Universidade do Minho
Escola de Engenharia

Mestrado Integrado em Engenharia Eletrónica Industrial e Computadores

Project I

Car Cabin Control

Professor: Adriano Tavares

Authors: João Faria A85632
Tiago Costa A86122

October 22, 2020

Contents

List of Figures

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 also 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.

Figure 1: Problem Statement Analysis

3 System

3.1 System Overview

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

Inputs:

- 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:

- 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;
- Relay, to block the car when it's activated, after a user-defined delay following the voice command recognition;

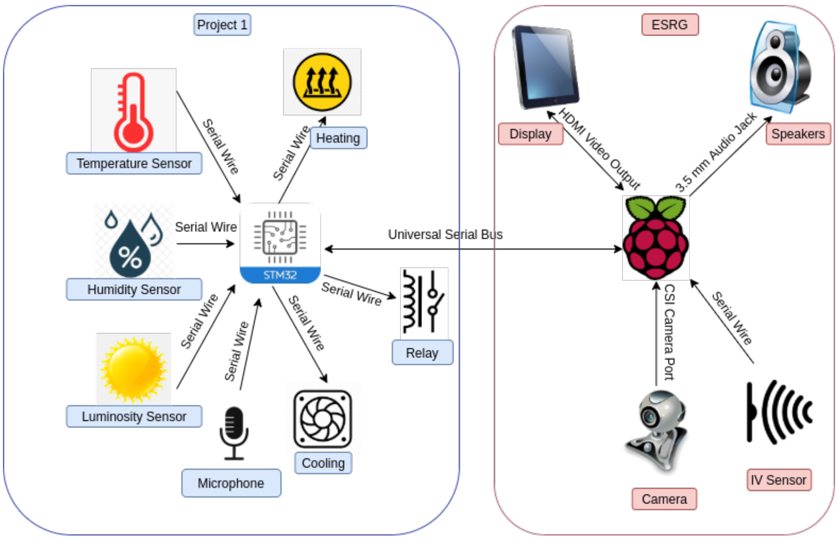


Figure 2: System Overview

3.2 Requirements and Constraints

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

3.2.1 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;
- Record and recognize voice/word and activate the relay after a defined time period;

Non Functional Requirements

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

3.2.2 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

- Use Reinforcement Learning;
- Use STM32F767ZI board;
- Use FreeRTOS;
- Use, at least, 3 sensors;
- Use object-oriented programming languages;
- Use 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;

3.3 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.

3.3.1 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. Figure below illustrates that:

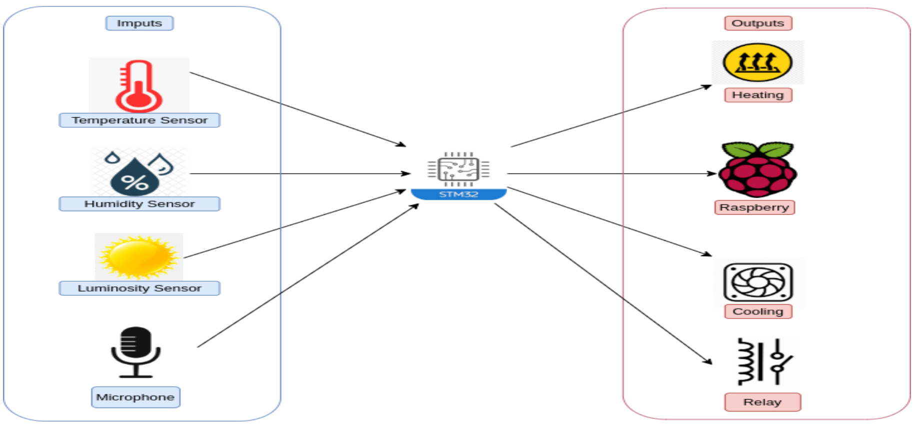


Figure 3: Hardware Architecture

3.3.2 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 and communicating with Raspberry, using an USB driver. The middle layer components are data acquisition and data processing. Finally, the application layer is composed of the reinforcement learning and communication part.

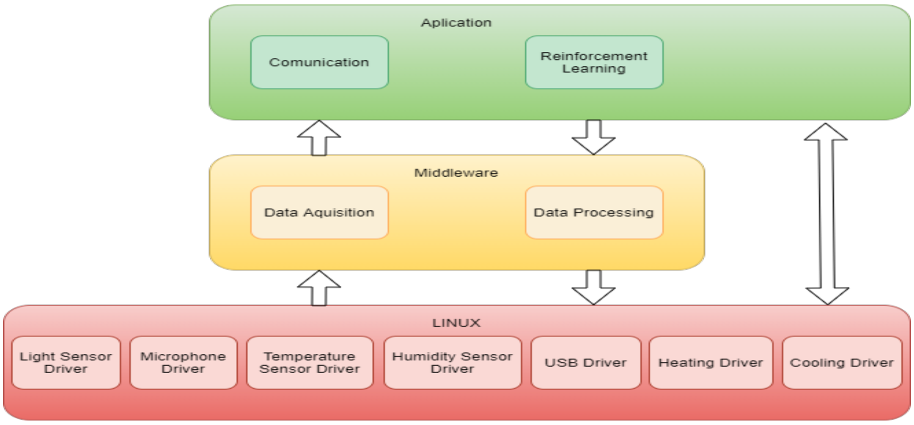


Figure 4: Software Architecture

4 Machine Learning

Machine learning is a form of AI that enables a system to learn from data rather than through explicit programming. As the algorithms ingest training data, it is then possible to produce more precise models based on that data. A machine-learning model is the output generated when you train your machine-learning algorithm with data. After training, when you provide a model with an input, you will be given an output. For example, a predictive algorithm will create a predictive model. Then, when you provide the predictive model with data, you will receive a prediction based on the data that trained the model.

4.1 Reinforcement Learning

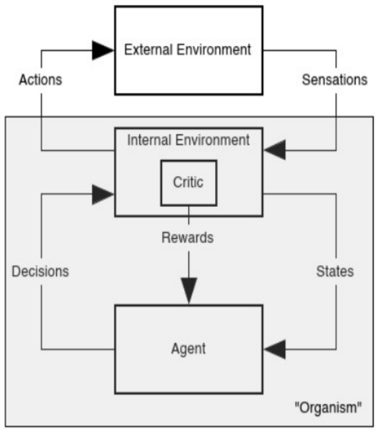


Figure 5: Gantt Diagram

Reinforcement learning is a behavioural learning model, an alternative branch to machine learning. The algorithm receives feedback from the data analysis, guiding the user to the best outcome. Reinforcement learning differs from other types of supervised learning because the system isn't trained with the sample data set, it learns through trial and error. Therefore, a sequence of successful decisions will result in the process being reinforced, because it best solves the problem at hand.

Within the scope of this project, the reinforcement learning will be applied to the PID controller (policy) to control the temperature inside the car cabin.

Given that, one proposes a reinforcement learning approach to the temperature control, that can acquire a set of rules enabling an agent to heat a room to the desired temperature at a defined time while conserving as much energy as possible.

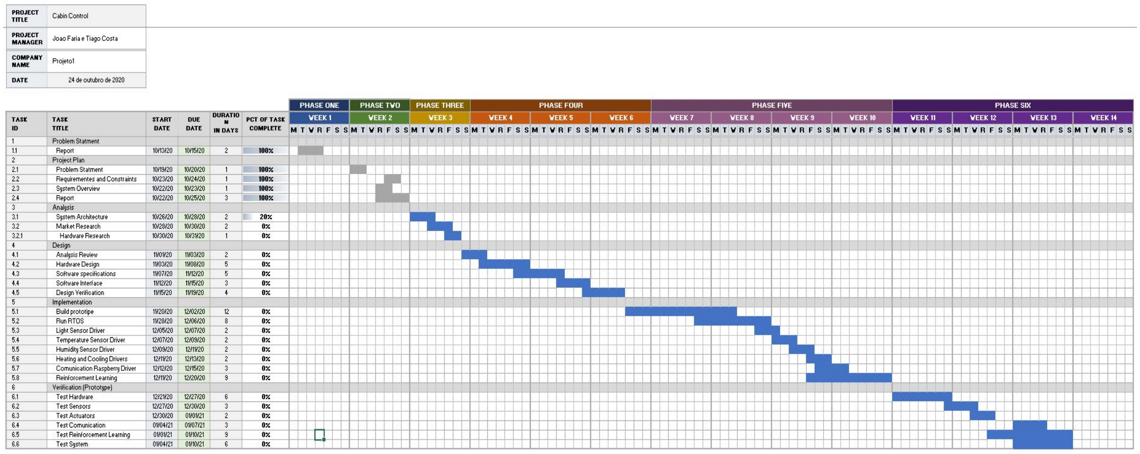
Because of that, the goal of reinforcement learning is to train an agent (software algorithm) to complete a task within an uncertain environment. In this case, the environment is stochastic since the user should be able to define the desired temperature and associated time duration, at any given time. The problem is sequential and time-dependent because the rate at which the current cabin temperature changes varies with how far it is from the reference value and the time window available for convergence.

The agent receives observations (in this case, the temperature measurements) and a reward from the environment and sends actions to the environment. The reward is a measure of how successful action is with respect to completing the task goal. In this context, the actions are the manipulated variables like the voltage in the thermal resistor or in the cooler fan and observations the user temperature referenced value.

The internal environment depicted is the STM-32 board where is located the learning algorithm that dictates (decision) the next state, influencing the external environment which is the car cabin, already mentioned.

Finally, the reward function should be established such that there is a compromise between minimizing steady-state error and the control effort, like, controlling cabin temperature to the desired temperature at a defined time while conserving as much energy as possible.

5 Gantt Diagram



PROJECT TITLE	Cabin Control
PROJECT MANAGER	Joao Faria e Tiago Costa
COMPANY NAME	Projeto1
DATE	24 de outubro de 2020

TASK ID	TASK TITLE	START DATE	DUE DATE	DURATION IN DAYS	PCT OF TASK COMPLETE
1	Problem Statment				
1.1	Report	10/13/20	10/15/20	2	100%
2	Project Plan				
2.1	Problem Statment	10/19/20	10/20/20	1	100%
2.2	Requirementes and Constraints	10/23/20	10/24/20	1	100%
2.3	System Overview	10/22/20	10/23/20	1	100%
2.4	Report	10/22/20	10/25/20	3	100%
3	Analysis				
3.1	System Architecture	10/26/20	10/28/20	2	20%
3.2	Market Research	10/28/20	10/30/20	2	0%
3.2.1	Hardware Research	10/30/20	10/31/20	1	0%
4	Design				
4.1	Analysis Review	11/01/20	11/03/20	2	0%
4.2	Hardware Design	11/03/20	11/08/20	5	0%
4.3	Software specifications	11/07/20	11/12/20	5	0%
4.4	Software Interface	11/12/20	11/15/20	3	0%
4.5	Design Verification	11/15/20	11/19/20	4	0%
5	Implementation				
5.1	Build prototipe	11/20/20	12/02/20	12	0%
5.2	Run RTOS	11/28/20	12/06/20	8	0%
5.3	Light Sensor Driver	12/05/20	12/07/20	2	0%
5.4	Temperature Sensor Driver	12/07/20	12/09/20	2	0%
5.5	Humidity Sensor Driver	12/09/20	12/11/20	2	0%
5.6	Heating and Cooling Drivers	12/11/20	12/13/20	2	0%
5.7	Comunication Raspberry Driver	12/12/20	12/15/20	3	0%
5.8	Reinforcement Learning	12/11/20	12/20/20	9	0%
6	Verification (Prototype)				
6.1	Test Hardware	12/21/20	12/27/20	6	0%
6.2	Test Sensors	12/27/20	12/30/20	3	0%
6.3	Test Actuators	12/30/20	01/01/21	2	0%
6.4	Test Comunication	01/04/21	01/07/21	3	0%
6.5	Test Reinforcement Learning	01/01/21	01/10/21	9	0%
6.6	Test System	01/04/21	01/10/21	6	0%

Figure 6: Gantt Diagram