Portada!!!!! 😊

Index

[Introduction 4](#_Toc136703383)

[State of the art – Benchmarking 5](#_Toc136703384)

[Ball tracking 5](#_Toc136703385)

[Type of communication 6](#_Toc136703386)

[Power of the motors 7](#_Toc136703387)

[Material of the design 7](#_Toc136703388)

[Development of the project 8](#_Toc136703389)

[Analysis of the problem and possible solutions. 8](#_Toc136703390)

[Mechanical design 9](#_Toc136703391)

[Electrical design 10](#_Toc136703392)

[Control and software 12](#_Toc136703393)

[Image processing 12](#_Toc136703394)

[PID control (Propotional Integral Derivative) 12](#_Toc136703395)

[Base movement 13](#_Toc136703396)

[Services 13](#_Toc136703397)

[Functional testing of the prototype 15](#_Toc136703398)

[Cost of the Project 16](#_Toc136703399)

[Conclusions 18](#_Toc136703400)

[Bibliography 19](#_Toc136703401)

# Introduction

The main objective is to create a balance plate using different mechanical, electrical and programing components. Furthermore, we want to implement a secondary function, we want to make the platform mobile to make stability more difficult.

We have had different inspirations to carry out this project. We started out wanting an entertaining, visually appealing, and complicated project. We liked the idea of having control over something having a resemblance to possible everyday control problems such as: the uncontrolled movement of a person with parkinson's or the constant balancing of the services of a waiter. In addition, we also think about the possible implementation of our project as a toy to captivate the interest of children.

This project is interesting because is made up of two parts. the first is based on control, specifically on two PIDs, which control the X and Y movement of the platform thanks to the vision of a camera located on top of the platform. With this we would have a good project, but in real life everything is more complex so we decided to make the mobile base, generating a speed, simulating the possible external forces that we could have.

So, we decided to put wheels on the base and make it mobile, controlling it with a wirelessly connected joystick.

# Team organization

The team is made up for:



Isaac

Product Owner



Hortense

Software Development



David

Mechanics Engineer



Aïda

Electronics Engineer

# State of the art – Benchmarking

This section supplies all the necessary knowledge to understand the background behind the different decisions made along this project. Specifically, we provide the different resource that were available during the process of development. We divided this section into four main concepts:

• Ball tracking

• Type of communication

• Power of the motors

• Materials of the design

## Ball tracking

Tracking the ball is one of the most important points and there are different ways to obtain the position in real time.

1. Vision-based tracking: This method utilizes computer vision techniques to track the ball's position and movement using a camera or multiple cameras. The ball is typically painted or marked with a distinct color or pattern to aid in its detection. By analyzing the video feed from the camera, the system can identify the ball's location by detecting its unique visual features. Techniques such as color-based segmentation, feature extraction, and motion analysis can be employed to track the ball's trajectory accurately. Vision-based tracking can be implemented using open-source libraries like OpenCV or specialized computer vision frameworks.
2. Sensor-based tracking: This approach involves using sensors embedded within the platform to track the ball's position. Different types of sensors can be used, such as accelerometers, gyroscopes, or infrared sensors. Accelerometers can measure the ball's acceleration, allowing estimation of its position over time. Gyroscopes measure angular velocity, which can be used to determine the ball's orientation and movement. Infrared sensors can detect the presence of the ball by emitting and receiving infrared signals, enabling continuous tracking. Sensor-based tracking can be useful in situations where visual tracking is not feasible or reliable, such as low-light environments or when the ball's appearance is not distinguishable.
3. Hybrid tracking: A combination of vision-based and sensor-based tracking methods can be employed for improved accuracy and robustness. This approach integrates the data from multiple sources to enhance the tracking performance. For example, vision-based tracking can provide precise position estimation, while sensor-based tracking can help compensate for occlusions or sudden changes in lighting conditions. By fusing the information from different sensors and vision systems, a more comprehensive and reliable tracking solution can be achieved.

It's worth noting that the specific implementation of ball tracking on a platform can vary depending on the application and the available resources. The choice of tracking method should consider factors such as accuracy requirements, cost, computational resources, and environmental conditions.

In our project we chose Vision based tracking, specifically with a raspberry camera. We considered the cost of the product and the ease of programming, so in the end we opted for this option.

## Type of communication

When communicating between a Raspberry Pi, Arduino, and motors, several parameters must be considered to ensure proper communication and control.

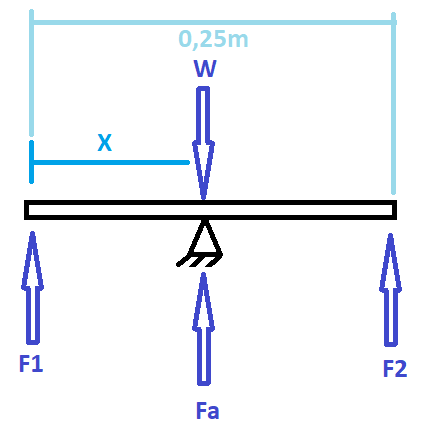
Data Protocol: Determine the communication protocol to use between the Raspberry Pi and Arduino. Common options include UART (Serial), I2C, SPI, or even using USB. Each protocol has its own advantages and limitations, so choose the one that best suits your project's requirements.

Motor Control Interface: Consider the type of motor you are using and the appropriate motor control interface. If you are using DC motors, you might need an H-bridge or motor driver module to control their speed and direction. The Arduino will receive commands from the Raspberry Pi and translate them into motor control signals accordingly.

Our decision has been to connect the motors that move the platform directly to the raspberry so that they have the highest possible speed. Being digital servos, we only need to send a PWM signal that we generate with the Raspberry. On the other hand, to move the wheels we use we have decided to use a wireless joystick, this sends the commands to the raspberry, then it sends them by Serial to the Arduino and this sends the information to an H bridge capable of sending the information to the DC motors.

## Power of the motors

First, we calculate the forces from the structure that we have for later to be able to choose the motors that we will need. We can see that the weight that we are going to support are 0,22125Kg.



To choose the DC motors we made a carton car with 5Kg to move, and it works, then we decide to buy these motors. In the “Photo X” we can see the test we did.

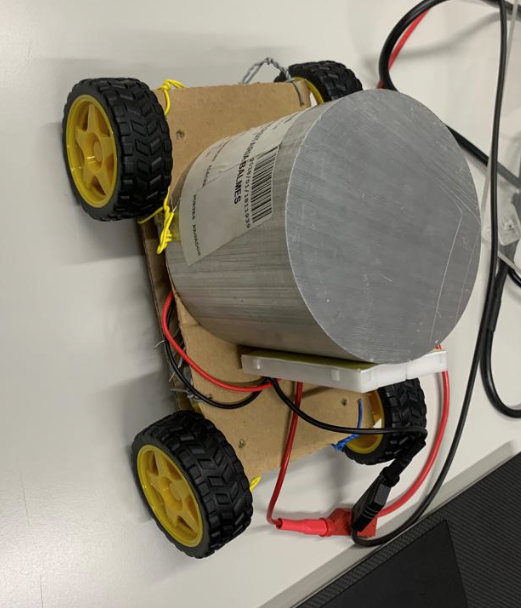


Photo X: test to choose the DC motors with 5 Kg.

## Material of the design

In this section we explain why we choose the materials to make the structure.

The base and the legs are printed in PLA because we think is a good material, we don’t put so much weight on and it’s economically.

The table is metraquilat because is a material that doesn’t much weigh and is easy to machine and move.

For the tail, the main idea was to do it with PLA, but the printer broke and we had to look for plan b and we took the arm of a lamp. It weighs more than we had thought but for the moment it holds up.

CAL FOTO?????????????????????????

# Development of the project

## Analysis of the problem and possible solutions.

In this part we analysis the possible solutions and the possible problems that we can find.

The material we must connect are:

Camera Raspberry Pi: this component we use it to scan the table to know where the ball is in every moment.

Raspberry Pi 3: control the two axis that move the table and process the image.

Arduino Nano: control the two wheels.

H-bridge: drive to control the motors dc.

Servomotors: moves the arms from the table in axis X and axis Y.

Motor DC: moves the wheels.

Joystick: we send the signal to move the scorpion where we want it.

In the “Photo x” we can see a simple electric scheme to know how we will connect all the components.

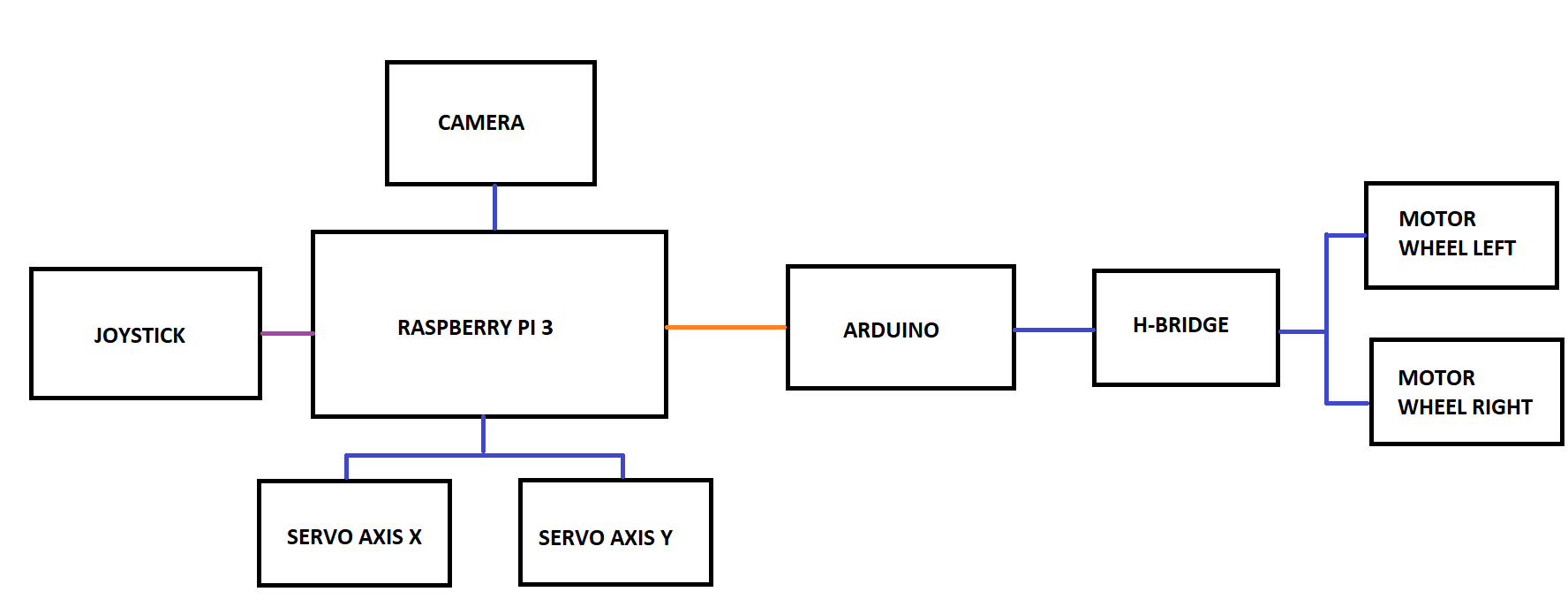


Photo x: Simple electric schemes.

That we can see in the “photo x” there are three different colors of lines:

When the line is purple, it means that the communication is by Bluetooth.

When the line is orange, it means that the communication is by serial port.

When the line is blue, it means that are connected.

During this memory in the part from “Electric design” we can see the electric schemes more explain it.

## Mechanical design

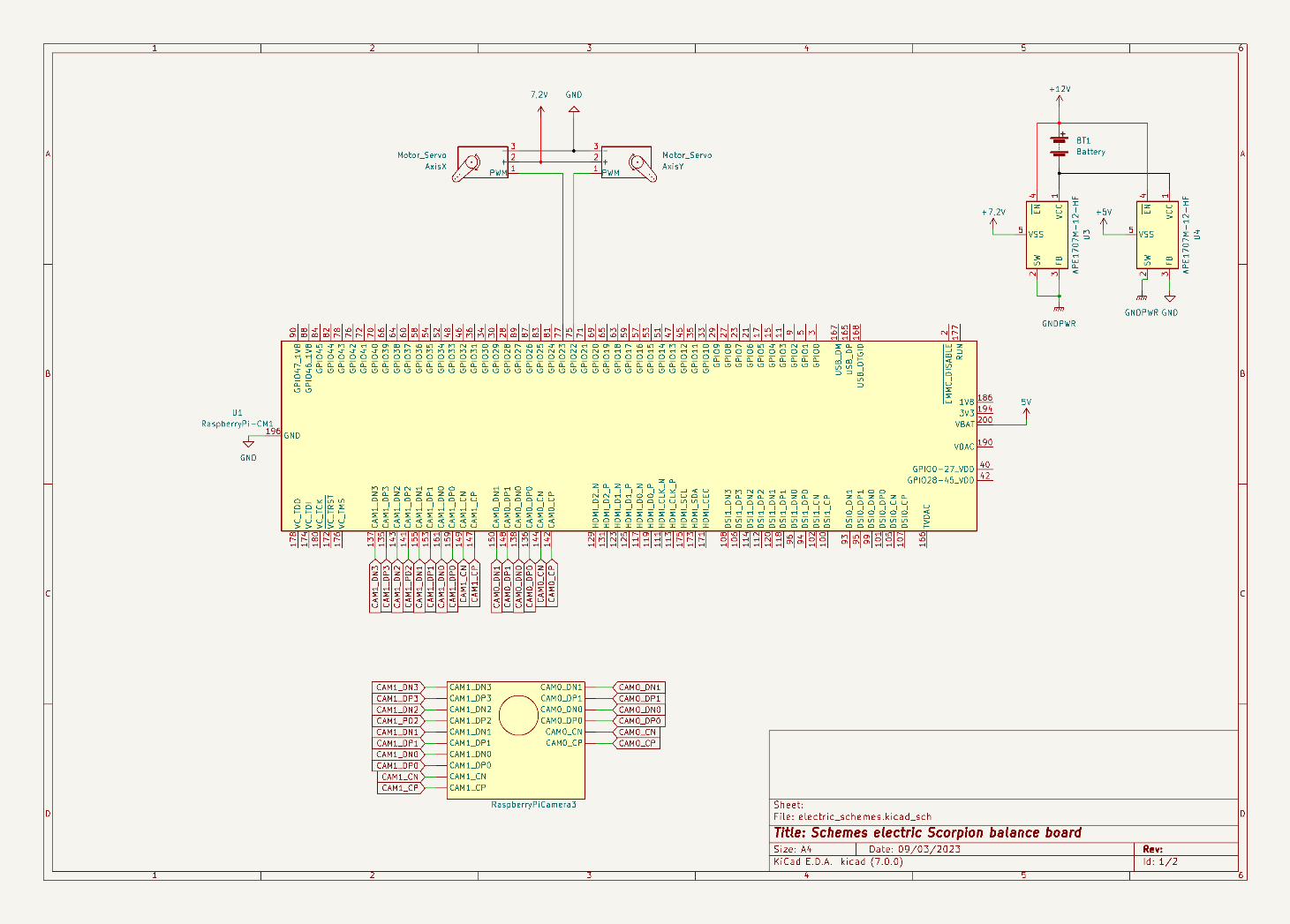
## Electrical design

As mentioned above, in this section we explain the electric schemes with more details.

We have a 12V battery with two regulators. One regulator is configurate with 7,2V to power the servomotors and the other regulator is configurate with 5V to power the Raspberry.

On the Raspberry Pi we connect the camera and the two servos. For the camera is connected in a special connector that there is for the camera, and the motors are connected in pin 77 and 75, as seen in the “Photo X”:

Photo X: Electric schemes. Raspberry Pi, servos, camera and battery connections.



The Raspberry sends to Arduino the value from the joystick and then the Arduino process this information and do the movement to the wheels.

The pins D2, D3 and D5 are the pins to control one wheel that are connected to the H-Bridge. The D4, D6 and D7 are the pins to control the other wheel and are connected to the H-Bridge too.

The H-bridge is power with 12V and is used to drive a load, such a as a DC motor, in both directions and it controls the flow of current to a load.

As seen in the “Photo X”:

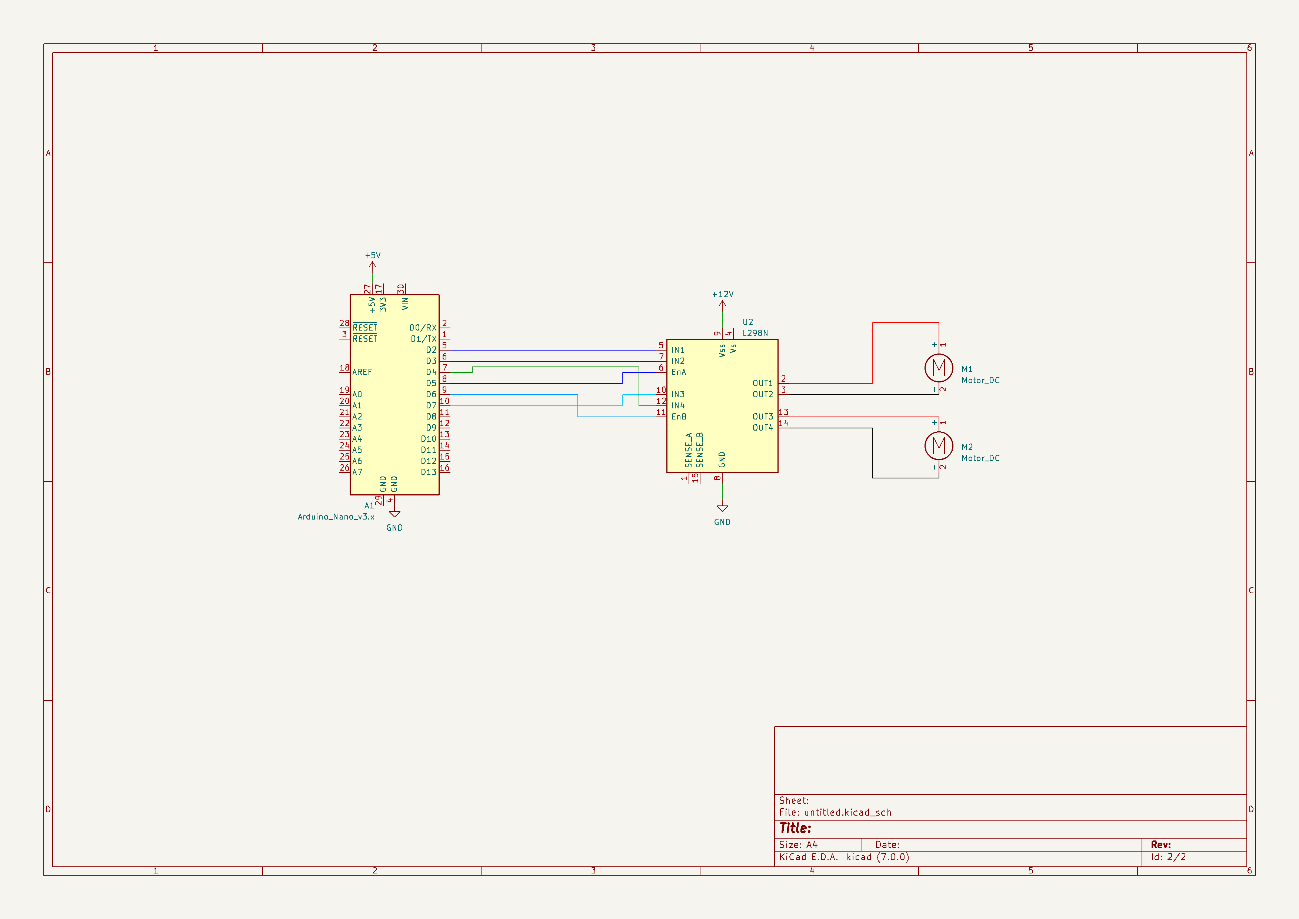


Photo X: Electric schemes. Arduino, H-bridge and two motors DC.

## Control and software

The software and control part of this project was divided in different sections that finally interconnect to create one entry point.

The main objective was to be able to capture the position of the ball with a camera and adjust the servos to return it to the default position, hence balancing.

The second part was to move the base of the project which is on wheels using a joystick.

The software is implemented in Python and Arduino’s C++.

### Image processing

The image capture was done using a RasberryPi camera v3, set to 30 frame per second attached to RasberryPi. The objective is to identify the ball on each frame and update it’s position. There were multiple attempts to do this using high level python libraries, but this proved to be a hindrance in terms of the processing speed.

As a solution the obtained frame is immediately converted to HSV (Hue Saturation Value). With the frame in this format we can calibrate it to only identify objects of a certain color. In this case the color of the ball is green therefore anything that is not in the acceptable range of green is a null pixel.

The HSV technique has solved two problems.The first one being noise due to multiple objects in the field of view of the camera including natural light. The second one , this technicle is enough to identify what kind of ball the program is looking for.

### PID control (Propotional Integral Derivative)

Once the position of the ball is known, we can apply the control to adjust. There are two PID controls for this project, one for the servo handling the x axis and another one for the servo handling the y axis.

Each time the position of the ball is captured , using the PID class, the servo values are calculated according to the difference between the current position and the default. As of now there is still room to properly fine tune the parameters of the PID.

### Base movement

As an additional functionality of the project, the base to which the balance board is attached, should be able to move on command using a joystick/controller.

To implement this, an arduino was used which allowed direct accces to the wheel motors. The Arduino program receives commands from the RasberryPi using a python program which captures the code of the pressed button on the controller.

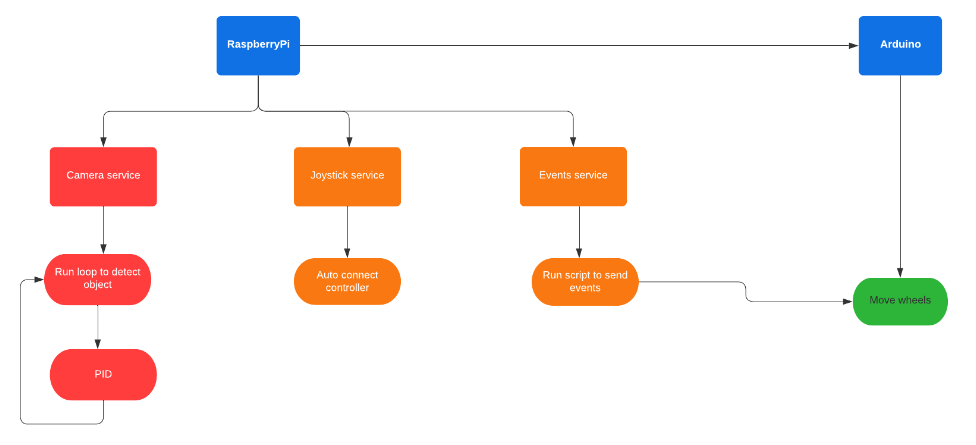
Depending on the number sent to the arduino, a movement function will be triggered.

The controller has to be wireless since the whole project is mobile. This requires capacity to auto connect the controller without a GUI. For this project, this connection was implemented as a service which tries to connect to a specific controller with bluetooth. It can wait until the device is available and stop waiting when the device is connected.

### Services

Given that the project will be mobile and without a GUI, it is crucial that all software components operate independently. In order to achieve this the software parts were divided in different services that launch when the microprocessor is turned on.

The following is an estimate flow of the software plan.



There are three main services that operate in parallel. They are separate to avoid dependecy which inevitablely more time than not blocks when one is not working.

* Camera service: This is the service that handles image capture and pid control. It’s always on as long as the microprocessor is turned on.
* Joystick service: This is the service that manages the autoconnection to the controller. It should be in standby until a controller is connected.
* Events service: This service is dedicated to sending commands of the controller from RasberryPi to Arduino through a USB channel.

## Functional testing of the prototype

# Cost of the Project

In this section we are going to explain the cost of the project.

First, we are going to specify the price of the materials that we need to do this project:

|  |  |  |
| --- | --- | --- |
| MATERIAL PRICE | | |
| PRICE | QUANTITY | MATERIAL |
| 59,45 € | 1 | RASPBERRY PI 3 |
| 31,70 € | 1 | RASPBERRY PI 3 CAMERA |
| 6,99 € | 1 | SD CARD for Raspberry PI 3 |
| 27,54 € | 1 | CARDAN |
| 16,88 € | 1 | DC Motors |
| 4,82 € | 1 | Metraquilat 250x250mm |
| 43,18 € | 2 | Servos base |
| 7,33 € | 1 | Arduino nano |
| 50,00 € | 1 | Others |
| 247,89 € |  |  |

To develop this project, we must add the price to the person that develop this project:

|  |  |  |  |
| --- | --- | --- | --- |
| DEVELOPMENT WORK | | |  |
| PRICE | HOURS | PERSON - FUNCTION | PRICE FUNCTION |
| 4500 | 100 | Hortense Utuje - Software development | 45 |
| 4500 | 100 | David Pfaffenrod - Technical engineer | 45 |
| 4600 | 100 | Isaac Lenin Acevedo Bau - Product owner | 46 |
| 4500 | 100 | Aïda Calero Chicharro - Technical engineer | 45 |
| 18.100,00 € |  |  |  |

The final price to development this project is **18.347,89€**.

If this product is going to sold in a store, as the development is already done, the final price is **397,89€,** only the material and the assembly of the technician would be charged.

# Conclusions

# Bibliography

*Ball Balancing PID system*. (29 de september de 2020). Obtenido de https://github.com/giusenso/Ball-Balancing-PID-System