EMBEDDED SYSTEMS
RESEARCH
GROUP

University of Minho
School of Engineering

Integrated Master in Industrial Electronics and Computers Engineering

Project 1

Zephyrus

Hand Motion-Controlled Remote Car

Authors:
Hugo Carvalho A85156
José Mendes A85951

Professor:
Dr. Adriano Tavares

October 2020

Contents

List of Figures

1 Problem Statement

Recent developments in research of motion-tracking and gesture-controlled robotic products have shone a new light on entertainment, special needs teaching and accessibility-focused applications. Products like motion-sensing console controllers, robotic arms, and prosthetics have popularized and extended the knowledge on gesture and impulse-based technologies throughout the past few years.

To provide a better understanding of the concepts associated with the forementioned technologies, the project will consist of a gesture-controlled car with a wristband as a remote controller. The user will be able to accelerate and steer the vehicle using natural wrist and finger movements.

2 Problem Statement Analysis

From the Problem Statement Analysis, one can extrapolate an overview of the main components, features they provide and relationships between them. The schematic in figure 2.1 describes a system comprised by:

- A **Local Computational Unit**, which will control the wheels of the car in accordance to the user inputs, which it receives from the Remote Computational Unit;
- A **Remote Computational Unit**, which senses user input and converts it into a digital format, sending it to the Local Computational Unit.

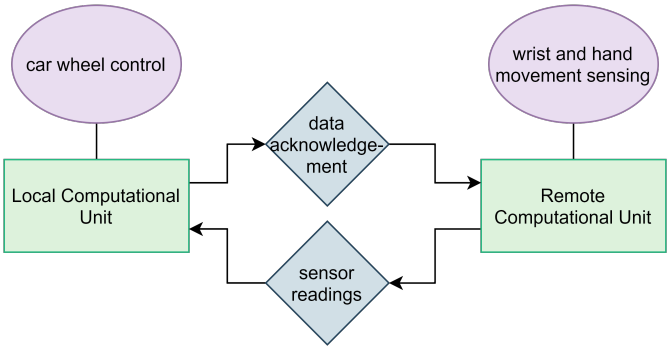

Figure 2.1: Problem Statement Diagram

3 System Overview

The diagram in figure 3.1 represents an initial big picture of the project to facilitate objective identification. Concerning the diagram, one can identify three main blocks:

- The **Local Board** block;
- The **Remote Board** block;
- The **External Environment** block.

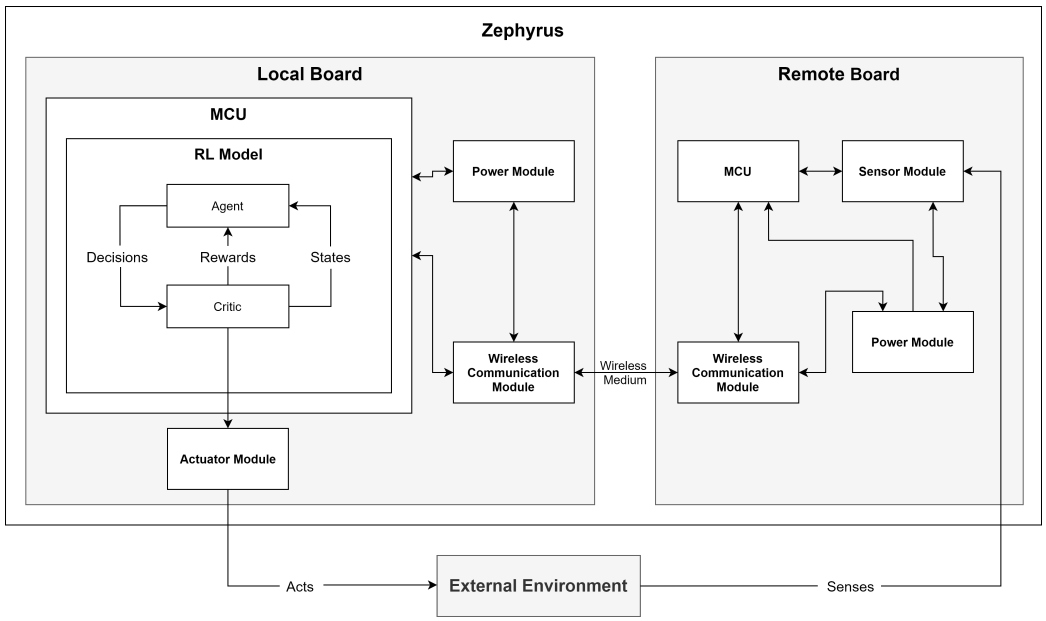

Figure 3.1: System Overview Diagram

3.1 Reinforcement learning

In figure 3.1 one can identify three main components of an Actor-Critic Reinforcement Learning Model:

- The **Actor/Agent** is the controlling entity of the system, which dictates actuator outputs based on the current state of the system, internal and external.
- The **Critic** is the entity that works to tune the behaviour of the Actor through a reward-based policy.

- The **External Environment** can be identified as everything that is not the Actor or Critic but is influenced by the Actor's actions and helps to stimulate it.

The model is time-dependent and sequential, since the reward calculation is a sum of all rewards from the past and the present. The environment is also stochastic because it's non-deterministic, meaning, the inputs change in time.

4 Requirements and Constraints

The development requirements are divided into functional and non-functional if they are main functionalities or secondary ones, respectively. Additionally, the constraints of the project are classified as technical or non-technical.

4.1 Requirements

Functional Requirements

- Sense the user inputs and generate the band outputs;
- Take the wristband inputs and generate the desired motor outputs;
- Use wireless communication between the local and remote systems;
- Control the establishment and closure of the connection from the remote system.

Non-Functional Requirements

- Have low power consumption;
- Have low latency in the connection between local and remote;
- Have a comfortable and practical wristband/glove;

4.2 Constraints

Technical Constraints

- Use the Nucleo-STM32F767ZI as the development board;

- Use at least three different types of sensors;
- Use FreeRTOS as the RTOS;
- Use a controller based on Reinforcement Learning following the Actor-Critic method;
- Use Keil MDK-ARM as the development environment;
- Use Object-Oriented Programming Languages;

Non-Technical Constraints

- Project deadline at the end of the semester;
- Pair workflow;
- Limited budget;

5 System Architecture

5.1 Hardware Architecture

The diagram in figure 3.1 represents an initial big picture of the project to facilitate objective identification. Concerning the diagram, aside from the External Environment, one can identify four main blocks:

- The **Local Board**. This unit is comprised by the main MCU (in the STM32 board) for housing the RL Model and intermediate its interactions with the world, as well as the Actuator Module, for driving the wheel motors, the Power Module, for power delivery and management, and the Wireless Communication Module, for facilitating the communication with the Remote Board through a wireless medium;
- The **Remote Board**. This unit contains the Sensor Modules network of the project for sensing the environment, its own MCU, whose job is to concentrate and multiplex the sensor network, sending data through its own Wireless Communication Module, and the Power Module which will provide all of the other modules with power.

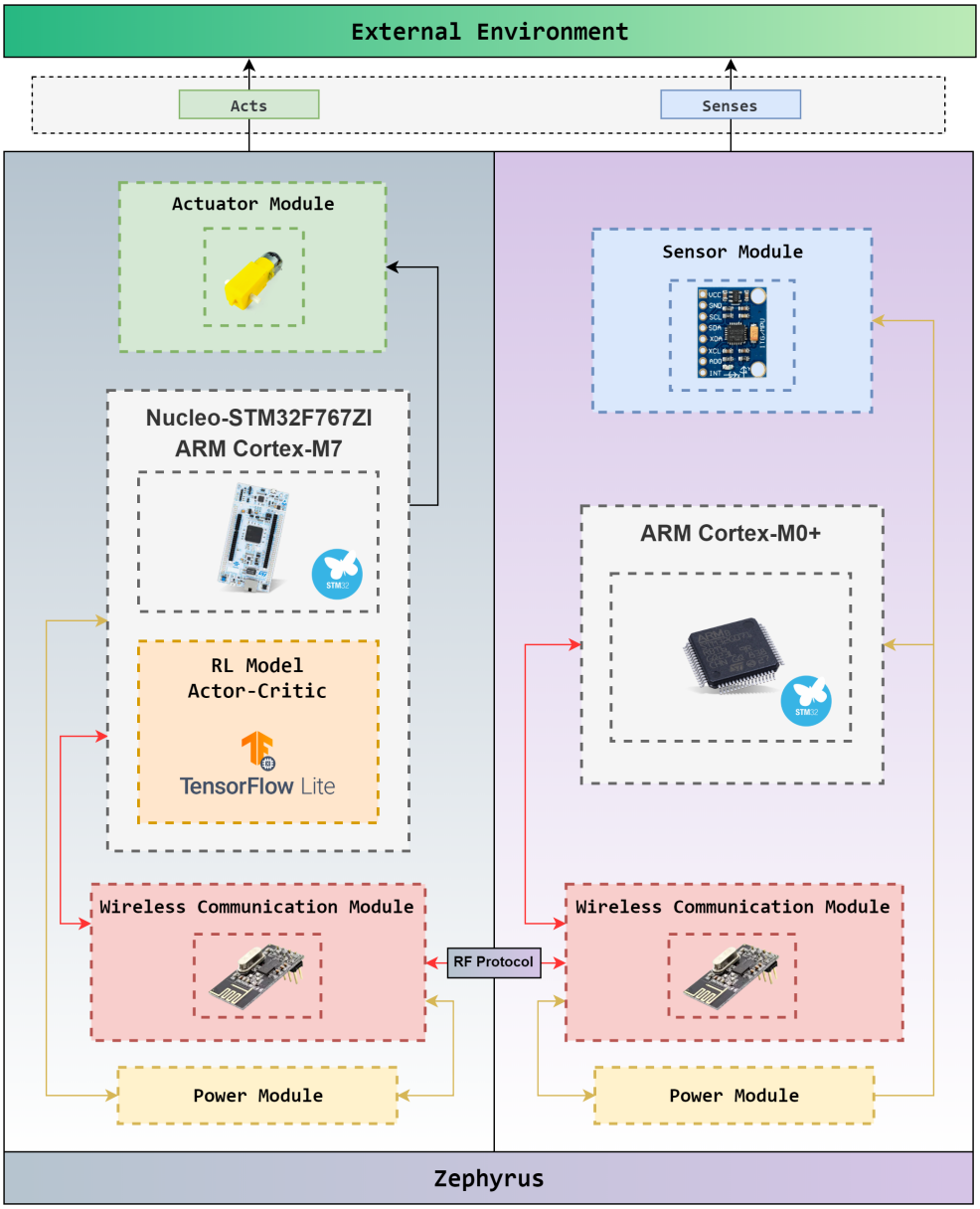

Figure 5.1: Hardware Architecture Diagram

5.2 Software Architecture

The software architecture (figure 5.2) is divided into three layers, those being:

- The **Operating System** layer, which is comprised by the Operating System drivers and Board Support Packages;

- The **Middleware** layer, which includes software for abstracting the lower level packages and streamlining the development of complex algorithms;
- The **Application** layer, where the core functionality of the program is built, with resource to the API's in the lower level layers.

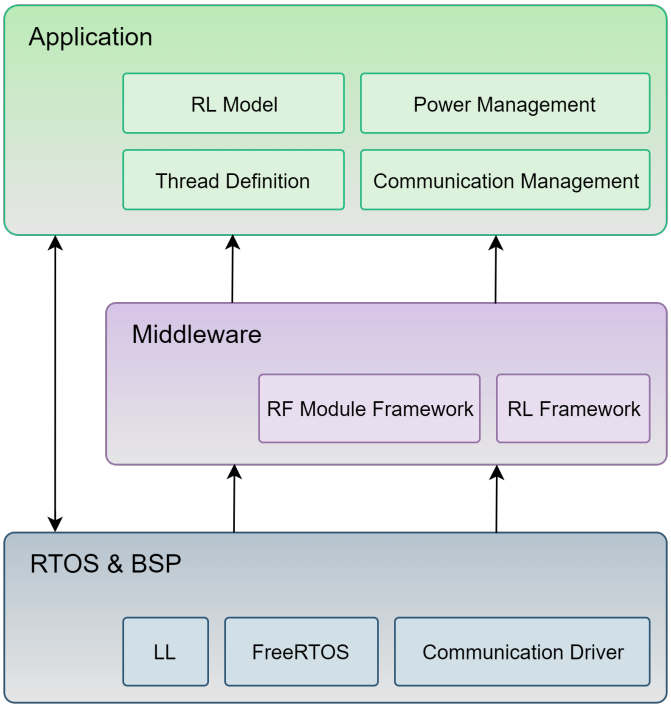

Figure 5.2: Software Architecture Diagram

6 Gantt Diagram

	Task Name	Duration	Start	ETA
1	**M0: Planning**			
1.1	Project Choice	2 days	15.10.20	16.10.20
1.2	Problem Statement	3 days	17.10.20	19.10.20
1.3	Project Plan	7 days	20.10.20	26.10.20
2	**M1: Analysis**			
2.1	System Overview	2 days	20.10.20	21.10.20
2.2	Requirements and constraints	1 day	22.10.20	22.10.20
2.3	System Architecture	2 days	23.10.20	24.10.20
2.4	System Analysis	2 days	25.10.20	26.10.20
2.5	Market Research	2 days	27.10.20	28.04.12
2.6	Estimated Budget	2 days	29.10.20	30.10.20
2.7	Diagrams	10 days	31.10.20	9.11.20
3	**M2: Design**			
3.1	Analysis Review	3 days	10.11.20	12.11.20
3.2	Hardware Specification	10 day	13.11.20	22.11.20
3.3	Component Shipping	14 days	23.11.20	7.12.20
3.4	Software Specification	10 days	18.11.20	27.11.20
3.5	RL Model Design and Training	8 days	28.11.20	5.12.20
4	**M3: Implementation**			
4.1	Hardware Testing	7 days	8.12.20	11.12.20
4.2	Software Implementation	32 days	1.12.20	31.12.20
4.3	Hardware Implementation	32 days	1.12.20	31.12.20
4.4	RL Model Conversion	5 days	1.12.20	5.12.20
5	**M4: Testing & Verification**			
5.1	Software Unit Testing	4 days	1.1.21	4.1.21
5.2	Software Integrated Testing	5 days	4.1.21	8.1.21
5.3	Hardwareware Unit Testing	4 days	1.1.21	4.1.21
5.4	Hardware Integrated Testing	5 days	4.1.21	8.1.21
5.5	Verification	3 days	9.1.21	11.1.21

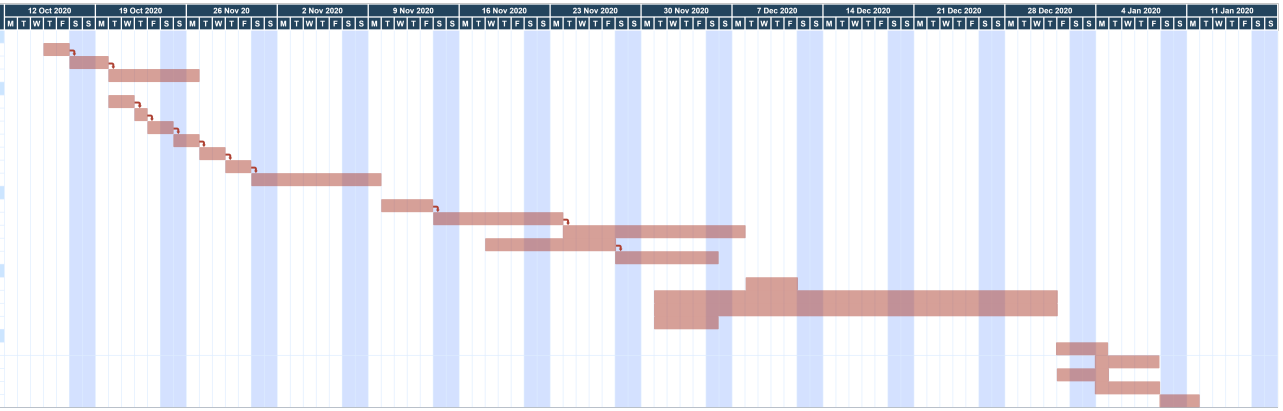

Figure 6.1: Gantt Diagram