CENTROALGORITMI

Universidade do Minho
Escola de Engenharia

SELF-BALANCING ROBOT

Authors:
Ailton Lopes A72852
José Oliveira A65308

Project Analysis
Project 1 MIEEICOM
2016/2017

Professor:
Adriano Tavares
José Mendes

Contents

Problem Statement

The system to be implemented in this project is similar to the inverted pendulum. The inverted pendulum system is an example frequently found in control systems manuals, the pendulum is mounted on a platform and is balanced by controlling the movement of the platform.

Since the launch of the Segway this type of system is being used in a larger proportion, particularly in the transportation field.

A Self-balancing robot it's a two wheel robot that balance itself on the vertical axis using their motors movements according to the angle and position where it is falling, the robot angle and position it's calculated in real-time using the information measured for the inertial measurement unit (IMU). The inertial measurement unit (IMU) is an electronic device that measures and reports a body's specific force, angular rate, and the magnetic field surrounding the body, using a combination of accelerometers and gyroscope.

Aims

This main purpose of the project is to design and implement an embedded system (Self-balancing robot) with the capability of balancing itself on the vertical axis, capable of resisting outside forces and performing commands sent by the user. The commands can be to move forward, move backward, turn in one direction, among others.

The communication between the embedded system and the user will be through Bluetooth.

Constraints

- Use the STM32F4DISCOVERY board;
- The development environment is Keil MKD-ARM;
- Use a PID controller integrating at least 3 sensors;
- 3 layers of hardware and respective software:
    1. Integrated DSP;
    2. ARM Cortex-M processor;
    3. Desktop application.

Requirements

- Acquire and store sensor data;
- Process the data to be used on the PID controller;
- Make the robot have a steady balance;
- Receive gain readjustment and movement commands from the PC;
- Send the sensor data to be displayed and stored on the PC

Market study

In the market we can find a big diversity of vehicle or robots that uses the same principles as the robot we pretend to implement.

The **Segway PT** is a two-wheeled, self-balancing, battery-powered electric vehicle invented by Dean Kamen. It is produced by Segway Inc. of New Hampshire.

Figure 1 – Segway PT

A **self-balancing scooter** or self-balancing two-wheeled board commonly referred to as a "hoverboard", is a type of portable, rechargeable battery-powered scooter. They typically consist of two wheels arranged side-by-side, with two small platforms between the wheels, on which the rider stands.

Figure 2 – self-balancing scooter

Steve Hassenplug's LegWay is a two wheel robot that uses RCX Lego. It balances using the IR Lego sensor to detect the floor proximity. Can follow a black line and/or spin in place. (Hassenplug, 2007)

Figure 3 – Steve Hassenplug's LegWay

Resources

1. **Hardware:**

In the implementation of this project the following hardware components will be used:

- STM32F4-Discovery,
- Microcontroller STM32F407VGT6,
- LIS3DSH 3-axis accelerometer,
- IMU MPU9255 (3-axis accelerometer; 3-axis gyroscope; 3-axis digital compass),
- Batteries,
- Bluetooth module HC – 06,
- DC Motors with encoder,
- USB to TTL converter,
- Motor Driver L298N.

2. **Software**

The software that are going to be part of the implementation of the project are:

- Keil uVision IDE,
- FreeRTOS.
- MATLAB,
- Visual Studio.

3. **User/Host**

To communicate with the robot, the user will use a computer running the application by the developers

System

The system its divided in two main segments, the Robot system and the Remote Desktop system.

Robot System

1. System Overview

The robot system is composed by a SRM32F4 board, to the board is connected sensors for the measurement of the position of the robot in real-time, a Bluetooth module for the communication with the desktop application and motors for the movement and control of the robot. To better control the motor it's uses in all motors encoders to measure the exactly position of the motor and the velocity. Since the board PWM do not have enough power to control the motor we are using a motor Drive that will control the motor.

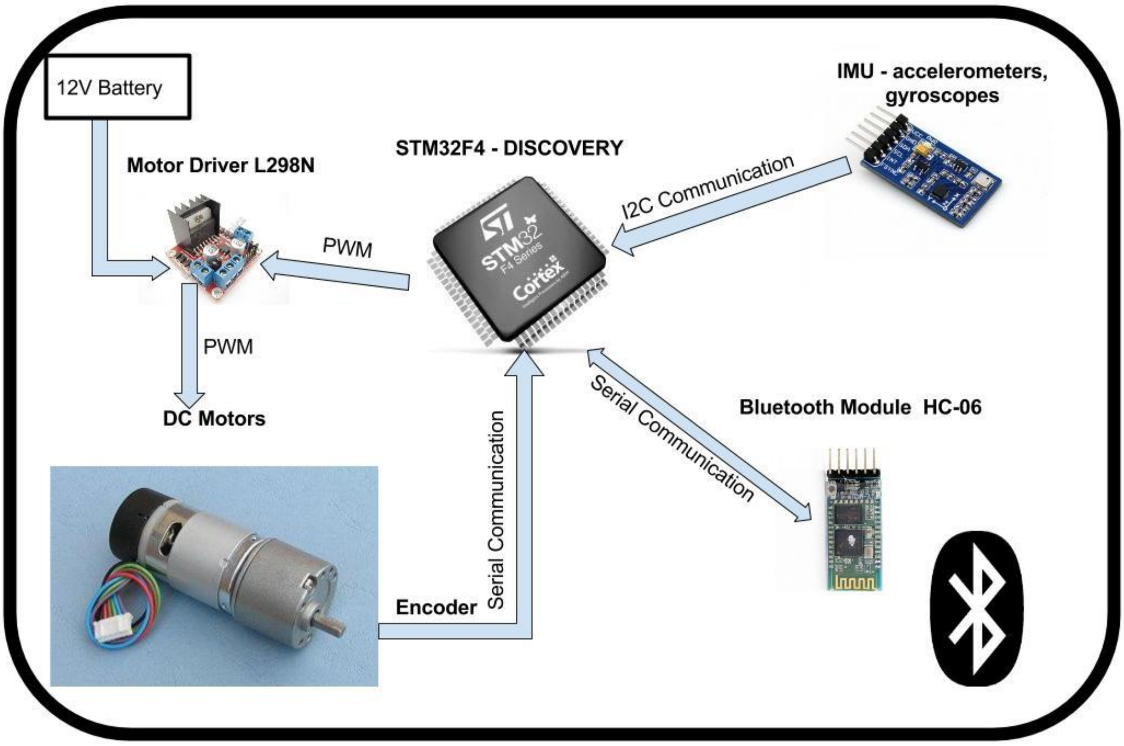

Figure 4 – Robot System overview

Remote Desktop

1- System Overview

The Remote Desktop system it’s a PC running the Desktop application and capable of sending Bluetooth data

Figure 5 – Remote Desktop System overview

System software stack

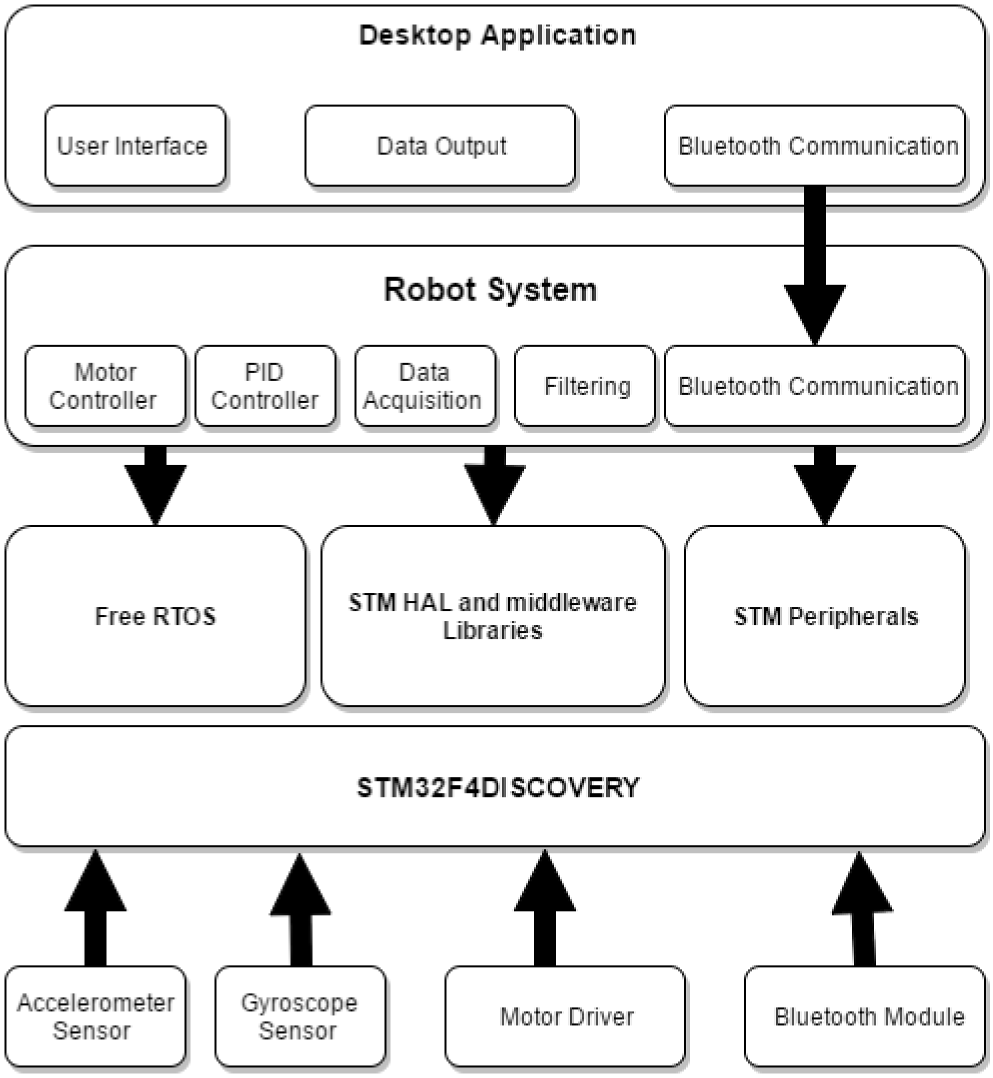

Figure 6 – System software stack

As we can see the system its composed by different layers starting on the top with the Desktop application that will communicate with the robot system via Bluetooth communication.

The Robot system composed by multiple tasks that will use some STM Peripherals and libraries, in the robot system it's going to be implemented multitasking provided by the FreeRTOS. The hardware layer its composed by the STM32F4DISCOVERY connected by the sensors, the Bluetooth shield and the motor driver.

Event tables

Robot System

Events	System response	Source	Type
Turn on Button pressed	Balance the robot	User	Asynchronous
New Command received	Process and perform the command	Remote System	Asynchronous
Falling (Angle with the vertical different than 0)	Get to a stable position	External forces	Synchronous
Sampling Period	Read the sensors	Local System	Synchronous

Table 1 – Robot System event table

Remote System

Events	System response	Source	Type
Data Input (Keyboard)	Process the data received	User	Asynchronous
Valid Command received	Send the command to the robot	User	Asynchronous
Invalid Command received	Send a warning to the user	User	Synchronous
Data received from the Robot System	Update de screen	Robot System	Asynchronous

Table 2– Remote Desktop System event table

The Table represent the possible events that can happens in the Robot system and the Remote desktop system, what are the system response, the source of that event and the event type.

Use cases diagram

The following diagram explains how the interaction happens between the different actor in the system, as shows the image below the Robot System has three main tasks to perform, Balance itself by measuring the robot position and adjusting the robot motors movements, perform commands sent from the Remote Desktop system and sending Data about the angle variation.

Robot System Use Case Diagram

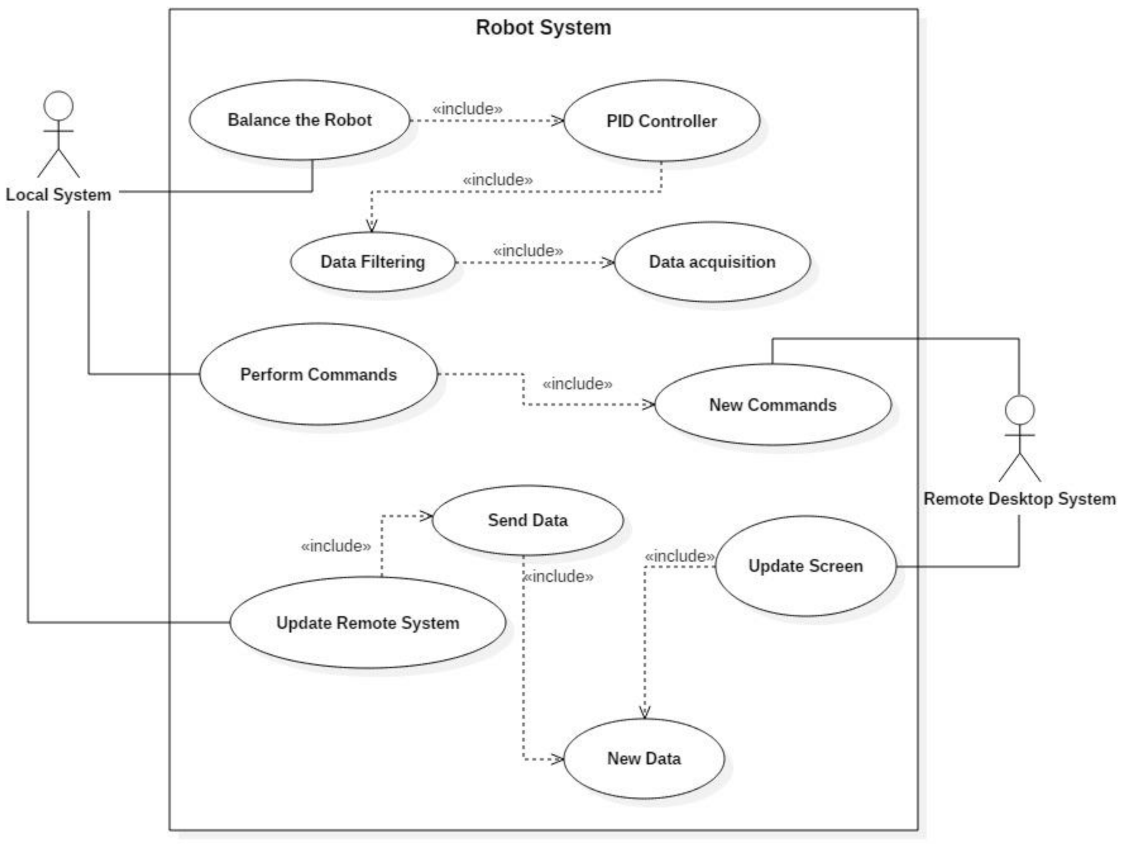

Figure 7 – Robot System Use Cases Diagram

Remote Desktop Use Case Diagram

For the remote system there's another actor that is the user that will be the one making the input of the commands in the remote system. The main goal of the remote system is send commands to the robot system and update the screen with the data received from the Robot system

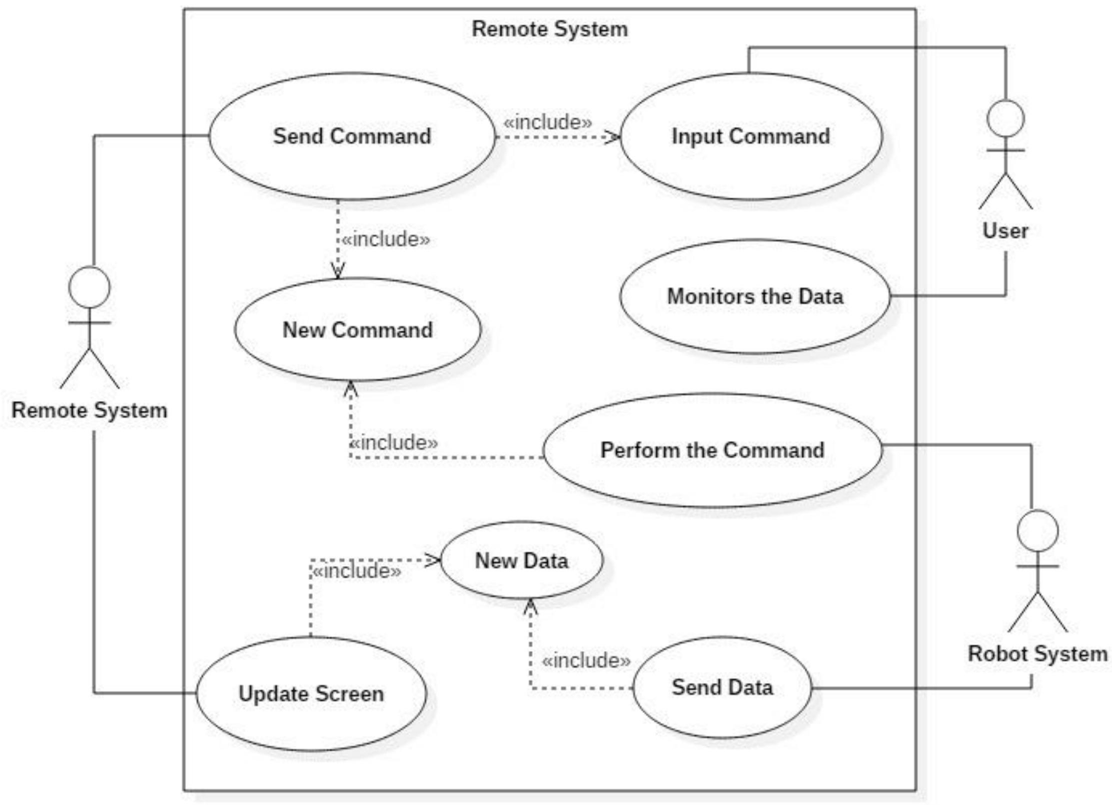

Figure 8 – Remote System Use Cases Diagram

State Diagram

Robot System

On the Robot System, the first step is acquiring the data from all the sensors, this data is then filtered so it can be applied on the PID controller. When connected with the remote desktop it will also send the sensor data via Bluethooth communication.

In the case that the robot is connected with the desktop application, commands can be received, as such, the system checks if the any data has been received from the bluetooth. If the application has sent commands data, the local system will read the commands and apply them on the PID controlller, otherwise it will keep just using the filtered sensor data acquired. The PID controller then actuates on the robot's motors and the cycle is restarted to keep the balancing

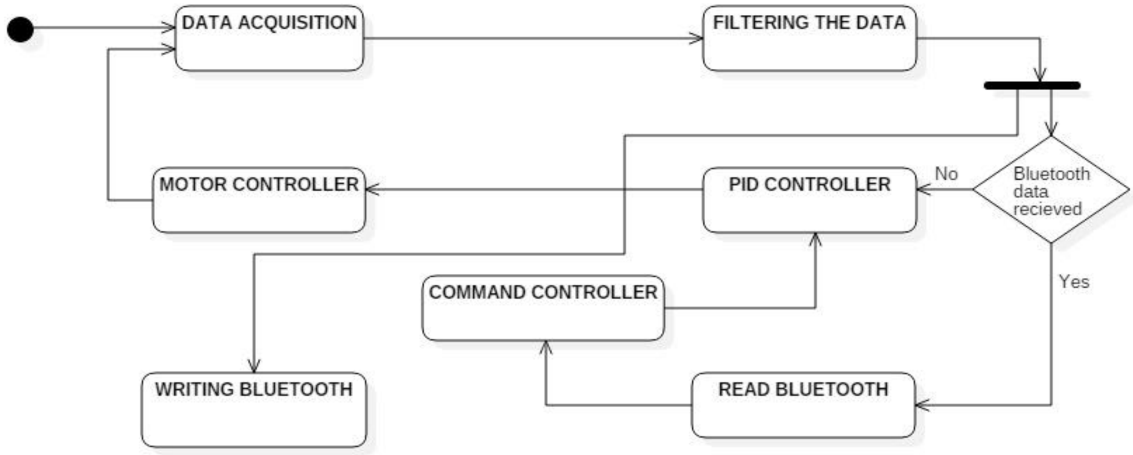

Figure 9 – Robot System Di state diagram

Remote Desktop System

The Remote system is going t to stay in a waiting state until it recieves new data from the Robot sytem to update the Screen (Data Output) or in case the User make the input of a new command where if valid the commands will ve sent to the Robot System.

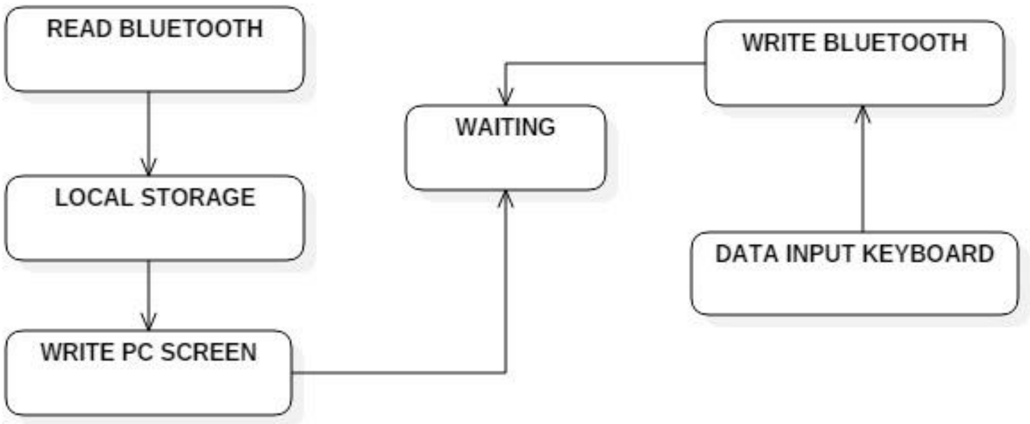

Figure 10 – Desktop System State Diagram

Sequence diagram

Robot Sequence Diagram

The image below shows the sequence diagram of the robot system, first step is to read the sensors (accelerometers and the gyroscope) and after reading the sensors the data it's filter to join the information from the different sensors to give a better measuring of the position of the robot, after knowing the position of the robot it's used that information on the PID controller and the output of the PID controller defines the movements of the robot.

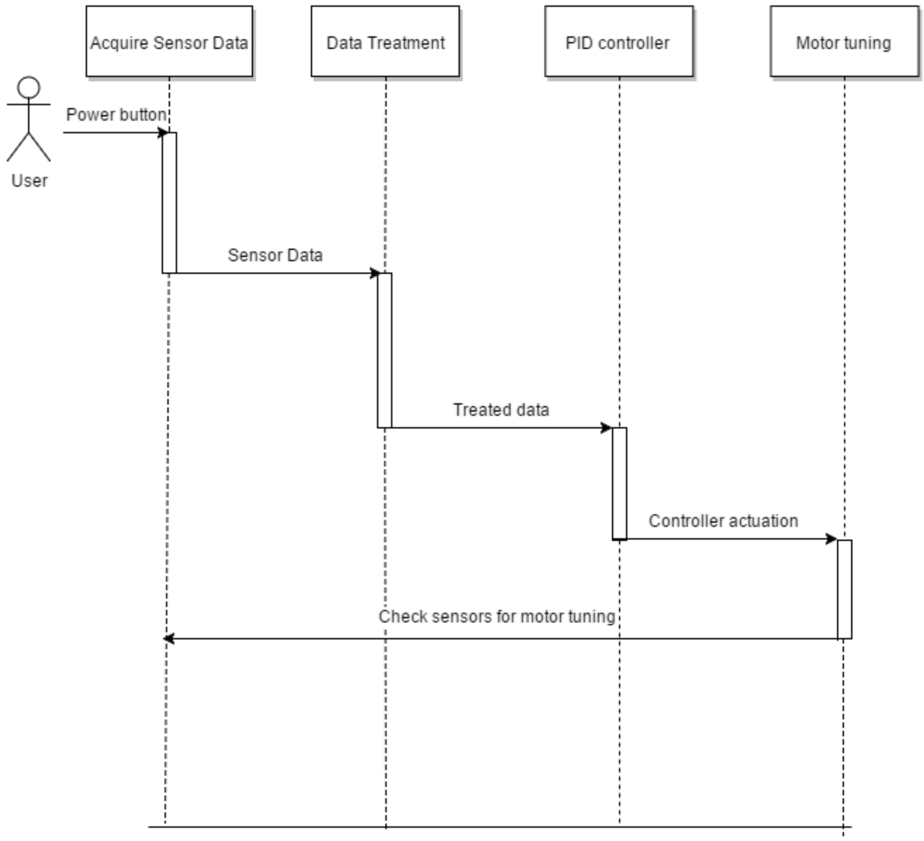

Figure 11 – Robot System Sequence Diagram

Application Sequence diagram

The application has two cases of a sequence diagram.

1 - The application sequence diagram it's as the following image shows, first the user makes the input of the command, the remote system verifies if the command is valid or not, if is a valid command the command is sent to the Robot System if not the remote system notifies the user by asking a new command

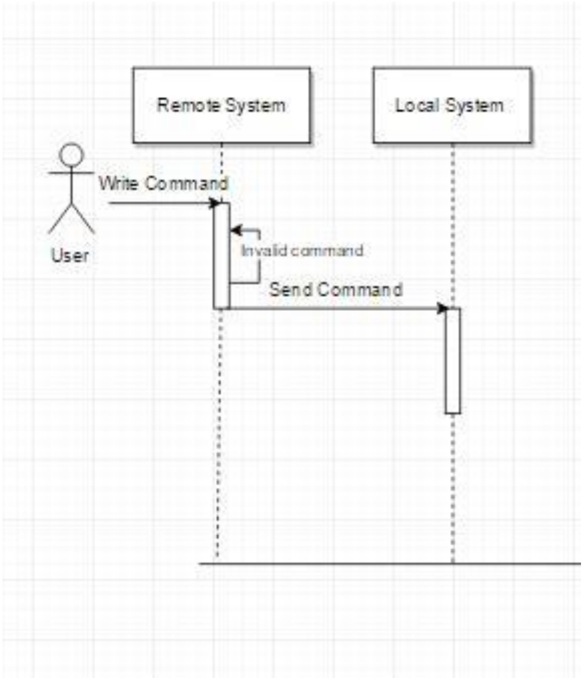

Figure 12 – Remote System Sequence Diagram 1

2- the second case it's when the remote system receives data form the robot system, first the robot system sends the data to the remote system, the remote system reads and saves the data locally and then update the application output (Screen).

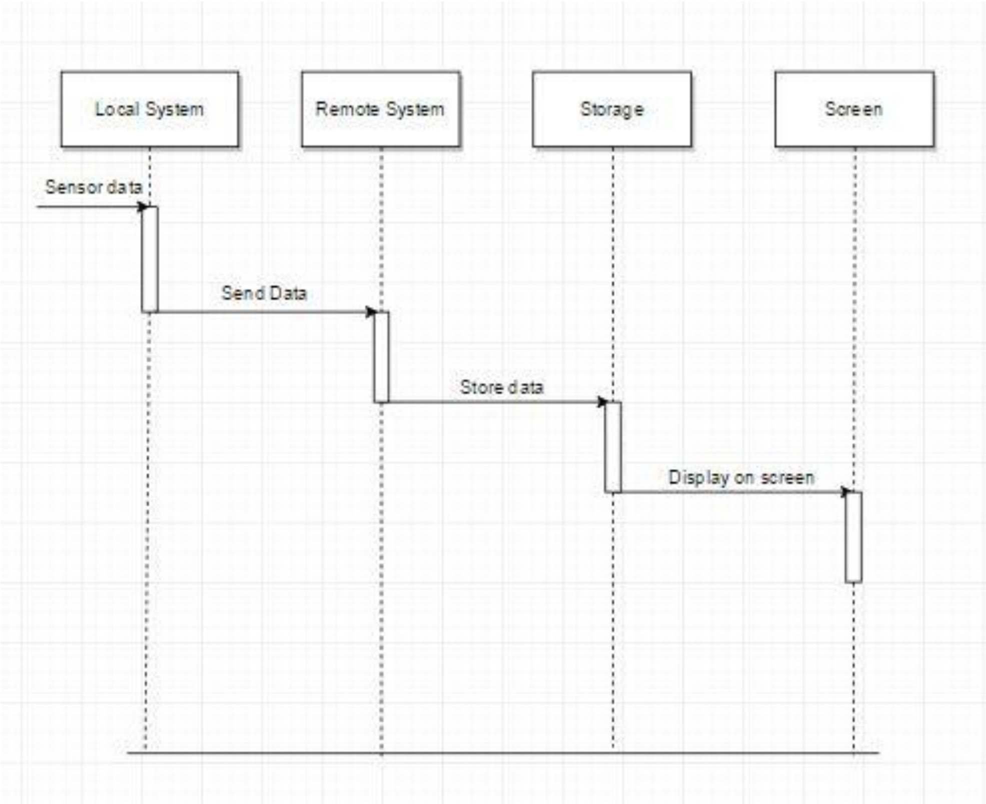

Figure 13 – Remote System Sequence Diagram 2

Tasks division

Task	Knowledge Overall (%)	To Learn Overall (%)	Developer(s)	Duration
Sensors interface and data acquisition	We already know how to Interface the distance sensor in on the microcontroller 20%	How the accelerometer and the gyroscope works and how control the STM32F4 GPIO. 80%	Ailton Lopes José Oliveira	10 days
Data treatment/Filtering	-	Learn the best filter to combine the data from the Gyroscope and the accelerometer and how to implement it on a microcontroller 100%	Ailton Lopes	14 days
PID controller Implementation	How the PID works and how to implement it 50%	How to implement the PID controller in a microcontroller 50%	Ailton Lopes José Oliveira	12 days
Motors Integration	How to control a motor using PWM 30%	How to use encoders to better control the motor and how to generate PWM using STM32F4 70%	Ailton Lopes José Oliveira	11 days
Communication system implementation -implement the Bluetooth communication system between the robot and the user PC	-	How the Bluetooth module works and how to interface it with the microcontroller and how to send and receive data through the Bluetooth module using the PC 100%	Ailton Lopes José Oliveira	9 days

Application Implementation – along with the previous task implement an application that allows the PC to send and receive data from the Robot.	Development of a C/C++ Application 50%	Communication between the application and the Bluetooth module. 50%	Ailton Lopes José Oliveira	10 days

Gantt Diagram

Task	Assigned To	Start	End	Dur
Project 1		21/9/16	20/1/17	122
Analysis Phase		21/9/16	19/10/16	29
Project Plan	Ailton e José	21/9/16	4/10/16	14
Analysis	Ailton e José	5/10/16	19/10/16	15
Design Phase	Ailton e José	19/10/16	3/11/16	16
Implementation Phase		2/11/16	12/1/17	72
Local System		2/11/16	12/1/17	72
FreeRTOS	Ailton e José	2/11/16	12/1/17	72
Sensor data acquisition	José	2/11/16	8/11/16	7
Data treatment	Ailton	2/11/16	8/11/16	7
PID implementation	Ailton e José	6/11/16	20/11/16	15
PID calibration	Ailton e José	20/11/16	12/1/17	54
Desktop Application		1/12/16	24/12/16	24
User interface	José	1/12/16	10/12/16	10
Bluetooth Communication	Ailton	1/12/16	10/12/16	10
Commands development	Ailton e José	10/12/16	24/12/16	15
Verification Phase	Ailton e José	16/11/16	13/1/17	59
Final Report Preparation	Ailton e José	16/11/16	20/1/17	66