Data and Telemetry for high efficiency vehicles using IoT

Vitória Mariana da Rocha Gasino

Summary

Based on the need to acquire, process, and display data obtained by a high mileage prototype vehicle, in 2022, within the Eco Octano energy efficiency team, we have developed an instrumentation and telemetry project. The project gives us the ability to capture the vehicle's speed, temperature, current, and voltage data, in addition to presenting them in real-time and saving them for future team analysis.

Some of the main characteristics taken into account for the development of the project were: Scalability; Low energy consumption and easy access to data. For the aforementioned points to be fulfilled, we only used microcontrollers for the vehicle instrumentation. For visualization of the data, a website was developed, in which the entire team could have access to the data in real-time, or close to real-time, with graphics dedicated to each type of data. The protocols and methods used for telemetry allow scalability and easy adaptation for future projects based on IoT since it uses the MQTT protocol. The project can acquire vehicle data, send it and display this data in real-time. It is also able to automatically generate PDF reports with graphs of the acquired data.

Project

The project was required to show the data in real-time and be able to be accessed remotely from different platforms. As the acquired information volume isn't that big, a system based on IoT(Internet of things) has been developed. Built based on the opportunity of utilizing the platform Azure, the ideas were constructed based on the IoT Hub, a service available in the platform. The communication protocol between the vehicle, the cloud, and the devices that show the data is the MQTT.

Microcontrollers

-TEENSY 4.0:

Teensy is a developing board that contains one microcontroller, whose programming is compatible with the Arduino platforms. The chosen type for the implementation in the Telemetry of the electric car was Teensy 4.0. Some of the reasons that explain this choice are the high speed of the data process, the big number of pinouts, and the reduced size.

-ESP32:

ESP32 is a microcontroller that is capable of communicating using WiFi, which was a fundamental characteristic for its implementation in the project. The ESP32 will receive the data by serial communication with Teensy 4.0, and send the information to the Azure IoT Hub.

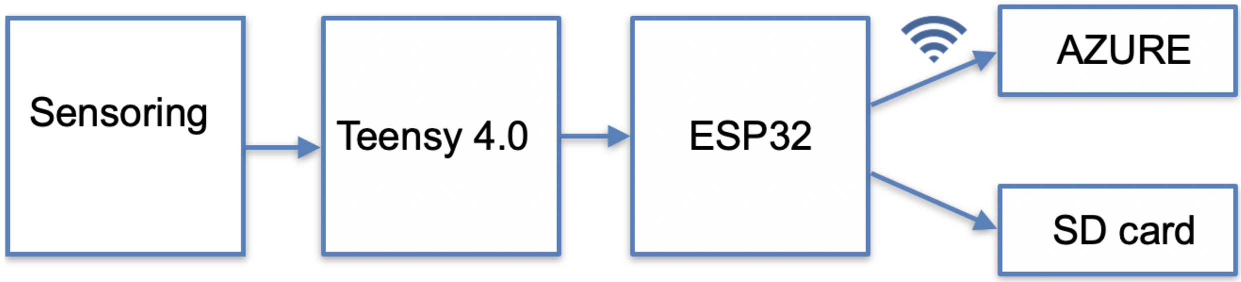

Figure 1: Fluxogram of the communication between the parts of the 2022 telemetry.

The communication steps can be divided: (i) communication between the microcontrollers, (ii)communication with the electric powertrain PCB, and (iii) Communication between ESP32 and Azure.

(i)Communication between the microcontrollers;
Using the serial method, the communication uses the serial ports. The Baud Rate is 115200.
(ii)Communication with the electric powertrain PCB;
Also serial, so it uses the serial ports as well. The Baud Rate is 115200.
(iii)Communication between ESP32 and Azure;
The communication with Azure is made using the libraries "AzureIoTHub.h" and "Esp32MQTTClient.h". A sample code, licensed and provided by MIT has been used as the basis.

Sensoring

The acquisition of the vehicle data was defined from the communication with the other subsystems. The founded sensoring needs in the vehicle was:

- Speed
    - Hall Effect Sensor;
- Motor voltage
    - Sent by the Electric Powertrain PCB;
- Motor current
    - Sent by the Electric Powertrain PCB;
- Motor temperature
    - Thermocouple sensor type K with module (MAX6675);

The current and voltage data Will be acquired by the electric Powetraind board and sent to the Telemetry board.

Circuit Board

There are some essential aspects that need to be taken care of when developing a board that manages sensible signals, such as sensing and telemetry. Problems like interference, noises, and difference of potential between the modules, are the factors that need to be taken into account in the development of the PCB.

The Telemetry circuit board needs to be capable of communicating with the sensors, buck converter, and microcontrollers. All the mentioned components must be connected at the PCB so they can communicate with each other.

In the development of the PCB, we added capacitors to do the signal filtering, close to the connections between the signal pins and the supply. The board design also guarantees the inexistence of trails with bigger or equal curves than 90º, preventing the reflexing phenomena. Another characteristic that was considered when the components were allocated was the distant positioning between the most sensible trails and the Buck, for example.

The PCB design was fully developed using the Proteus software, and printed by a company specialized in printed circuit boards.

Some components were created inside the software in order to supply the project's needs.

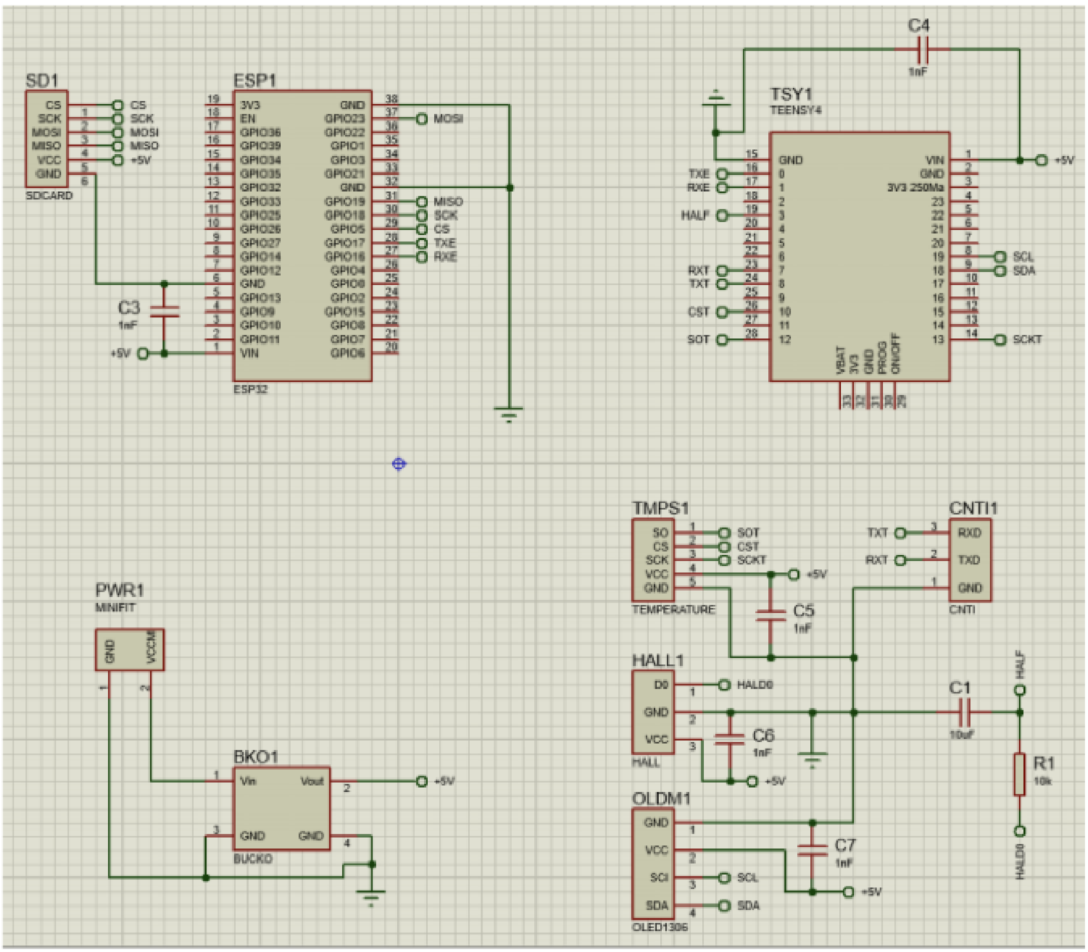

Figure 2:Schematic of the developed board.

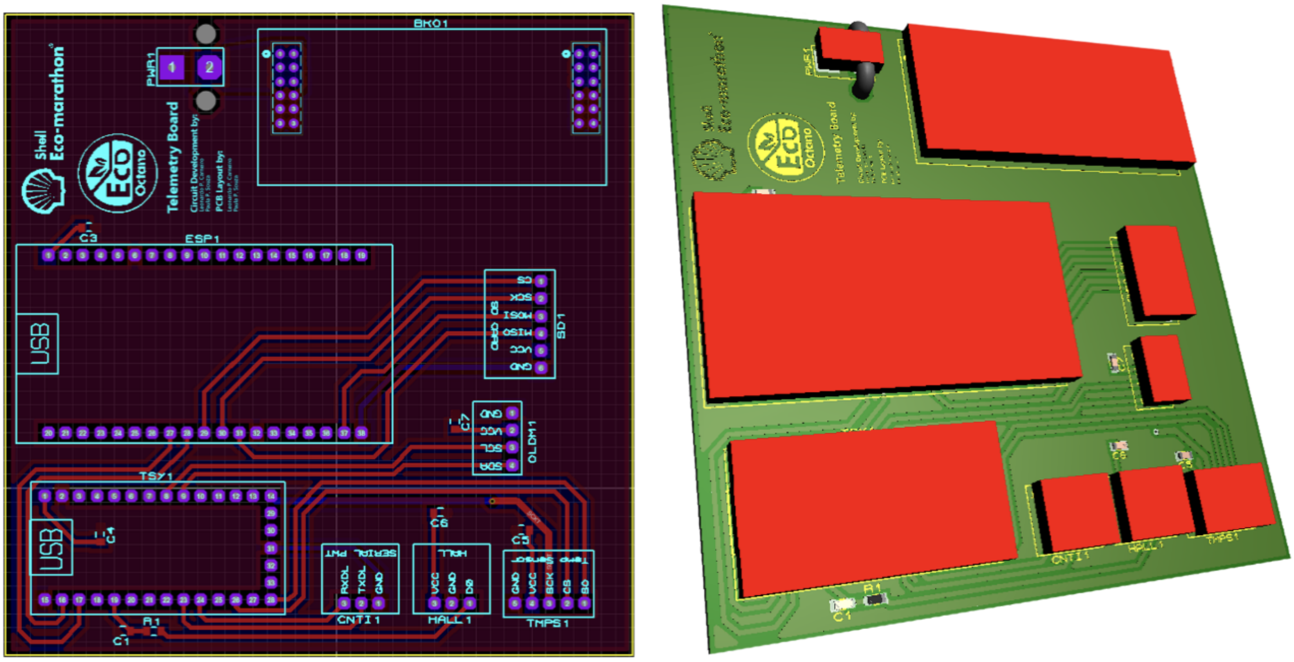

Figure 3:PCB layout and 3D model of the developed board.

Components

The SMD(Surface Mounted Device) components used were:

- Teensy 4.0;
- ESP32;
- Thermocouple Sensor type K (MAX6675);
- Hall effect sensor 3144E;
- 5x Capacitors 1nF;
- 1x Capacitors 10 µF;
- 1x Resistor 10k Ohms
- Soldering bar;
- Molex connectors;

All the components are SMD 0805.

Manipulation and exposition of the obtained data

For real-time visualization, we choose to use the Microsoft cloud platform and Azure, for storage, handling, and display of the data acquired by the sensors on the PCB. The large amount of documentation, tutorials, and available services in the platform provided by the company made it possible to plot graphs in many styles on a website (local or public) for the performance analysis of each race. When installed, the system's principal objective is to show the cars' speed to the team for easy reading and interpretation. Also, the pilot will have a non-WiFi dependent system that will also show the speed in real-time in order to help the driver to drive in a way that guarantees greater vehicle efficiency.

The data flux is represented in the figure and may be divided into two parts: on the left, we have the local connections in the PCB, and on the right, the data when it is sent by WiFi to Azure.

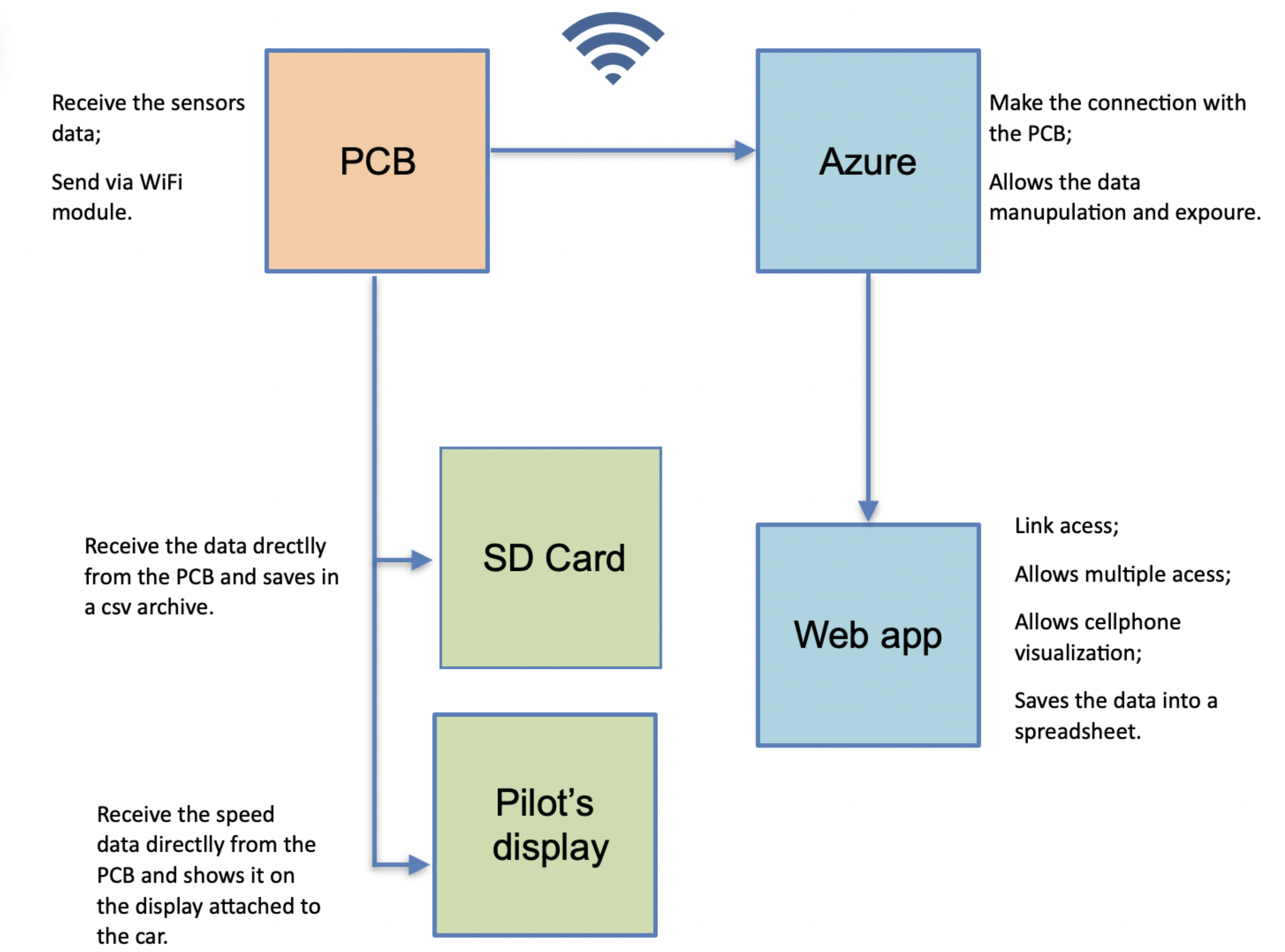

Figure 4: Web App developed final result.

The second stage consists of the manipulation, organization, and publication of the data. This step was developed based on the codes and tutorials provided by Azure. At the end of the configuration and adaptation of the environments, we have the result publication of a web application with the dynamic supply by the data from sensors present on the PCB.

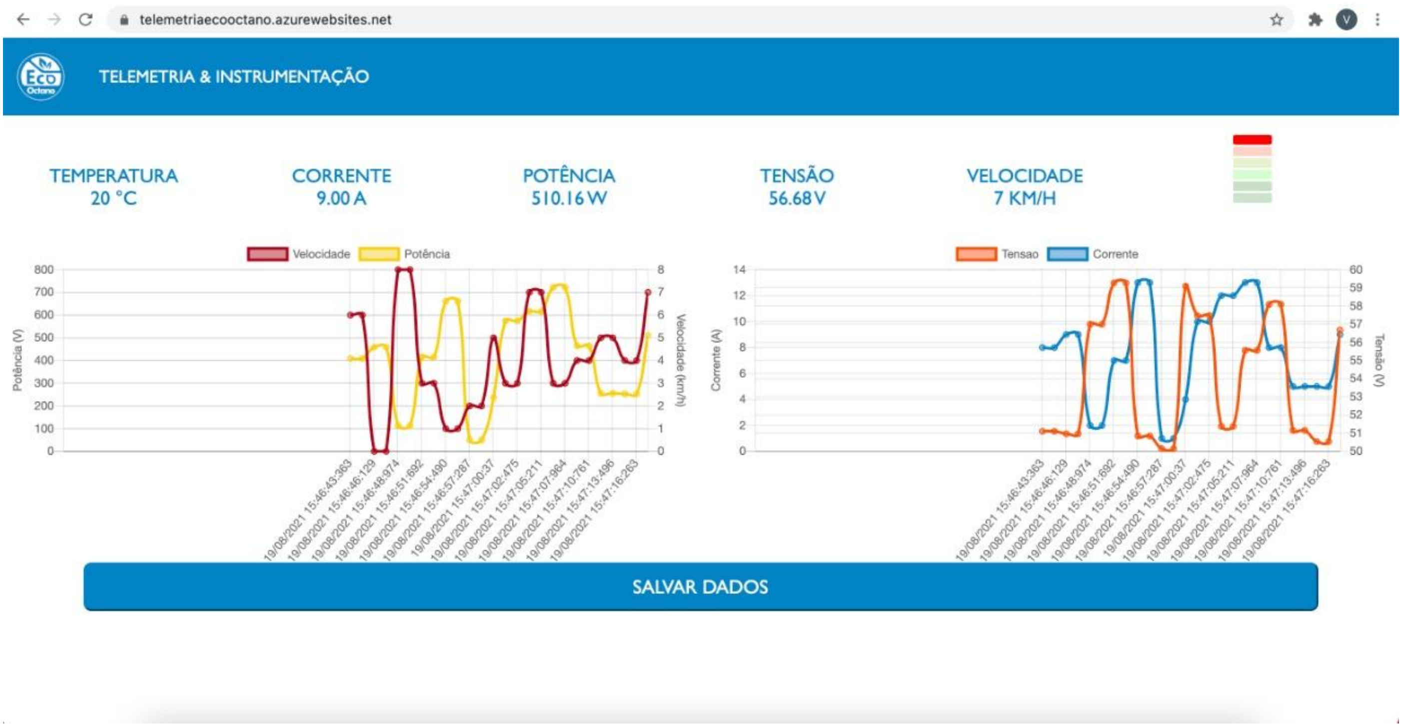

Figure 5: Developed Web App final result.

Showing, in order, the values of temperature, current, power, voltage, and speed. In the superior right corner, we have vertical bars that reference the current value, following a color scheme that shows when the level is dangerous.

In the middle of the page, we have two graphs, one of speed and power, and the other of voltage and current. At the bottom of the page, the button “Salvar dados” allows saving all the shown data and organizing it in a spreadsheet. The main purpose of having a button to save data after each race, is to make it easier to feed a database with the vehicle's performance on each lap. When compared to efficiency, they can help to determine the actions that guarantee the best result on the track. In addition, this data can be compared to the data saved on the SD card, directly from the PCB, in order to verify the reliability of the data displayed on the website.

Also, the pilot will have access to the speed information of the vehicle. When the driver has access to a speedometer, both his driving and the communication with the team in the box are improved. The pilot will have - in a suitable place inside the prototype - a display attached to the car that shows speed, for quick and easy reading. The information comes directly from the PCB without the need to use WiFi. Therefore, the data sampled for the pilot is not hostage to a possible signal drop.