INNOVATIVE PROJECT

Teacher: Thierry Monteil

Introduction and Context of the Project

As part of our fifth year of engineering studies at INSA Toulouse, we were led to carry out an Innovative Project, an initiative aimed at applying the theoretical knowledge acquired throughout our course to a concrete Internet of Things (IoT) project. This project took place over one semester and was carried out in collaboration with a Toulouse company, Maillon Mobility, specializing in the development of new means of transport.

The objective of the project was to integrate smart sensors into their main prototype, the Maillon Capitol, an electric tricycle intended for peri-urban mobility. Through this initiative, we adopted an Agile methodological approach, favoring iterative development in several sprints, in order to ensure better task management and progressive adaptation to the technical challenges encountered.

Choosing the Project Direction

During the initial brainstorming phase, several avenues for improving the tricycle were considered:

- An obstacle detection and prevention system to improve vehicle safety.
- A solution aimed at enhancing driver safety, in particular by integrating an intelligent protection mechanism.
- A system for measuring the user's heart rate to adjust the tricycle's electric assistance according to their physical effort.

After a feasibility study, we decided to focus on the third option, which seemed more concrete and feasible in the time available. The goal was to design a connected ecosystem integrating:

- A smart watch to retrieve the user's heart rate.
- A web application to display the data and allow the setting of the effort level (walk, sport, etc.).
- An ESP32 microcontroller responsible for receiving this information and adjusting the tricycle's motor assistance accordingly.

Task Distribution and Agile Methodological Approach

Our team, composed of four engineers, followed an Agile approach to structure the development process. We defined several sprints, each corresponding to a specific phase of the project:

- **Sprint 1** – Planning & development of the general architecture.
- **Sprint 2** – Establishing communication between the smartwatch and the application.
- **Sprint 3** – Integration and testing with the ESP32 microcontroller.
- **Finalization** – System performance analysis and adjustments.

Each team member was assigned a specific task:

- Two members worked on the test bench and the integration of the ESP32 with the tricycle motor.
- I was responsible for developing the user interface and establishing communication with the smartwatch API. My task involved developing a web application in React to display real-time data on heart rate, calories burned, and stress levels. These data were retrieved through a Python script using the Garmin Connect API, deployed with Flask to ensure seamless communication between the smartwatch and the application.
- Another member managed communication between the ESP32 and the phone, as well as back-end development for displaying cardiac data on the application.

Although the Agile approach allowed us to adapt to challenges encountered, hardware constraints and delays in receiving components limited its effectiveness, requiring us to adjust our sprints multiple times.

Implementation and Technical Challenges

Communication Between the Smartwatch and the Application

The first major difficulty involved integrating the smartwatch. Initially, we attempted to use the Garmin Connect API to retrieve the user's heart rate. However, this required an enterprise API key, which we could not obtain within the given timeframe. Faced with this constraint, we explored alternative solutions and ultimately decided to integrate an external heart rate sensor.

Establishing Communication Between the Phone and the ESP32

Another technical challenge was creating a robust and efficient link between the phone and the ESP32 microcontroller. Several communication options were considered:

- **Wi-Fi**: High energy consumption and dependence on a network connection.
- **NFC**: Too limited in terms of range and data throughput.
- **4G/5G**: Too complex and not suitable for the project.
- **Bluetooth Low Energy (BLE)**: Selected for its low energy consumption and compatibility with the ESP32.

The BLE implementation required developing two software modules:

- A C++ program on the ESP32 to receive data from the phone and transmit motor assistance.
- A TypeScript interface on the mobile application to send information to the microcontroller.

However, real-time data transmission between the microcontroller and the mobile application remains an unresolved challenge. We continue to explore different approaches to optimize this transfer smoothly and without excessive latency.

Analysis and Lessons Learned

This project has been an enriching experience in several ways:

- **Application of theoretical concepts** – We were able to put into practice concepts learned in class, particularly IoT communication, wireless protocols, and embedded application development.
- **Agile approach in a real-world context** – Sprint-based project management allowed us to organize our work efficiently, although material constraints sometimes required unexpected adjustments.
- **Development of a complete embedded solution** – I particularly enjoyed working on the interaction between hardware and software, especially the BLE communication between the microcontroller and the application.
- **Challenges with sensor integration** – The inability to access the Garmin API forced us to explore alternative solutions, highlighting the importance of adaptability in engineering.

Future Prospects and Improvements

Although we successfully developed a functional prototype, some improvements remain necessary:

- Optimizing communication between the microcontroller and the application to ensure smooth real-time data transmission.
- Integrating a dedicated heart rate sensor such as the Pulse Sensor coupled with an ESP32 microcontroller to avoid dependence on third-party APIs.
- Developing a native Android version of the application to enhance user experience and reduce connectivity constraints.

Conclusion

This project was a pivotal experience in my training as an engineer in automation and electronics. It allowed me to strengthen my skills in embedded development, IoT communication, and Agile project management. Despite the challenges encountered, it taught us to find alternative solutions to technical constraints while developing a structured and collaborative approach to teamwork.

We are now continuing our research to improve real-time data transmission, a key element for fully integrating the system in an industrial context.

Skill Matrix:

Service Oriented Architecture	**Expected**	**Estimated**
Solve a problem in a creative way	4	4
Develop the first stage of innovation	4	4
Understand production, validation, distribution, acceptability, and aftermath of innovation	4	4
Structure and lead an innovative project	4	4
Analyze a real-life problem	4	4
Suggest a technological solution to a problem	4	4
Implement a prototype to solve the problem	4	4
Present and debate (in English) the technical choice made	4	4
Produce a report (in English) for the developed project	4	4
Multi-disciplinary students work as a team	4	4