



Universidade do Minho
Escola de Engenharia

Mestrado Integrado em Engenharia Electrónica Industrial e Computadores

Project I

Car Cabin Control

Professor: Adriano Tavares

Authors:

João Faria A85632

Tiago Costa A86122

October 22, 2020

Contents

1	Problem Statement	4
2	Problem Statement Analysis	4
3	Market Research	5
3.1	Target Market.....	5
3.2	What is in the market?	6
4	System	7
4.1	System Overview	7
4.2	Requirements and Constraints.....	8
4.2.1	Project Requirements.....	8
4.2.2	Project Constraints	8
4.3	System Architecture	9
4.3.1	Hardware Architecture.....	9
4.3.2	Software Architecture	10
5	Reinforcement Learning.....	11
6	Gantt Diagram	12
7	System Analysis	12
7.1	Big Picture	13
7.2	Events	13
7.3	Use Cases.....	14
7.4	State Chart.....	15
7.5	Sequence Diagram	15
8	Mobile System.....	17
8.1	Events	17
8.2	Use Cases.....	17
8.3	State Charts	18
9	System Stack Diagram	19
10	Expected Budget	20

List of Figures

Figure 1: Problem Statement Analysis	5
Figure 2: Generic Wireless Temperature Controller	6
Figure 3: Dorman Climate Control System	6
Figure 4: System Overview	7
Figure 5: Hardware Architecture.....	10
Figure 6: Software Architecture	10
Figure 7: Reinforcement Learning.....	11
Figure 8: Gantt Diagram	12
Figure 9: Big Picture	13
Figure 10: Use Cases Diagram	14
Figure 11: State Chart Diagram	15
Figure 12: Sensor Data Handling Sequence Diagram	16
Figure 13: Secure Mode Activated Sequence Diagram	16
Figure 14: Use Cases for mobile application	18
Figure 15: Sequence Diagram for mobile conection	18
Figure 16: Sequence Diagram for mobile control	19
Figure 17: System Stack Diagram	20

List of Tables

Table 1: Cabin Events Table	14
Table 2: Mobile Events Table	17
Table 3: Expected Budget.....	20

1 Problem Statement

In a world where IoT is already ubiquitous, especially in the automobile industry, one seeks safeness but also a more comfortable driving experience. Besides that, a lot of automotive brands put a lot of focus on the infotainment part, giving detailed information about the car system. However, motor vehicle theft is still a problem that does not seem to diminish with the technological evolution of this particular niche. Over the last decade, in the United States, the number of vehicles each year stolen stagnated around 700000. With that in mind, in this course, the system developed will revolve around the car's cabin management, allowing the user to control for example its lights' gradient, temperature, humidity, as well as act as a security for the driver: in case of auto theft, the system will block the car after a user-defined time delay, prompted by an emergency button pressed, or by recognition of a user-defined voice command.

2 Problem Statement Analysis

Since we are connecting both (Embedded Systems and Project1) projects, in Project1, the STM-32 will be the centre of data processing. The STM must be able to communicate with Raspberry Pi to allow the driver to have absolute access and control about what is happening inside and around the car. Therefore, the STM board will process the data collected by the sensors (temperature, light, humidity and microphone) and present it in the display for reading the respective measurements as well as to be able for the user to control them. In the diagram below is represented a draft of the project features. Also, the user can interface with the system via bluetooth-based mobile application where he can read sensors data and send commands to define a given reference temperature to be attained in a defined deadline.

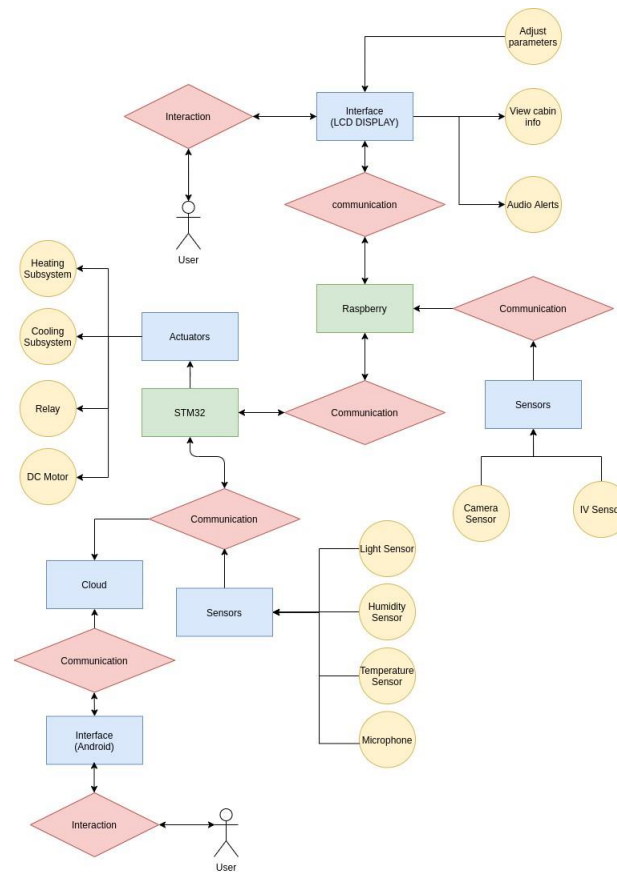


Figure 1: Problem Statement Analysis

3 Market Research

3.1 Target Market

It's important to perform a market research to analyze and understand competitor products and also how to best fit customer needs.

The developed product seeks to provide more comfort in the driving experience by providing a user-friendly mobile interface for temperature and ambient light control, and humidity feedback.

The product also has a built-in carjacking detection system that once activated, the infractor only has a brief time window to drive the vehicle until it reached the blocking state, that is, the vehicle becomes unable to be driven. To conclude, it is suitable to everyone that possess a automobile vehicle, especially, one with a closed cab.

3.2 What is in the market?

After a brief search in this automobile particular niche, one discovers some available products with similar functionalities.



Figure 2: Generic Wireless Temperature Controller

This product allows temperature measure, display and control locally or remotely via internet connection. It's wallet-friendly too.

Yet, it doesn't affect the brightness on car display nor measures its humidity, and forget about any system carjacking detection.



Figure 3: Dorman Climate Control System

In this case, the product above is able not only to control cab temperature but also airflow and has an A/C built-in. However, the product can't connect via Bluetooth, it is more expensive than ours, and it is only compatible with specific cars like Cadillac, Chevrolet and GMC.

Overall, our product combines the best of both worlds by providing the aforementioned services in a user-friendly way, not being restricted to any particular vehicle and not too costly.

4 System

4.1 System Overview

To read and control the forementioned features, the input/output list is below described:

Inputs:

- Bluetooth Connection for the user to control given temperature;
- Temperature Sensor, to measure the temperature inside the cabin; • Humidity Sensor, to read humidity inside the cabin;
- Light Sensor, to register light intensity inside or outside the cabin;
- Microphone, for voice detection and later recognition with emergency word;

Outputs:

- Bluetooth connection for displaying measurements back to user;
- Heating subsystem, to rise the temperature according to the user reference value;
- Cooling subsystem, to diminish temperature faster;
- Raspberry to communicate the information back to the display;
- Motor, to block the car when it's activated, after a user-defined delay following the voice command recognition;

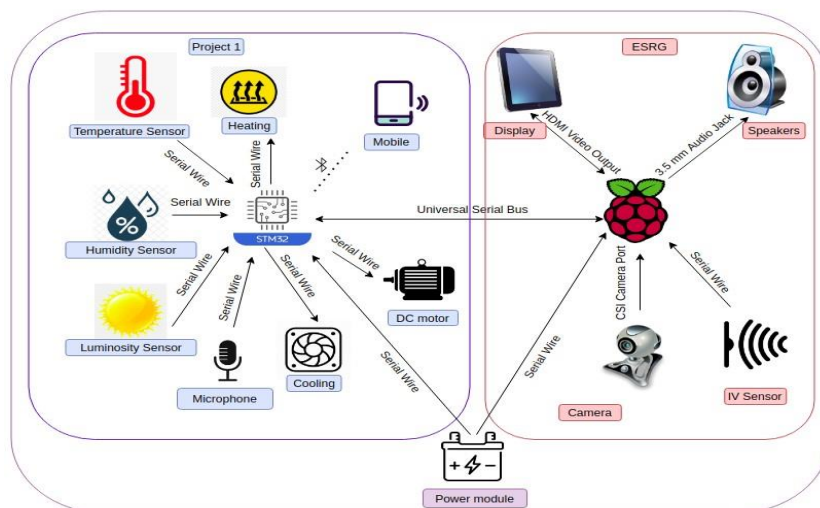


Figure 4: System Overview

4.2 Requirements and Constraints

Here we have 2 different types of requirements, functional requirements and non-functional requirements. The functional requirements must specify what the system is capable of doing and the non-functional requirements specify how the system will perform a certain action.

4.2.1 Project Requirements

Functional Requirements

- Measure and control the temperature inside the cabin, via a Reinforcement Learning PID as the control unit, and using the heating and cooling as actuators;
- Measure the humidity of the car;
- Read the light outside the car and change the brightness of the display accordingly, simulating a typical car dashboard;
- Allow user to connect via Bluetooth to system and receive sensor measurements, as well as be able to send commands to control cabin temperature within a pre-defined time.
- Record and recognize voice/word and cut power supply to the motor after a defined time period;

Non Functional Requirements

- Low cost;
- Physical Robustness;
- Reliable;
- Interoperability;
- Power Efficient;
- Safety;

4.2.2 Project Constraints

The constraints define the product limitations. They can be technical and non-technical. Technical constraints are associate with the technical part of the project whilst non-technical ones are project management related.

Technical Constraints

- Reinforcement Learning;
- IoT;
- STM32F767ZI board;
- FreeRTOS;

- Use, at least, 3 sensors;
- Object-oriented programming languages;
- Keil MKD-ARM (uVision) as the IDE;

Non-Technical Constraints

- Group composed by 2 members;
- Deadline near end of semester;
- Limited budget;
- Time to prototype;
- Soft Real-Time System;

4.3 System Architecture

The system hardware and the software architecture will be planned in more detail below to accomplish all the product features described in the aforementioned topics. This planning is part of a methodology that recommends breaking the global problem into minor problems to help to understand how things work independently. Finished the preparation, it is easier to build the project components in a structured fashion. So, the next figures present the established solution to fit the referenced architectures.

4.3.1 Hardware Architecture

In this section we take a view of Hardware Architecture and how the inputs and outputs are grouped and connected to the main processor that in this case is the board STM32F767ZI.

Temperature, light and humidity are measured by its respective sensors, processed in STM-32 and sent back to user.

App commands allow change of current temperature by communicating with STM-32 via bluetooth module and then actuating on the heating and cooling subsystem.

The board can send alerts back to Raspberry and receive commands from it to actuate on motor.

The car blocking system works by recognizing a specific word via microphone and then stopping the motor after a delay. Figure below illustrates that:

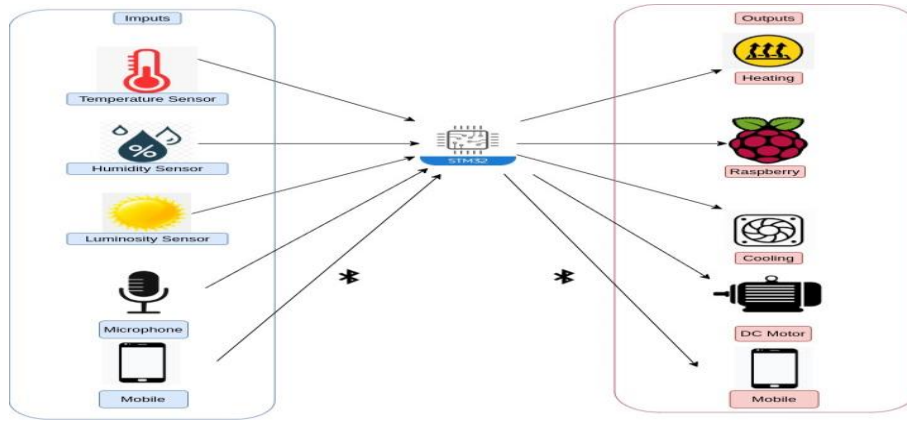


Figure 5: Hardware Architecture

4.3.2 Software Architecture

In the software architecture, specifically on the lower layer are represented the device drivers required for light, temperature, humidity sensors as well as for heating, cooling, motor with PWM driver, USB driver for raspberry communication and I2S for microphone interface.

The middle layer components are composed of sensor data acquisition and data processing modules, a reinforcement learning framework for the controller and finally bluetooth framework and database management for user communication and log sensorial information to the cloud, respectively.

Finally, the application layer is comprised of a graphical user interface and bluetooth communication for end user interaction.

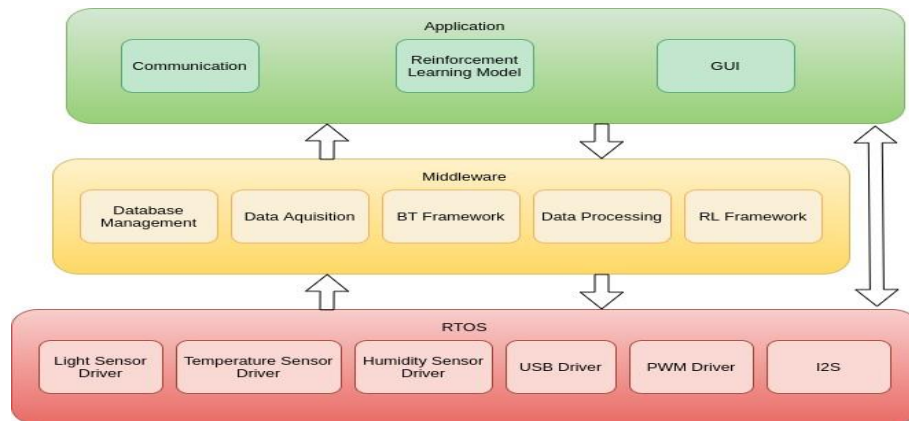


Figure 6: Software Architecture

5 Reinforcement Learning

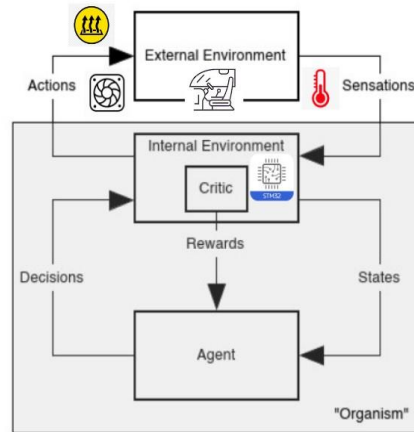


Figure 7: Reinforcement Learning

This project will have a reinforcement learning component. In this context, the goal is to train by trial and error a software algorithm (agent) to act as a temperature PID controller (policy) dictating actuator outputs (decisions) based on the current state, internal or external.

For example, the sensations are inputs signals from the temperature sensor and actions the voltage on the heating and cooling subsystems. The external environment is the cabin inside temperature while the internal environment is STM-32 board.

The critic component is responsible for fine-tuning the reward function that is responsible for changing the current temperature to user-reference value meeting defined deadlines while also conserving as much energy as possible. In conclusion, the problem is stochastic since the user can change at any given time reference temperature and time-dependent, sequential because the reward functions depend on past and present input.

7.1 Big Picture

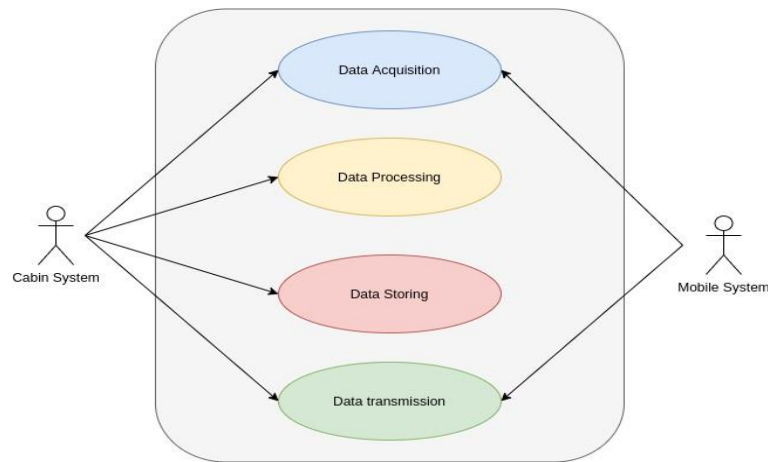


Figure 9: Big Picture

As illustrated in figure 9, the remote system, the cabin which has STM32, is responsible for data acquisition of the temperature, light, humidity sensors and user input regarding the carjacking detection system.

It is then processed and stored in it not only to allow bidirectional communication with the local system, the Bluetooth mobile app but also to apply the reinforcement learning algorithm over the temperature values stored, changing input voltage of the actuators and sending values over to the raspberry for display.

7.2 Events

For the local system, that is the cabin system it is composed of two types of events, synchronous for sensor data sampling and transmission to the mobile system and asynchronous for user input like enabling automatic light. The secure mode is triggered outside of both mobile and cabin systems. Note that all mobile-related events are only executable if a Bluetooth connection is established in the first place.

All information exchanged to mobile goes to display attached to Raspberry as well.

Event	System Response	Source	Type
Secure Mode Triggered	Set timer to block car	User	Asynchronous
App Commands Received	Put data in buffers to be read	Mobile System	Asynchronous
Connection Established & Sensors Data Available	Put data in buffers to be sent to mobile and Rasp	Cabin System	Synchronous
Sensors Sampling Time Elapsed	Read sensors data	Cabin System	Synchronous
Reference Temperature Changed	Initiate ML temperature control	Cabin System	Asynchronous
Connection Requested	Establish a Bluetooth connection with mobile	Mobile System	Asynchronous
Rasp Commands Received	Put data in buffers to be read	Raspberry	Asynchronous

Table 1: Cabin Events Table

7.3 Use Cases

Concerning the local use cases, as depicted in the following figure 10, the cabin system will interact with user and mobile system, acquiring data to process, store and send it back to the respective actuators, detailed in the diagram, only after a connection request is accepted.

The cab system is receptive of raspberry commands.

The RL actor is outside of the system acting as a PID controller only for temperature.

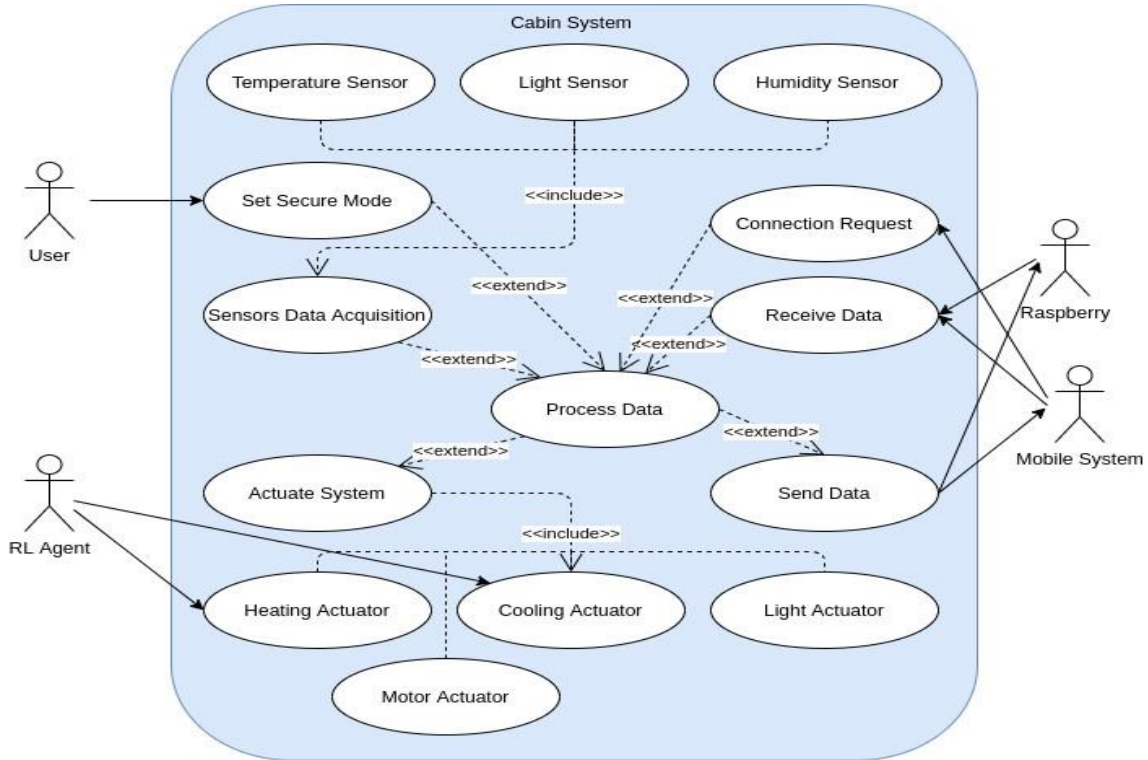


Figure 10: Use Cases Diagram

7.4 State Chart

The internal car system starts by performing an initial configuration regarding I/O, then enters an idle state where it can jump to the following states: reading sensors when sampling time elapsed, activate secure mode after the external trigger of the button pressed, receive via USB commands from raspberry and mobile phone only after an ACK is received when performing handshake related to Bluetooth connectivity.

After that, data is then processed and stored in the local system for actuation and transmission back to mobile and raspberry, returning then to the idle state, as shown in figure 11.

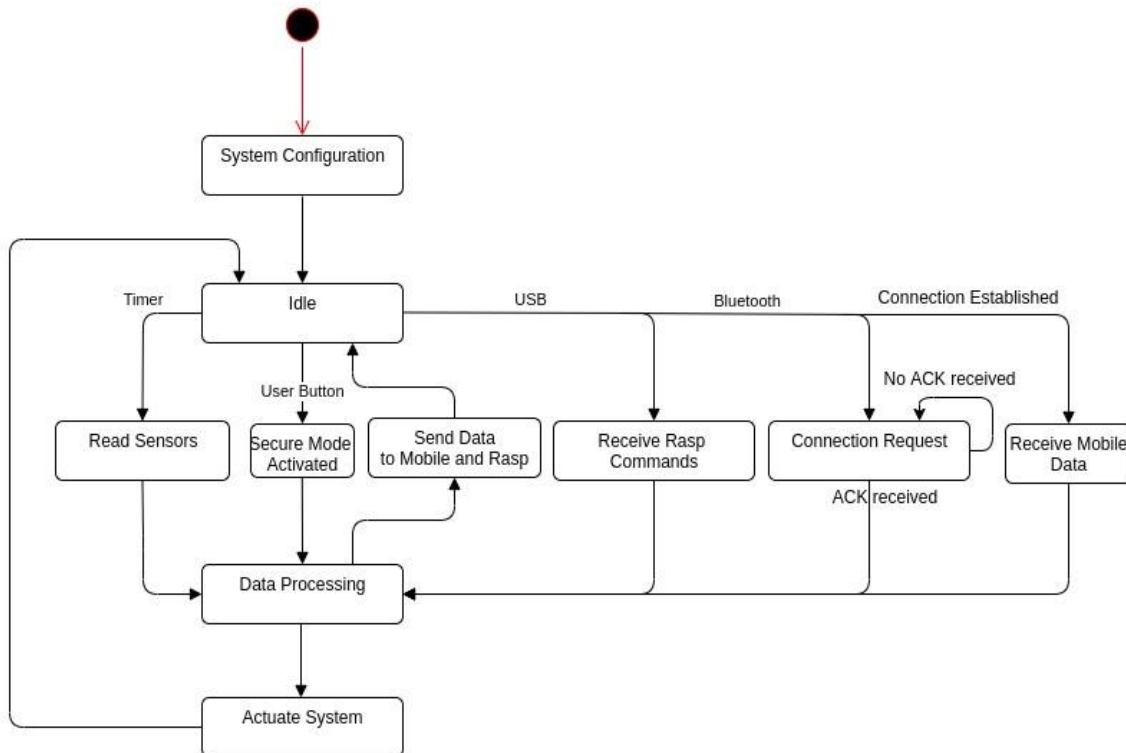


Figure 11: State Chart Diagram

7.5 Sequence Diagram

In figure 12 it is depicted the sequence of actions from the moment user sets a new reference value for temperature and light, to sensor data display back to him.

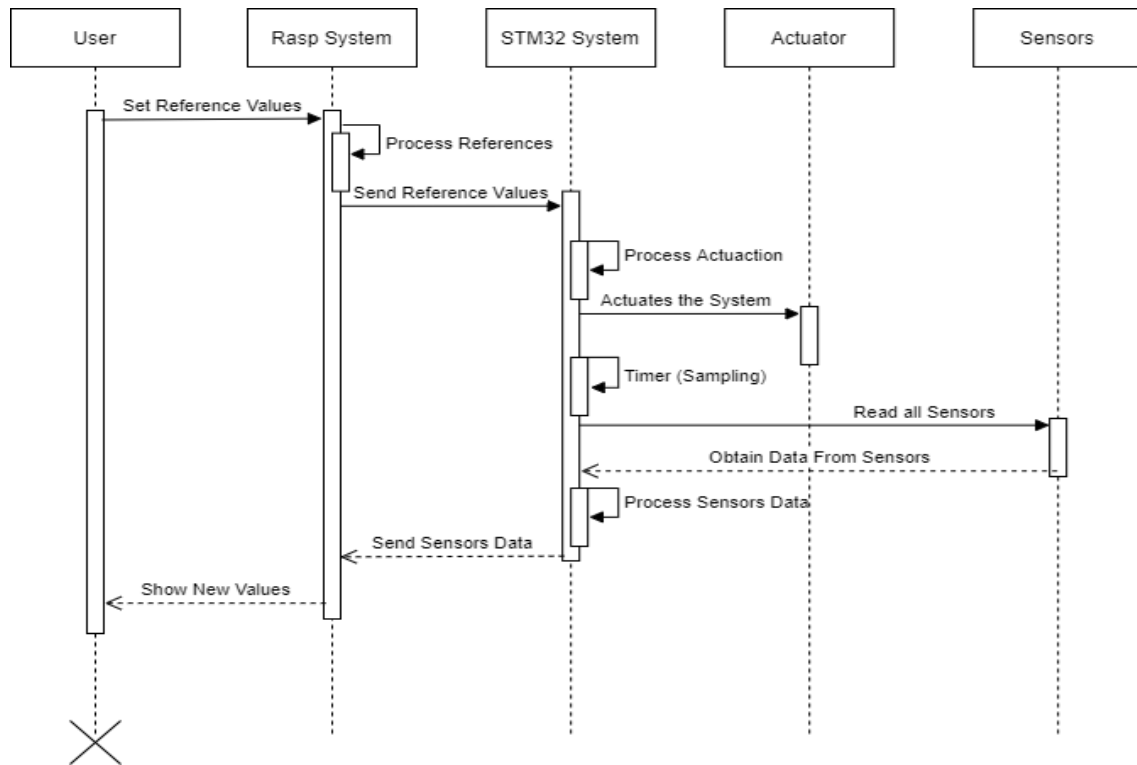


Figure 12: Sensor Data Handling Sequence Diagram

In figure 13, the sequence illustrates what happens when the user sets secure mode by pressing the button. That enables a timer that when it reaches the end, blocks the car, cutting off power to the motor. Then, another countdown starts to unblock the system.

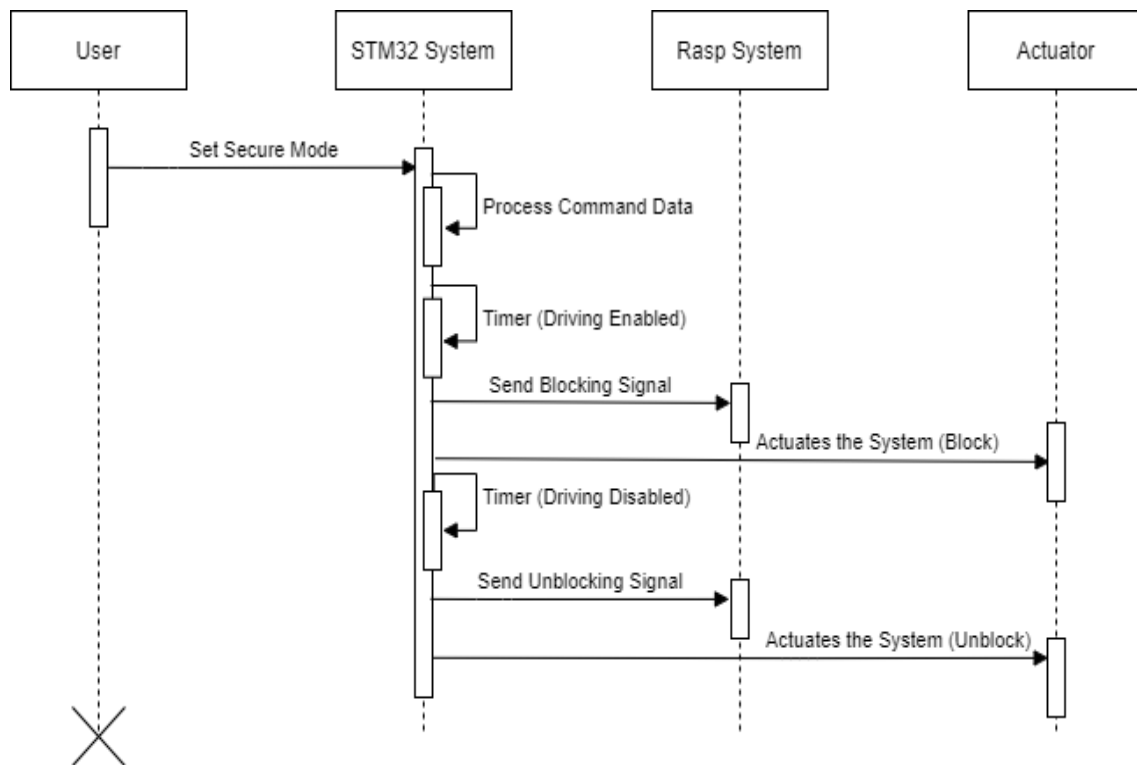


Figure 13: Secure Mode Activated Sequence Diagram

8 Mobile System

For the mobile system, the main events are depicted in the table below:

8.1 Events

Event	System Response	Source	Type
Bluetooth Enable Button Pressed	Turn on Bluetooth & scan near devices	User	Asynchronous
Cabin Device Button Pressed	Send a Bluetooth Request for Cabin System	User	Asynchronous
Data Available & Sensors Update Time Elapsed	Display Sensor Data	Mobile System	Synchronous
Reference Temperature Changed	Send Reference Value for RL control	User	Asynchronous
Brightness Toggle Button Pressed	Toggle between automatic/manual display	User	Asynchronous
Manual Brightness & User Input	Send commands to change the brightness according to slider position	User	Asynchronous

Table 2: Mobile Events Table

Note that almost all operations are triggered by the user and the only event that is system reactive is displaying data values, in a synchronous way, after an initial connection to Bluetooth server counterpart in the cabin system.

8.2 Use Cases

Text.

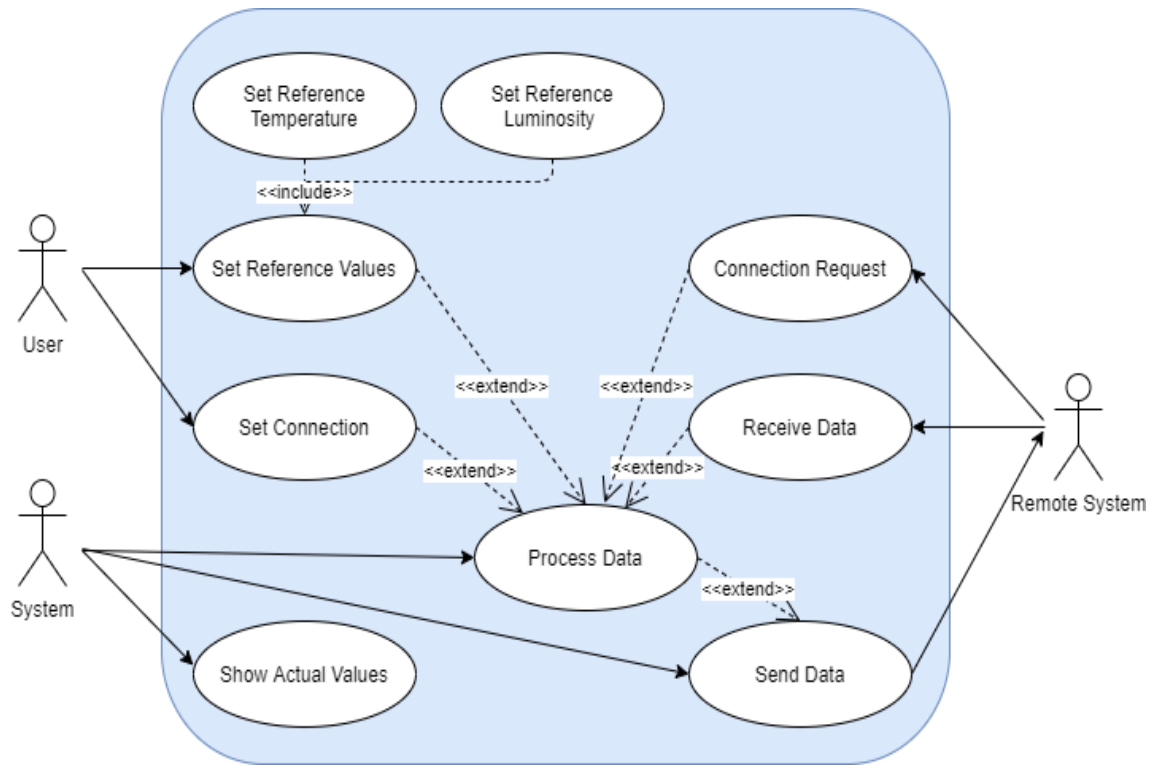


Figure 14: Use Cases for mobile application

8.3 State Charts

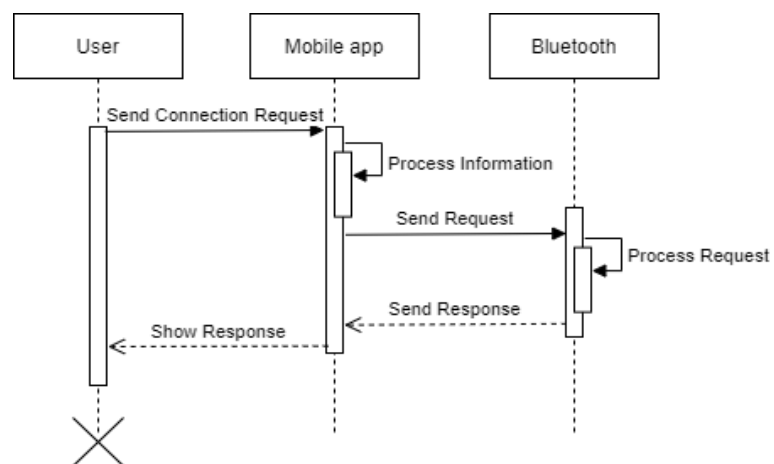


Figure 15: Sequence Diagram for mobile conection

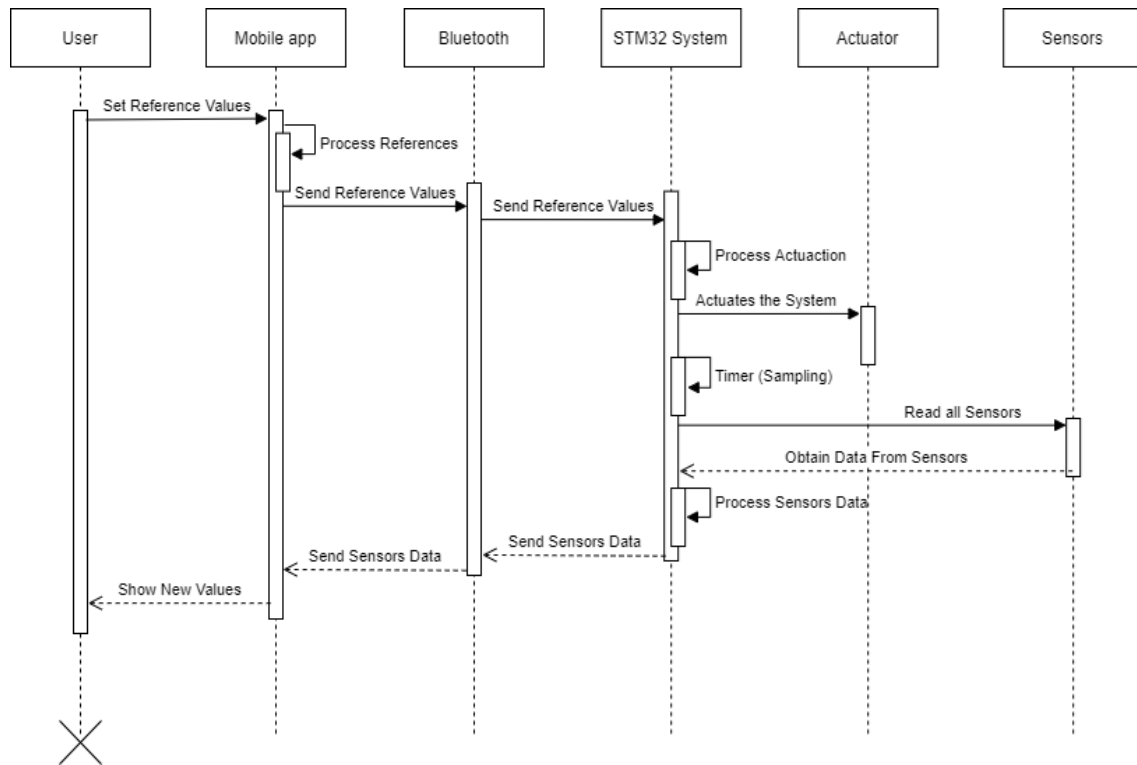


Figure 16: Sequence Diagram for mobile control

9 System Stack Diagram

The system consists of the main processing unit (STM32), a light sensor (TSL2561), a temperature and humidity sensor (DTH11), a switch, a Bluetooth module (HC05), a fan, an electrical resistance, and an engine.

The sensors will obtain information about the environment inside the vehicle, which will be further processed by the processing unit, which in turn, will act on the heating and ventilation elements according to the reinforcement learning algorithm.

All the information will be transmitted to the remote system by Bluetooth, as well as giving instructions for the desired values of the three variables measured.

This information and adjustments can also be viewed and performed using an existing Linux application in the infotainment system, for this communication, there is a USB connection.

The safety system will be activated by the switch that will start a timer, during this period the driver will be able to safely leave the vehicle, at the end of this time, the engine will stop running and the vehicle will be immobilized, causing the carjacker leaves the site, starting a second timer during which the vehicle is inoperative, when it finishes, the vehicle can be driven again.

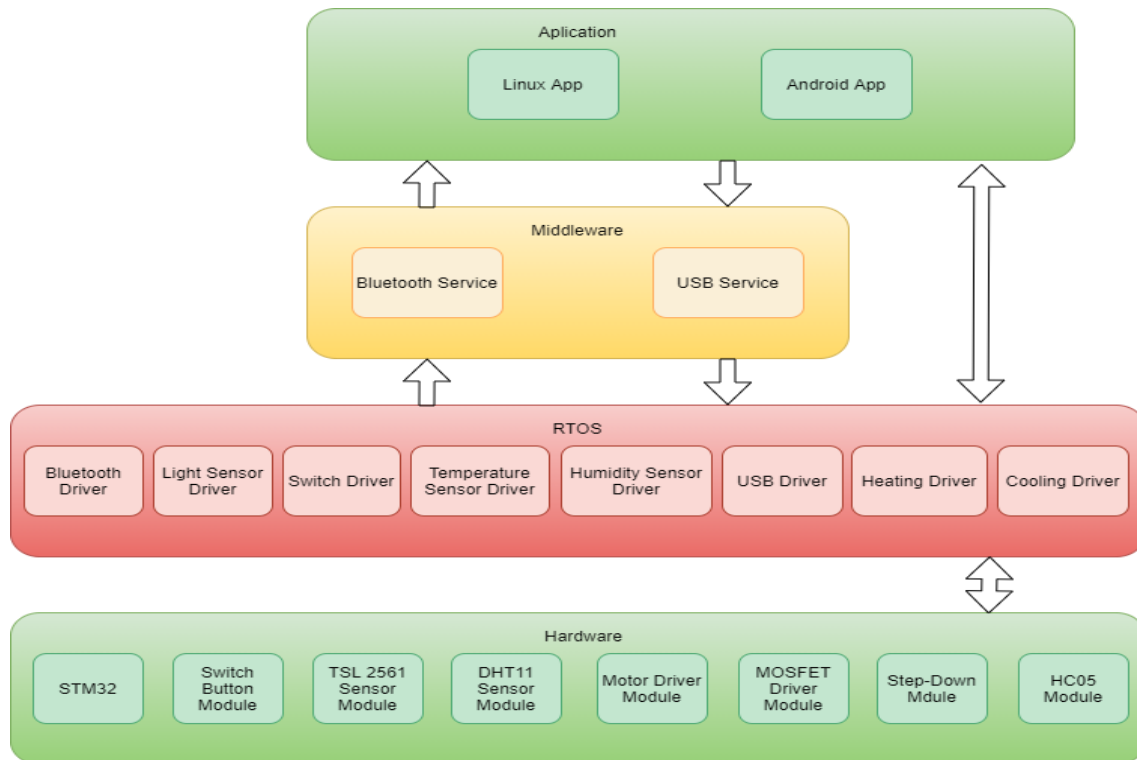


Figure 17: System Stack Diagram

10 Expected Budget

Text.

Quantity	Product	Price Unit (€)
1	STM32F767ZI	28
1	HC05	5
1	Luminosity Sensor	7,5
1	Temperature and Humidity Sensor	2
1	PWM Motor Driver	2
2	PWM Driver	1
1	Switch Module	1
1	Step-Down	1
1	Materials to buil Model	30
Total		80

Table 3: Expected Budget