*Team Siegel*

# Team Description Materials

*Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina - Campus Criciúma, Brazil*

INSTITUTO
FEDERAL
Santa Catarina

Câmpus
Criciúma

**Team: Siegel**

**Team e-mail:** sigiel.ifsc@gmail.com

**Team mentors:**

Guilherme Amorim Schmidt (Mechatronics Engineering Masters)
guilhermeamorimschmidt@gmail.com
Lucas Artur Dutra Junior (Mechatronics Engineering Masters)

Thiago Viebrantz Ferreira (Mechatronics Engineering 2th semester)

André Soratto (Mechatronics Engineering 2th semester)
soratto.andre@gmail.com

**Team members:**

Technical high-school level (Integrated Mechatronics 2nd year):

Felipe Elton Pazini Savi - felipe.elton18@hotmail.com

Flavio Maccari Frasson - flaviofrasson@outlook.com

João Henrique Aléssio - joaohenriquealessio@gmail.com

Kauan Biring Fontanela - kauanfontanela09@gmail.com

Ludmila Silveira dos Santos - ludvieira11@hotmail.com

Pedro da Silva Cordeiro - pedrodscordeiro@outlook.com

Technical high-school level (Integrated Mechatronics 3rd year):

Elias Mendonça Filho - eliasmenfilho@gmail.com
Arturo Brando Moreno Burigo - arturoburigo@hotmaill.com

## 1.1 Introduction:

Creating a robot is a real challenge, it requires a team with an extensive knowledge in coding, mechanics and electronics. Our 2020 team is composed by high school students and recent graduates, which are ready to learn and work with each step of a robotic project planning and constructing to have the opportunity to participate of RMRC 2020.

The goal for this year competition is improving what we have learned in our previous three experiences on creating rescue robots that are able to pass through complex terrains and rescue victims in circumstances where people can't.

As we have new members this year, we are going to work with features we already have, developing and adding new ideas to upgrade the concepts of our robot, as well as we will be working on developing the solutions for tasks we didn't solve yet.

This year our inspiration for creating our robot's structure is the same from last year, Team TUPAC's 2018 robot. The main idea is to have four supplementary belts, all capable of being independently angled by a servomotor, along with the main crawler, which is fixed in relation to the rest of the body. The appeal of such a form factor lies in its high mobility, being capable of traversing inclined terrain and jagged edges with ease. What we modified from Team TUPAC's design is primarily the shifting some components to the top (except the motors that need to be on the bottom), besides changing all the components to ones we can buy, either in ISO, DIN or ANSI standard, instead of JIS.

## 1.1 The Team:

Siegel's history began in April 2016, with only two members, whose original goal was to take part in the Brazilian Olympics of Robotics (OBR). In 2017, the team wanted to participate in RoboCup, going to Nagoya's edition. A lot was learned with the competition, leading the team to apply for the next year, 2018. Siegel was again approved to take part in RoboCup, this time in Montréal. 2018 edition added some considerable knowledge, especially due the problems that were faced and the interaction established with teams from all over the world. The experience ended in a huge incentive to the team for 2019, to come with some brand-new ideas and motivation to make them happen in Sydney. Sydney was a hard year. The motors burned out in the first day of competition, and took hard work from the team to repair. But even with the obstacles, Siegel reached 3rd place in Robocup 2019. This year, Siegel has new integrants which are really engaged to make it in Bordeaux.

The 2020 team is composed by: Arturo Brando Moreno Burigo, Elias Mendonça Filho, Felipe Elton Savi Pazini, Flavio Maccari Frasson, João Henrique Aléssio, Kauan Biring Fontanela, Ludmila Silveira dos Santos, Pedro da Silva Cordeiro (high school level); Thiago Viebrantz Ferreira and André Soratto (graduation level). The mentors are: Guilherme Amorim Schmidt, Lucas Artur Dutra Junior, André Soratto and Thiago Viebrantz Ferreira.

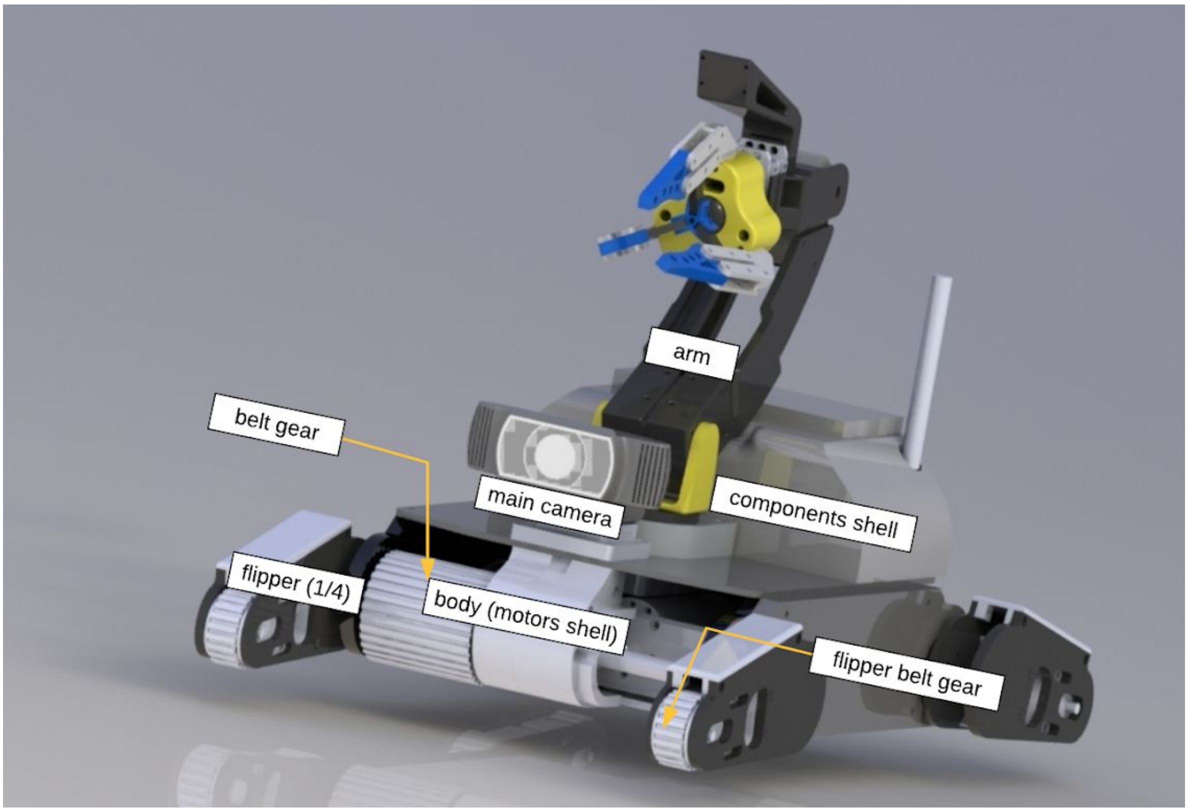
Siegel's objective consists in keep developing and learning, competing and overcoming our difficulties, always sharing the advances and solutions achieved with the community.

## 2. System Description:

The robot's structure is similar to that most common on Major, which got to Siegel's attention through its use by Team TUPAC in 2018. It basically consists of four "flippers", that is, secondary mobile belts, with the capability of being independently angled, along with the two main fixed belts. This should grant better mobility and power to go up almost vertical surfaces, such as stairs. **Figure 1** is the 3D rendering for the prototype, and **Figure 2** is the same render with the definition of the robots components.

**Figure 1**

**Figure 2**



## 2.1 Hardware:

### 2.1.1 Mechanics:

The robot is divided into a few sections, each responsible for different tasks. They shall each be explained in their own section.

The main materials are: aluminum, for all the power transmission, such as gears, shafts and axles; ABS, for the housing and supporting of the components; and polyurethane for the belts.

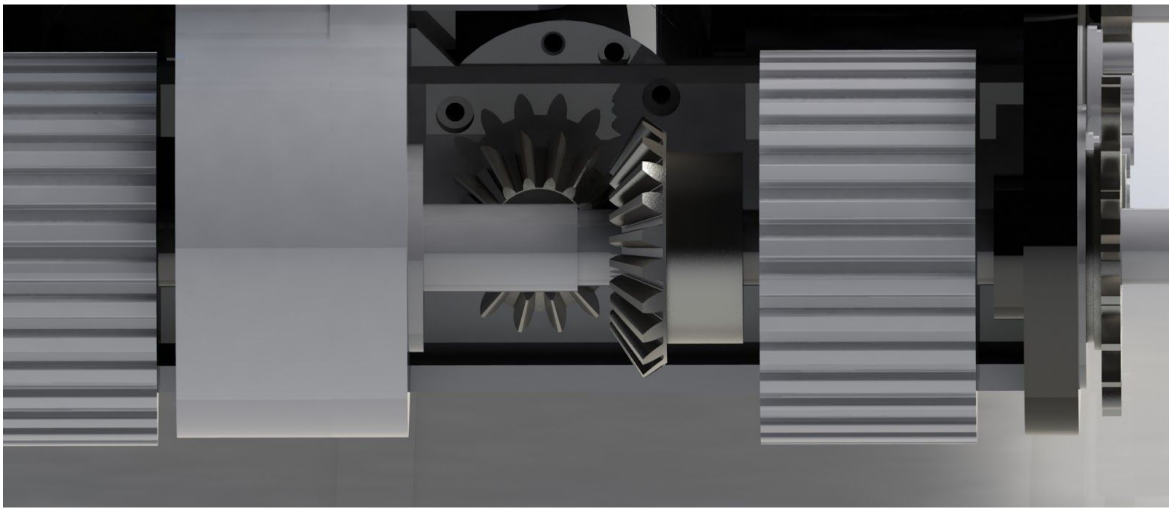
The design was done using SolidWorks2014

#### *2.1.1.1 Main crawlers:*

There are two fixed tank threads on the robot (in the sense they cannot move in relation to the rest, if they move, everything moves). These are called main crawlers and are responsible for the majority of the traction. They have approximately 15.795mm2 of ground contact area each.

The motion of the pulleys comes from two brushed DC motors, and is transmitted through a gearbox on the inside, with reduction ratio 8:11. (**Figure 3**).

**Figure 3**



#### *2.1.1.2 Flippers:*

There are also four mobile tank threads, one on each corner, to provide more complex motion which can be seen on **Figure 4** and **Figure 5**. Their belts spin synchronized with its side's main crawler's belt, as they are both on the same shaft. To angle the flipper, there are gearboxes in the outside panels of the robot (**Figure 6**). The reduction ratio is 3:5.

**Figure 4**

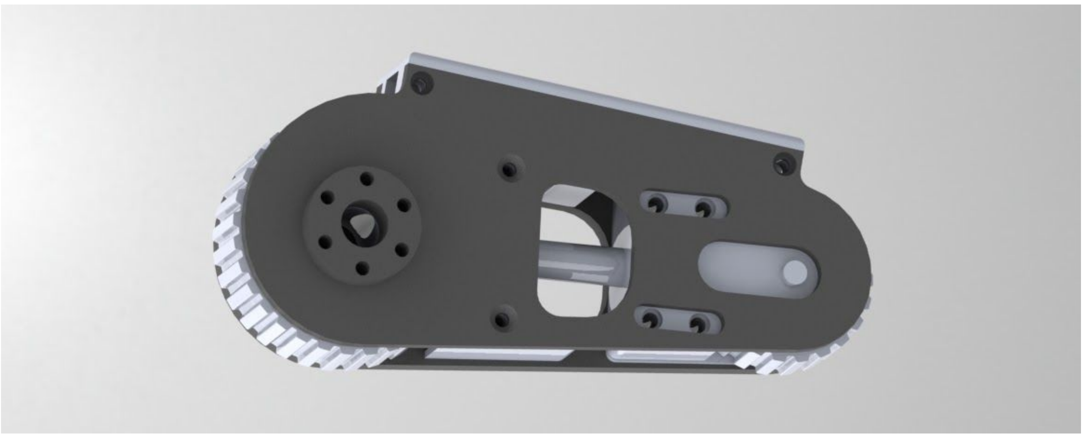


**Figure 5**

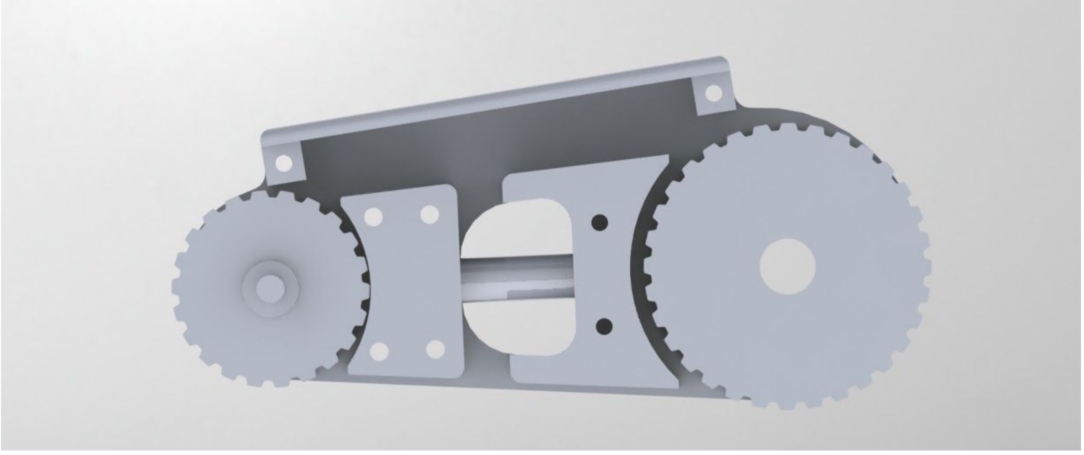
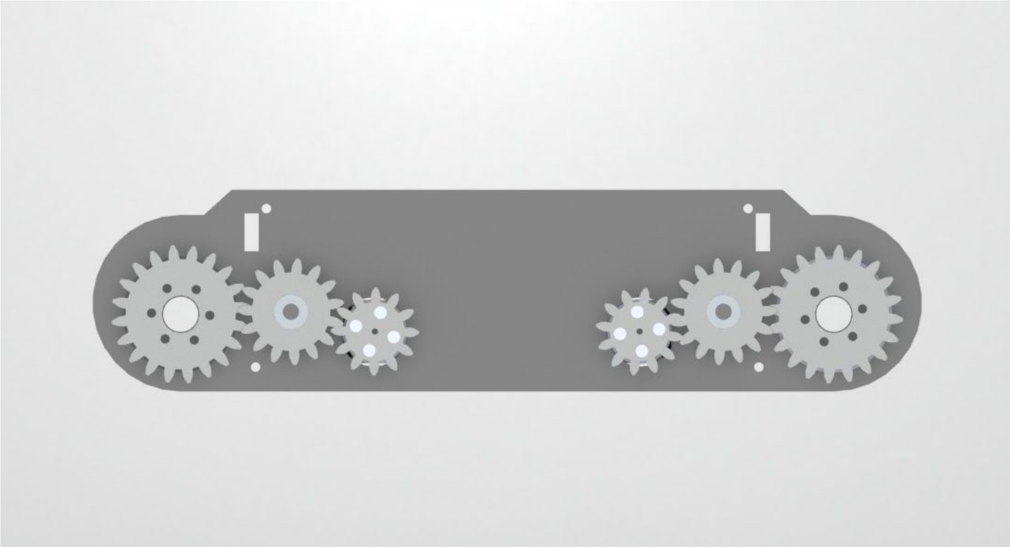


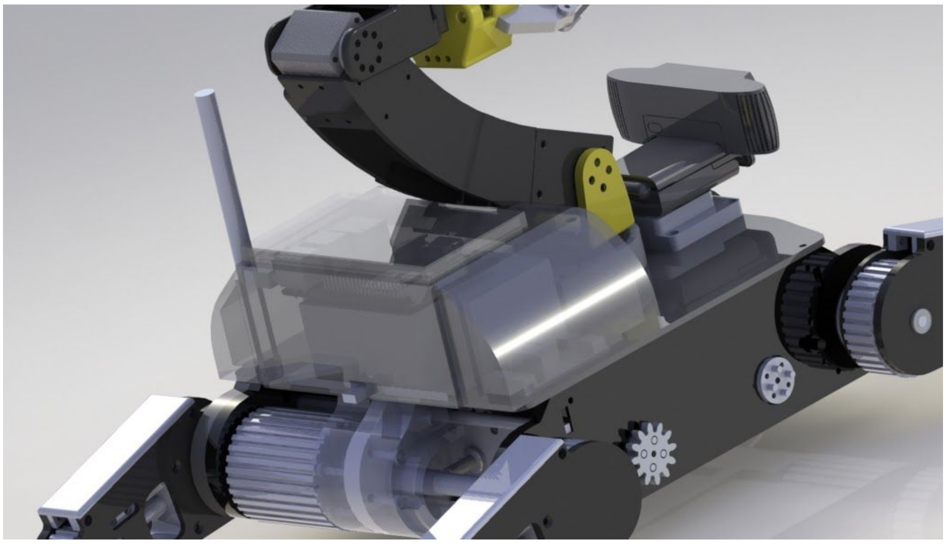
**Figure 6**



#### 2.1.1.2 Shell:

On top is a shell which contains the main components such as the main board and the communication antenna, a space to put components on. The batteries, the controller board, the sensors, and the arm are mounted here (**Figure 7**).

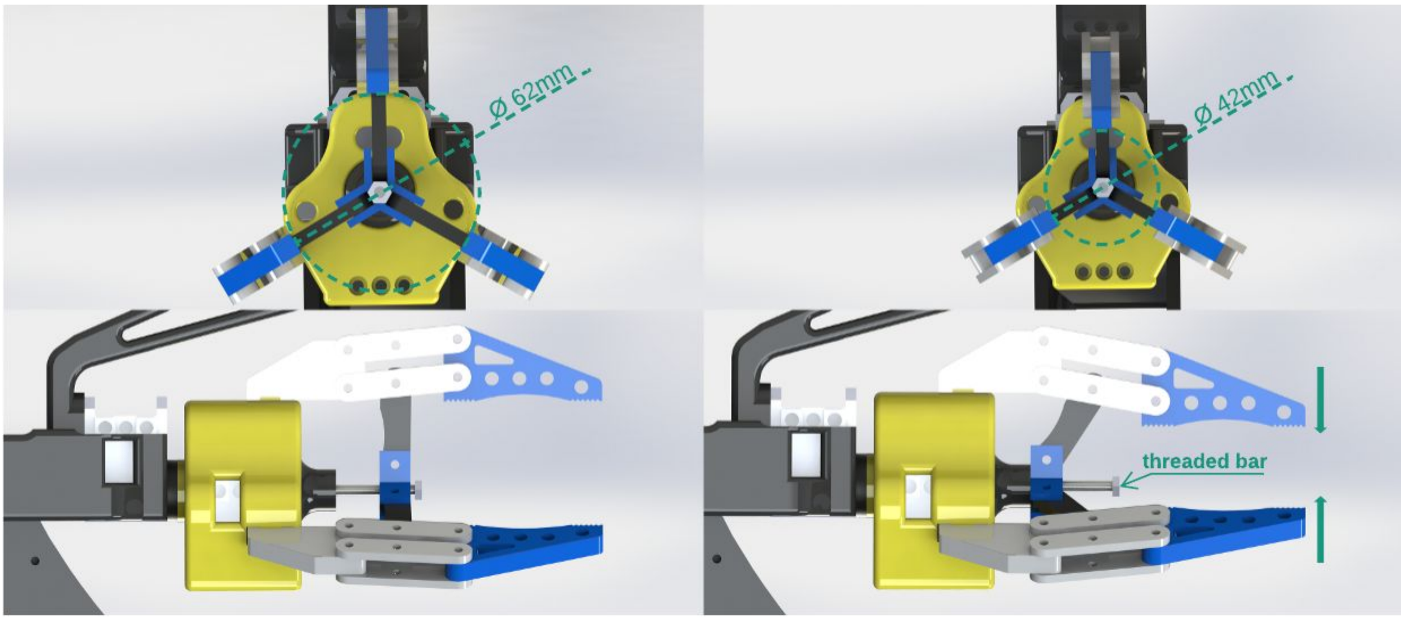
**Figure 7**



### 2.1.1.3 *Manipulator:*

The manipulator is a 3-DOF robot arm with a rotating wrist (**Figure 1** illustrates best). Once the main board is turned on, the claw is setted to closed position. After that, the claws state (open/closed) depends on the operator. The controller has two buttons for controlling the claws: one for opening and one for closing. While pressing the opening button a motor inside the arm's fist rotates a threaded bar clockwise, which is responsible for opening or closing the claw. Likewise while pressing the close button, the bar rotates counterclockwise, what closes the claw.

**Figure 8**




### 2.1.2 Electromechanics:

For the actuation, 3 types of motors were chosen. One is a simple brushed DC motor, with 3650rpm and 1380g.cm of stall torque, of which there are two. These are used for rotating belts. Next, there are 4 servo motors SPT5325LV-360 to control each flipper angle. For the robot's arm we have 4 dynamixel high precision motors: three XL430 and two XL320.

### 2.1.3 Electronics:

#### *2.1.3.1 Components:*

As power supply, three LiPo batteries. Two of them with 11.1V and 2200mAh capacity, which are responsible for the main motors alimentation. And one of them with 7.4V and 1800mAh capacity for the other components.

As for sensing, there are 1 gyroscope/accelerometer sensor, the MPU-6050 and 1 analog gas sensor (MQ-135), coupled with a 16-bit analog to digital converter, the ADS1115. The monitoring will be done with two Logitech C920 Pro usb cameras, one on the front and one on the back, plus one on the arm, a raspberry pi camera module.

#### *2.1.3.2 Controls:*

The single board computer the team chose is the same as last year, the Asus Tinker Board. The choice was made based on its higher performance when compared to the Raspberry Pi 3 used by the team in RMRC 2018. An added bonus was its compatibility with Raspberry Pi hardware, as both boards share the same form factor.

The Asus Tinker Board is cheaper compared to some other single board computers on the market and fits the team's needs. It has the same number (40) of GPIO connections as a Raspberry Pi 3 and contains double the RAM, with higher bandwidth networking (1Gb instead of 100Mb), which isn't crippled by being shared with the USB 2.0 bus, like on the Raspberry Pi 3.

In 2018, the team worked with a Raspberry Pi 3. There were serious problems before and during the competition that significantly altered our final results, to worst. Because of these problems, Siegel decided to change the single board computer to a better option and came across Asus Tinker Board. As said before, the two single board computers have a remarkable resemblance in layout, size, and operation. Like the Raspberry PI, the Tinker Board runs an actual OS, giving it basic multitasking capabilities and all the tools of the Linux environment, with which the team is now familiar.

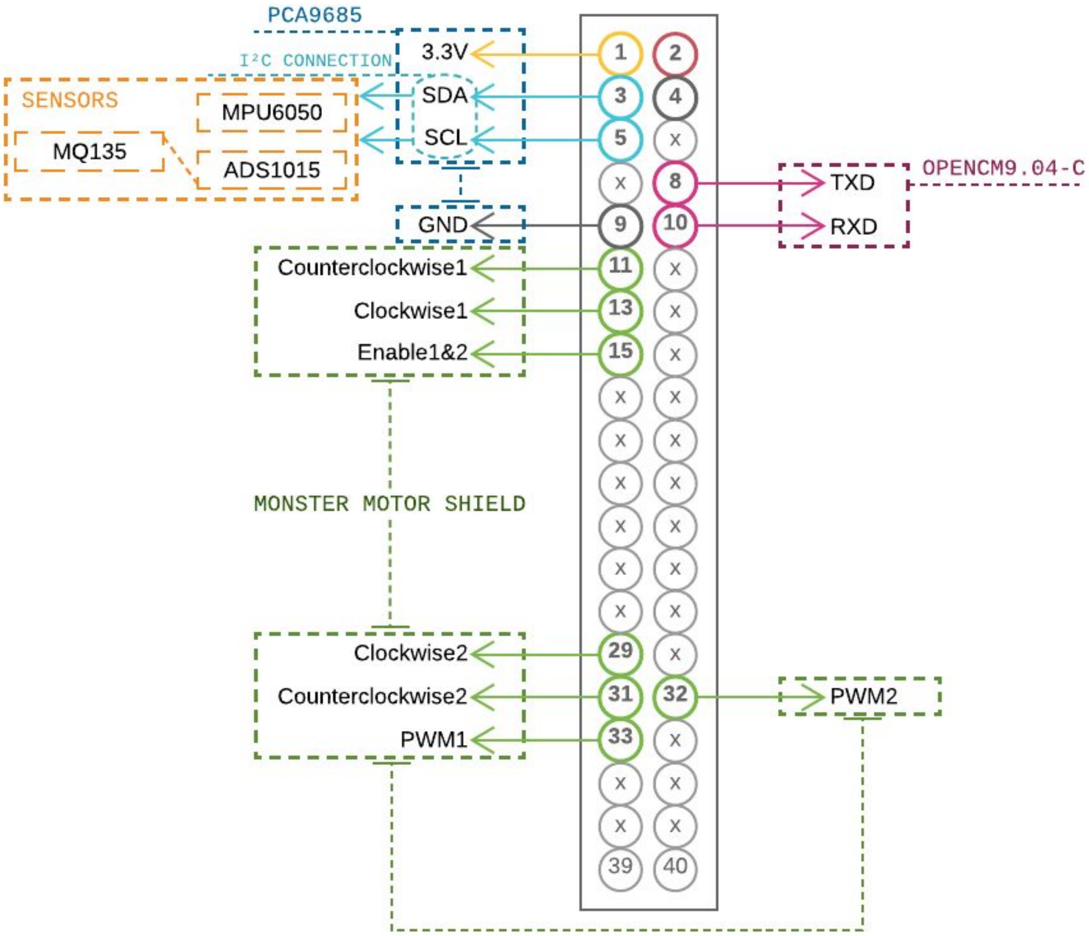
#### *2.1.3.2 Signal propagation:*

This year, as the components are all placed in a small shell, the team decided to wire the components manually. At this moment is being considered using a pcb connected directly into the main board pinout that would distribute the connections to better positions in order to avoid tangling the wires. But nothing decided yet.

##### *2.1.3.2.1 ASUS Tinker Board connections:*

**Figure 9** shows ASUS Tinker Board's GPIO connections as used by the robot.

**Figure 9**




### *2.1.3.2.1 Dynamixels' control:*

For controlling the dynamixels responsible for the arm's movement it's going to be used a OPENCM 9.04-C board, which is programmed for interpret bytes signal sent by the main board and control each one of the Dynamixels according to what it receives.

### *2.1.3.2.2 DC Motors' Control:*

For controlling the DC motors, H bridges with PWM will be used. Their function is providing bidirectional control of the motors and allowing for speed regulation. In this case, Motor monster shield is the board used (wired as **Figure 9**) to control the motors.

### *2.1.3.2.3 Sensors:*

All sensors use I²C, divided in two links, as the controller board has two I²C pin pairs. On one link, the four distance sensors will be connected, and on the other, the analog gas sensor with the AD converter, along with the gyroscope.

## 2.2 Software:

The main programming language used to control the robot is Python3 and C++, with Python used to the main program, such as PyQt and C++ used on OPENCM.

For choosing the development environment were considered the accessibility and the practicality on programming and, eventually, changing and improving the program according to the necessities, what took the team to choose the fresh Python3, which is pretty accessible, didactic and easy to work with.

### 2.2.1 Architeture:

There is a program that needs to be run on the operator's computer to spawn the other functions of the robot. That is the HMI program, whose job is, beyond just interfacing, organizing all processes and threads.

The program running in the operator's computer the one that send commands and receive data from the robot.

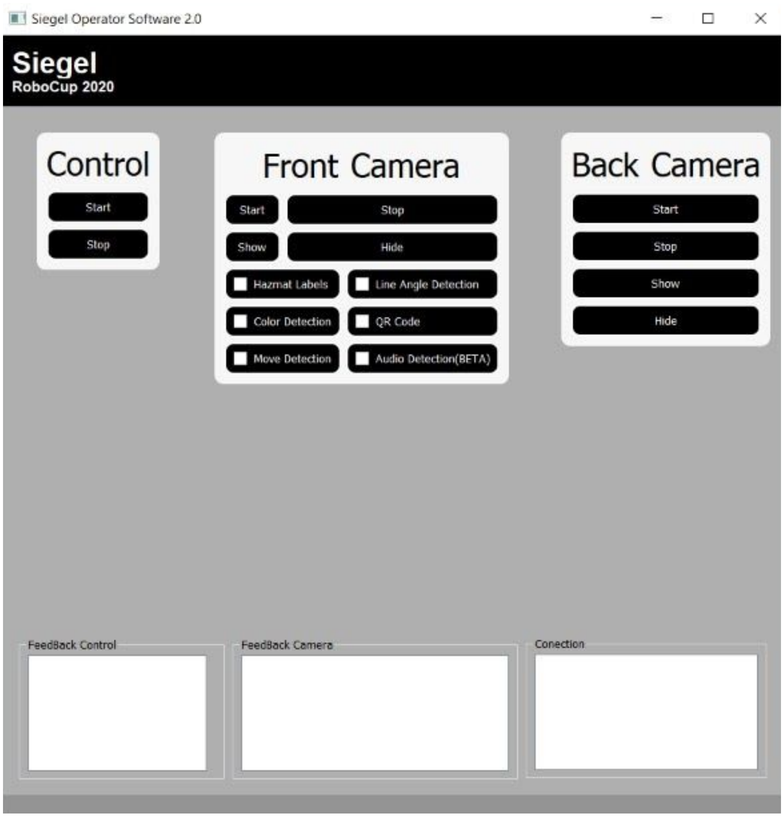
Only one process runs on the robot, a Pyhton program to interpret commands sent from the operator's computer and send sensor's and camera's feedback to the operator.

All the communication between operator's PC and the robot is wireless. The commands are all send via socket (python library for network interface) in byte format. The data transmission happens in different ports: one for each camera, one for commands sending, and one for each sensor.

### 2.2.2 Human Machine Interface (HMI):

On **Figure 10** a sketch of the HMI is shown.

**Figure 10**



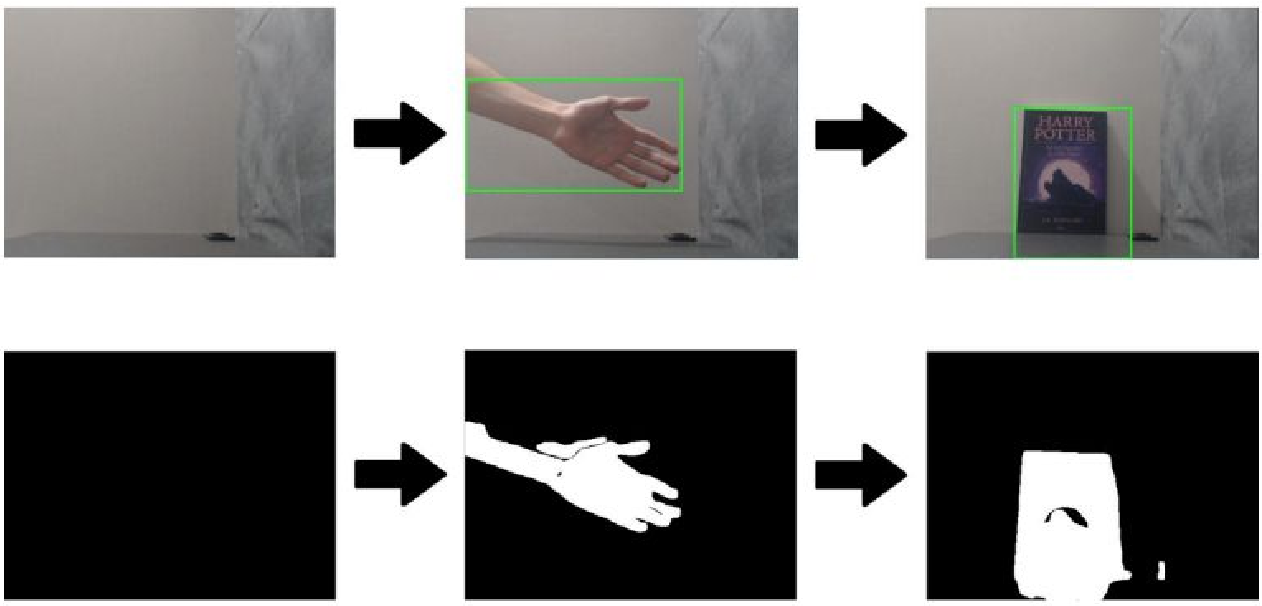

The HMI will function as follows: on the left side, we have the switches to start/stop the controller's command reading and it's feedback right below; in the middle we have the camera control panel, with switches to start/stop and show/hide, the check boxes can enable or disable the feedback for its function, which appears below the panel; and in right is the control panel for the back camera with its basic switches, and at the bottom is the connection status feedback.

### 2.2.3 Subprocesses:

#### *2.2.3.1 Motion detection:*

The movement detection will be done using OpenCV, which has software the team has already worked with to do this. **Figure 11** shows an example of one of the tests that have already been performed. When the camera is initialized, the first frames are used as a template with which the next frames that the camera captures are compared. From this comparison, any new objects that appear in the new frames are superimposed, as shown in the below images at line 2 and finally, marked in a raw video, shown in the images at line 1.

**Figure 11**



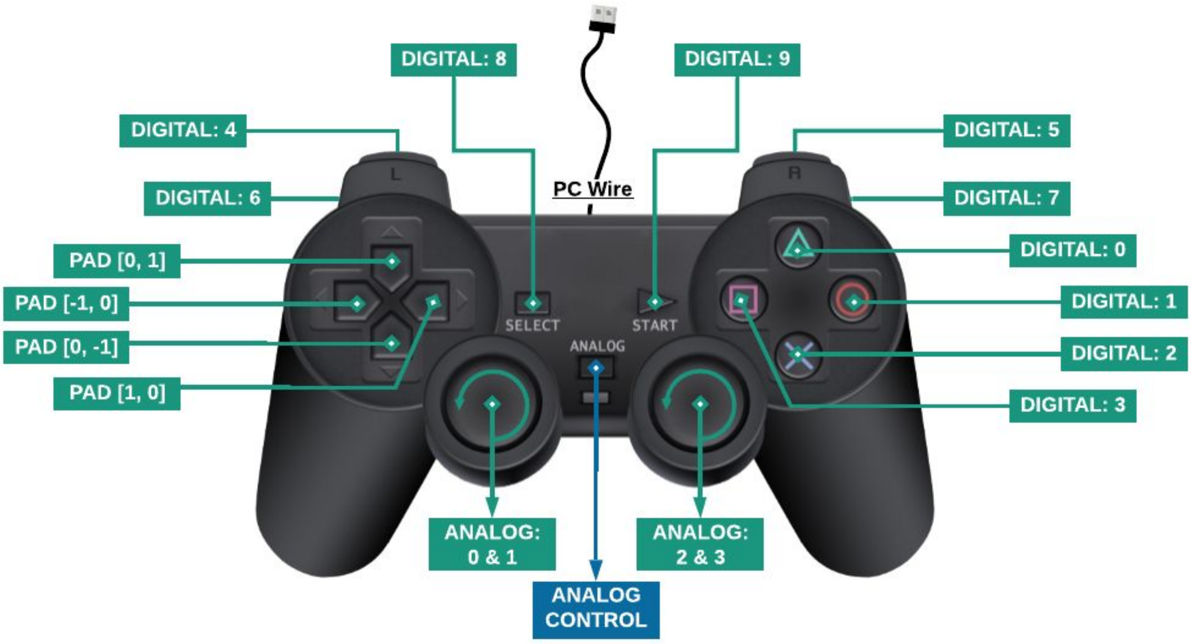

#### *2.2.3.2 Color detection:*

The Color Detection will use OpenCV, for actually detecting the colors, and for detecting the digits numbering the colors, OCR shall be used.

#### *2.2.3.2 Teleoperation:*

The Teleoperation program will be done with PyGame, python library that's able to read gamepad input. For the gamepad, a DualShock model controller will be used. The bumpers and triggers (R1, R2, L1 and L2) shall each select one of the four flippers and raise it on button press, unless the cross key is pressed, making the top buttons lower the flippers. The button mapping is exactly the same as if positioned on top of the robot, viewing from behind. All the controller's mapping is in **Figure 12.**

**Figure 12**




| | |
|---|---|
| DIGITAL: 0 | Open robot's hand |
| DIGITAL: 1 | Axis rotation |
| DIGITAL: 2 | Servo rotation inverter |
| DIGITAL: 3 | Close robot's hand |
| DIGITAL: 4 | (De)activate servo id:0 |
| DIGITAL: 5 | (De)activate servo id:1 |
| DIGITAL: 6 | (De)activate servo id:2 |
| DIGITAL: 7 | (De)activate servo id:3 |
| DIGITAL: 8 | ------------------------------ |
| DIGITAL: 9 | ------------------------------ |

| | |
|---|---|
| ANALOG: 0 | Distribute pwm (left-right) |
| ANALOG: 1 | Set pwm to motors 1&2 |
| ANALOG: 2 | Add analog value to hand's X axis position |
| ANALOG: 3 | Add analog value to hand's Y axis position |
| PAD [0, 1] | Arm Z axis UP |
| PAD [0, -1] | Arm Z axis DOWN |
| PAD [1, 0] | Clockwise rotate fist |
| PAD [-1, 0] | Counterclockwise rotate fist |
| Switch joysticks between ANALOG & DIGITAL signal | |

For the actual movement of the robot, a mathematical function will be used to translate from analog input to motor rotation, that will decompose the vector and distribute the Y coordinate according to the X one, so the robot will move in the general direction pointed to by the analog stick.

With regards to the arm's control, the left analog is responsible for moving the claw in X and Y position, while Z position is controlled by two D-Pad's arrows. The angles of each joint shall be calculated by the program.

### 2.3 Setup and strategy:

As the main goal for this year is improving what we already had last year and developing solutions to the abilities our past robot was not able to perform, pretty much of the setup is only an upgrade of last year's setup (**Figure 13**). For some improvements we are working with:

**Figure 13**

- Better material and shapes for the thank treads.
- Better wireless communication.
- Mechanical design upgrade.
- Concentration and protection of the electronic components.
- Better power stability.

The robot's layout and design (**Figure 1**) were influenced by Team TUPAC's 2018 model. It was an amazing approach to the challenges of the competition, which inspired Siegel. We also used outside knowledge to be able to build the robot with all of its functionalities.

## 3. System Description:

After reviewing the problems faced in previous editions and searching for new methods to be applied, the team got into a new idea of mechanical structure for the robot. The 3D drawing was made and it is now finalized, leading to the confection process. All of the metal parts, including gears, shafts and axles, were machine at the institution, with bench lathes and mills. All of the bearings were bought, and the fastening elements purchase has been scheduled. The motors are in the team's possession, along with the cameras and the controller board.

As of writing this, the team has been stopped for while by coronavirus pandemy. As soon as things get back to normal and the printings are all done, then everything is ready for the first prototype.

## 4. Conclusion:

The team has been working on the robot for about nine months. In this period, all the members have acquired new knowledge from the most diverse areas, as is expected of a mechatronics project, being a multi-field area of study.

Our plans from this point to the competition are to enhance everything that has been already done, do more testing on the robot and to learn more about the areas involved on the project, so that we can make our robot a superior one. We really hope that this year we can achieve better results.

We have done our best to this point and will continue to do so henceforth, so we can present a robot to solve the tests and trials proposed by RoboCup. That is our goal.

## Appendix

| Component | Unitary Price (USD) | Quantity | Total Price (USD) |
|---|---|---|---|
| **ELECTRONICS** | | | |
| Asus TInker Board | $61,41 | 1 | $61,41 |
| Logic level converter | $1,97 | 1 | $1,97 |
| ADC 16-bit ADS1115 | $7,43 | 1 | $7,43 |
| Accelerometer and gyroscope MPU-6050 | $5,22 | 1 | $5,22 |
| Toxic gas sensor MQ-135 | $5,74 | 1 | $5,74 |
| Monster motor shield | $30,00 | 1 | $30,00 |
| Cameras C920 | $70,00 | 2 | $140,00 |
| Raspberry Pi camera module V2 | $29,95 | 1 | $29,95 |
| Dynamixel XL-320 | $32,00 | 2 | $64,00 |
| Dynamixel XL-430 | $49,00 | 3 | $147,00 |
| DC Motors | $39,31 | 2 | $78,62 |
| Eletronics Total: | | 16 | $571,34 |
| **MECHANICS** | | | |
| Gears | $8,50 | 16 | $136,00 |
| Bearing MR115ZZ | $0,96 | 12 | $11,52 |
| Bearing ECX1055 | $3,08 | 6 | $18,46 |
| Bearing 6805 | $1,33 | 4 | $5,33 |
| Bearing 6901 | $4,03 | 1 | $4,03 |
| Mechanics Total: | | 39 | $175,33 |
| Final product: | | $746,67 | |