# Team 3: RoboCops

## Interdisciplinary Robot Competition

Project Supervisor: **Alessandro CRESPI**
Professor: Prof. **Auke IJSPEERT**
Team Coach: **Nicolas FURRER**


**Florian COMTE** - 346006
**Michel ABELA** - 339421
**Alessandro MINGHELLI** - 342216


**June 15, 2025**

# Contents

# 1 Introduction

## 1.1 Presentation of the Competition

The Interdisciplinary Robot Competition is an Internal Competition at EPFL (Interdisciplinary Project in EPFL Masters) where the goal is to make a fully autonomous robot able to detect, collect and bring to a collection point Duplos, while localizing in an Arena, avoiding obstacles and overcoming different challenges. The Robot has 10 minutes in the Arena to make a maximum of points by collecting Duplos.

## 1.2 Rules

First of all, let's present the Arena. We can find many Duplos distributed across different zones:

- **Zone 1** : This is the main zone composed of a normal carpet. Each Duplo of this zone is worth 10 points and there are 15 Duplos to be found. Multiple obstacles (plants, trash cans, couch, cardboard box, small wood table) can be seen within this zone.
- **Zone 2** : This zone is made of a fluffy carpet, which is pretty hard to navigate. There are 6 Duplos within this zone that count for 20 points each. There is also 1 obstacle.
- **Zone 3** : This zone is accessible through a door, that can be open by clicking on a button. There are 6 Duplos within this zone that count for 30 points each. There is also 1 obstacle
- **Zone 4** : This zone is placed on an elevated platform (30 cm approximately) and can be accessed through a ramp or stairs. There are 6 Duplos within this zone that count for 40 points each. There is also 1 obstacle.

There is a collection point where Duplos need to be dropped to get points. If the Duplos are dropped in the orange zone (1 $m^2$), 100% of the points addressed to the Duplos are collected. If they are collected in one of the three green squares of 1 $m^2$ each near the orange zone, only 50% of the points are collected. If the Duplos stays in the Robot, they are worth 25% of the original points. You can see a schematic of the arena in figure 1.

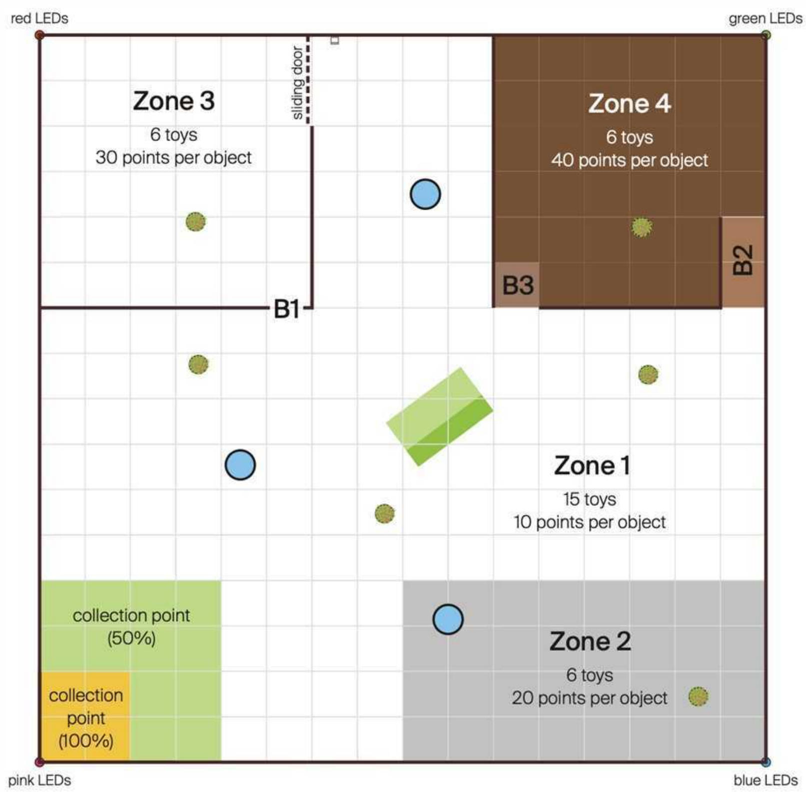



Figure 1: Schematic overview of the Arena

Moreover, those are the main rules:

- The Robot has only **10 minutes** in the Arena to perform the tasks.
- The Robot must be **fully autonomous**, except from start, stop and emergency. No remote controlling or data processing is allowed during the Competition. It must be done on board.
- There is **no limit in weight or size**, but the **minimum distance between walls and obstacles is 50 cm**, which we will consider for the dimensions of our Robot.
- **Power source must be on board** (own battery).
- Flying solutions are forbidden.
- All the components must be bought on the **virtual budget** (1500 CHF of a virtual catalog) or from the **real budget** (750 CHF to buy articles online, at a local shop or to manufacture pieces by 3D printing or laser cutting at SPOT/SKIL)
- During the Competition, we will be able to move the Robot after 15 seconds if it get stuck. We will be able to move its center in a 25 cm radius from the position it was stuck.

Those are the main rules we will consider for our functional analysis and the conception of our Robot. The complete list of rules can be found in the official rule book.

## 1.3 Strategy

Our strategy is based on storage : we want to be able to store as much as possible Duplos (between 6 and 8) for each back and forth at the collection point. Our Robot will first go to zone 3 and zone 4, before collecting Duplos in zone 1, to maximize the number of points. We will neglect the Duplos in zone 2 as going in the carpet implies to many limitations for the number of points it provides (20 points per Duplos compared to 30 for zone 3 and 40 for zone 4). Moreover, we are not considering the Duplos over the obstacles. Our objective is to make a system that can take a large number of Duplos, sufficiently fast and that can grab and release Duplos in the most efficient way. We will present our solutions to achieve our strategy in this report, considering the functional analysis, the budget management, and finally the mechanical, electrical and software architectures.

## 1.4 Team Composition and Objectives

Our team is composed of two students in the "robotique" master : Michel Abela and Alessandro Minghelli, which have great basis to work in this project. Florian Comte, a student in the "informatique" master, is completing the team, to help develop the software architecture of the Robot. When beginning this Competition, our objective was to break record with efficiency and storage, and we were all implied during the complete duration of the semester to make the best Robot possible to try to break records.

## 2 Functional Analysis

We first made a functional analysis to understand what are the main tasks that our Robot will need to make during the Competition.

### 2.1 Duplos Storage and Ejection

For the Storage, our objective was to have a lot of space to stock a maximum number of Duplos. For this to happen, we were reflecting on two solutions:

1. A large Robot with a lot of space in length.

2. A space more compact than the first idea, but where we can stack Duplos on each other.

We decided to go for the second choice, as we wanted to maximize the compactness of our Robot while taking the maximum number of Duplos. Furthermore, even if we are higher than the Arena walls, the LiDAR can be locked at 180 degrees, and that's what was done finally.
For the Ejection mechanism, we were thinking about those solutions:

1. Door opening and moving back and forth to release Duplos, but we find this solution not robust and fast enough for our objectives.

2. Storage floor that moves by 90 degrees to let Duplos out. This solution could have cause serious mechanical problems with the number of Duplos we wanted to pick.

3. Use of a Door and a Conveyor Belt System to release Duplos. After testing this with "Dycem" in week 4, we knew this was the path we would follow for the rest of the semester (Section **3.2**). During the Competition, it worked really well as a release method, even if the "Dycem" is a difficult material to place on a conveyor. It is also bringing motors to control, but we managed to keep everything working in a great way.

### 2.2 Picking Up Duplos

To pick up Duplos, and after seeing a video of the 2024 Robot winner "Team 5 - BlockBusters", we have seen that a Brush Mechanism was really effective. We thought about other solutions, such as an Arm, or a Shovel that would turn of 90 degrees to grab the LEGO. However, we decided finally to go for a Brush Mechanism (Section **3.1**).

To innovate compared to last year, as we wanted to stack Duplos one onto each other, we needed to have a system to bring up those Duplos after catching them. We first wanted to implement a Slope Conveyor System, but as we will discuss in section **6.1.2**, it was not robust and reliable enough after our first testings. We decided then to mix up some of our ideas, and to bring Duplos up with a Lift Mechanism (Section **3.3**) on top of a large Conveyor Belt, to directly eject Duplos in the Storage Space in the same way we eject them out of the Robot at the end of a trip. Furthermore, the Brushes (Section **3.1**) needed to be specifically designed to be able to throw Duplos over this Lift system.

### 2.3 Navigate in the Arena (Mechanically)

We needed to found a way of motion for the Robot to fulfill our objective. Multiple choices were discussed throughout the semester, such as:

1. System with 4 wheels powered by MAXON Motors. This was our priority at some point, but due to budget constraints (MAXON Motors, with ESCON 24/2 Controllers and the MM8 Adapters are each worth 182.1 CHF, which for 4 motors is worth 728.4 CHF) is worth and fixed priorities (we did not needed it as two wheels were powerful enough to push the Robot in Zone 4 + We decided not to go in Zone 2 with Brushes), we changed our mind.

2. At the end of the semester, we hesitate to incorporate a Chain Mechanism to be able to go to Zone 4 without problems. However, due to time constraints, budget constraints and the fact that we've seen that our prototype was able to climb in Zone 4 with a large chassis, we prefered to reflect on an easy solutions with two motorized Wheels and two passive Wheels.

3. One of the main idea was simply to incorporate two motorized Wheels and two passive Wheels. We finally went for that idea. This choice wasn't easy to make, as our Robot was pretty large at the end, but we found an intelligent way to implement this system (Section **3.6.3**).

The other solutions, such to use MAXON Motors with 1:18 Gearbox and not 1:60, were not great due to the weight of our Robot. We needed a lot of Torque.

## 2.4 Mechanical Schematic after Functional Analysis and Dimensions Analysis

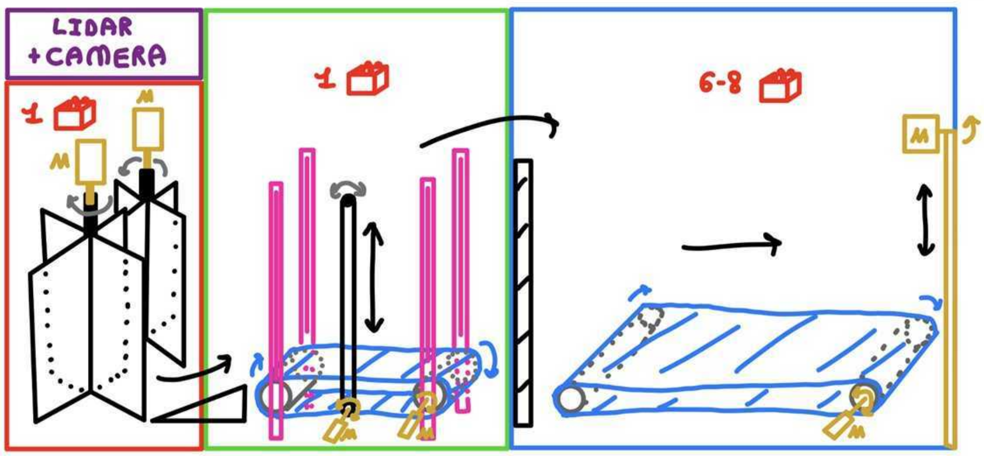



Figure 2: Internal Mechanical Schematic (Presented at Milestone 3, One Motor has been Added for the Lift to be Functional in the Final Design): Three Parts Design

At the Milestone 3, we were estimating distances and making a lot of Schematics to have a rough idea of the final Dimensions of our Robot. In fact, we were estimating the Conveyor System (Section **3.2.2** and **3.4.1**) to be 32 cm (Roller) x 20 cm x 17 cm (height of Storage) to be able to stock 8 to 10 large Duplos structures. We were estimating that the Lift Mechanism (Section **3.3**) was going to be 26 cm x 15 cm and that the Brushes (Section **3.1**) will be taking 14 cm of diameter. Considering we were thinking to put the Electronics on top of the Brushes, our rough estimate was a Robot of 55 cm x 40 cm at Milenstone 3. At the end, we needed some space between the Storage Space and the Lift Mechanism, the Conveyor of the Lift was 16 cm of length instead of 15 cm and we managed to use 18 cm for the Storage length instead of 20 cm. At the end, our Robot is measuring 62 cm x 47.7 cm (with the Wheels), which is definitely bigger but closed to our estimate. the height of the Lift is negligible (Section **3.3**) as our LiDAR was placed in an intelligent way in the front of the Robot and is locked at 180 degrees at 27 cm from the Ground. Also, we realized that we were able to pick up more than 10 Duplos with our final Storage Space, and that we were doing our tests with too big Duplos.

## 2.5 Duplos Detection

For Duplos Detection, our objective was to recognize and localize Duplos accurately and in real time, in order to guide the pickup mechanism efficiently. We directly decided to use a depth camera with a neural network because we wanted to apply some machine learning learned things in a concrete project and that the OAK D Lite was available in the virtual stock.

## 2.6 Obstacle Detection

In order to navigate autonomously, the Robot needed to reliably detect obstacles, both static (walls, structures) and dynamic (other robots, dropped Duplos).

We considered the following methods:

1. **Camera-based Obstacle Detection:** This would have involved detecting obstacles using visual cues. However, the reliability of this method is reduced under changing light conditions, and it adds computational complexity.

2. **Infrared or Ultrasonic Sensors:** These sensors are low-cost and simple, but have limited range and field of view, and tend to produce noisy data in crowded or reflective environments.

3. **2D LiDAR Scanning:** A robust and fast method that provides real-time distance measurements in a wide horizontal plane, capable of detecting both nearby and distant obstacles.

We chose the third option: a 2D LiDAR system positioned at 27 cm from the ground and fixed at 180 degrees. This configuration allowed us to detect surrounding obstacles efficiently and to build a dynamic local map. It worked especially well in detecting Arena walls, other robots, and obstacles left on the ground. The wide field of view and high accuracy of the LiDAR made it an ideal solution for real-time navigation and safety.

## 2.7 Localisation and Navigation

For localisation and navigation, the Robot needed to accurately estimate its position and orientation within the Arena and navigate through different zones while avoiding collisions.

Several strategies were discussed:

1. **Odometry Only:** Using wheel encoders to estimate movement. This method is simple but accumulates error over time, especially due to slippage or uneven terrain.

2. **LiDAR-Based SLAM:** Using LiDAR data to build a map and localize within it.

3. **Sensor Fusion:** Combining odometry with LiDAR and IMU data to correct drift and maintain an accurate estimate over time.

We opted for a hybrid approach based on **sensor fusion** (third option), combining:

- **Odometry**, for short-term displacement estimation;

- **IMU**, to detect orientation changes and improve heading estimation;

- **LiDAR**, to periodically correct position through scan-matching.

This strategy proved to be the most robust and computationally efficient. It handled dynamic obstacles and minor drift, keeping the Robot on track during complex tasks such as Duplos collection and return to drop-off zones. At first we only used Lidar and odometry but the results were not really good. IMU is really key!

# 3 Mechanical Design

## 3.1 Collecting Mechanism: TPU Brushes

For the collecting mechanism, we decided to opt for TPU brushes. We took the same mechanism for the brushes as the Team 5 of 2024 - "BlockBuster", and we modified the brushes to include them in the best possible way in our Robot.

Those TPU brushes measure 14 cm in diameter approximately and are a powerful way to catch a large number of Duplos of different sizes.

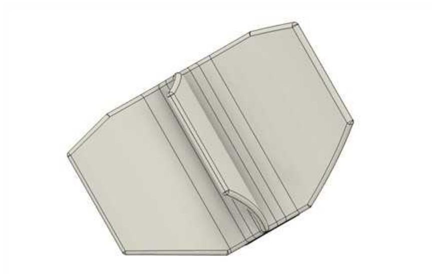


Figure 3: TPU Brush

Our biggest innovation in this regard is the fact that we wanted to rotate the brushes by 30 degrees to make the Duplos directly go on top of the Conveyor System of the Lift. To do this, we cut at 30 degrees the Brushes to make them closer to the ground. We also cut it by 15 degrees on the other side of the Brush if we wanted to change the angle afterward. We decided to choose a 25% infill for the TPU, to make it sufficiently strong to bring Duplos inside the robot and to catch them effectively. This was really effective. You can see how it turned out in figure 3.

The axis of the brushes in PETG and the motor holders in TPU remain unchanged compared to the "BlockBuster" team. We changed however the mechanical play inside of the TPU brush to make it fit better inside the axis (2 mm of mechanical play). This axis mechanism was really great, as it allows to include the coupler for the motor and to adjust the height of the brush with a threaded cap. You can see in figure 4 the axis mechanism of the brushes and in figure 5 the threaded cap.

Figure 4: Brushes Axis

Figure 5: Threaded Cap of the Axis

Figure 6: Lateral Support of the Brushes Mechanism, with holes to choose the Angle

However, we also changed the support of the brushes, that links the system to the basis of the robot. There are two supports in MDF 4 mm that are directly linked to a piece with holes that allow us to choose between angles of 10 to 30 degrees with a range of 5 degrees for each hole. We decided to choose 30 degrees in the final robot, as it worked pretty good in our tests. You can see how it looks in figure 6.

We did a similar mechanism for the PETG pieces placed in between of the MDF supports, that are holding the brushes. This allows us to choose the distance between the two brushes, from 16 cm to 22 cm with a range of 2 cm between each teeth. We choose 18 cm in the final version. There are two pieces for each brush; one of them is linked to the MDF supports, and the other is contained inside the first one and connected to the TPU motor holder. This piece can be adjusted in the axis of the supports to choose the distance between the brushes. We restricted the mechanical play to 0.2 mm in the axis going inside the robot, and added a mechanical play of 1 mm for the axis linking the two brushes in the teeth holes to allow the brushes to shift a little bit and help the Duplos absorption. You can see how it looks in figure 7

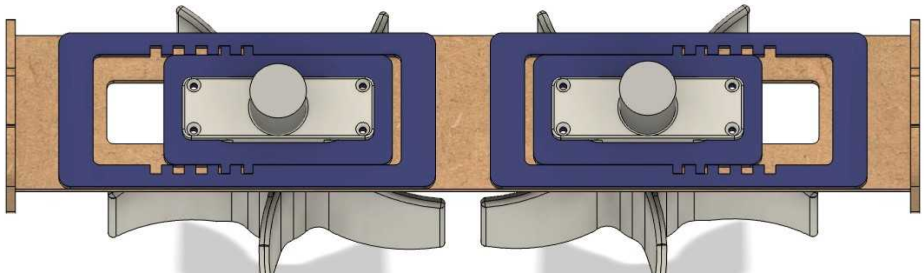


Figure 7: Brushes Support Pieces, with teethes to choose the distance in between Brushes

Finally, you can see in figure 8 the complete Brushes Mechanism in a CAD version and the first assembled version in figure 9 (the implementation in the Robot is explained in more details in section **3.6**).

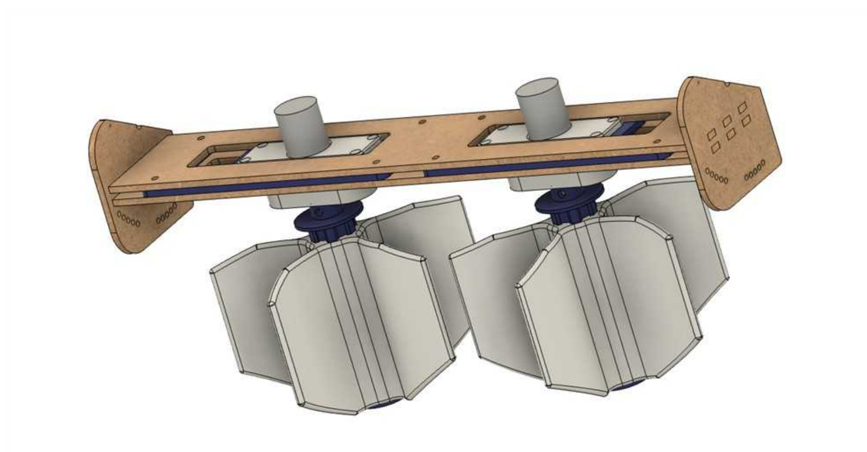


Figure 8: Overall Brushes Mechanism CAD Visualization

Figure 9: Brushes Mechanism Assembled (without Fixation to the Robot)

## 3.2 Conveyor Mechanism

To bring Duplos from one side to the other in our Robot on the longitudinal axis, from the entrance to the storage space, we first use Brushes, then we use two conveyor mechanisms. One of them is on top of a lift, the other one is our storage floor. The lift will allow us to stack Duplos on top of each other in the storage space. On the other hand, the conveyor of the storage space makes us able to eject Duplos at the Collection Point. We can see how important conveyors are in our Robot. In this regard, we will develop how we implemented customized conveyors in our final solution.

In this section, we will discuss what are the main components included in both conveyors.

### 3.2.1 Mat Surface Material Choice: ”Dycem”

When we first designed the conveyors, we were uncertain on the material to use as a surface. For our Robot, we first imagined that the lift mechanism was a conveyor slope of 30 to 45 degrees. To make it, we finally chose to go for the ”Dycem” material, used in medical environment on the ground. This material is anti-slip and is able to grab Duplos pretty well. However, we realized during those testing that the slope was not reliable and sufficiently robust. At this point in the semester (end of week 4), we decided to opt for a lift mechanism (and not a conveyor slope) to bring Duplos up to stack them in the storage area. Despite this change, we decided to keep the ”Dycem”, as it catch well Duplos, which is really great after collecting them with Brushes or to stack them in the storage space. You can see the ”Dycem” material in figure 10.

Figure 10: ”Dycem” Material

### 3.2.2 3D Printed Rollers

We will now discuss the roller that we 3D printed. First of all, our goal was to make a compact solution for the motorized roller and for the passive roller of the conveyors. We made our tests with ”Peko Rollen” from the virtual budget as a passive roller. However, we decided to opt for a similar design than the motorized roller that we 3D printed (PETG), as it was more compact and less expensive (a Peko Rollen is 40 CHF).

For the two rollers, we made two tubes of 40 mm of diameter that are able to connect from their extremities. In the center of the conveyor (at the contact point between the two pieces), the diameter is a little bit bigger (42 mm), to help keeping the ”Dycem” in place. For both the passive and the motorized roller, one of the side is passive. The other side of the passive roller is in fact still passive, but for the motorized roller, the other side includes space for a customized motor holder to get inside and a coupler hole to fix the motor inside the conveyor.

Figure 11: Female Tube CAD visualization (Storage Area)

Figure 12: Male Tube CAD visualization (Storage Area)

Figure 13: Roller Fixation part I

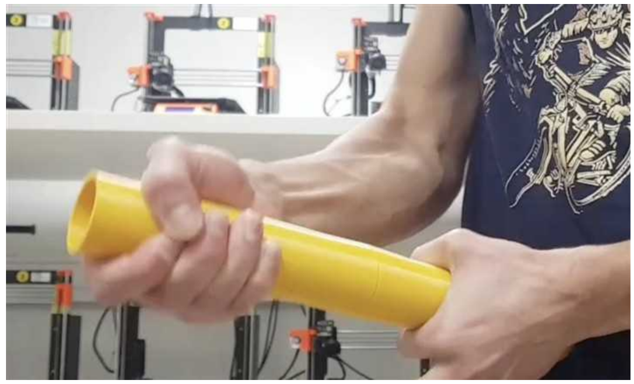


Figure 14: Roller Fixation part II

The passive sides of those conveyors are simple : we included by mechanical tightening 10x35x11 ball bearings, were we can include a 10 mm axis fixed to a basis. Those axis are simple M10 screws for the storage area conveyor, but we will see that we made custom 3D printed parts for the Lift Mechanism in section **3.3**.

Figure 15: CAD Example of a Passive Roller side (Storage Area)

The motorized size includes a hole to be able to include an hexagonal coupler. Also, there is space for a 3D printed (PETG) DC motor holder to get inside the roller. This makes the system really compact. This 3D printed holder has 3 holes in the outside that make us able to fix it with the conveyor to the basis with M3 screws and nuts.

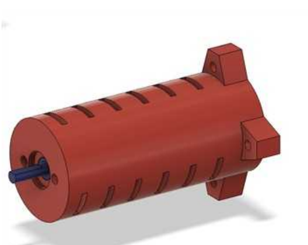


Figure 16: Pololu DC Motor (75:1) Holder

Figure 17: Motorized Roller Coupler Hole (Lift Mechanism)

Figure 18: Motorized Roller, Motorized side (Storage Area)

For both conveyors, the most complicated task was to fix the "Dycem" around the conveyors, such that they are nor too tense, nor too relaxed, and without too much inconstancy. We finally decided to fix them with dual-sided adhesive tape, after extensive testing with different type of bonding methods (such as gluing, adhesive tape, clip). It was the most robust solution. However, there is still a mechanical problem, that makes the mat moving left or right depending on the DC motor moving direction. We can fix this with the software, by making the "Dycem" mat going in the opposite direction after each task.

## 3.3 Collecting Mechanism: Lift Mechanism

With the Brushes, Duplos are directly sent on a Lift Mechanism, where a conveyor system as described in section **3.2** is mounted. The objective is to bring the Duplos upward, then to send them with the conveyor in the storage area. Bringing them up helps us to stack Duplos on top of each other and not only having Duplos on the ground. Moreover, we use the conveyor to help the Duplos get on the lift and not staying close to the brushes before activating the Lift. In figure 19, you can visualize a CAD version of the Lift Mechanism.

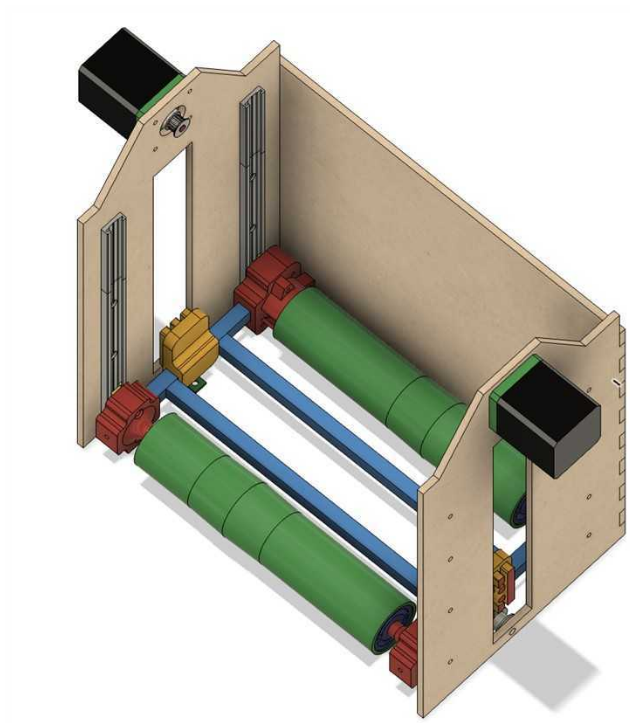


Figure 19: CAD visualization of the Lift Overall Mechanism

As the main objective of the lift is to allow to store more Duplos in the storage area, it should not take too much space in the Robot. That's why we made this lift the most compact possible. The conveyor includes the motor in its motorized roller, as mentioned in section **3.2.2**.

We then created 3D printed parts (PETG) that act as a link between IGUS rails and the rollers. For the passive side of the rollers, we made a cylindrical axis that can enter in the 10x35x11 ball bearings. A special part with 3 holes can receive the DC motor holder.

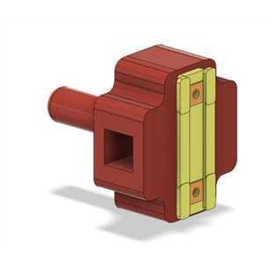


Figure 20: Passive Side Conveyor Holder: IGUS rail link and hole for MAKERBEAM to connect to the Lift

Figure 21: Passive Side Conveyor Holder: Cylindrical axis to connect to the Conveyor

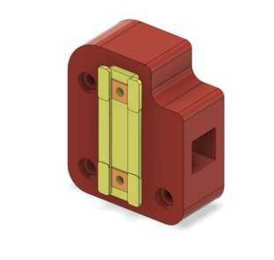


Figure 22: Motorized Side Conveyor Holder: IGUS rail link and hole for MAKERBEAM to connect to the Lift

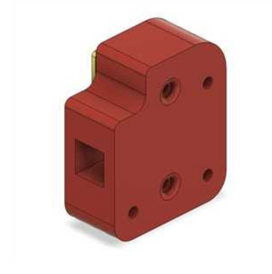


Figure 23: Motorized Side Conveyor Holder: Holes to connect with M3 Screws to the DC Motor Holder

Those parts are also linked between each other with MAKERBEAMS. We can adjust the distance between the rollers accordingly. Attached to those MAKERBEAMS on the side, we can also found 3D printed parts that allows us to link the conveyor to the chains and to control the Lift with Stepper Motors (more info in section **4.1**).

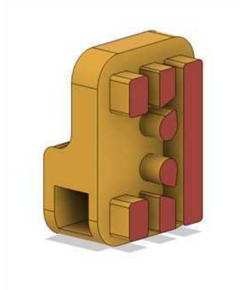


Figure 24: One of the Parts Linked to the Chains, linked to MAKERBEAMS taking up the whole Lift Structure

The IGUS rails are fixed to MDF 5 mm parts, and are screwed in a way that the Lift moves smoothly, without friction. The complete structure is linked to the base of the robot with MDF teethes from the back of the Lift (laser cut with MDF 5 mm), which is linked in the same way to the sides of the Lift.

For the chains, we choose them to be T2.5, and we take this into account for the lower passive link and for the coupler linked to the Stepper Motor. For the lower passive link of the chain, we designed a little part to fix it to the ground from both sides. This part is not necessary, as the link is screwed at the Lift MDF 5 mm plate; however, by fixing it directly to the ground, it is definitely more robust.

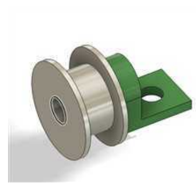


Figure 25: Little Part to fix the Passive Chain Link to the Base of the Robot

Furthermore, the Lift is able to bring Duplos up by 24 cm in this phase. (from the beginning of the Lift) When prototyping it, we realized the size was way to big to bring only one Duplo. That's why we finally decided to go for a Lift Mechanism of 18.5 cm x 29 cm for the Lift Mechanism without Stepper Motors and containing a Conveyor of 16 cm x 20 cm. The 29 cm in larger will allow us to include cables and some electrical components, as the Robot walls are separated from each others by 40 cm. We will see this in section **3.6**.

Figure 26: Lift Final Conveyor Mechanism Mounted (without Dycem and Walls)

You can see in the figures 27, 28 and 29 how the Lift Mechanism looks like in the final version of the Robot. In fact, it is very compact. We can see the "Dycem" material mounted on the Conveyor and the IGUS rails, as well as the Stepper Motors and the T2.5 chains. More information on the implementation of the Lift in the complete Mechanics of the Robot will be discussed in section **3.6**.

Figure 27: Lift Mechanism implemented in the Robot

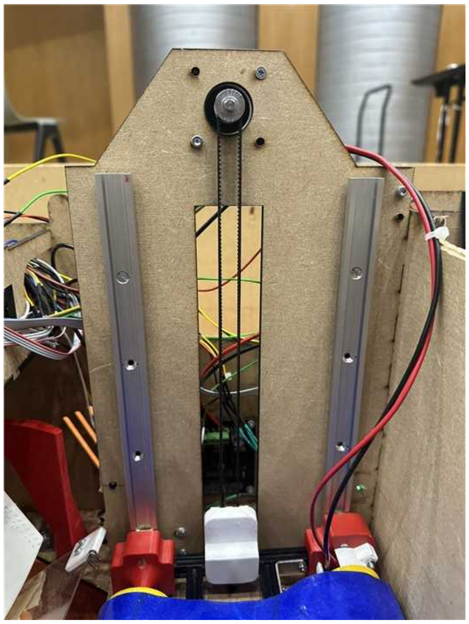


Figure 28: Side View of the Lift Mechanism

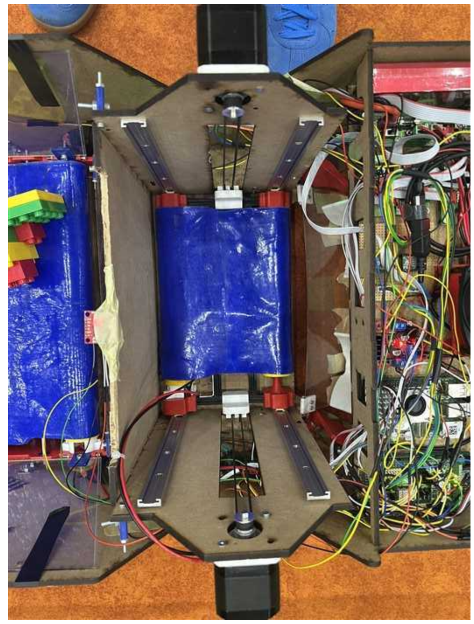


Figure 29: Lift Mechanism seen from the Top of the Robot

On the MDF 5 mm plate, we included some 5 mm of diameter holes for IR sensors to analyse when a Duplo structure enters the Lift but also to guarantee that he reaches the back of the Lift. The emitters are always placed on the opposite side of the receivers to analyze when the communication is stopped. There are two receivers for one emitter in the back of the Lift, for more robustness. One was sufficient in the front. The cables linking the emitter and the receivers through their board can pass in the space between the Lift and the Storage Space (section **3.6.1**). The boards and emitters/receivers are glued to the MDF 5 mm with hot glue.

## 3.4 Storage Area and Ejection Mechanism

### 3.4.1 Conveyor System Implementation in Storage Area

For the implementation of the Conveyor System in the Storage Area, we chose to use 31 cm large Rollers (as presented in section **3.2**) with a "Dycem" mat covering a distance of approximately 19 cm (Considering the Rollers centers are distant by 15.3 cm). We also fixed 2 mm PET walls around the Conveyor System on 3 sides, fixed on the MDF 4 mm of the Robot Walls and on the MDF 5 mm of the back of the Lift. In the back of the Robot, next to the Conveyor System, there is a Acrylic 3 mm Door that opens when getting to the Storage Area to eject Duplos with the Conveyor System (Section **3.4.2**). To be able to fix the Conveyor System to both the basis and the different walls, we crafted 3D printed Conveyor Holders. It follows the shape of the Robot, as you will see in section **3.6.1**. Those two Holders let pass M10 screws to holds the passive sides of the Conveyor System (Section **3.2.2**) and can attach to the Motor Holder of the Conveyor on one point of one of the Holders. There is also a hole to let pass the cables of the DC Motor 75:1. Furthermore, we wanted our "Dycem" to not propagate too low considering the number of LEGOs getting on it. To compensate this, we integrated two steel bars going from one side to another. They are integrated in the Conveyor Holder with 5x14x5 ball bearings, such that they are able to turn with the Conveyor movement when needed and to support LEGOs on the Conveyor. Those bars are spaced by 53.75 cm approximately, and the top of them is placed 7 mm below the top of the Conveyor Rollers. Moreover, they are centered in this part. In figures 30, 31 and 32, you can see how we designed the Storage Area in a CAD version. Furthermore, you will see where everything is attached in a better way in section **3.6.1**.

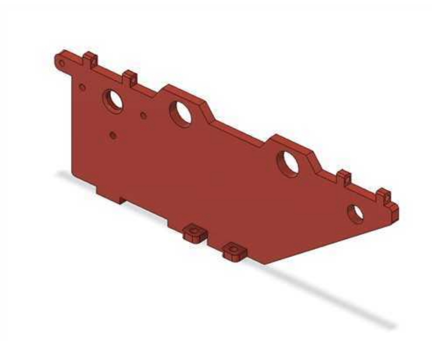


Figure 30: Conveyor Holder for the Passive Conveyor Fixation

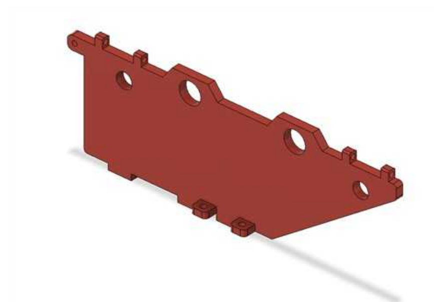


Figure 31: Conveyor Holder for the Motorized Side Fixation

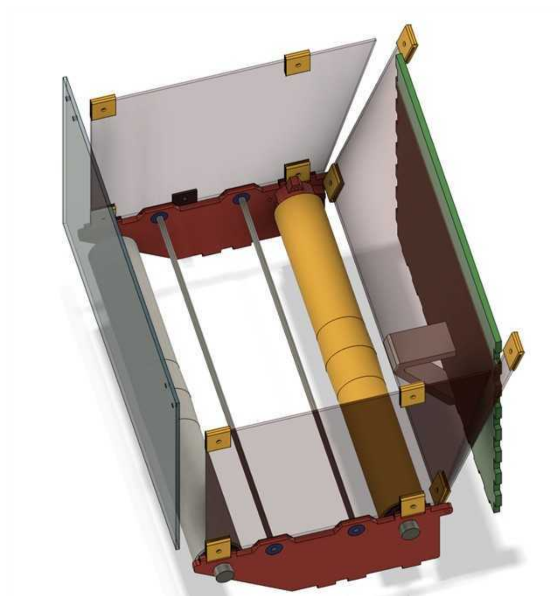


Figure 32: Overall CAD Visualization of the Storage Area

Also, you can see the complete implementation in the figure 33 the complete implementation. You can also notice that the Motor Holder has been placed in the other side of the Robot in the final Robot. This has been done on purpose such that the two Conveyors DC Motors 75:1 (Lift Mechanism and Storage Space) are placed on the same side, next to the DFRobot 2x15A Motor Driver.

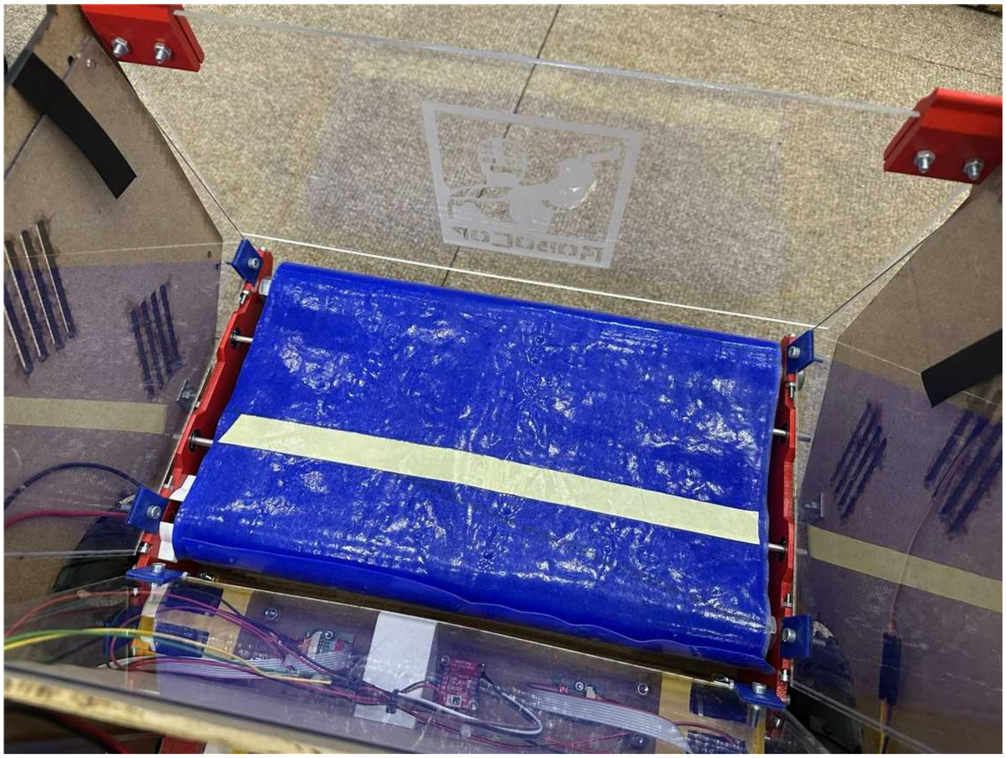


Figure 33: Final Conveyor Storage Area Implementation

#### 3.4.2 Door Mechanism

The Door Mechanism is pretty simple. We used a Reely Numerical Servo S-5252-030, 28 Ncm to open it and maintain it closed for all the 10 minutes. For that, we designed two Door Holders, one which is coupled with the Servo Motor and the other one which is passive, letting a M4 Screw holding it with the side of the Robot. The Servo Motor is linked to the other side with a Motor Holder placed behind the MDF 4 mm of the wall, which contains also a rectangular hole to let the Servo Motor pass through. The Door has a 18.2 cm x 39.2 cm size and is also fixed with M4 screws to the Holders. We also Laser Engraved with a DXF file a "RoboCop" picture in the back of the Robot. You can visualize in figure 33 the "RoboCop" image engraved, and in figures 34, 35, 36 and 37 how the parts holding the Door to the walls of the Robot are designed in our final CAD version.

Figure 34: Passive Door Holder, with M4 Fixation

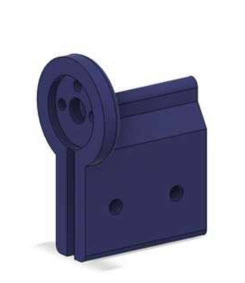


Figure 35: Motorized Door Holder (with Coupler Hole)

Figure 36: Motor Holder, with Space for the Servo Motor to get in and M4 Fixations

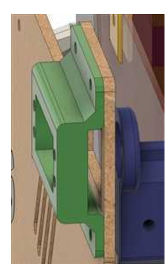


Figure 37: Motor Holder Placement Compared to the Left Wall of the Robot

## 3.5 Electronic Box

Our Robot is pretty complicated: it has 9 motors to control and needs to catch information from multiples sensors (IR, IMU) and components (Camera and LIDAR), as we will discuss in section **4.1**. It is thus essential to reflect on a great Electronic Box, were the main boards and electronic components will be located, in a safe way and in a great place to provide a robust cable management overall. We decided to include this electronic box in between the side walls of the robot (40 cm of distance) and on top of the Brushes, to avoid contacts with the Lift and the Storage Area. Furthermore, we reflect on an intelligent way to include the Camera and the LIDAR close to this box.

The final result is a MDF 4 mm box with an inside volume of 40 cm x 14.2 cm x 60.1 cm, as you can see in figure 39. We decided to put a Plexiglass 2 mm on top that can easily be removed by hand to make changes. Moreover, it is magnetically attached at 2 points, to be sure that the plate won't get out during runtime. It also includes a hole to let space for the LIDAR to go through. We also made a custom engraving referencing the film "RoboCop", as you can see in the final version of the top plate in figure 38.

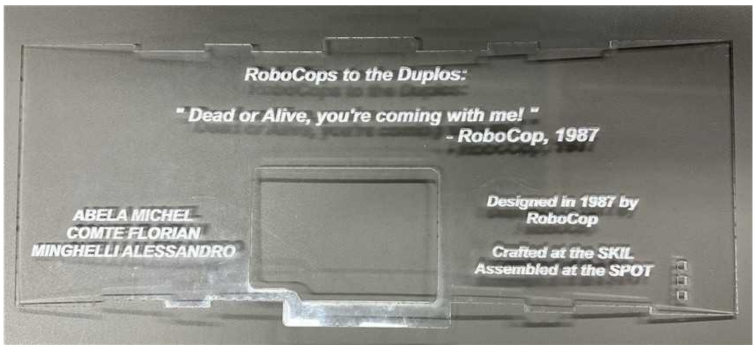



Figure 38: Electronic Box Plexiglass Top Plate (Engraved, Final Version)

On the Bottom Plate, holes to place our electrical components and the LIDAR support are available. In figure 40, you can see how we included the electronic in the Box.

On the plate in the side, holes for the Camera and windows are available to let cables get through. We can see in figure 39 the bigger windows on the side, that are linked to the side of the lift, were 5 cm on each side are available for electrical cable management going to the motors or other boards that needed to be out of this Box, such as the DFRobot Motor Controller, controlling the two conveyors. We will discuss the boards that are not inside the Electronic Box in section **3.6**.

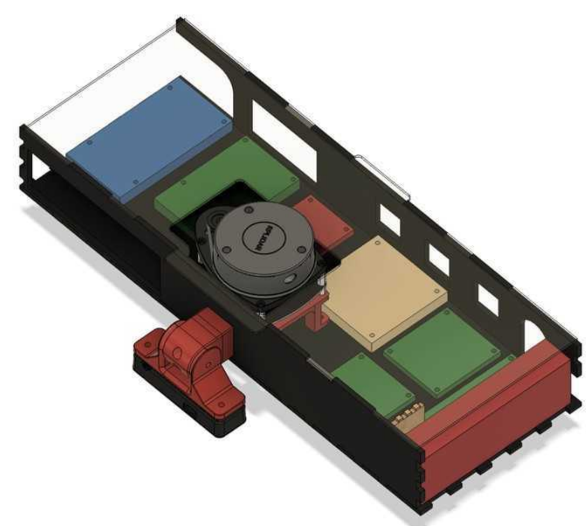


Figure 39: Electronic Box CAD Visualization

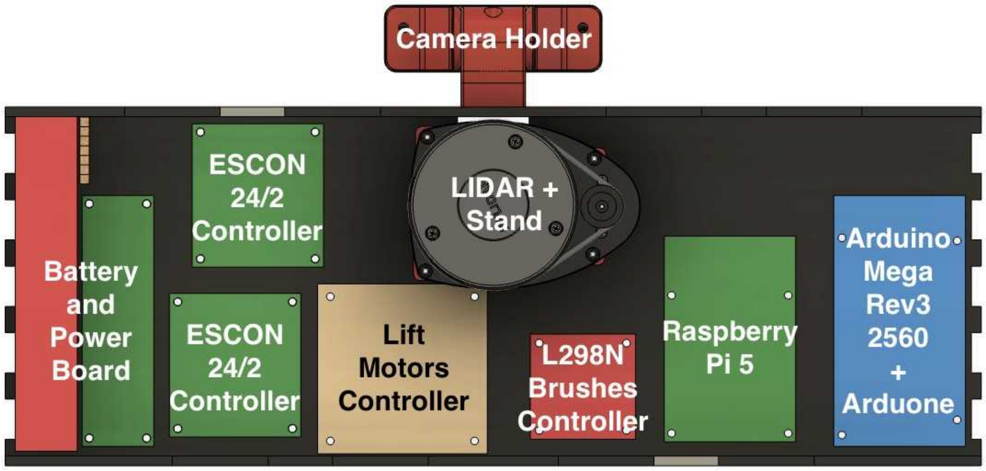



Figure 40: Electronic Components Disposition

The LIDAR eye is located at 29 cm from the ground, with its head being just out on the top of the electrical box. We made a custom 3D Printed (PETG) stand such that the body of the LIDAR is in the electronic box, and that the LIDAR head is positioned just outside. You can see the cad of this stand in figure 41. Furthermore, you can notice the curvature on the side near the Camera Holder, letting screws get in, but also the holes between the feets of the stand letting cables go through.

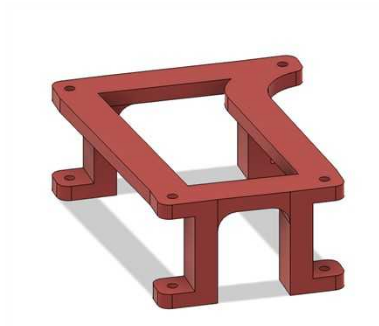


Figure 41: LIDAR Stand CAD Visualization

Finally, the Camera Holder is located at the front of the Box, next to the LIDAR. Those two components are placed in a way such that they are close to the Raspberry Pi 5. A hole in the front MDF 4 mm plate allows cables to access to the Camera for both information and power. The Camera Holder is made in two parts, in a way that we can adjust the position of the Camera angle. You can see a representation in figure 42.

Figure 42: Camera Holder CAD Visualization

## 3.6 Overall Mechanical Design

### 3.6.1 Storage Area and Lift Mechanism/Walls Link

To link the Storage Area to the Lift Mechanism, we use 4 PET 2 mm Holders to fix 2 of them on the Conveyor Holder on one side, and the 2 others on the side of the Lift Mechanism MDF 5 mm (by the Outside). This PET 2 mm plate (192 cm x 330 cm) has a U-Shape at the top, to fit with the back of the Lift. M3 screws are linked to those basis and M4 screws are fixing the PET 2 mm from both sides. For the walls, we were supposed to use a similar principle, but we chose to use scotch to link it to the wall and to fix the Conveyor Holder in the same way as before. For the two walls of the Storage Area, we used PET 2 mm with a triangular shape next to the Lift, to follow the angle of the PET 2 mm plate fixed on the back of the Lift. Those plates are large by 23.5 cm approximately on the top, 19.2 cm on the bottom and have an height of 16.1 cm. In the final Robot, we needed to cut by 3 mm the bottom of those plates, because of the changes we made to the Conveyor Holders in week 14 (explained in section **6.1.4**). You can see how this looks in a CAD version in figure 43 and in a final version in the assembled Robot in figure 44.

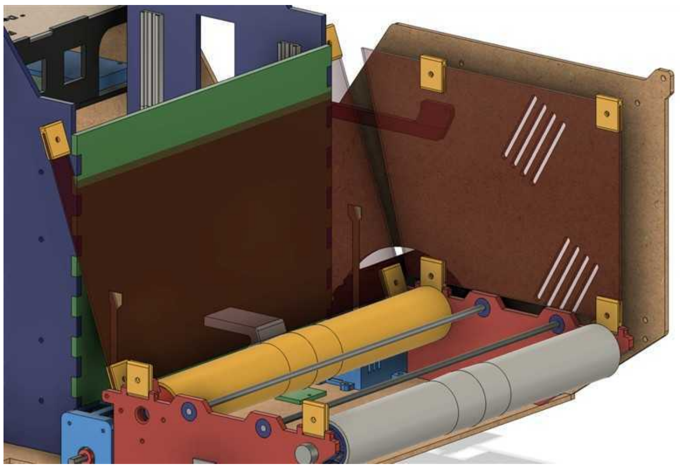


Figure 43: Lift Mechanism/Robot MDF Walls and Storage Space Link with PET 2 mm

Figure 44: Lift Mechanism/Robot MDF Walls and Storage Space Link with PET 2 mm in the Final Robot

Furthermore, we intelligently used the space between the Storage Space and the Lift Mechanism and under the PET 2 mm to incorporate the IMU on the basis of the Robot, but also the two MAXON Motors Holder. This Space was initially supposed to be for the Storage Space, but we implemented the PET 2 mm especially to have some space to put the Motor Holders and MM8 Adapter such that the Robot measures 47.7 cm large and could enter in Zone 4 4 (50 cm between the walls). Also, we use it for the Cable Management, especially for the IR of the Lift Emitter-Receiver link, the IMU and the Motors Parts (figures 45, 46 and 48). We also included a support of this PET 2 mm glass in PETG, to be sure that pressure of the Duplos won't bring any little Duplos under the Conveyor Belt (figure 47).

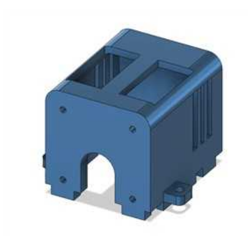


Figure 45: MAXON Motor Holder from the Front: Fixation to the Basis of the Robot and Fixation Holes (M4 screws)

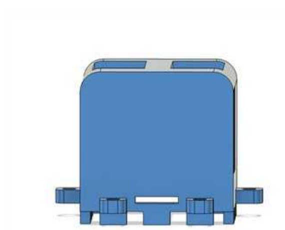


Figure 46: MAXON Motor Holder from the Back: Hole for the Cable to Pass

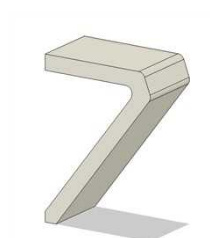


Figure 47: Support of the PET 2 mm Between the Lift and the Storage Space

Figure 48: Overview of the Back of the Lift with IMU, Motors Holders and Support Part

### 3.6.2 Entrance of the Duplos

We made an Entrance Slope pretty steep, with an angle of 35 degrees from the Ground. We use the Power tested of the Brushes and the option to put the Brushes at a 30 degrees angle to make this choice, which has largely influenced in a positive way the compactness of the length of our final Robot, without compromising the efficiency of the brushes. The basis of the Robot let this PET 1 mm glass of 11.7 cm x 28.1 cm go in direction of the Ground. We made two Supports Placed next to the Lift Mechanism Walls where we can hold the Ramp with a M3 screw. The ramp is linked to those supports with M2.5 screws. The support is sufficiently solid in shape and in infill to not break, with M4 screws fixing it to the basis of the Robot (figure 50). Furthermore, as we encountered problems with the Slope turning around in the Robot, we fixed an elastic under the Ramp, from one side to another, with circular parts on top of the L-Brackets connecting the Walls to the Basis. We also added two 3D Printed parts of 7 cm in front to glide over little obstacles such that the Slope don't get stuck. We tested different variations of this parts before choosing this one in particular (figure 49). We can see the final CAD version of the slope in figure 51.

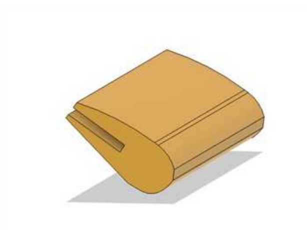


Figure 49: Front Support for the Slope

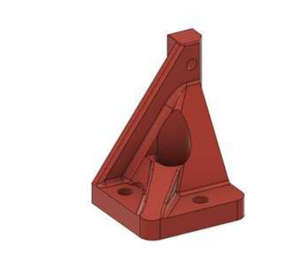


Figure 50: Support for the Slope: Linked to the Basis

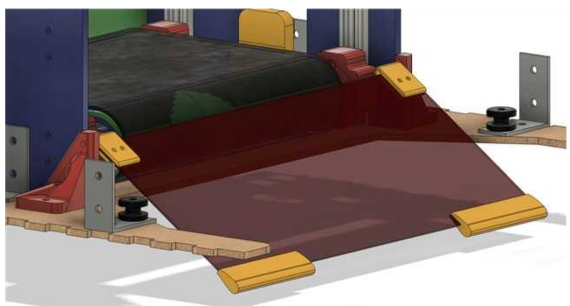


Figure 51: Slope Overall Mechanism, we can notice the Slope Holder, the Elastic Holders over the L-Brackets, and the Front Support

Furthermore, we included a Guide at the front, fixed to the Wall of the Robot such that the Duplos are pushed mechanically in direction of the Brushes (figure 53). We also included an internal protection such that bigger Duplos don't turn around in the Brushes and get stuck between the walls and the Brushes, potentially stopping the Brushes for the run (figure 52).

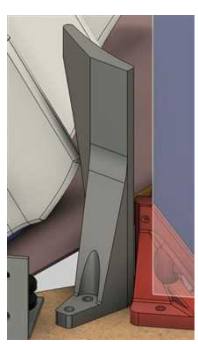


Figure 52: Internal Protection

Figure 53: Duplos Guide in the Final Robot

#### 3.6.3 Basis and Walls of the Robot

For the Basis and the Walls of the Robot, we used 4 mm MDF on each side. They are linked with L-Brackets and MDF teethes. The plates contain all the necessary holes and teethes to fix the parts mentioned earlier, as well as space for the Wheels (**WildTumper 120x60 mm wheel, 4 mm shaft**) and to see the Lift from outside. We also let some space in the front to allow the Entrance Slope to pass, and we only advance in the side of the Entrance Slope by the amount we decided to put the Passive Wheels Holders (figure 57). We determined the length between the back of the Basis plate and the MAXON Motors to be 15 cm after testing a first version in Zone 4. The Motorized Wheel and the Passive Wheel (**50mm omni wheel**) are spaced by 37 cm. There are also some holes for cables to pass on the Walls. Furthermore, we engraved "RoboCops" on the Side Walls of the Robot. Notice that on the Right Wall, there is the space and holes for the DFRobot 2x15A board, driving the Conveyors from that side (figure 55).

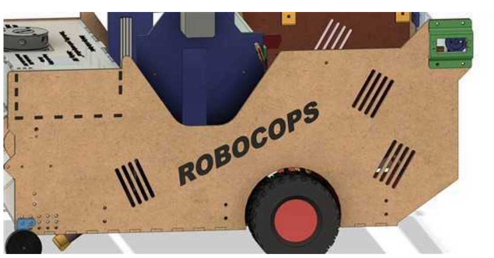



Figure 54: Left Wall of the Robot


Figure 55: Right Wall of the Robot

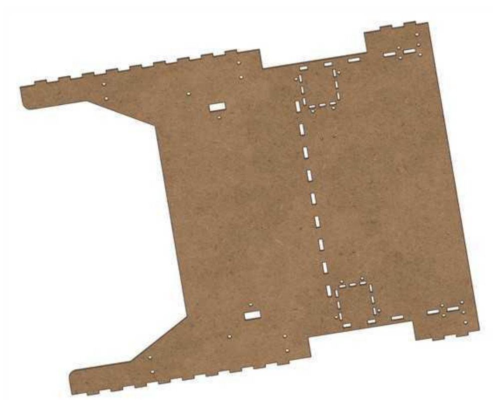


Figure 56: Basis of the Robot

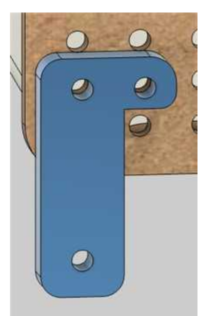


Figure 57: Passive Wheel Holder (3D Printed with Infill 90% for Robustness)

#### 3.6.4 Final Touches and Final Robot Model

Finally, let's discuss about some of our final touches to make our Robot fully prepared for the Competition. First of all, we implemented an Ultrasonic Sensor in the right side of the Robot (figures 58 and 59) to be able to detect more precisely if the Door of Zone 3 has been opened. To do this, a 3D printed Stand was made and attached to the right wall with two M3 screws. The direction is adjustable with the Holder part of the Stand with an M3 screw and nut linking it to the other part.

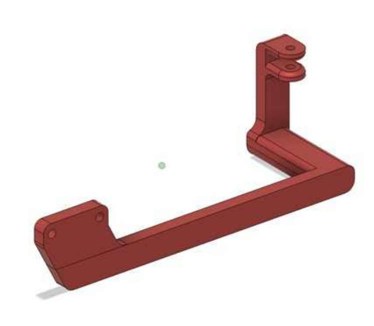


Figure 58: Ultrasonic Sensor Stand: Fixation to the Right Wall and Link to the Holder

Figure 59: Ultrasonic Sensor Stand: Holder

Finally, we added thread-lock to the coupler screws of the Brushes and the Wheels, to be sure that those two important modules won't get out during the Competition duration. We also put wood glue to fix some teeth MDF parts (such as the Electronic Box link to the Walls or the Brushes Mechanism) and hot glue to fix the IR Sensors and some cables. We also scotched and glued other parts, such as the Storage PET 2 mm Side Walls or the two guiding parts in front of the Entrance Slope. In figure 60, you can visualize a final version of our CAD model.

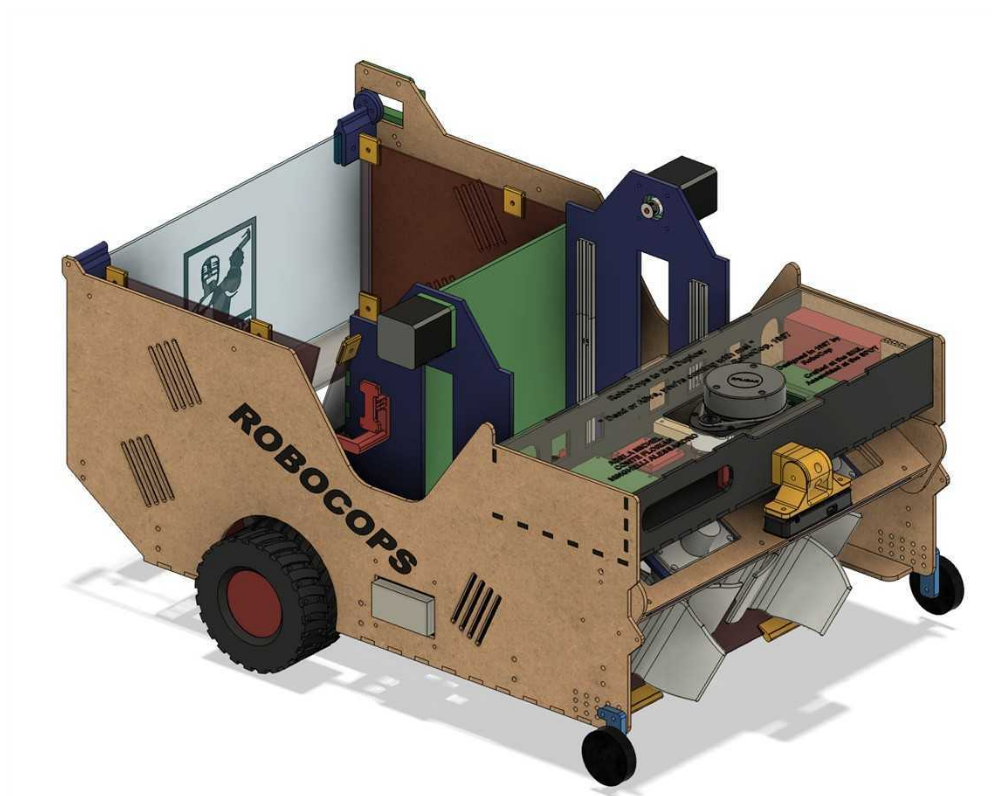



Figure 60: Final CAD Model

# 4 Electrical Design

## 4.1 Overall Electrical Design and Components Choices

For the Electrical Design of our Robot, we reflected a lot to make it compact, even with the large number of motors we need to control. In fact, considering all subsystems, we have to control 9 motors. To make this possible, we made an Electronic Box, that has been described in section **3.5**, where we can find the biggest part of our electronic components. With the exception of the motors, the only components that are out of this electrical box are the DFRobot 2x15A driver (which can be found next to the conveyors), the sensors (IMU, Ultrasonic Sensor, IR Sensors) and the EC Flat to MM8 adapter (due to distance constraints with the MAXON motors). In the page **27**, you can see how we control our Robot in an Electrical Schematic.

The main Electrical Components of our Robot are:

**Power Boards and Computation**

- **Raspberry Pi 5**: We chose a Raspberry Pi 5 (8GB RAM) for the largest number of computation from our Robot (except some computations done directly by the Camera). It allows us to do everything we want.
- **Arduino Mega 2560 Rev 3 + Arduone v1.21**: The Arduino allows us to interface with all our electrical components. With all the motors we control, the Arduino Mega 2560 Rev 3 was a necessity. The Arduone Power Board on top allows us easily to connect each of the motors to its Arduino pins, but also to power pins, with GND, 5V pins but also 11.1V accesses coming from the Battery.
- **LIPO Battery 3S 4000mAh + Power and Protection Board**: This battery gives us approximately (nominal voltage) 11.1V. It allows us to power all our Robot Electrical components. During the end of the Competition, we had two of them to be able to test continuously the Software implemented in the Robot.

**Motors implemented in our Robot**

- **MAXON Motors with Encoder EC 32 Flat + Gearbox 1:60**: We have two of them to control the Robot Motorized Wheels. As we have a pretty heavy Robot, the 1:60 Gearboxes on the two wheels allow us to move quite fast compared to the weight of our Robot, but more importantly to give us enough torque to climb the slope that leads to Zone 4.
- **Polulu DC Motors 75:1**: We have four of them to control the Brushes and the Conveyors.
- **Stepper Motors 42HD4027-01, 1.8 degrees, 0.4Nm**: We have two of them to control the Lift Mechanism from both sides.
- **Reely Numerical Servo S-5252-030, 28 Ncm**: Make us able to open and close the Back Door, but also to keep the Door closed at each time.

**Motors Drivers and Controllers**

- **ESCON 24/2 Controller and Motherboard with EC Flat to MM8 Adapter**: Allow us to connect and control easily our MAXON Motors. We have two of them, one for each motor. (Configuration for the ESCON 24/2 explained in more details in section **4.2**)
- **DFRobot DRI0018 2x15A Driver**: Make us able to control with sufficiently high current our two Conveyors (Pololu DC Motors 75:1)
- **L298N Driver**: Make us able to drive the two Brushes (Pololu DC Motors 75:1).
- **Custom Board** to control the Lift Stepper Motors **(with Stepper Drivers A4988 2A)**. This board allows us to fix with crimped cables more safely all the cables needed for the Lift Mechanism, both in term of power and program (Arduino Link). For each Stepper, we use the same "Enable" (Blue), one "Step" (PWM - Green Cables), one "Direction" (Yellow Cables), a 5 V alimentation for the Drivers and 11.1 V for the motors to work. The objective is to be able to directly replace a Driver if there is a problem, but also to allow a better cable management.

**Sensors, Localisation, Obstacles and LEGO Detection**

- **RP LIDAR A1M8**: We use this component to localize ourself in the arena with SLAM, allowing the Robot to know how to move from one Zone to another in the Arena, but also to avoid obstacles.
- **OAK D-Lite Luxonis**: We use this Camera with a pre-trained and augmented YOLO model able to detect Duplos. This Camera can do computations on board. Normally, an IMU is integrated in this Camera. However, some version were sold without it, and it is unfortunately the case for our Luxonis. To fix this problem, we integrated an IMU on the Robot, next to the Storage Area and the MAXON Motors.
- **Ultrasonic Sensor HC-SR04**: We use one of them to verify more clearly if Zone 3 door is opened after pressing the button, if not, the Robot will try again.
- **Arduino IR Sensors**: We decided to put three of them, with the emitter on one side of the Lift Mechanism and the receiver on the other side. One of them is at the front, and the two others are at the back of the Lift (with only one emitting and two receiving, to check edge cases): by software, we made a Duplo advance in the conveyor of the Lift until he reaches one of the two back sensors, before making it going up. This allows us to be sure that the Duplo is in the good place and catched.
- **IMU 9DOF SparkFun ICM-20948**: We are using an IMU to be able to do SLAM in a better way, as our Camera did not provide an integrated IMU.

## 4.2 Configuration of Wheels MAXON Motors

Here you can find how we configured in the Setup Wizard of ESCON Studio our two ESCON 24/2 boards controlling our two MAXON Motors EC 32 Flat with Gearboxes 1:60. We are using those two of them to provide Differential Drive. Here are the steps to follow in the Setup Wizard:

- Motor Type: Maxon EC motor
- Motor Data: 427 Speed Constant / 8.8s / 4 pole pairs
- System Data: 10 000 RPM / 1.06 A / 3.06 A
- Detection of Rotor Position: Digital Hall Sensor / Maxon
- Speed Sensor: Available Hall Sensor
- Mode of Operation: Speed Controller (closed-loop)
- Enable: Enable & direction / Digital Input 2 & 3 / both High Active
- Set Value: PWM set value / Digital input 1 / 10% –> 0 RPM / 90% –> 10 000 RPM
- Current Limit: Fixed Current Limit / 3.06 A
- Speed Ramp: Fixed Ramp / 10 000 RPM/s / 10 000 RPM/s
- Offset: Fixed Offset / 0 RPM
- Minimal Speed: 0 RPM
- Digital Input/Output: Digital Input 1 –> PWM - Set Value / Digital Input 2 –> Enable / Digital Input 3 –> Direction / Digital Input/Output 4 –> None
- Analog Input: Analog Input 1 –> None / Analog Input 2 –> None
- Analog Output: Analog Output 1 –> Actual Current Averaged / Analog Output 2 –> Actual Speed Averaged
- Analog Output 1: 0V –> -3.06A / 4V –> 3.06A RPM
- Analog Output 2: 0V –> -10 000 RPM / 4V –> 10 000 RPM

After the Setup Wizard, you can run the Regulation Tuning in Auto Mode.

## 4.3 Arduino Pins Table

In figure 61, you can visualize to which Arduino Pins each Modules are linked.

| Pins | Type | I/O | To... | |
|---|---|---|---|---|
| D6 | PWM | Output | MAXON Motor 1 (DIGITAL INPUT 1) | Through ESCON |
| D7 | PWM | Output | MAXON Motor 2 (DIGITAL INPUT 1) | |
| D36 | Digital | Output | MAXON Motor 1 (DIGITAL INPUT 2 : Enable) | |
| D37 | Digital | Output | MAXON Motor 2 (DIGITAL INPUT 2 : Enable) | |
| D34 | DIgital | Output | MAXON Motor 1 (DIGITAL INPUT 3 : Direction) | |
| D35 | Digital | Output | MAXON Motor 2 (DIGITAL INPUT 3 : Direction) | |
| A4 | Analog | Input | MAXON Motor 1 (ANALOG OUTPUT 1 : Actual Current Averaged) | |
| A5 | Analog | Input | MAXON Motor 1 (ANALOG OUTPUT 2 : Actual Speed Averaged) | |
| A6 | Analog | Input | MAXON Motor 2 (ANALOG OUTPUT 1 : Actual Current Averaged) | |
| A7 | Analog | Input | MAXON Motor 2 (ANALOG OUTPUT 2 : Actual Speed Averaged) | |
| D50 | Digital | Output | Lift Enable | Custom PCB |
| D48 | Digital | Output | Lift Motor 1 Direction | |
| D42 | Digital | Output | Lift Motor 2 Direction | |
| D46 | PWM | Output | Lift Motor 1 Step | |
| D44 | PWM | Output | Lift Motor 2 Step | |
| D24 | Digital | Output | Brushes Motor 1 Direction In1 | With L298N |
| D30 | Digital | Output | Brushes Motor 2 Direction In4 | |
| D26 | Digital | Output | Brushes Motor 1 Direction In2 | |
| D28 | Digital | Output | Brushes Motor 2 Direction In3 | |
| D4 | PWM | Output | Brushes Enable 1 PWM | |
| D5 | PWM | Output | Brushes Enable 2 PWM | |
| D27 | Digital | Output | Conveyor Motor 1 Enable EN1 (for Direction) | With DFRobot |
| D29 | Digital | Output | Conveyor Motor 2 Enable EN2 (for Direction) | |
| D8 | PWM | Output | Conveyor Motor 1 PWM | |
| D9 | PWM | Output | Conveyor Motor 2 PWM | |
| D2 | PWM | Output | Door Servo Motor 1 | |
| D3 | PWM | Output | Door Gate Servo Motor 2 | |
| D45 | Digital | Input | Echo Ultrasound | |
| D47 | Digital | Output | Trigger Ultrasound | |

Figure 61: Arduino Pins Table

## 4.4 Electrical Box In the Final Robot

You can see how the Electrical Box is implemented in the Robot final version in figure 62.

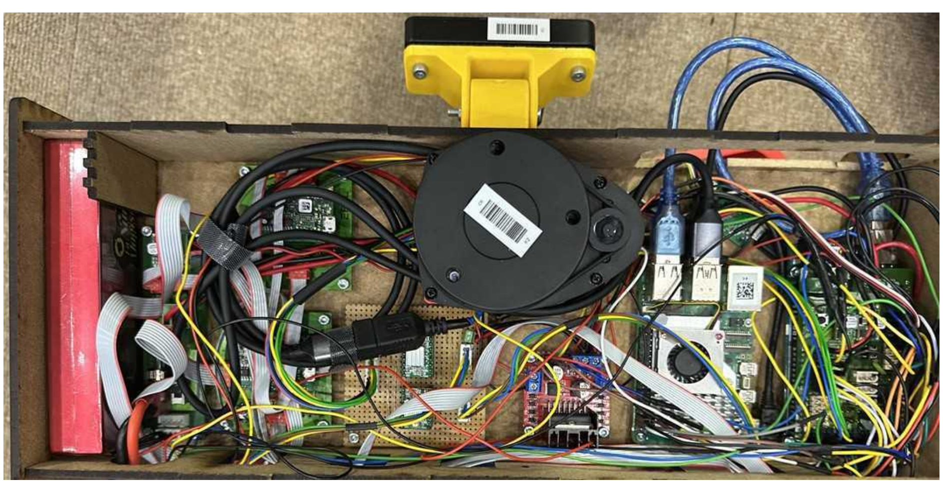
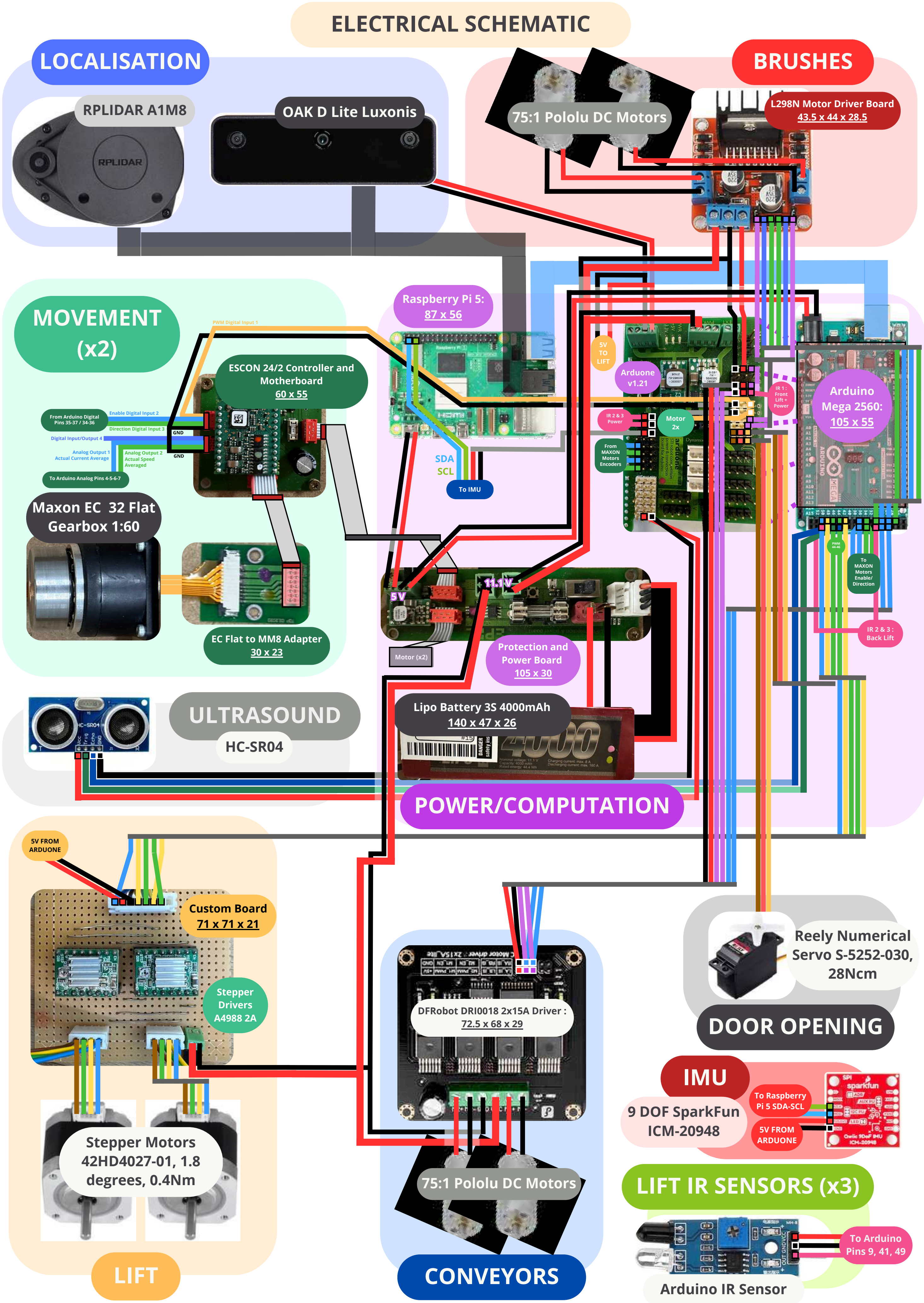


Figure 62: Electrical Box Assembled and Connected



ELECTRICAL SCHEMATIC
LOCALISATION
RPLIDAR A1M8
OAK D Lite Luxonis
BRUSHES
75:1 Pololu DC Motors
L298N Motor Driver Board
43.5 x 44 x 28.5
MOVEMENT (x2)
PWM Digital Input 1
ESCON 24/2 Controller and Motherboard
60 x 55
From Arduino Digital Pins 35-37 / 34-36
Enable Digital Input 2
Direction Digital Input 3
GND
Digital Input/Output 4
Analog Output 1 Actual Current Average
Analog Output 2 Actual Speed Averaged
GND
To Arduino Analog Pins 4-5-6-7
Maxon EC 32 Flat Gearbox 1:60
EC Flat to MM8 Adapter
30 x 23
Raspberry Pi 5:
87 x 56
SDA
SCL
To IMU
5V TO LIFT
Arduone v1.21
IR 2 & 3 Power
Motor 2x
From MAXON Motors Encoders
IR 1 : Front Lift + Power
Arduino Mega 2560:
105 x 55
PWM 44-46
To MAXON Motors Enable/ Direction
IR 2 & 3 : Back Lift
5V
11.1V
Motor (x2)
Protection and Power Board
105 x 30
Lipo Battery 3S 4000mAh
140 x 47 x 26
ULTRASOUND
HC-SR04
POWER/COMPUTATION
5V FROM ARDUONE
Custom Board
71 x 71 x 21
Stepper Drivers A4988 2A
Stepper Motors 42HD4027-01, 1.8 degrees, 0.4Nm
LIFT
DFRobot DRI0018 2x15A Driver :
72.5 x 68 x 29
75:1 Pololu DC Motors
CONVEYORS
Reely Numerical Servo S-5252-030, 28Ncm
DOOR OPENING
IMU
9 DOF SparkFun ICM-20948
To Raspberry Pi 5 SDA-SCL
5V FROM ARDUONE
LIFT IR SENSORS (x3)
To Arduino Pins 9, 41, 49
Arduino IR Sensor

# 5 Software Architecture

The software architecture of our robot is designed around modularity, real-time responsiveness, and efficient hardware utilization. The following subsections outline the technologies selected, vision and control strategies, and the structure of our software stack across the different computing platforms onboard the robot.

## 5.1 Technologies Selection

- **ROS 2 Jazzy**: Utilized on the Raspberry Pi with C++-based packages to handle robot control, sensor integration, and inter-process communication.
- **C++ for Arduino**: Employed for low-level motor and sensor control on the Arduino Mega, ensuring efficient real-time operations.
- **YOLOv11 Nano Model**: Deployed directly on the OAK-D Lite camera, leveraging its edge computing capabilities to perform object detection and send inference results to the main processor.

## 5.2 Computer Vision

We utilize an Oak-D Lite camera equipped with a `YOLOv11-Nano` model, trained on an augmented version of last year's winning dataset. The dataset is publicly available at: `roboflow.com/robocops/duplo-merged-v3/2`, where reproduction instructions are also provided. The model has been simplified to a single classification label: **Duplo**. Obstacle detection was deliberately excluded from the network, as the LiDAR sensor is deemed sufficient for this purpose. Thanks to the onboard inference capabilities of the Oak-D Lite, all neural network processing is offloaded from the Raspberry Pi. This allows the Raspberry Pi to focus solely on navigation and localization tasks.
An example of the model's detection output is shown in Figure 63.

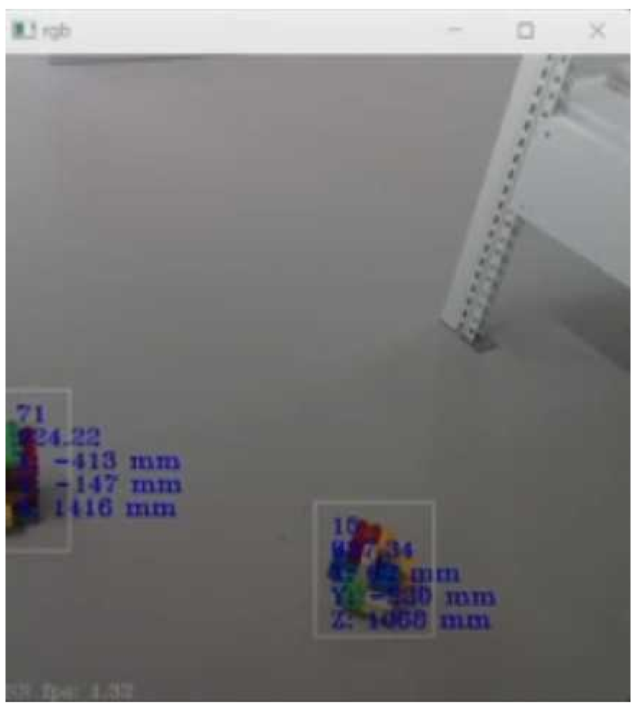



Figure 63: Example of detection done by our model

## 5.3 Arduino Code

The Arduino firmware serves as the hardware interface between the Raspberry Pi and the robot's actuators and sensors. It implements a binary serial communication protocol designed for efficiency and reliability. We implemented different routines that are used for the unloading, the button opening, the on-slope movements and the lift activation.

### Command Byte Structure

The command message is composed of the following bytes:

- **Bytes [0–1]**: 16-bit signed integer representing `command_maxon_left` speed, offset by +10000.
- **Bytes [2–3]**: 16-bit signed integer representing `command_maxon_right` speed, offset by +10000.
- **Byte [4]**: Bitfield containing several command flags:
  - Bit 0: `command_capture`
  - Bit 1: `command_unload`
  - Bit 2: `command_button`
  - Bit 3: `command_slope_up`
  - Bit 4: `command_slope_down`
  - Bit 5: `command_emergency`

### State Byte Structure

The state message is composed of the following bytes:

- **Bytes [0–1]**: 16-bit signed integer representing `state_maxon_left`, offset by +10000 from the encoder value.
- **Bytes [2–3]**: 16-bit signed integer representing `state_maxon_right`.
- **Byte [4]**: Bitfield indicating the current active routines and statuses:
  - Bit 0: capture active
  - Bit 1: unload active
  - Bit 2: button routine active
  - Bit 3: slope up routine active
  - Bit 4: slope down routine active
  - Bit 5: emergency active
- **Bytes [5–6]**: 16-bit unsigned integer indicating the number of Duplos captured.
- **Bytes [7–8]**: 16-bit integer representing the distance measured by the back ultrasound sensor.
- **Bytes [9–10]**: 16-bit integer for `state_other`, reserved for future use.

### Code Structure

The following files make up the Arduino-side implementation of the RoboCops project:

- `robocops_arduino.ino`: Contains the main control loop and manages serial communication with the Raspberry Pi.
- `maxon_driver.*`: Provides control logic for Maxon motors, including PWM generation and direction setting.
- `maxon_encoder.*`: Handles encoder readings and estimates wheel speed.
- `servo_driver.*`: Controls the angle of the servomotor used for object manipulation or storage.
- `l298n_driver.*`: Interfaces with the L298N H-bridge to control brushed DC motors.
- `dri_driver.*`: Manages motor control using the DRI motor driver module.

- lift_driver.*: Implements all the interfaces used to control the lift.
- ir_sensor.*: Reads data from IR sensors and performs signal averaging for line-following functionality.
- back_ultrasound_sensor.*: Measures distance behind the robot, primarily used in button routines.
- routines.*: Encapsulates higher-level robot behaviors such as capturing objects, handling slopes, and button interactions.

## 5.4 Raspberry Pi 5

The Raspberry Pi 5 (8GB RAM) is the central computing unit of the robot, responsible for high-level decision-making and Nav2 navigation. All ROS 2 packages, algorithms, and behavior logic are executed here except for the neural network computation.

#### System Architecture and Package Layout

The project follows a modular ROS 2 (Jazzy) architecture, composed of the following main packages:

- robocops_camera: The robocops_camera package is used to process the camera detections and publish the position of the duplo in the arena using the localization. It uses internal processes to verify the published camera detections and then publishes the official detections on a topic.
- robocops_gazebo: The robocops_gazebo package provides Gazebo simulation support for the RoboCops robotics system. It launches the simulation environment, spawns the robot at a configurable initial pose, and bridges ROS 2 and Gazebo topics using ros_gz_bridge.
- robocops_teleop: This package provides keyboard-based teleoperation functionality for controlling the RoboCops robot via ROS 2 topics.
- robocops_control: The robocops_control package provides control interfaces for the RoboCops robot, specifically for controlling the robot's wheels through ROS 2 control systems. It interfaces with the robot hardware and provides control commands for wheel speed and motion.
- robocops_brain: This package handles high-level decision-making and robot task coordination on a behaviortree.cpp base.
- robocops_msgs: The robocops_msgs package defines custom message types used across the RoboCops ROS 2 system. These messages facilitate communication between perception, decision-making, and control components of the robot.
- robocops_description: The robocops_description package contains the full robot description for the RoboCops robot, including its physical structure, sensors, and control interfaces. This package plays a critical role in both simulation and real-world deployment by providing a centralized source of truth for the robot's configuration.
- robocops_navigation: The robocops_navigation package is responsible for enabling autonomous navigation capabilities on the RoboCops robot. It leverages the ROS 2 Navigation Stack (Nav2) to plan paths, localize within a map, and avoid obstacles while navigating in the arena.

#### Behavior Tree Execution

Robot behavior is orchestrated using a custom Behavior Tree implemented with the BehaviorTree.CPP library. Key features include:

- **Modular Node Structure**, with nodes handling:
  - Navigation goals
  - Detection results
  - Lift state monitoring
  - Command triggering (e.g., unloading, button pressing)

- **Failure Handling**: Fallback nodes manage action failures.
  - Example: If the ultrasonic sensor detects the door is still closed, the tree retries the button press.
- **ROS 2 Integration**: Communication between the behavior tree and other modules uses ROS 2 topics, services, and actions, promoting modularity and robustness.

### YOLO Inference Integration

While inference is performed directly on the OAK-D Lite camera, the Raspberry Pi handles additional processing through the following steps:

- **Subscribes** to a ROS 2 topic published by the OAK-D Lite camera, which provides onboard inference outputs.
- **Filters** out false positives by:
  - Requiring multiple consecutive detections for confirmation.
  - Clustering detection centers to verify consistency.

  (For more details, refer to the `robocops_camera` package.)
- **Sends** the position of detected Duplos to the navigation module and behavior tree, which uses the information for high-level decision-making (e.g., triggering a "capture" action).

# 6 Management

## 6.1 Time Management: Gantt Chart and Progression During the Semester

### 6.1.1 General Structure

At the beginning of the semester and after our first Functional Analysis, we made a Gannt Chart to be able to follow our progression trhoughout the semester and have some Time Management. Our Gannt Chart included important dates and Milestones, different phases from the Design to the Final Testings but also a large number of tasks, including the percentage of completion, the dates and number of days of each task. We will present to you the Gannt Chart of week 4, when we got our Milestone with our coach, to analyse how well did we manage to follow the Gannt Chart and what was not predicted at that point in time. In figure 64, you can notice a complete vision of the Gannt Chart, with different colors representing the different phases, considering the Conception, the First Testings, the Assembling Phase and the Final Testing. We also included important dates to localize ourself, such as the opening of the Arena, the Design Review Midterm, or the date of the Internal Testing Competition. In red on the top, you can notice the Competition day and week. In orange, you can see the dates we are supposed to work on different tasks of the First Testings phase. Those are only two examples, but you can see that we thought about a large number of tasks that were to be done.

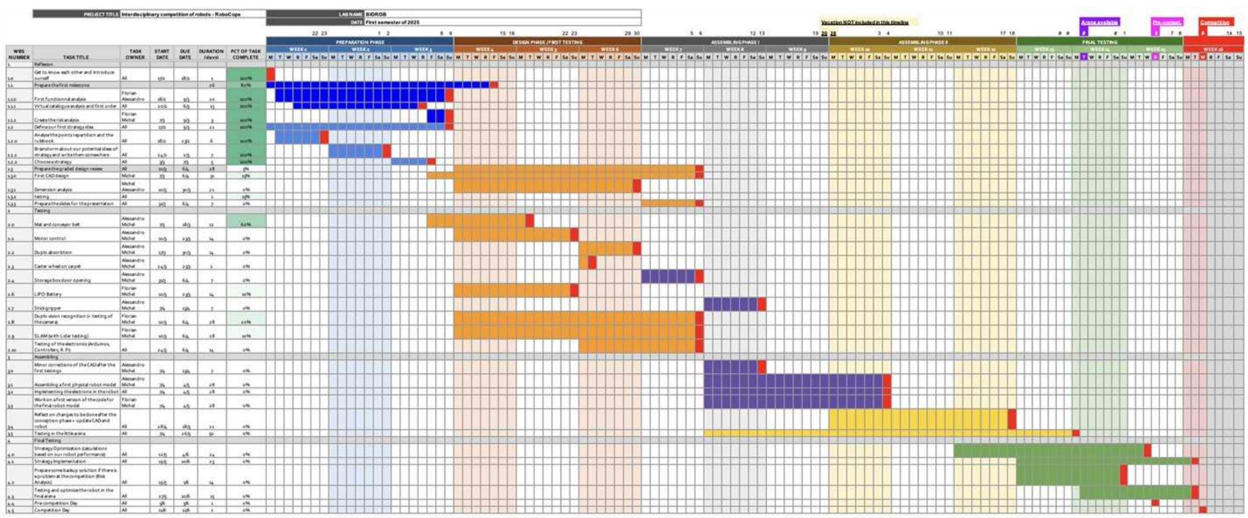


Figure 64: Overview of our Gannt Chart at the End of Week 4

We will now develop a little bit those phases to compare our Timeline to what happened during the semester. Let's use the first weeks as an example of the structure (figure 65).

| WBS NUMBER | TASK TITLE | TASK OWNER | START DATE | DUE DATE | DURATION (days) | PCT OF TASK COMPLETE |
|---|---|---|---|---|---|---|
| 1 | Reflexion | | | | | |
| 1.0 | Get to know each other and introduce ourself | All | 17/2 | 18/2 | 1 | 100% |
| 1.1 | Prepare the first milestone | | | | 26 | 80% |
| 1.1.0 | First functionnal analysis | Florian Alessandro | 18/2 | 9/3 | 20 | 100% |
| 1.1.1 | Virtual catalogue analysis and first order | All | 20/2 | 6/3 | 15 | 100% |
| 1.1.2 | Create the risk analysis | Florian Michel | 7/3 | 9/3 | 3 | 100% |
| 1.2 | Define our first strategy idea | All | 17/2 | 9/3 | 21 | 100% |
| 1.2.0 | Analyse the points repartition and the rulebook | All | 18/2 | 23/2 | 6 | 100% |
| 1.2.1 | Brainstorm about our potential ideas of strategy and write them somewhere | All | 24/2 | 2/3 | 7 | 100% |
| 1.2.2 | Choose a strategy | All | 3/3 | 7/3 | 5 | 100% |
| 1.3 | Prepare the graded design review | All | 10/3 | 6/4 | 28 | 5% |
| 1.3.0 | First CAD design | Michel | 7/3 | 6/4 | 31 | 15% |
| 1.3.1 | Dimension analysis | Michel Alessandro | 10/3 | 30/3 | 21 | 0% |
| 1.3.2 | Purchasing new needed items for testing | All | | | 1 | 15% |
| 1.3.3 | Prepare the slides for the presentation | All | 31/3 | 6/4 | 7 | 0% |

Figure 65: Tasks and Information of the First Weeks of the Semester

During the First Weeks, we were doing the components analysis, but also we were planning to work on the first dimensions of the Robot. We can see which task is due for which date but also the member its assigned to, with a percentage of completion and the duration. This process allows us to have clear ideas of what is to be done next. In fact, we will see in the next phases that even if we weren't able to follow strictly our Gannt Chart and that we have been in difficult moments during some phases, this Gannt Chart gives us a great overview of the steps that we needed to follow.

### 6.1.2 Testing Phase

This phase of the project was really important. We needed to see how our first Ideas will work, but also think about potential changes in the overall Design of our Robot. Globally, during this phase, the objective was on one side to test different Mechanical system, such as a Conveyor Belt Mechanism to use as a ramp to bring up Duplos and to release them, testing the Brushes and think of the best way to implement them such that the Duplos finish on the Conveyor, and more. On the other side, we were testing our Electrical and Software components: LIDAR, training of a YOLO Camera model, First Implementation of the MAXON Motors to control a first version of the Robot. You can see the Gannt Chart tasks defined at week 4 in figure 66.

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | Testing | | | | | |
| 2.0 | Mat and conveyor belt | Alessandro Michel | 7/3 | 18/3 | 12 | 60% |
| 2.1 | Motor control | Alessandro Michel | 10/3 | 23/3 | 14 | 0% |
| 2.2 | Duplo absorbtion | Alessandro Michel | 17/3 | 30/3 | 14 | 0% |
| 2.3 | Caster wheel on carpet | Alessandro Michel | 24/3 | 25/3 | 2 | 0% |
| 2.4 | Storage box door opening | Alessandro Michel | 31/3 | 6/4 | 7 | 0% |
| 2.6 | LIPO Battery | Florian Michel | 10/3 | 23/3 | 14 | 10% |
| 2.7 | Stick gripper | Alessandro Michel | 7/4 | 13/4 | 7 | 0% |
| 2.8 | Duplo vision recognition (+ testing of the camera) | Florian Michel | 10/3 | 6/4 | 28 | 20% |
| 2.9 | SLAM (with Lidar testing) | Florian Michel | 10/3 | 6/4 | 28 | 10% |
| 2.10 | Testing of the electronics (Arduinos, Controllers, R. Pi) | All | 24/3 | 6/4 | 14 | 0% |

Figure 66: Tasks and Information of the Testing Phase of our Gannt Chart (Week 4)

This phase was supposed to last until week 7, but finally, as we have a pretty complex Robot in a Mechanical perspective, creating all those modules and eventually finding new modules to compensate for the one that weren't reliable and robust expanded this phase to week 11.

The main problems were:

- The conveyor system we designed was really great and the "Dycem" we found was a great material. However, use a ramp to bring Duplos up on another conveyor was not only not robust enough, it was also implying a way too large Robot. This is why we needed to implement a Lift Mechanism (Section **3.3**), which expanded by 4-5 weeks the Testing Phase, as it is a pretty complicated module to Design. However, we don't regret taking our time to design it, because it is compact and robust. In figures 67, 68 and 69, you can notice the main problems that we encountered with the Conveyor prototype.

Figure 67: Test of Ramp Conveyor System on Week 4

Figure 68: Fail With a Steep Angle of 30 Degrees part I

Figure 69: Fail With a Steep Angle of 30 Degrees part II

- The fact to have a large Robot that couldn't be manufactured fast implied that we needed to think about a way to test in some conditions a first version of our Navigation code and simulate the LEGO Detection. To simulate this, we made two MDF prototype: one in week 5, with a Camera Support, Two Motorized Wheels and a Caster Wheel; and the other one in week 9-10, bigger, to simulate the Final Dimensions and test the ramp of Zone 4 with different Motorized Wheels and Passive Wheels distances. We also made a first CAD version of the Robot by week 10-11 to be able to test the Robot with RViz and Gazebo on simulation.

Figure 70: Prototype of Week 5, with Camera Stand

Figure 71: Camera Testing in the Little Arena with First Prototype

Figure 72: Prototype of Week 10, to test the Wheel Distances and the Passive Wheels, as well as Slope Climbing (I)

Figure 73: Prototype of Week 10, to test the Wheel Distances and the Passive Wheels, as well as Slope Climbing (II)

- We were really dependent of the opening hours of the SKIL and the SPOT. Some of the facilities, such as the Electrical Prototyping Place (where we can buy cables and electrical components) or the Mechanical Prototyping Place (where we can Laser Cut, buy Screws, Machining...), are opened only during the week from 8 to 20, and there are some difficulties to find places or to access to the machine without waiting a lot of time (with the large number of MAKE project). The great think was that we were able to 3D print pretty much everything when we wanted to, but also to access to some of the SPOT facilities at pretty much all time.
- With all the Electronic Modules to implement, the Electrical Schematic and Implementation was crucial and definitely took some time (Section **4.1**).

Finally, we were able to test all our module by the end of week 11. At this point in time, we fixed a goal for the team: have a first assembled model of the Robot for the opening of the Arena at the latest. In fact, the next phase (Assembling Phase) that was supposed to take 4 to 5 weeks took in reality only 2 weeks. We will see that we weren't able to test in the Little Arena as predicted, but we were determined with the Gazebo simulation and a final Robot assembled to test our Robot from the 27 of May directly in the Final Arena.

#### 6.1.3 Assembling Phase

For the Assembling Phase, we worked on all the last steps to have a final CAD: modification of some Mechanical Parts to be able to fit a Robot to the great dimension (especially for the Robot to move up the ramp of Zone 4), Implementation of the Entrance Slope on the Robot to be able to combine it with the Brushes and send LEGOs up on the Lift directly (Section **3.6.2**), creation of the Electronic Box (Section **3.5**), and the first Overall Design. During this period, we were also working on the Navigation and a first version of the Behavior Tree (Section **5.5**) on the Software side of things. We manufactured a lot in few days to meet our deadline of having a Robot assembled for the 27 of May. We had a first version assembled at this date, but we needed to remanufacture it because we were stuck into the slope of Zone 4. For this, we move the Storage Area Conveyor System up by a bit, to be able to cut a big part of the back of our Robot (by 2 cm). This change has not influenced that much our Storage Space, and made our Robot able to climb the ramp of Zone 4. You can notice those changes in figures 74, 75 and 76.

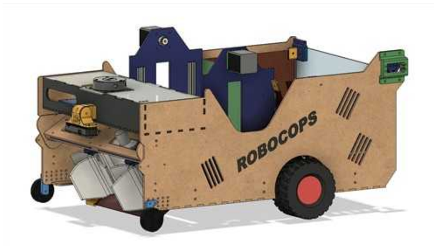



Figure 74: First CAD Version of our Robot, tested the 27 May in assembled version: was not able to climb the Ramp of Zone 4

Figure 75: First Version of our Robot, tested the 27 May: was not able to climb the Ramp of Zone 4


Figure 76: Final CAD version: we were able to climb on Zone 4 with the changes made to the back of the Robot

#### 6.1.4 Final Testings

This phase corresponds to the last two weeks before the Competition, when we needed to prepare both for the Internal Competition and the Final Competition. We were extensively testing the Robot, while implementing great Arduino Routines (Section **5.4**), Navigation (Section **5.5**), and LEGO Detection (Section **5.3**). We were also making replacement parts or Mechanical tweaks to our Robot, to compensate for the newly discovered problems, such as:

- Support for the Storage Area Plexiglass Separation between the Lift and the Conveyor to compensate if little LEGOs get stuck in this Area (Section **3.6.1**)
- IR Sensors Implementation (Section **3.3**) to verify when a LEGO structure entered in the Lift. We first wanted to use an Ultrasonic Sensor, but it wasn't robust enough.
- Use of an Ultrasonic Sensor to analyze if the door of Zone 3 is opened, with the Stand previously placed next to the Electronic Box, when we expected to use the Ultrasonic Sensor for the Lift.
- Mechanical Parts to guide the LEGO in the Brushes Direction (Section **3.6.2**)
- Elastic under the Entrance Slope of the Brushes to prevent the Slope to turn around inside of the Robot (especially when going in Zone 3 little ramp and in Zone 4 ramp, section **3.6.2**)
- Modification of the Entrance Slope (Section **3.6.2**) and Back Door (Section **3.4.2**) side size (- 1 cm) due to the newly implemented design
- Prepare Replacement 3D Printed or Laser Cutted Parts (PET 1 mm Entrance Slope, Passive Wheels Support, Electrical Cables...) to be able to manage Risks

The fact that our Robot was pretty much ready 2 weeks before the Competition allows us to implement those type of fine tuning and to really take time to test extensively our Software. However, as we will discuss more in section **7**, we weren't able to implement the routine of Zone 4 perfectly due to time constraints.

## 6.2 Budget Management

For the budget, we were allowed to either take some parts from BioRob in a "Virtual Budget", or to buy at local shop/online with EPFL in a "Real Budget". We are also allowed to buy screws, to Laser Cut and buy MDF, PET, Acrylic and to 3D Print in PETG at the SPOT/SKIL facilities.

### 6.2.1 Virtual Budget

For the Virtual Budget, we can use up to 1 500 CHF. However, if everything happens to a part that we were using, it goes in the Real Budget. In fact, we need to keep some margin in the Real Budget in case of problems to mitigate Risks. We used **1 308.25 CHF** in the Virtual Budget, which makes our Robot already pretty expensive. In fact, the MAXON Motors with Controller ESCON 24/2 and the MM8 Adapter are each worth 182.1 CHF, which for two of them is worth 364.2 CHF. The two passive Omni Wheels of 50 mm are worth 72 CHF. The LiDAR has a price of 100 CHF and the Camera OAK D Lite of 150 CHF. Considering all the other components, our Robot is already pretty expensive, but we managed to keep 191.75 CHF in the Virtual Budget balance in case of problems.

### 6.2.2 Real Budget

For the Real Budget, we have 750 CHF that we can use. We have bought:

- Strack, 3DWare, Digi-Key and Mouser online purchases (such as Lift Belts, "Dycem" Material): **155.27 CHF**
- OBI: **28.4 CHF** (Testing Cables, Materials for Mat, L-Brackets)
- 3D Printing at SPOT/SKIL: **174.06 CHF**. As we have a lot of modules and we needed to test a lot of them, this section is from where come our main expenses. The TPU Brushes were especially expensive, for 35.9 CHF (For testing the infill and make the final versions of the Brushes)

This brings up a total of **357.73 CHF**. However, we have not been able to take note in a great way through the semester a lot of purchases we made at the Mechanical Workshops of the SKIL or the SPOT. What we weren't able to count by ourselves are: MDF, PET and Acrylic plates, screws and nuts, IGUS Rails and MAKERBEAMS steel bars for the Storage Space Conveyor System and some testing couplers materials. However, with the budget left in the Real Budget, we were pretty sure it was in the 357.73 CHF left with some margin to mitigate Risks if a Virtual Budget part broke.

# 7 Performances and Discussions

## 7.1 Final Results

During the Competition, our Robot was able to go autonomously in Zone 3, to grab Duplos but also to release them as expected. He made some researches in Zone 1 too, which was pretty great. However, the routine for the Robot to climb in Zone 4 was not fully implemented, and we weren't able to exploit the full potential of our Robot. We needed to switch to "Demo Mode", not autonomous, to show how the Robot would have gone in Zone 4. **We made 150 points at the Competition, to finish at the 3rd place.** Even if we weren't able to collect Duplos in Zone 4, we are really happy to have been able to implement a great software in a such complex Robot. It was a really fun experience. We know that our Robot would have got the ability to have even more points if we implemented in a better way the routine to go on Zone 4, but we are heavily satisfied with our performance.

## 7.2 Possible Improvements

Our Robot was really great in a Mechanical perspective, however, it is pretty big. In fact, we can store a large number of Duplos, but maybe a little bit too much for our Strategy and the tasks to be done. A smaller Robot would have implied an easier Navigation and Localization to develop, and we could have gone to Zone 4 more easily. For the rest of the Robot and with the time we had for working on it and for the Competition, we are really satisfied of our work. Furthermore, we should have take into account all little purchases at the SKIL and SPOT, as there is no website to visualize the purchases for the moment (except for 3D printing).

## 7.3 Advices for Future Teams

For the future teams, we have some advices from our experience this year:

- Test the ideas as soon as possible in the semester, and always think of ways of making things compact. Furthermore, try to have a Robot ready as soon as possible, or at least to have all the modules ready separately (with Arduino routines), as we have done with our Lift, Conveyors and Brushes Mechanisms.
- There are only 2 weeks available to test the Robot in the Arena. In fact, we weren't able to implement the Zone 4 routine correctly due to time constraints. Having a Robot model ready to test on the day the Arena opens is the way to go, as you would be able to directly test the Robot. During the last days, it will be more difficult to test the Robot full-time, as other groups will be there too.
- Always think about what is needed to work on the project during Week-Ends or Public Holidays. Laser Cutting at SKIL/SPOT, screws and coaches are not available at those times. Furthermore, it is sometimes difficult to Laser Cut or to do some Machining at the SPOT, as a lot of people are working on other MAKEs project. In fact, it is good to take in mind that the Mechanical Processes may be delayed, and to have some time margin to compensate for those types of problems. It is also important to do those thinks as soon as possible, and to not wait even more.
- Be aware that buying parts online can also take some time, and that you can do only one purchase per website each 3 weeks.
- Use an IMU: the code will be a little bit more complicated, but it will help a lot for Localization.
- Navigation and Software implementation takes time: Simulation with RViz and Gazebo or prototypes are really important steps for good Software progression throughout the semester.

# 8 Acknowledgments

First and foremost, we would like to extend our sincere gratitude to our project supervisor, Alessandro Crespi. His availability and expertise have been crucial in helping us advance throughout the semester. Alessandro's advice and encouragement as well as his passion for the robotic field have been greatly appreciated and have significantly contributed to our progress.

We are also thankful to the Professor Auke Ijspeert and the BioRob lab who gave us the opportunity to work on such a motivating and innovative project. Professor Ijspeert's gives us important feedbacks both at the Midterm Presentation and at the Internal Competition, which helped us during the project.

Our thanks also go to our teaching assistant, Nicolas Furrer, which we were able to get feedbacks from once a week. It was really helpful, as he had already participated in the Competition and was able to give us precious advices.

We would also like to thanks the coaches of the SPOT and the SKIL for their availability and help throughout the semester. They were able to give us the great materials and to help us understand how all the machines works.

Finally, we also want to thanks all the other teams. Working next to them and compete against them was a great experience, especially during those last two weeks testing in the Arena. The teams were helping each others, and at the end of the day, it was more than a Competition!

Figure 77: Picture with all the Teams after the Competition


Figure 78: Final Robot in the Arena I

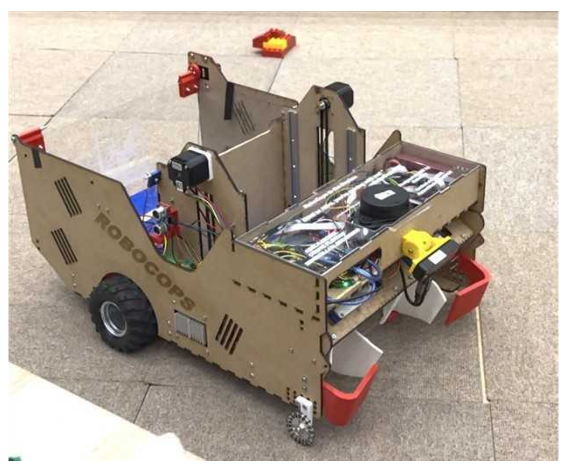



Figure 79: Final Robot in the Arena II