July 13, 2023

Leoracer

Lorenzo Balugani
Master's Degree student @ Alma Mater Studiorum - University of Bologna

Contents

1 Introduction

This document serves as the project report for the Leoracer project, which was created for the Alma Mater Studiorum's "Laboratory of Making" course, held by Professor Renzo Davoli and Professor Federico Montori in 2023.

This report will contain information pertinent to the project, such as the problem description, the implementation and development process description and the summary of the issues that have arisen during the project lifecycle. Additionally, it will contain a section all about future improvements.

Special thanks to Matthew Heironimus for his joystick library [2], that was key in the development of this project.

2 The problem and the solution

This project aims at solving a very specific and obscure issue that may be of not a lot of interest to many, but it's still an useful exercise as a first project that involves electronics, 3D design and 3D printing.

Leoracer is a purpose made controller that aims to make available to everyone that possesses the correct equipment to experience the arcade version of the videogame "Star Wars Episode 1: Racer": the arcade machine wasn't extremely widespread when it was created, and as time passes it has become more and more rare.

Before going too much in depth, let's understand what's the game about in the first place, and what made the arcade version stand out. In the game, the player controls one of the several "Pod Racers" in different races around the galaxy, hence this game behaves for the most part as a racing game. The pod racers are vehicles propelled by several engines, and canonically they "turn" by lowering the thrust of half of the engines, like an airplane moving on the ground. Other than turning, the pod racers can tilt on their axis and pull their nose up or down to better handle air time.

In the computer release and on the console ones, the control over the pod is provided by keyboard, mouse or the analog sticks of the controller. In the arcade version, as portrayed in [1], the user is given two throttle controls that can be manipulated in order to make the vehicle change direction, being the key stand-out feature of the arcade release, along with better graphics.

Leoracer was hence developed in order to bring this control scheme to the PC release, while trying to improve the original product. Leoracer's throttle control can tilt forwards/backwards and sideways, allowing the user to not just control the thrust and the direction, but also the pod racer's tilt. The controller features 3 buttons, one of which is also mapped as the combination of both throttle buttons and a toggle, that allows the control of the pod's orientation.

3 Developing the solution

3.1 3D Modeling

In order to create such a particular controller, a custom model was needed. The design was made using Solidworks, a powerful parametric CAD software aimed at mechanical 3D modeling. Using Solidworks allowed, through its constraints system, to test the design before printing: this allowed the resolution of many issues before printing the components, and guaranteed the alignment of screw holes and etcetera.

The main inspiration for the controller was a radio transmitter for RC planes, that had two analog sticks that behaved in the desired way. The final design is basically a cylinder with a missing section of its surface, allowing the throttle to be mounted on a potentiometer - allowing it to tilt in all the needed directions.

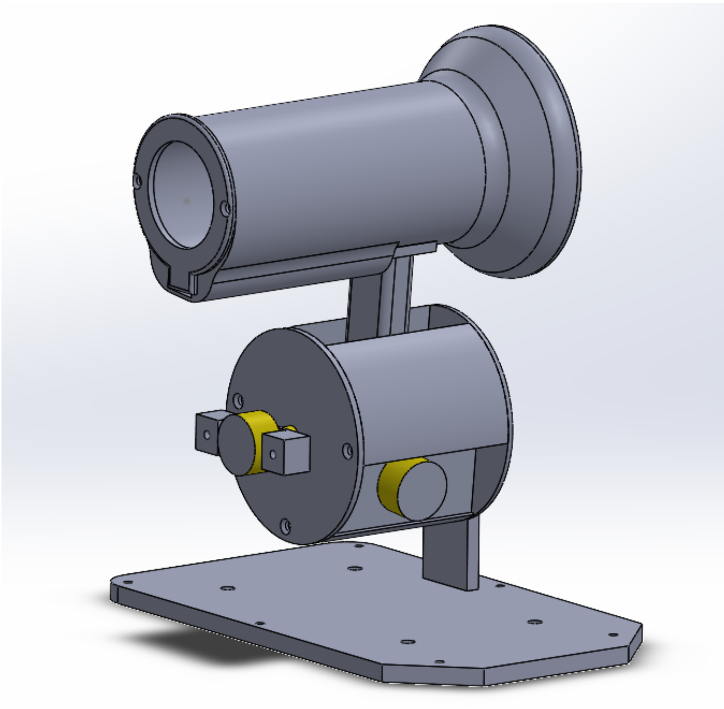

Figure 1: The rotating cylinder of the throttle

The cylinder is held in place both by the potentiometer that handles forward tilting - which attaches to the external case - and a holder that sits on top of the bottom of the controller.

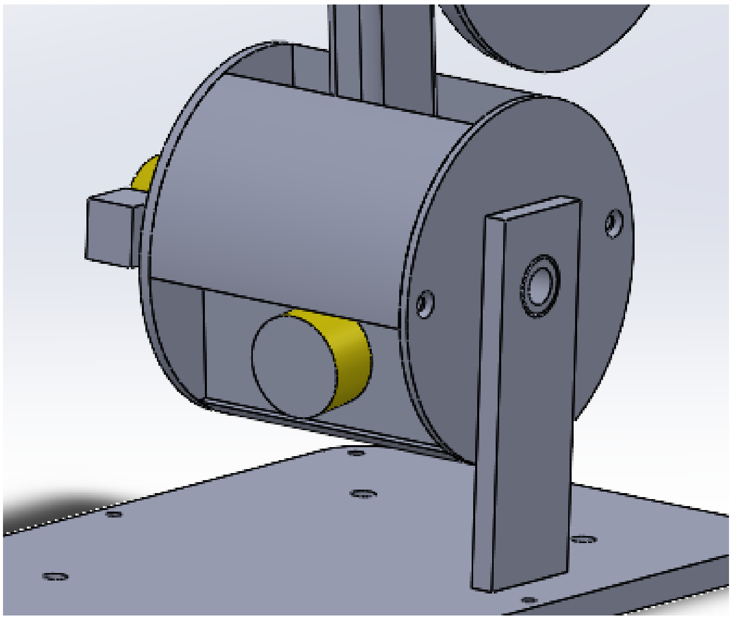

Figure 2: The support for the rotating cylinder

The throttle is made of two components: the “stick” and the “handle” itself. Both allow a wire to be routed inside of them, which will connect to the button placed inside the handle, and exit from the cylinder using the hole that can be seen fn figure 2. The staff is locked in place by a screw, that is inserted on the larger side of the handle.

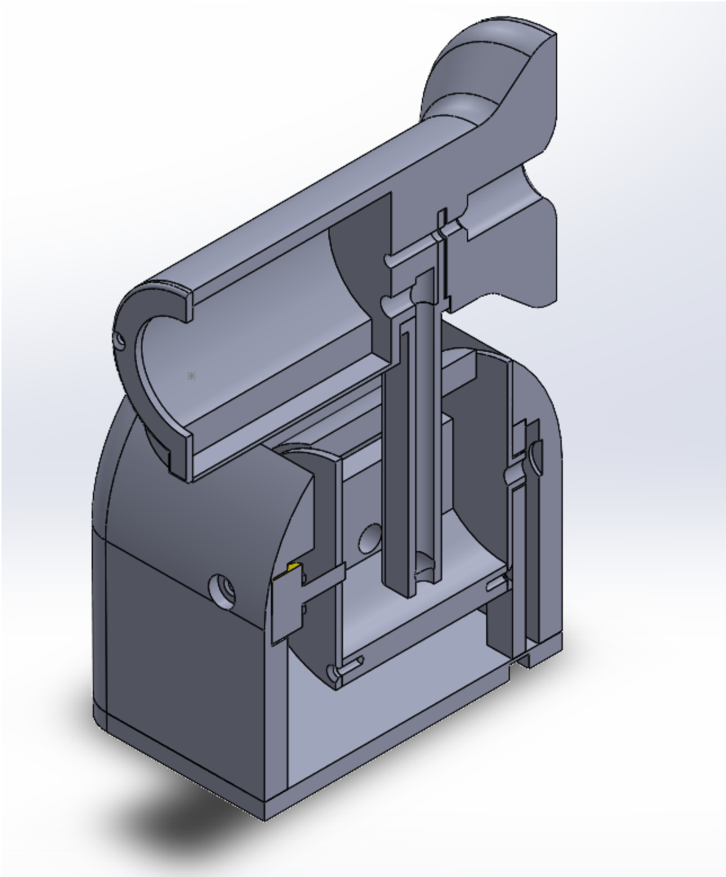

Figure 3: A slice of the controller, showing the button's wire channel

Due to the nature of the controller, two throttles will be needed. These are symmetrical internally, while externally they have a slightly different case.

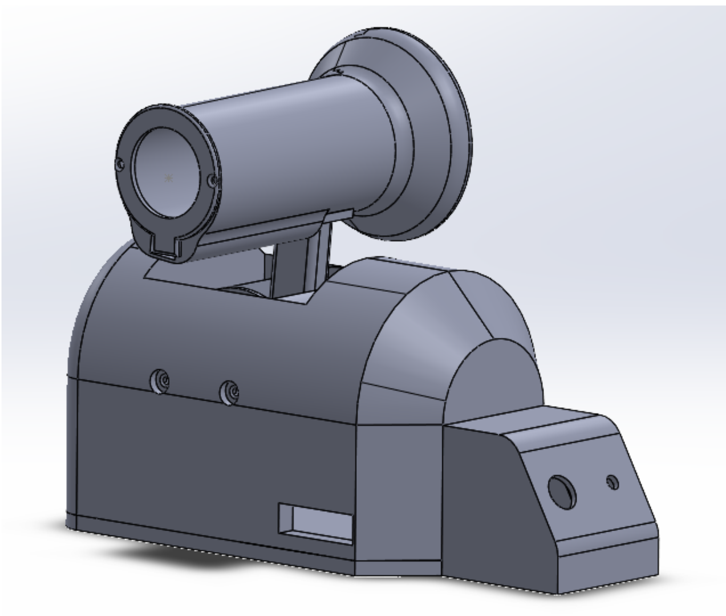

Figure 4: The right part of the controller

The Arduino Leonardo can fit inside the case of the right part of the controller.

3.2 3D Printing

All the parts were printed on a Lotmaxx SC-10 Shark, a FDM 3D printer. In order to make all components stick together, brass screw inserts were embedded - with the help of a soldering iron - in the provided holes of all the prints, and M3 screws were used. The controller is held in place on the table by small suction cups.

3.3 Circuit design

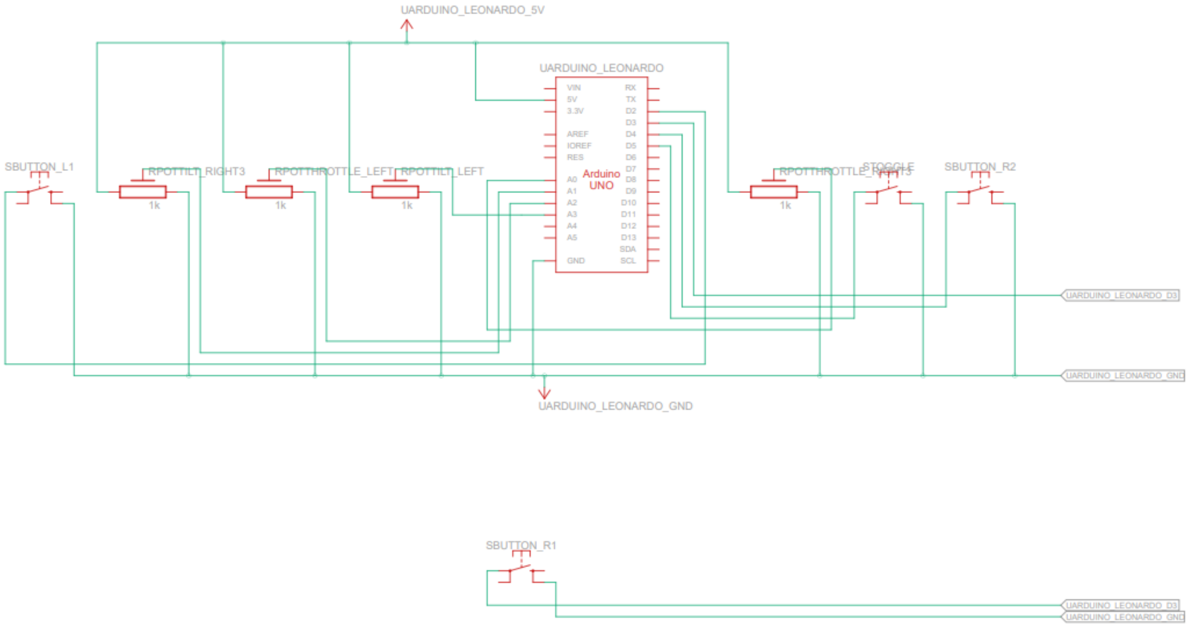

Figure 5: The schema of the controller

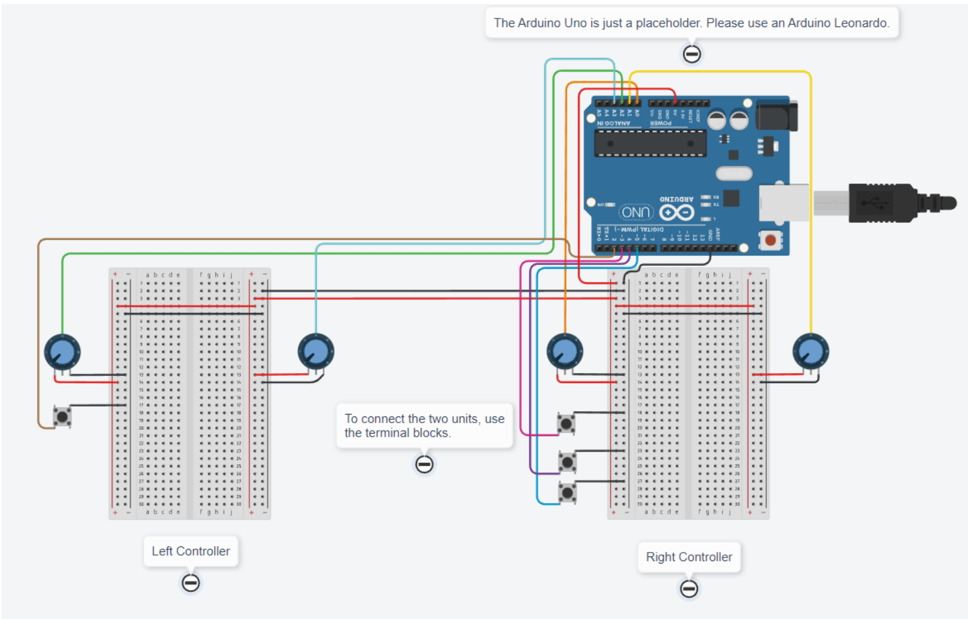

Figure 6: An intuitive version of the schema

The circuit requires to solder just the buttons, the resistors and the terminal blocks that connect the two parts, and all other connections can be made using jumper wires and breadboards. Additionally, a custom cable needs to be made to connect both parts of the controller. Since the Arduino is configured with INPUT_PULLUP, the buttons do not require resistors to properly work.

3.4 Firmware development

Two different pieces of firmware have been developed: the main firmware, which is the one that interfaces with a PC as a D-Input device, and the calibrator, which assists with the calibration routine.

3.4.1 Calibrator

The calibrator allows the user to calibrate the controller by placing one axis at a time at its maximum or minimum, then pressing the right throttle button. The calibration data gets then saved to the EEPROM, and will get loaded by the controller firmware. The calibration order is at it follows:

1. Right Throttle (Min/Max).
2. Right Tilt (Min/Max).
3. Left Tilt (Min/Max).
4. Left Throttle (Min/Max).

3.4.2 Controller

The controller firmware hevily relies on the Arduino Joystick Library [2], that allows him to act as any D-Input controller, with many customizable parameters. The firmware loads calibration data from the EEPROM, and then uses it to map each axis to a percentage. The axis are evaluated as it follows:

- X-Axis (turning): percentages of the two throttle potentiometers are subtracted, and the resulting value gets passed as the value of the X axis;
- Y-Axis (attitude): the toggle status gets evaluated, and Y is set to the maximum or the minimum, accordingly;
- Z-Axis (throttle): the highest percentage of the two throttle potentiometers gets used;

For tilting, two buttons (4 and 5) are used, since the game consideres it a discrete value. If the two tilt controls have a percentage lower than 15 (or higher than 85), the corresponding button gets pressed.

4 Using the controller

Once built and calibrated, the controller is straightforward to use, as it gets picked up as a standard D-Input controller. The buttons can be bound ingame without much of an issue. A video of the game being played with the controller can be seen on Youtube [3].

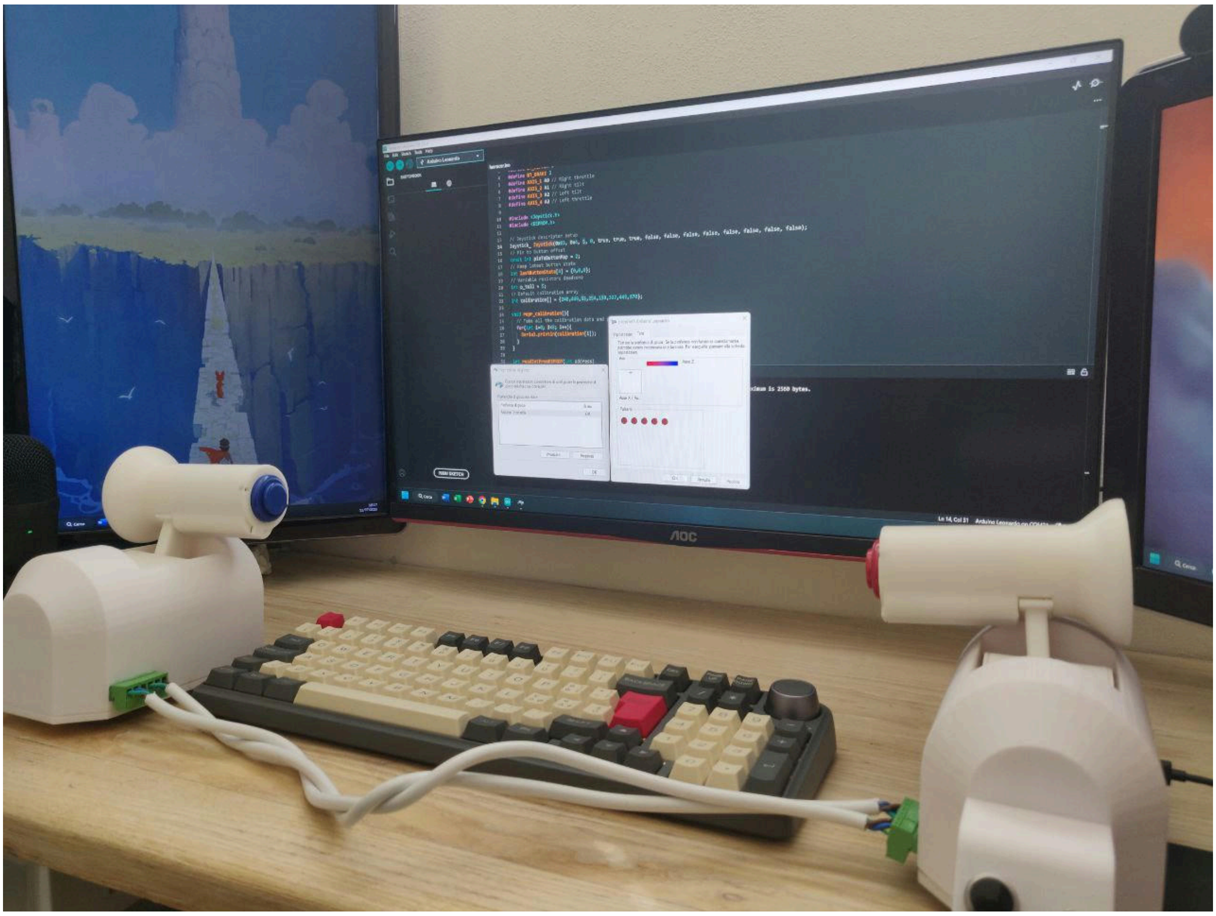

Figure 7: The Leoracer Controller

5 Project availability

The project is available under the GPL v3 license on the github repository [4]. The readme contains a detailed parts list and a set of instruction for assembly.

6 Challenges

This project was the first experience with developing from scratch and printing a complex model that had to accomplish some mechanical challenges. It also was the first project that involved soldering and - in general - electronics.

7 Known issues

- Due to how the in-race menu works, it's not possible to control it using the keyboard, as it's meant to be controlled by the Y axis of the controller. In order to regain control, press CTRL+J to momentarely disable controller input.
- Suction cups do not work on all surfaces.

8 Future improvements

The controller, as is, works great. However, several improvements could be made in a future revision, such as:

- Encoders instead of potentiometers, providing better accuracy and longevity.
- More buttons on the throttle handles, in order to increase button count.
- Stiffer throttle tilting, perhaps achieved using springs.
- Reduced bulkyness of the case.
- An optional mountuing system to allow the controller to stick to any surface.

Bibliography

[1] Dr. Retro. *Star Wars Episode 1 Arcade Racer Cabinet Gameplay*, (2014). [Online Video]. Available: https://www.youtube.com/watch?v=g0bF3kAVur4

[2] Matthew Heironimus, “Arduino joystick library.” [Online]. Available: https://github.com/MHeironimus/ArduinoJoystickLibrary

[3] Lorenzo Balugani. *Star Wars Episode 1 Racer With a Custom Controller*, (2023). [Online Video]. Available: https://www.youtube.com/watch?v=g0bF3kAVur4

[4] Lorenzo Balugani, “Leoracer repository.” [Online]. Available: https://github.com/Nemesis-FT/leoracer