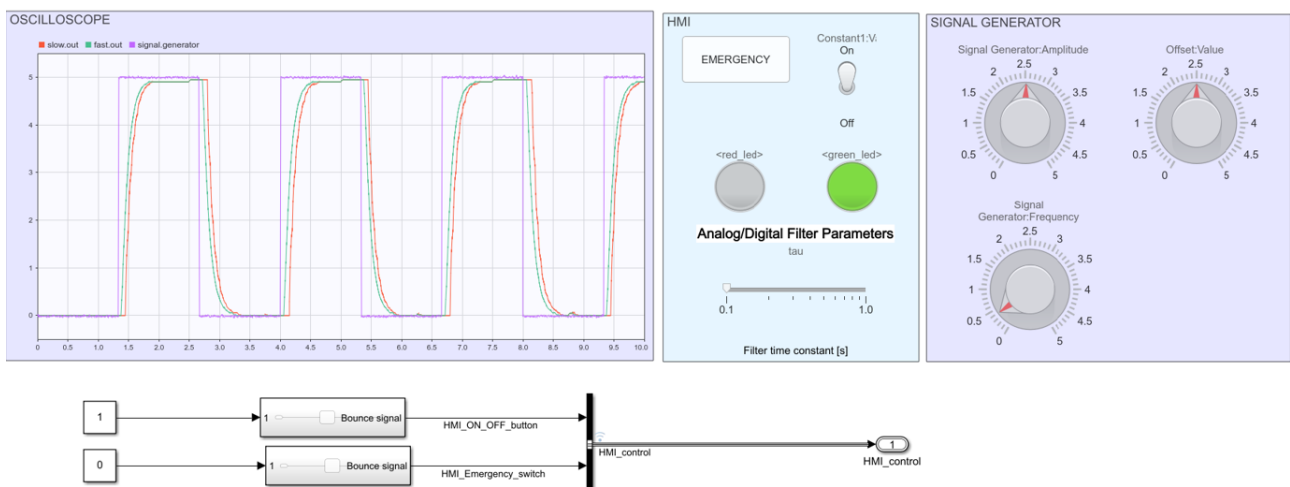


Figure 5 HMI

This was followed by simulation, where it was ascertained that the model behaved correctly and the Technical requirements were implemented correctly.

In simulation, key bouncing was also reproduced in order to test the implemented debouncing strategy.

Below are some of the results obtained during the simulation.

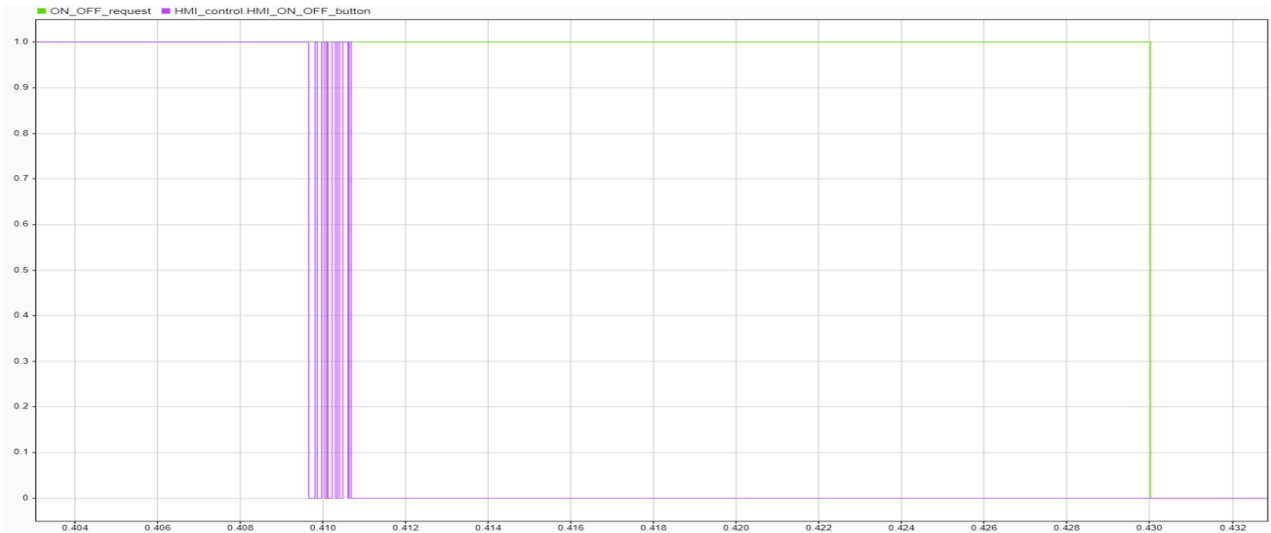


Figure 6 Debouncing

T01-02 Power ON (Sequence viewer)

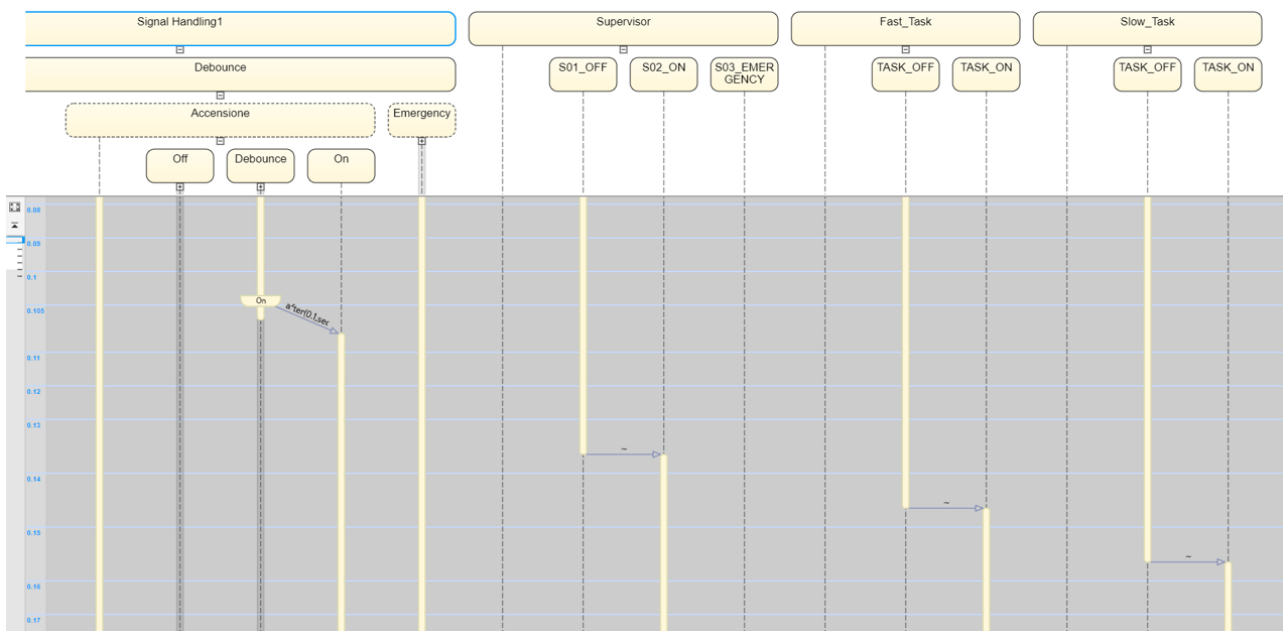


Figure 7 Power ON

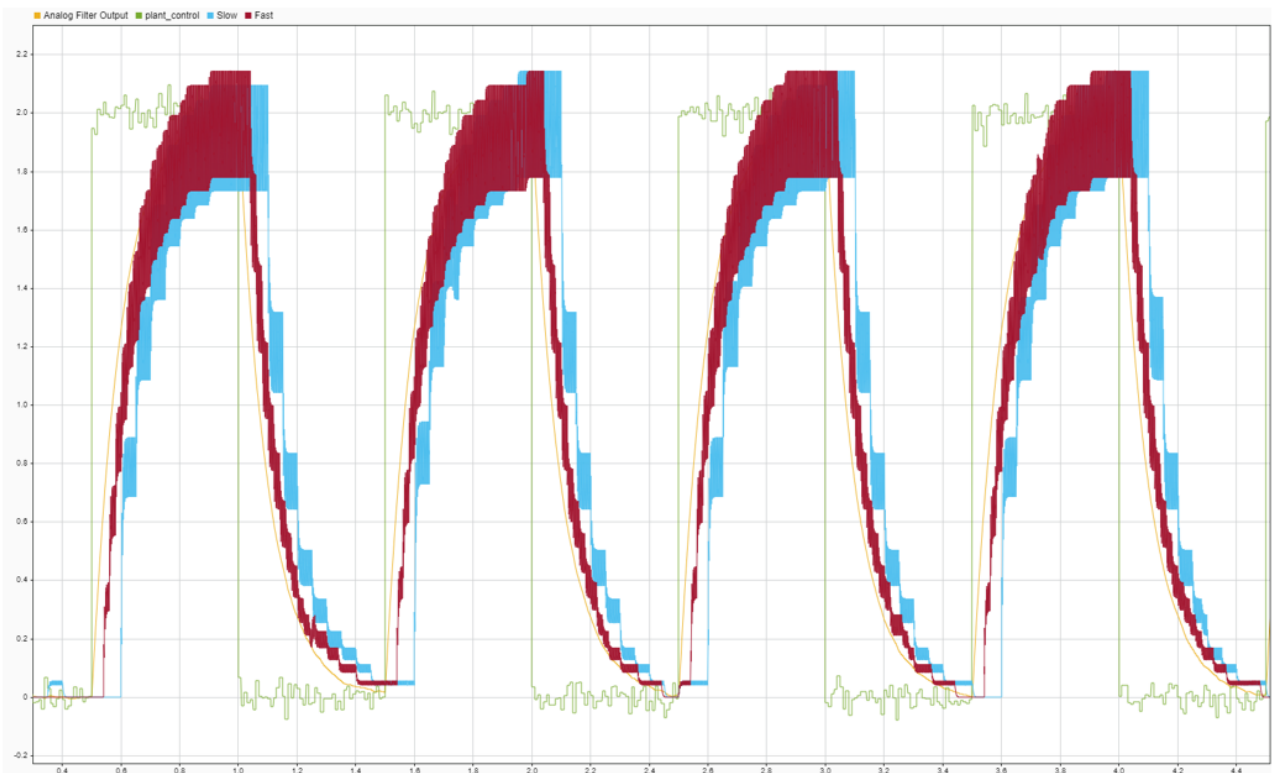


Figure 8 Results

Once the simulation and verification phase has been completed, we move on to rapid control prototyping, here the code is generated via the embedded coder and uploaded to dSpace before moving on to the test bench.

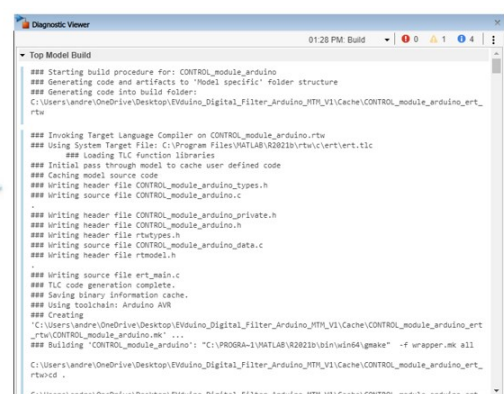
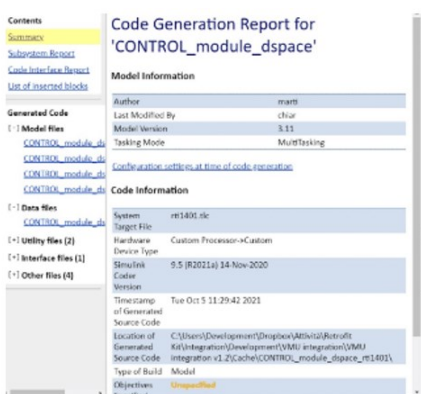


Figure 9 Code generation

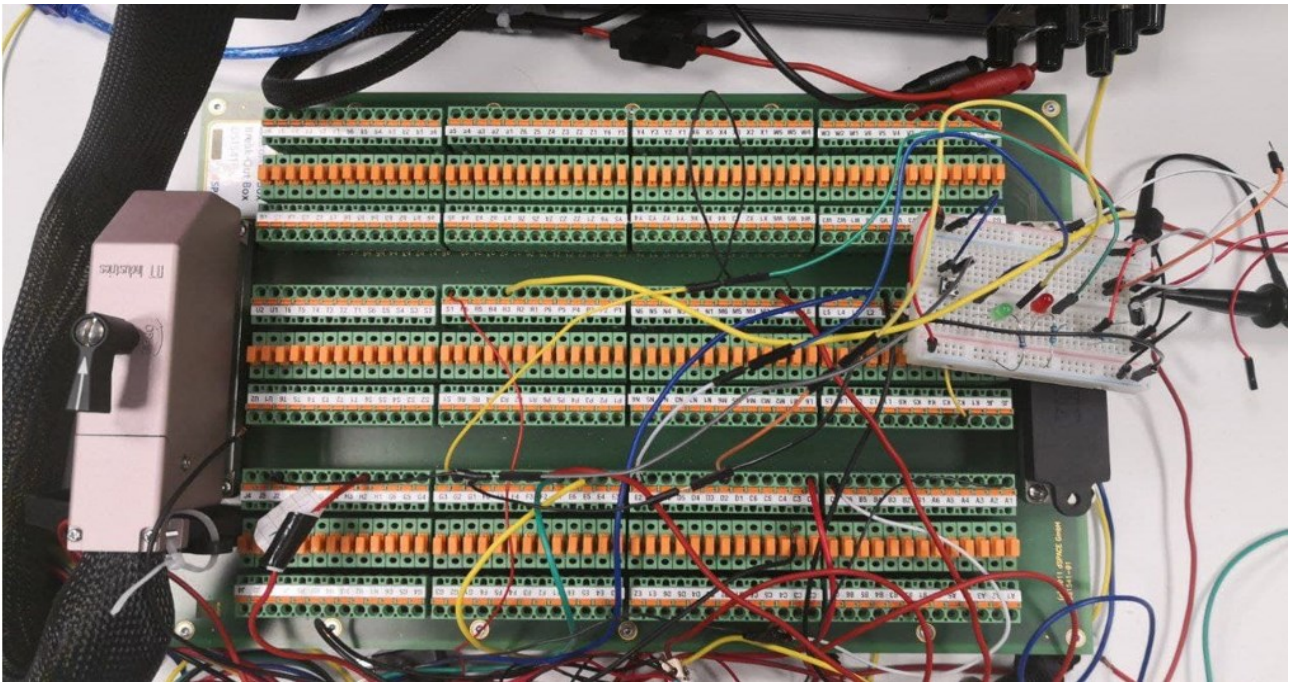


Figure 10 Test bench

On the bench it was possible to test the control logic in real time and verify the technical requirements.

Rapid prototyping systems (in this case dSPACE) not only provide the necessary hardware but also software for logging and tuning.

The software used is [ControlDesk](#).

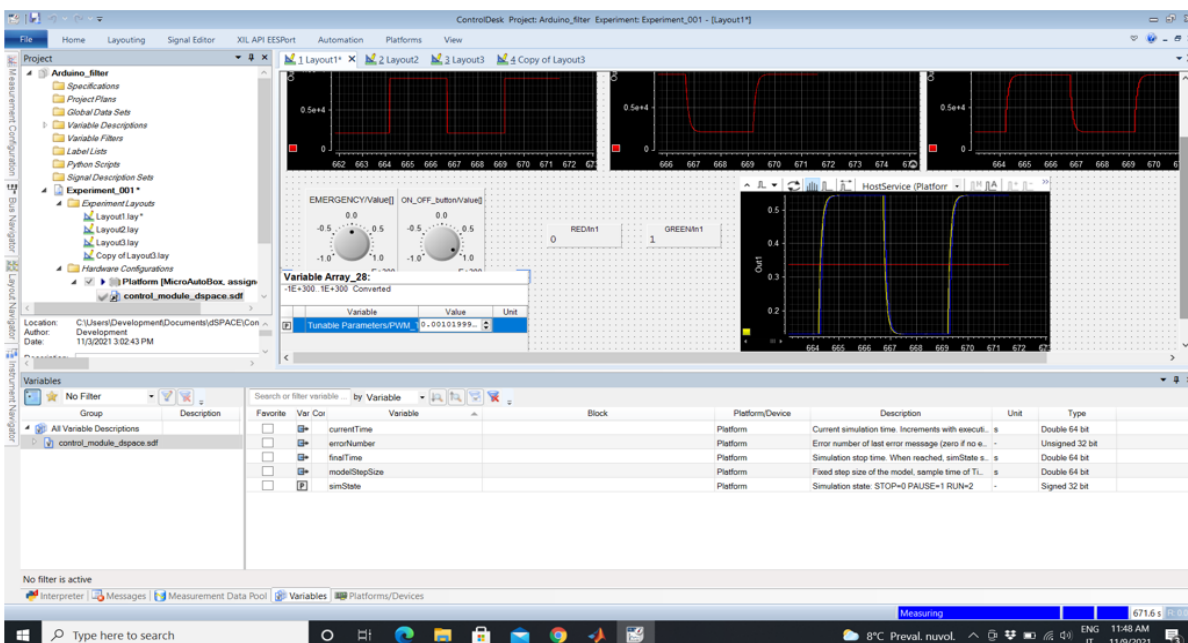


Figure 11 ControlDesk

The next step was to generate the code and upload it to the Evduino VMU. To debug the code on the board, the HighTec programme was used with the procedure outlined in the [EVduino MultiTask IDE - User Manual](#).

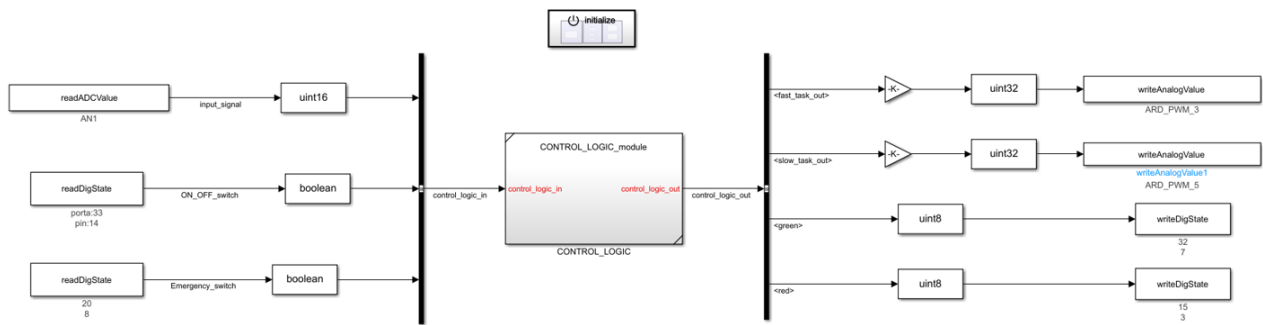


Figure 12 Control Module - EVduino Support Package

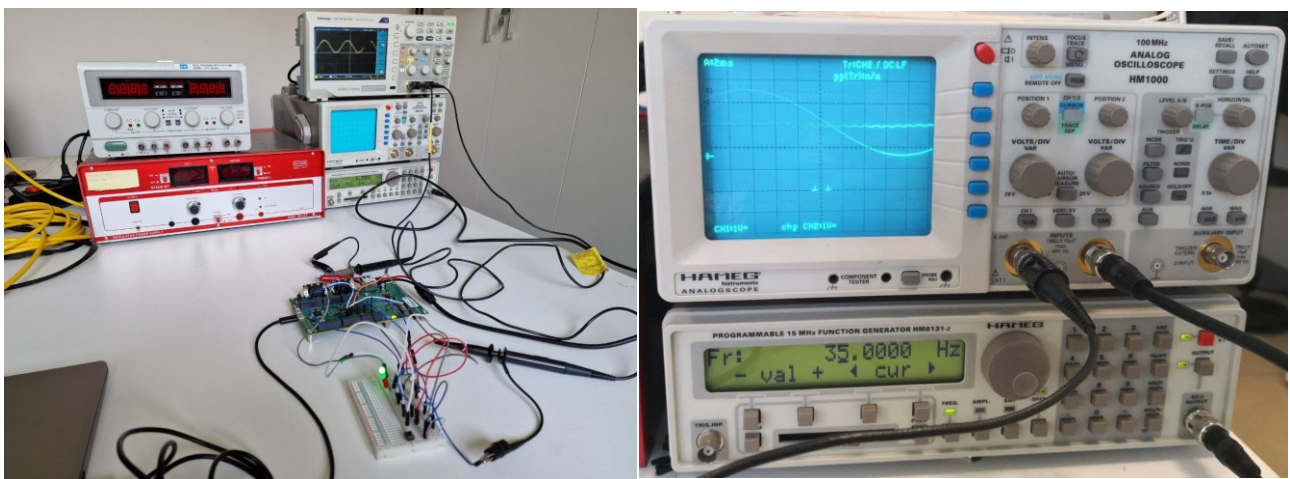


Figure 13 Test bench

The full digital filter tutorial can be found [here](#).

Chapter 3

Application: Panda 4WD-H

THE COMPETITION:

[Panda Raid](#) is an international long-distance (about 3000 km) rally raid in the Sahara Desert in which 300 cars race against one another for 7 days to win.

THE GOAL:

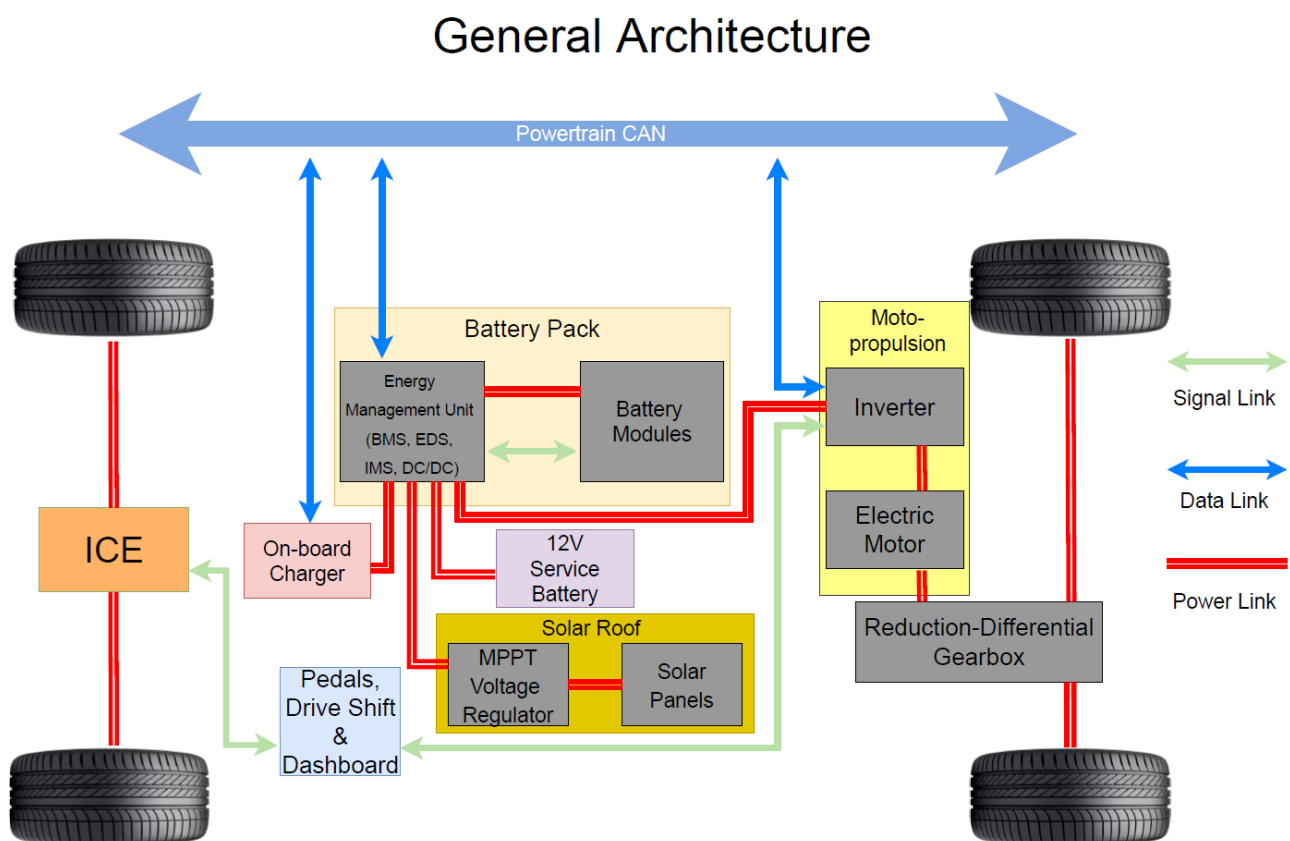


Figure 14 General Architecture

Design a 4 Wheel Drive Hybrid powertrain for a Fiat Panda 1st generation to race in a desert raid competition.

The goal is to create a hybrid system that can deliver enough performance to overcome a classic Panda AWD in such a challenging environment, despite the extra weight.

Furthermore, in this project is possible to acquire and apply new skills both from a programming and mechanical point of view to design a hybrid vehicle which can race in a rally raid competition called “Panda Raid”.

The team had to focus on three areas:

- Implementation of the software and the coding in order to exploit a VMU (Vehicle Management Unit) to control the hybrid system both for the torque split strategy and for the battery pack management
- Design the rear axle which will host the electric motor
- Design the vehicle from a dynamic point of view in such a way that it can overcome all the most demanding part of the race

The same working methodology was used for the design of the Panda 4WD-H.

The first phase of the project was the drafting of [functional requirements, technical requirements and the definition of the vehicle architecture](#).

MAIN STATES

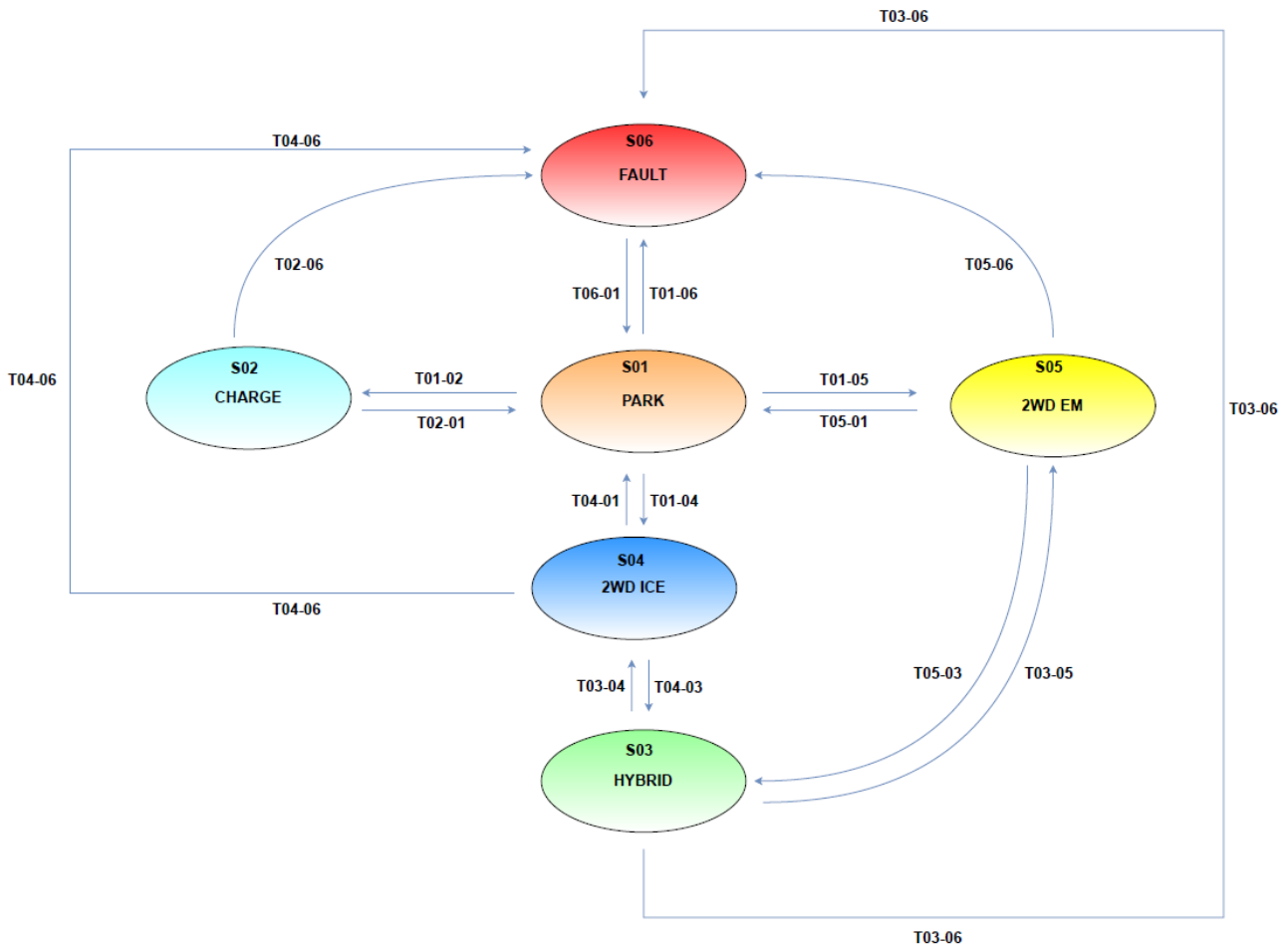


Figure 15 Main States

Once all vehicle functions and controls had been defined, the design of the electrical system was moved on.

Starting with the PowerBox of the electric Panda, the design focused mainly on how to integrate it and the modifications necessary to adapt it to the hybrid vehicle.

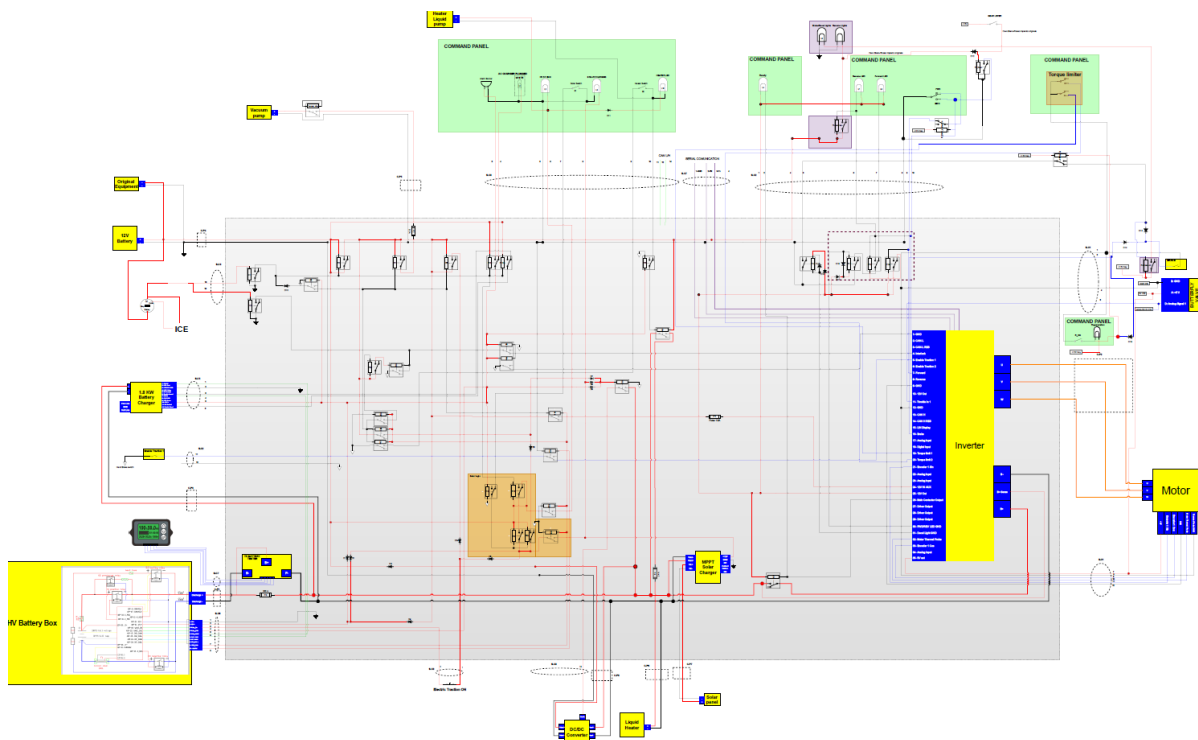


Figure 16 Electrical scheme

On the bench in the laboratory, it was verified that the changes made to the electrical system are correctly interpreted by the inverter.

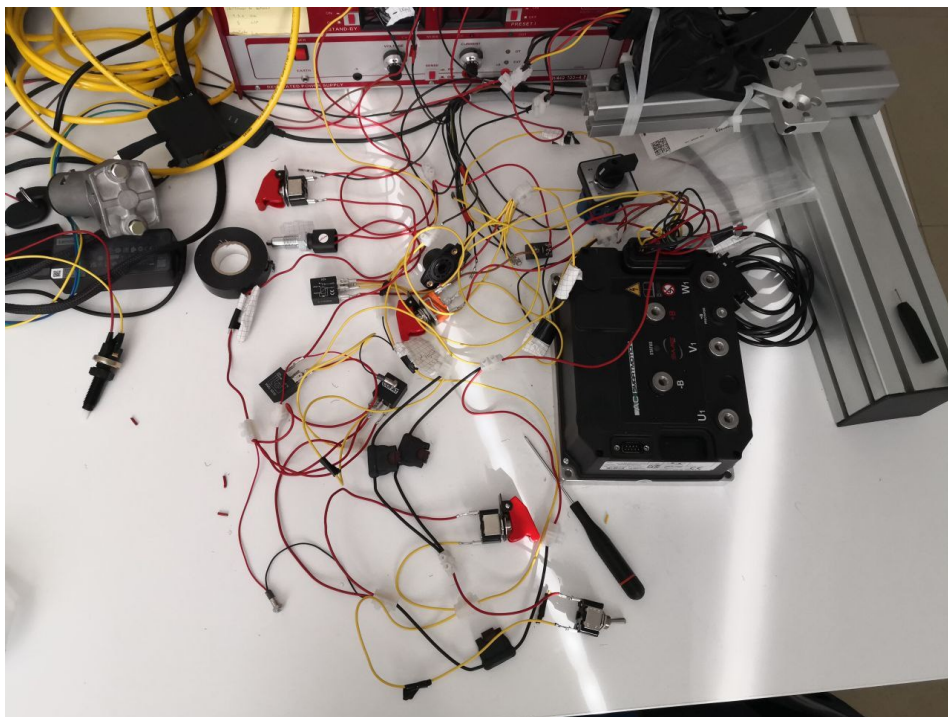


Figure 17 Test bench

Detailed documentation of the [circuit diagram](#).

Parallel to the design of the electrical diagram, the dashboard was developed in which all the controls for operating the electric powertrain in manual mode are housed.

The study of the [dashboard](#) took ergonomics into account, as the controls must be easily accessible to both the pilot and co-pilot in all conditions of use.

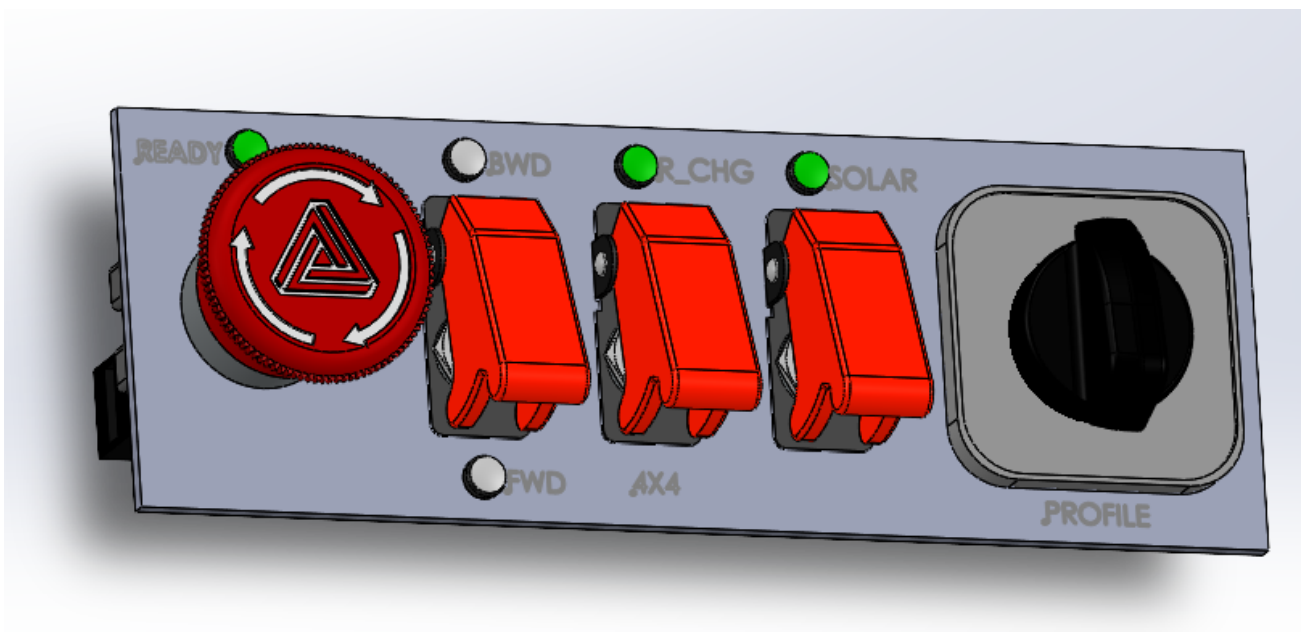


Figure 18 Dashboard

Starting with the dimensions of the Panda's car trunk, the [PowerBox-Battery-Tank layout](#) was then defined.

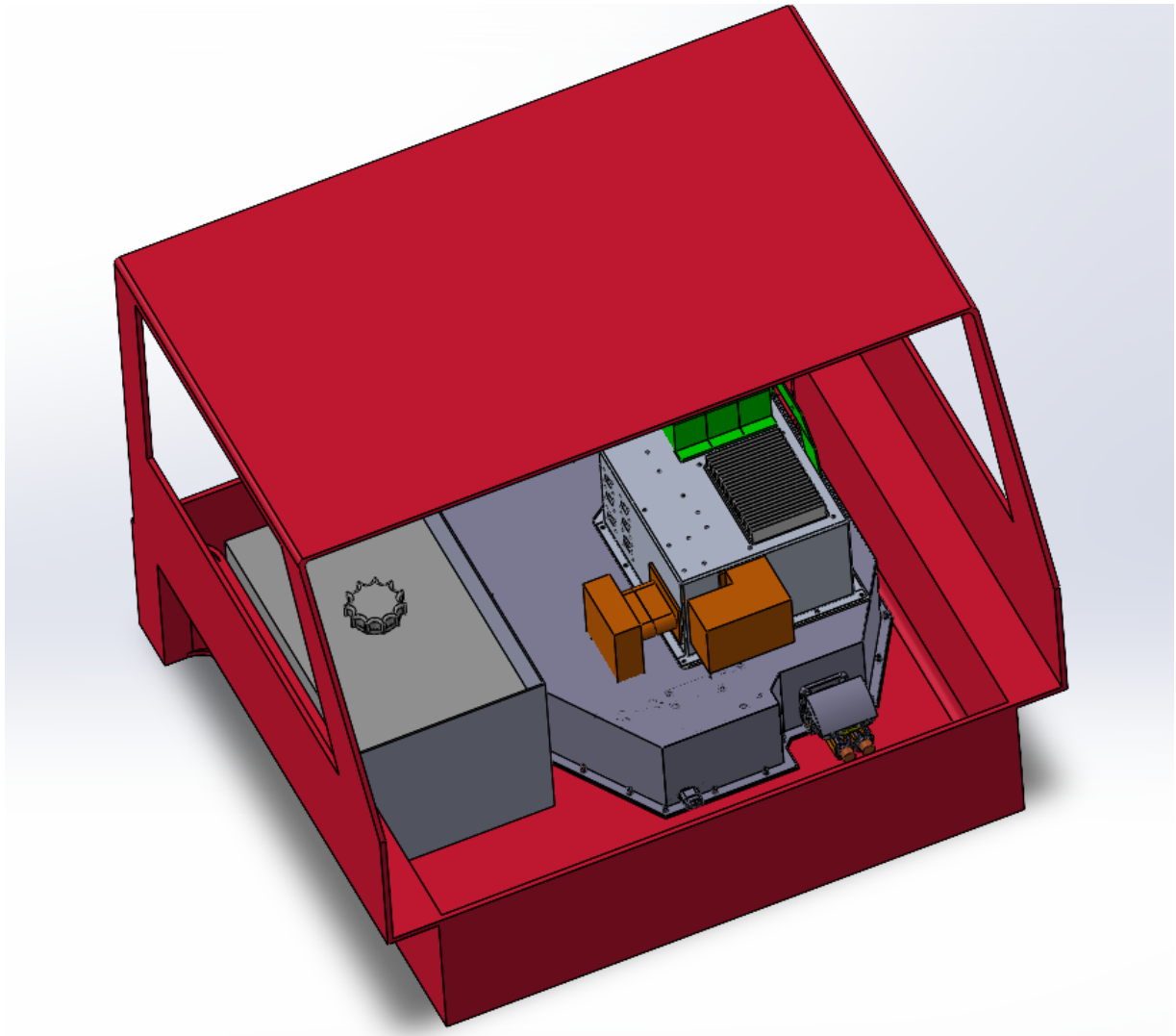


Figure 19 Layout

For the positioning of the components, account was taken of the weight distribution due to the positioning of the motor-differential unit, we can see that it is offset on the left side. The components will be secured by means of belts.

Once the arrangement in the car trunk was defined, it was possible to define the length of the cables for wiring the components.

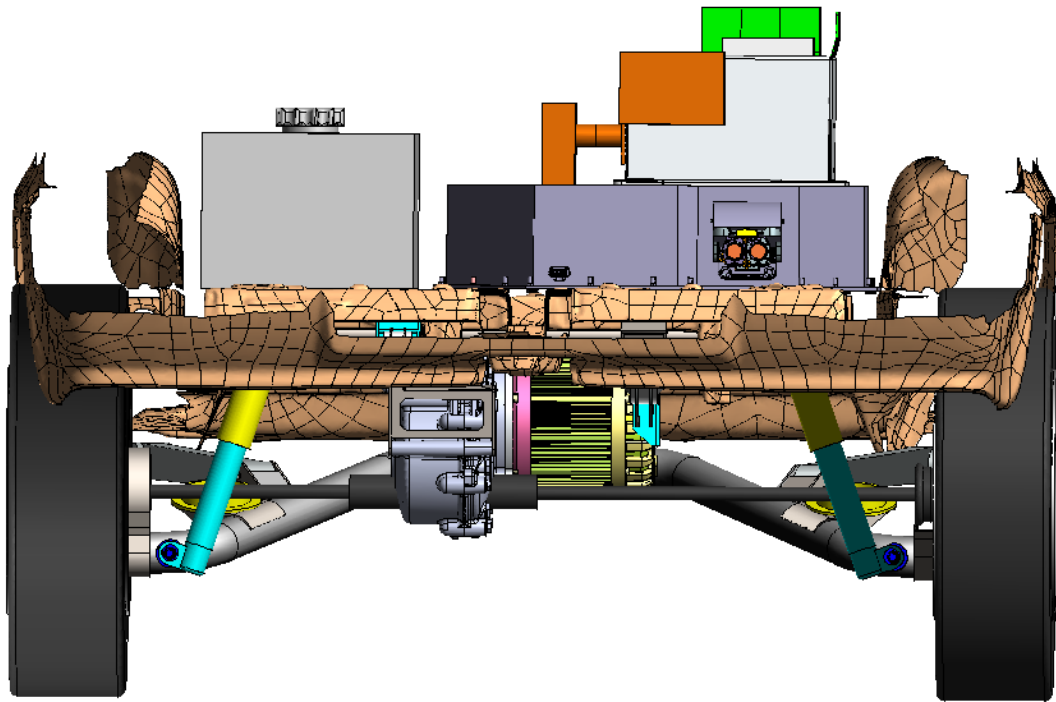


Figure 20 Complete rear end

Future works

Panda 4WD-H

Work will proceed along two parallel paths:

- Bench testing of the electric powertrain and subsequent assembly of the prototype with road tests
- Implementation on VMU of the hybrid mode traction substate:
 1. Torque split.
 2. Anti-slip.
 3. Idle mode.
 4. Manual/automatic control switch.