Portada!!!!! 😊

Index

[Introduction 3](#_Toc4520470)

[State of the art – Benchmarking 4](#_Toc1394362884)

[Development of the project 5](#_Toc948222986)

[Analysis of the problem and possible solutions. 5](#_Toc348218987)

[Mechanical design 6](#_Toc965079496)

[Electrical design 7](#_Toc527143293)

[Control and software 8](#_Toc1897722858)

[Functional testing of the prototype 10](#_Toc1456662051)

[Cost of the Project 11](#_Toc868118644)

[Conclusions 12](#_Toc1769352431)

[Bibliography 13](#_Toc2096567337)

# 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.

# State of the art – Benchmarking

# 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.

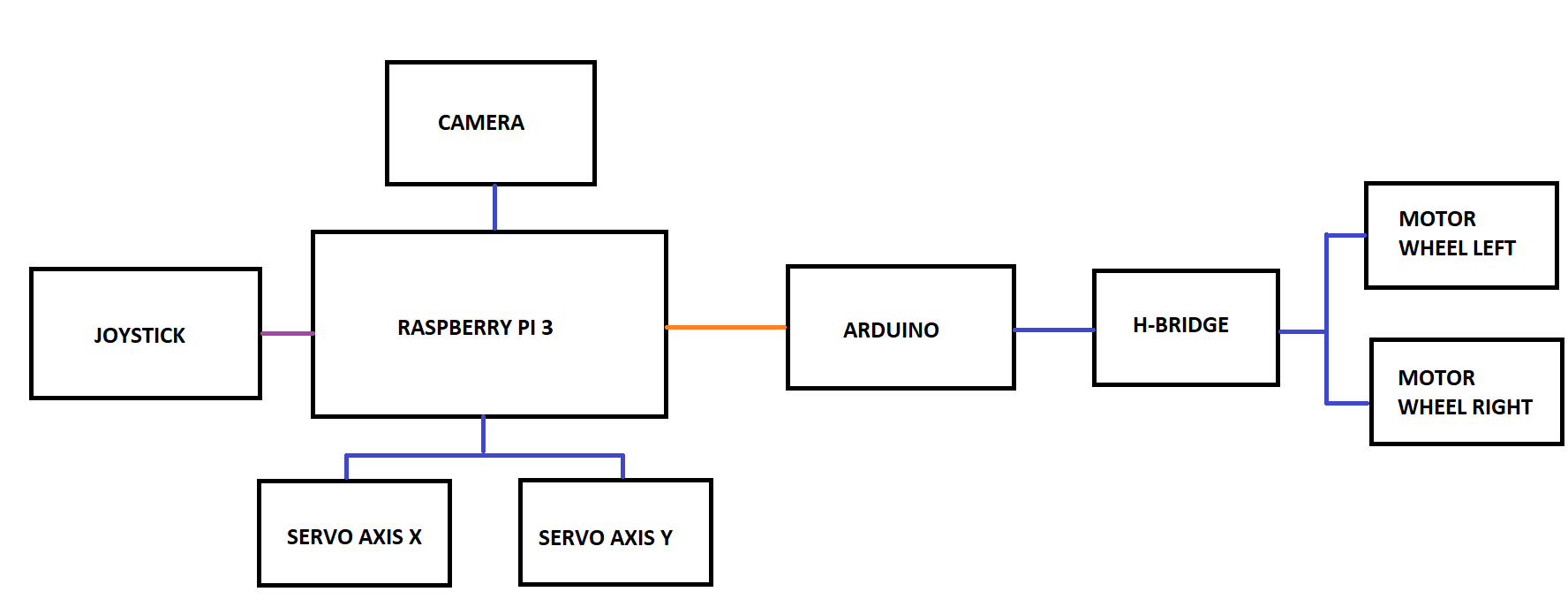


Photo x: Simple electric schemes.

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

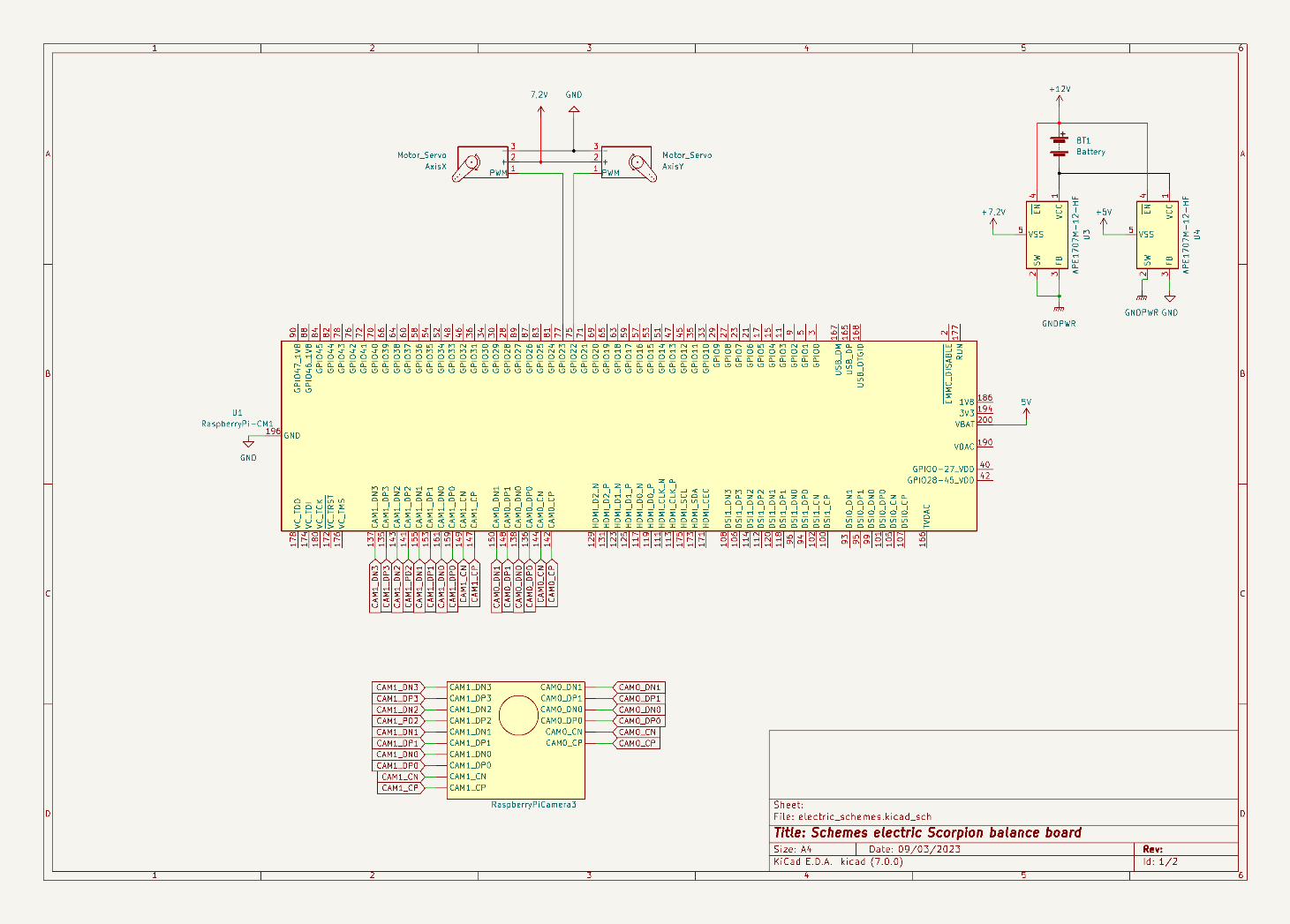
## Electrical design

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”:

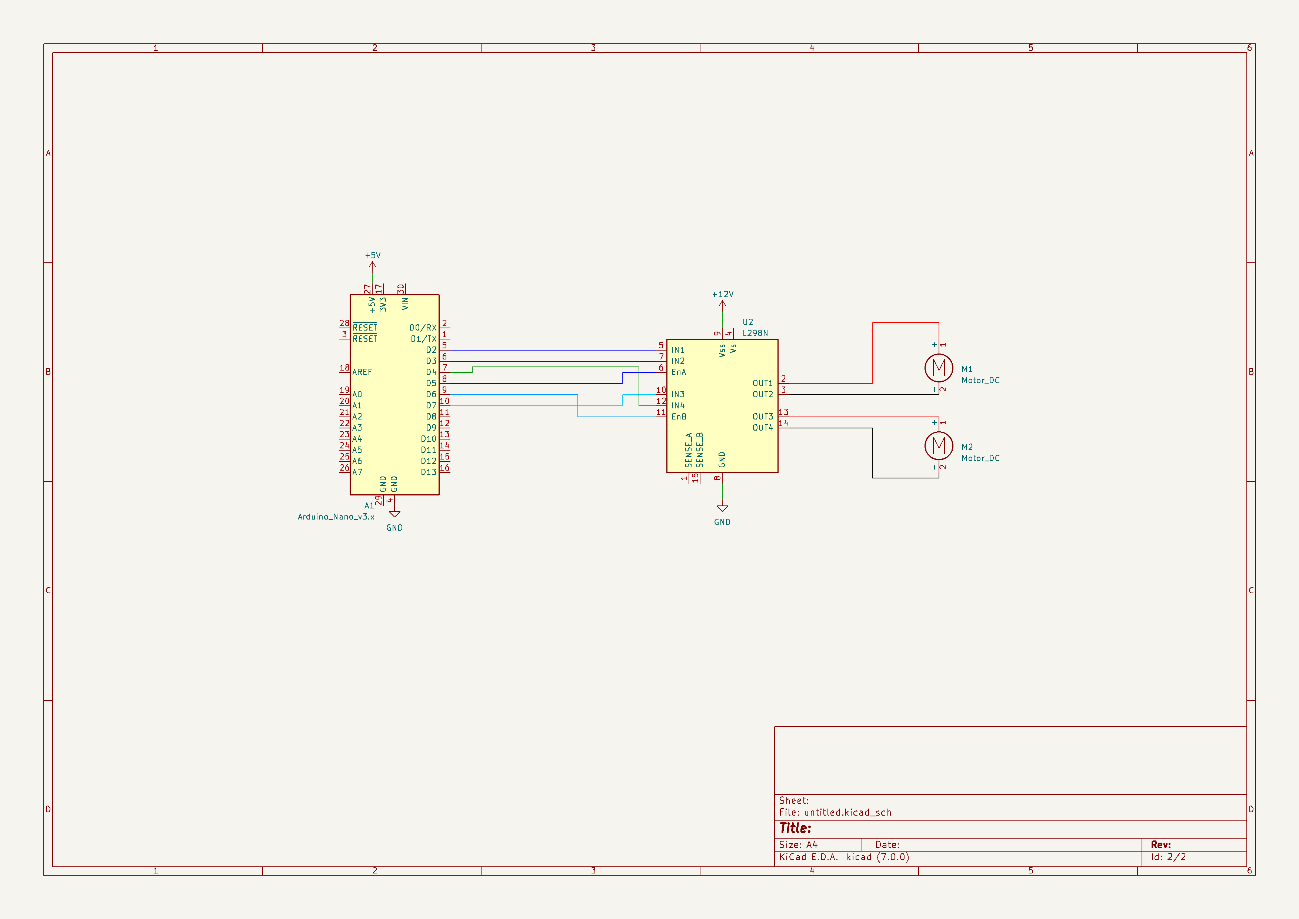


Photo X: Electric schemes. Arduino, H-bridge and two motors DC.

## 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++.

1. 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.

1. 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.

1. 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.

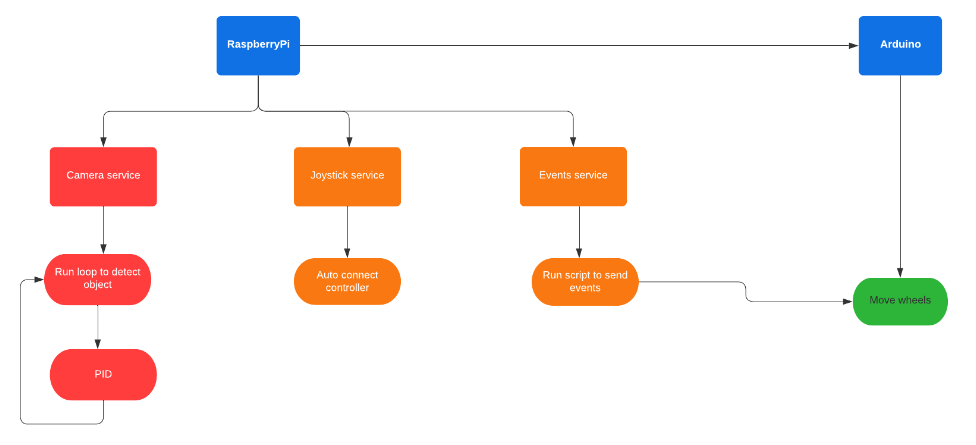
Depending on the number sent to the arduino, a movement function will be triggered.

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.

1. 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