Robotic Manipulator for Chocolate Boxes Packing

Francisco García, Ricardo Yépez
Escuela de Mecatrónica, *Universidad Internacional del Ecuador*
Av. Simón Bolívar y Av. Jorge Fernández, Quito, Ecuador,
frgarcia@uide.edu.ec, riyepezzu@uide.edu.ec

***Abstract*—Robots are helpful for the industry in the means that they ca be used for the automation of different processes. In this case, a robot is meant to be used for the automation of the chocolate packaging process. The correct construction of a serial link robot can result in the creation of a helpful tool to increase the efficiency of the industrial processes. The construction of this robot is characterized by having 4 degrees of freedom, a two-finger gripper and the implementation of different types of control based on the forward and the inverse kinematics, that allows the user to control the robot in a very understandable and easy way. Also the robot can be programmed to create routines to automatize the packaging of a box of chocolates. The robot has also a chocolate transportation unit, which is a conveyor belt, used to move the chocolates underneath the robot so it can grip and place them in the box. The tests performed on the robot, show a reliability of 75% with the routine, and an error of 5% while controlling it manually. These results show that the robot has an acceptable value of reliability, that can be increased by modifying the design. The manual control results, show a low error, that can be concluded in a reliable mode to work on.**

***Index Terms*—4 DOF Robot Manipulator, Automatization, Forward Kinematics, Inverse Kinematics, Two-finger Gripper.**

I. Introduction

THE idea of this project is to create just one robotic arm that can do the process of storing and packing the product, instead of having a line of robots with different tasks. This means that the user can leave an empty box and indications of storage to the robot. In this way the robot arm will start to grab each chocolate one by one and place them in an empty position inside the packing box.

In this way, it is proposed to carry out a movement algorithm capable of establishing point coordinates of positions considered empty, where the robot arm will place the product, in order to determine occupied positions, and later discard them and place the new products in unoccupied positions.

This robotic arm will have a R-R-R configuration, which has 3 degrees of freedom provided by the joints. Also, the effector of the robot will be a gripper, that opens and closes grabbing and depositing the chocolate in a desired position. Likewise, it will be designed in such a way that, through communication with the operator, it will provide control of each moving point of the robot arm, through a user interface that allows establishing a percentage value within a bar, to determine the level of arm position within your work zone.

Therefore, a communication interface will be implemented in the same way that will allow the entry of desired coordinates within the workspace, that is, the operator will enter the desired coordinates and then the algorithm, through iterations, will determine the most appropriate task space, that the arm must follow to meet the objective.

Index Terms: 3 DOF Robot Manipulator, Automatization, Forward Kinematics, Inverse Kinematics, Two-finger Gripper

A. *Main Objective*

Create a robotic arm that automates the packaging of chocolates.

B. *Specific Objectives*

- Design, build and implement a gripper capable of efficiently gripping the spherical dimensions of a specific chocolate.
- Allow the implementation of the chosen actuators in such a way that the structure of the arm does not look robust and provides ease of maneuverability of the actuators.
- Create a visual interface in which the user can control the robot functions at his desire.
- Test the prototype and determine its results considering the position of each placed chocolate, avoiding overlapping ones in another, or placing them outside the box in wrong positions.
- Investigate a method to control the movement of the robot based on forward and/or inverse kinematics theories.

II. Theoretical Framework

Cobots or collaborative robots have been revolutionized the food industry in the last decade. Nowadays, these robots for agriculture or food have achieved an increase in productivity and an improvement in the efficiency of processes that has opened up new possibilities in these sectors. Their main objective is simple to understand, they help companies improve their competitive advantage and reduce their costs by enabling the automation of essential processes that contribute to improving worker safety, speeding up production and improving productivity.
One of the concepts most used by robot arms is ”Picking and Placing”, which is carried out with specific products within each industry, thus resulting in one of the main tasks that a robot arm can carry out autonomously. Understanding this becomes of vital importance to see how each industry develops and improves the quality and efficiency of its products.
An example of this is the cocoa industry in Ecuador. It is

estimated that 37% of cocoa is exported to countries such as Switzerland, in which this import represents 60% of its chocolate production, making Ecuador its first main cacao importer [1]. On the other side of the coin is Pacari, which is the main chocolate company in Ecuador, its production estimates 49% of cocoa use of national production. Therefore, if the application of this research is considered, it can be said that there is a wide range of implementation of this type of project due to the fact, Ecuador is worldwide known as a chocolate country [2]. Where companies like Pacari could implement robot arms to package their chocolates.

To consider the main design, the prototype was based on the robot arm of the KUKA brand [3], which is a company that is dedicated to the design and manufacture of different robot arms for each industry. The design was based on a robot arm that has most of the motors as close as possible to the ground, so that in this way the weight and center of gravity are extremely low and all the force of the motors can be used to perform the specific task. Fig 1

Fig. 1: KUKA Robotic Arm Models

III. METODOLOGY

A. *Theoretical Phase*

In order to develop the project, some considerations have to be taken into account. First of all, the gripper of the robot have to be designed and analyzed so it surely can grip the chocolates.

Second, a proper and reliable method to control the movement of the robot has to be developed. This movement has to be fluid, easy to control and also simple to understand. At last, the generation of a Jacobian matrix can also be useful in order to find the accelerations of the project.

*1) **Gripper of the robot**:* To achieve the purpose of gripping the chocolates, a gripper has to be designed. For this particular case, there are two alternatives to make this gripper, which differentiate from each other in the number of fingers that they have. The gripper can be constituted by two or three fingers, and each of them have they're advantages and disadvantages.

On the one hand, a three-finger gripper may provide a higher stability and improve the gripping action because of the number of fingers, but can result a little bit more complex to design and program, and also a little more expensive.

On the other hand, a two-finger gripper is easier to control and program, because of the mechanism, and also can result a little bit cheaper, but because it has just two fingers, the grip may not be the best one.

For this project, the chosen option was to design a two-finger gripper and to control it by the means of a 360º servomotor. This gripper has been designed in such way that the chocolate fits properly without making the robot struggle during the gripping process.

*2) **Movement of the robot**:* There are two ways in which the robot can be controlled which are by applying the forward or inverse kinematics concepts, that, in simple terms, refer to control the robot by inputting angles or coordinates respectively.

On the one hand, to obtain the forward kinematics of this particular robot, it is necessary to use the concepts of the Denavit-Hartenberg convention. For this, first a frame analysis of the robot has to be performed, in order to obtain the different parameters to be placed in the Denavit-Hartenberg table. The analysis of the Choco-Bot is shown in Fig. 2.

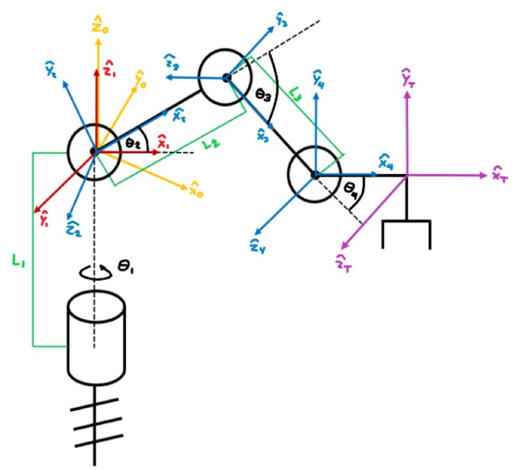


Fig. 2: Denavit-Hartenberg frame analysis for the Choco-Bot

After making the analysis shown in Fig. 2, the **Denavit-Hartenberg** parameters for the robot can be obtained. The parameters for the Choco-Bot are shown in the Table I.

TABLE I: Denavit-Hartemberg Parameters for the Choco-bot

Denavit-Hartemberg Parameters				
Link	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	90º	0	θ_2
3	L2	0	0	$-\theta_3$
4	L3	0	0	θ_4

The next step in order to obtain the **forward kinematics** of the Choco-Bot is to obtain the homogeneous transformation matrix of the gripper with respect to the base. To do so, first the general form of the homogeneous transformation matrix of one link respect to the previous one, given in the Equation 1. has to be used.

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

To obtain each of the matrices, a script in MATLAB was executed. This script is basically a function that has as arguments the a_{i-1}, α_{i-1}, d_i, and the θ_i parameters given on the Table I. Once these matrices were obtained, a post-multiplication operation had to be performed in order to find the homogeneous transformation matrix of the final frame of the robot with respect to the reference point. The matrix that was found is shown in Equation 2.

$$ {}^{0}_{4}T = \begin{bmatrix} \sigma_3 & 0 & \sigma_2 & \sigma_3\sigma_1 \\ \sigma_2 & 0 & -\sigma_3 & \sigma_2\sigma_1 \\ 0 & 1 & 0 & L_2 sin(\theta_2) + L_3 sin(\theta_2 - \theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

where:

$$ \sigma_1 = L_2 cos(\theta_2) + L_3 cos(\theta_2 - \theta_3) $$
$$ \sigma_2 = sin(\theta_1) $$
$$ \sigma_3 = cos(\theta_1) $$

Since the position that wants to be obtained is the gripper respect to the reference point, a translation matrix has to be found to represent the pose of the gripper with respect to the frame 4 of the robot. This matrix is represented by the Equation 3, that has no rotation, and in which L_4 and L_5 represent the distance from the center point of the gripper to the frame 4 of the robot in the X and Y axis respectively.

$$ {}^{4}_{G}T = \begin{bmatrix} 1 & 0 & 0 & L_4 \\ 0 & 1 & 0 & L_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

As it was done before, the pose of the gripper with respect to the reference point can be found by operating the ${}^{0}_{4}T$ and the ${}^{4}_{G}T$ matrices, as it is shown in Equation 4.

$$ {}^{0}_{G}T = {}^{0}_{4}T \times {}^{4}_{G}T \begin{bmatrix} \sigma_3 & 0 & \sigma_2 & \sigma_3\sigma_1 \\ \sigma_2 & 0 & -\sigma_3 & \sigma_2\sigma_1 \\ 0 & 1 & 0 & \sigma_4 + L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4) $$

where:

$$ \sigma_1 = L_4 + L_2 cos(\theta_2) + L_3 cos(\theta_2 - \theta_3) $$
$$ \sigma_2 = sin(\theta_1) $$
$$ \sigma_3 = cos(\theta_1) $$
$$ \sigma_4 = L_2 sin(\theta_2) + L_3 sin(\theta_2 - \theta_3) $$

By obtaining the matrix of the Equation 4, it is possible to control the robot by inputting the angles θ_1, θ_2 and θ_3, which are the ones in the base, arm and forearm respectively. The other values represented as variables are actually known:

- L_2 and L_3 : Lengths of the arm and forearm, both of them of 120 mm.
- L_4: Distance in X axis from frame 4 to gripper, with a value of 58.5 mm.
- L_5: Distance in Y axis from frame 4 to gripper, with a value of 28.45 mm.

On the other hand, to control the robot by inputting coordinates is necessary to obtain the **inverse kinematics**. For this, a geometrical method has been applied.

The first thing to do is to change the view of the robot to the upper view, so the X and Y axes can be seen. From here, a triangle can be set as in Fig. 3.

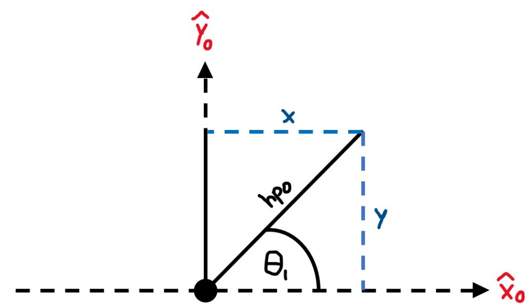


Fig. 3: Upper view triangle

$$ h_{po} = \sqrt{x^2 + y^2}; \theta_1 = \tan^{-1}(\frac{y}{x}) \quad (5) $$

After obtaining the equation for θ_1, the view can be changed to the front view, as shown in Fig. 4. From here, two triangles can be obtained to find θ_2 and θ_3 equations.

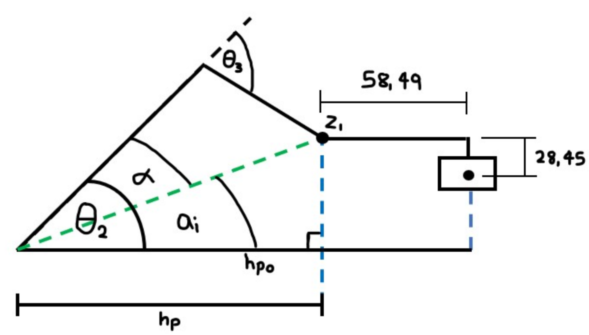


Fig. 4: Front view geometrical relations

The obtained triangles are shown in Fig. 5. Using these triangles will result in obtaining the equations for θ_2 and θ_3.

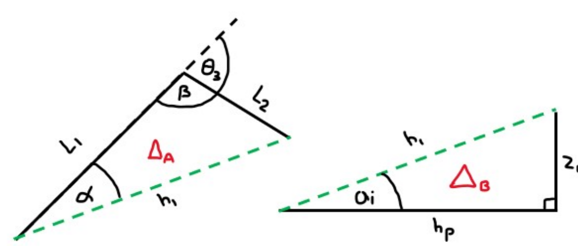


Fig. 5: Triangles to obtain equations for inverse kinematics for this robot

For the triangle A of Fig. 5, the Equation 7 can be made in order to solve it and find θ_3. On the other side, the triangle B is required to obtain the angle 1 to obtain the expression for θ_2, so the Equation 6.

$$
\begin{aligned}
a_1 &= \tan^{-1}(\frac{z_1}{h_p}) \\
h_1 &= \sqrt{hp^2 + z_1^2} \\
\alpha &= \cos^{-1}(\frac{h_1^2 + L_1^2 - L_2^2}{2 * h_1 * L_1}) \\
\theta_2 &= a_1 + \alpha
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
\beta &= \cos^{-1}(\frac{L_2^2 + L_1^2 - h_1^2}{2 * L_1 * L_2}) \\
\theta_3 &= 180 - \beta
\end{aligned} \tag{7}
$$

On the other hand, there is the iterative method for inverse kinematics, the last method that was discussed was the geometric method, while the iterative method is presented by the following code:

```
    clc
clear all
close all
x = -220;
y = 140;
z = 30;

c = 1;
j = 1;
er = 5;
sQ1 = [0,0,0,0];
sQ2 = [0,0,0,0];
sQ3 = [0,0,0,0];
sQ4 = [0,0,0,0];
spx = 0;
spy = 0;
spz = 0;
tic
while(1)

    for i = 2:1:4
        sQ1(i) = rad2deg(atan2(y,x));
        sQ2(i) =randi([0 139]);
        sQ3(i) =randi([-138 -50]);
        sQ4(i) = abs(sQ3(i))-sQ2(i);
    end


    for i = 1:1:4

        phi1 = sQ1(i);
        phi2 = sQ2(i);
        phi3 = sQ3(i);
        phi4 = sQ4(i);

        [O_T_T] = DH_Chobot(phi1, ...
            phi2,phi3,phi4);
        spx(i) = O_T_T(1,4);
        spy(i) = O_T_T(2,4);
        spz(i) = O_T_T(3,4);

    end

    for i = 1:1:4
        Min(i) = abs(x-spx(i))+
        abs(y-spy(i))+abs(z-spz(i));
    end

    j = 1;

    for i = 2:1:4
        if (Min(i)<Min(j))
            j = i;
        end
    end

    sQ1(1)=sQ1(j);
    sQ2(1)=sQ2(j);
    sQ3(1)=sQ3(j);
    sQ4(1)=sQ4(j);

     fprintf("c:%d   a1:%d   " + ...
         "a2:%d   a3:%d   a4:%d   " + ...
         "x:%0.2f   y:%0.2f   " + ...
         "z:%0.2f \n",c ...
         ,sQ1(1),sQ2(1),sQ3(1),sQ4(1), ...
         spx(1),spy(1),spz(1))


   if ((abs(x-spx(1))<=er) && (abs(y-spy(1))<=er)
           && (abs(z-spz(1))<=er))
       break;
   end
    if (c>1800)
         break;
    end
    c = c + 1;
end

phi1f = sQ1(1);
phi2f = sQ2(1);
phi3f = sQ3(1);
phi4f = sQ4(1);
toc
```

This is an iterative method, which randomly generates values for DOFs at each iterationj, and later, those values are used to solve by the Denavit Hartemberg method, which gives values in X,Y and Z that are compared with the values of desired coordinates, and it is established if it is close or not, if it is close, those values are saved, so that they are not erased with the new iteration, which generates other values, and thus if the new values are closer to the solution, they are keep only those. In this way the method approximates the solution by iterations. The algorithm is designed to have a maximum iteration value of 1000.

Either of these methods can be used to achieve the main objective of this project, that is to package the chocolates in a 2x2 box.

*3) **Jacobian obtention** :* To obtain the Jacobian matrix for the Choco-bot, the homogeneous transformation matrix of the gripper respect to the reference (Equation 4) is used.

Using the definition of the Jacobian Matrix, after applying partial derivatives in the position vector of ${}^{0}_{G}T$, the resultant matrix is the one stated in Equation 8.

$$J = \begin{bmatrix} -s(\theta_1)\sigma_2 & -c(\theta_1)\sigma_1 & L_3(\theta_2-\theta_3)c(\theta_1) \\ c(\theta_1)\sigma_2 & -s(\theta_1)\sigma_1 & L_3(\theta_2-\theta_3)(\theta_1) \\ 0 & L_2 c(\theta_2)+\sigma_3 & -\sigma_3 \end{bmatrix} \tag{8}$$

where:

$$\sigma_1 = L_2\sin(\theta_2) + L_3\sin(\theta_2-\theta_3)$$
$$\sigma_2 = L_4 + L_2\cos(\theta_2) + \sigma_3$$
$$\sigma_3 = L_3\cos(\theta_2-\theta_3)$$
$$s = sin$$
$$c = cos$$

B. Design/Analysis Phase

The design of this project has been done in the Inventor CAD for the mechanical parts, and in Proteus for the electronic. The design is divided in four different categories, which are: structure of the robot, gripper of the robot, conveyor transportation system, and the electronic part of the robot.

*1) **Structure of the robot**:* In order to make the structure of the robot, the components that were going to conform this project had to be analyzed. To control the movement, it was stated that the Choco-Bot will have three NEMA-17 stepper motors. Taking this into account, it was important to analyze how the shape and the weight of these motors is going to affect the performance of the robot.

In the first place, to move the base of the robot, a gear-band relation is used to improve the torque of the motor, like shown in the Fig. 6. This relation is also used for the remaining stepper motors that control the arm and the forearm. All the gears that are set for the movement of the motors have a ratio of 4.5, which allows the motors to improve the torque from 12 N*m to 54 N*m.

That is why the stepper motors are located in the lower part of the robot, near the center of gravity, as shown in the Fig. 7. This was done to avoid excessive loads in the structure of the robot, and at the same time to improve the equilibrium of the robot itself.

By implementing this idea, also it can be noticed that the shafts of the motors were not communicated in any way to the joints of the arm and the forearm of the robot. To communicate the motors with the respective joints, first a gear-band relation was designed to be in communication with the motor shaft.

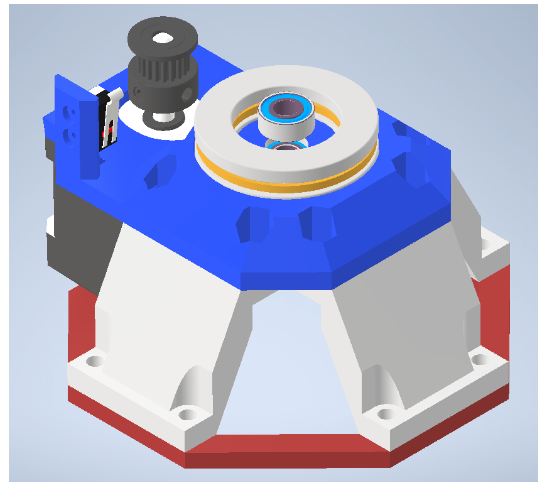


Fig. 6: Gear-band system for the base stepper motor

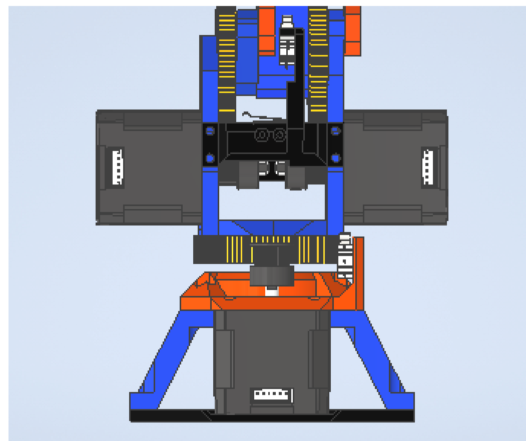


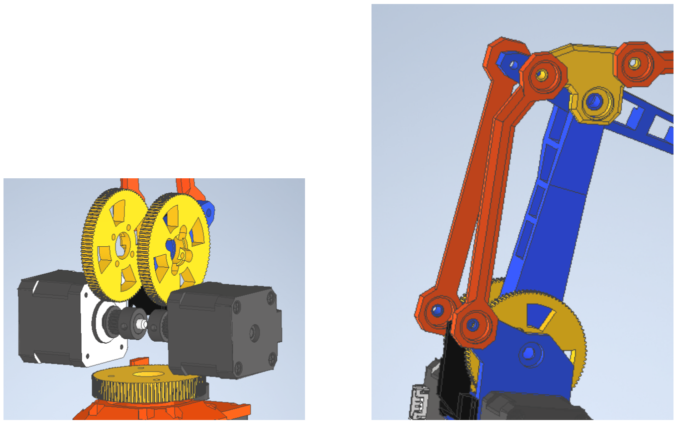
Fig. 7: Stepper motor distribution in the structure of the robot

This gear is the one in charge to communicate pieces added to a linkage system that creates the movement for each joint as it can be seen in Fig. 8. This method helps the motor to translate the rotary motion of the shaft to each joint.

The linkage system that was designed to communicate the joints of the arm and the forearm has also another purpose. It is required for the project that the gripper remains in a stabilized position, that's why another bar was designed to correct the angle of the gripper to be horizontal most of the time, as it is shown in Fig. 9. This can be done because of the elbow piece (in Fig. 9b, the upper yellow piece).

*2) **Gripper of the robot**:* For the design of the gripper, the shape of a chocolate has been taken into account to design two pieces called nails, which are the one in charge to grip the chocolate. By designing those pieces, the result for the gripper was obtained as the one in the Fig. 10 In order to obtain a better grip, is was thought from the beginning that some elastic material can adapt to the form of the chocolate providing stability, so one of this nails was designed to put a rubber exercise band in tension so it can function as a cushion for the chocolate.

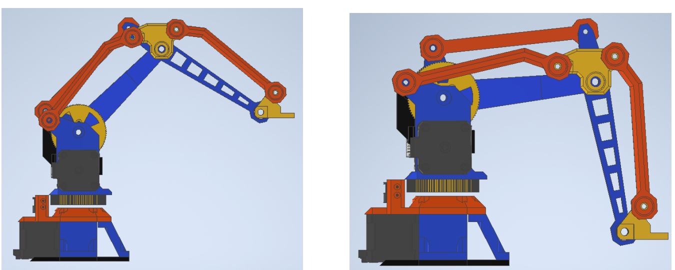
The remaining design of the gripper consists in the mechanism of the opening and closing. It's based on a gear-rack



(a) Gear mechanism of the stepper motors

(b) Linkage communication for the joints

Fig. 8: Communication system between the stepper motors and the arm and forearm joints of the robot



(a)

(b)

Fig. 9: Stabilizing system of the gripper. It is shown in (a) and (b) that the gripper remains in the same position

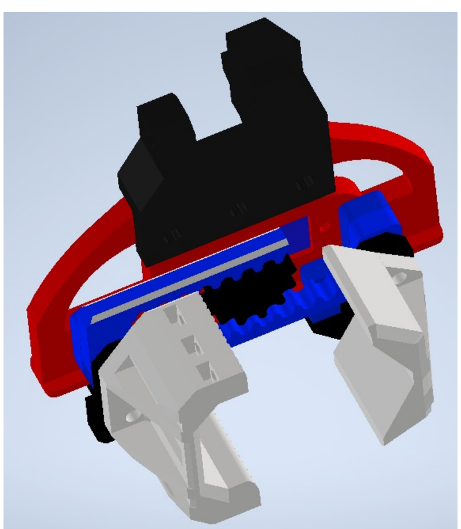


Fig. 10: Design of the gripper

relation that is motioned by a micro servomotor. The other parts that compose the gripper have been designed in a way so the components would fit without any problem, and the pieces themselves would couple to the robotic arm perfectly.

The gripper also counts with to spaces to put limit switches. The purpose of these switches is to identify when the gripper is opened or closed.

This gripper is also designed to grip something to work, since the closed limit switch is located under the rubber band cushion in the nail. If nothing is gripped by the robot, and the gripper is closing, the gripper will not stop since it has nothing in the inside to hit the limit switch. This limitation of the project can be seen in Fig. 11.

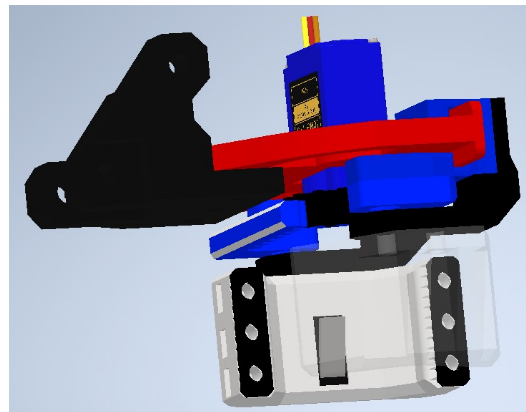


Fig. 11: Limit switch located in one of the nails

The complete design of the Choco-Bot with the gripper is shown in Fig. .

*3) **Conveyor System Design**:* The conveyor system is another important part of the project, since it is the one that automatizes the process of delivering the chocolates to the robot.

It is designed with 6 cylinders, 5 of them to be in the middle of the conveyor to help with the movement of the band and 1 of them to be the one that provides the movement. This mechanism can be seen in Fig. 12.

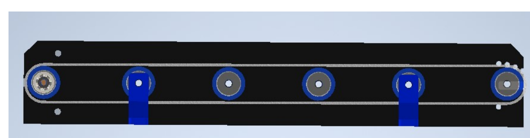


Fig. 12: Side view of the conveyor. The left cylinder is the one in charge of the motion

This system is also designed to work with a NEMA-17 stepper motor, and the selected band for the design was made of polyurethane.

The purpose of this conveyor belt is to in the future implement a chocolate dispenser that puts the chocolate in the transporting system to move it underneath the robot, automatizing the process completely. The complete design of the conveyor system can be seen in Fig. 13.

*4) **Circuit Enclosure**:* To keep all the electronic components safe, an enclosure (seen in Fig. 14 has been designed to protect and to keep them in one position. This enclosure has enough space to enter new components if desired, and also has cuttings in the sides for the IR diode and the power supply. Also it provides a comfortable site for

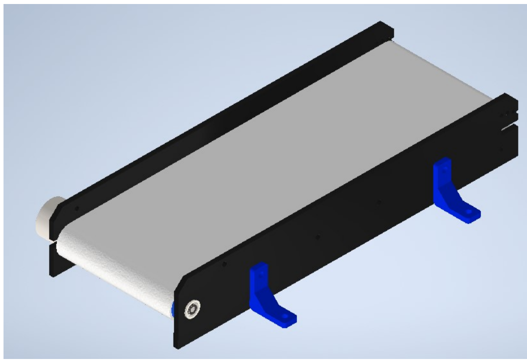


Fig. 13: Conveyor system design

the LCD screen, from which the user can read and interact with it with not a single complication.

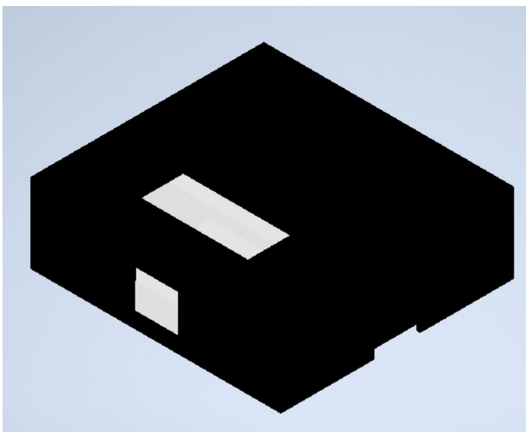


Fig. 14: Enclosure for the electronic components

*5) **Whole Design**:* The final design is complemented with the union of each section, and a last piece called "Package" is added, which is in charge of establishing the locations for each chocolate within the process. Fig 15

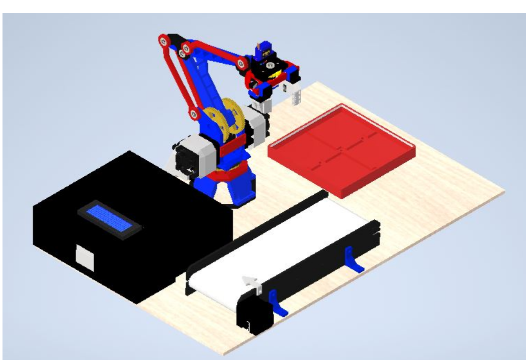


Fig. 15: Whole Design

*6) **Electronic Part**:* For the electronic part, the first thing was to select a microcontroller that could fit in the project. An Arduino MEGA was selected due to the high number of pins so all the connections can be made, and, if that's the case, to add more components without trouble.

In order to provide the movement to the motors of the robot, 4 drivers A4988 were used. This drivers allow the user to change the speed, direction and the energizing of the motors. Also to cool the 3 motors that are placed in the arm, 3 4010 fans (1 per motor) have been used.

To implement the interface a 20x4 LCD screen has been used, and for the human-machine communication, an IR diode with a remote controller were implemented. The selected screen is communicated with the microcontroller using the I2C serial in order to save pins. The purpose of selecting a large screen is to develop an interface that is easy to understand and manipulate.

An schematic circuit of the project has been designed using Proteus, a software in which the simulation of the connections was made. This schematic is shown in Fig. 16.

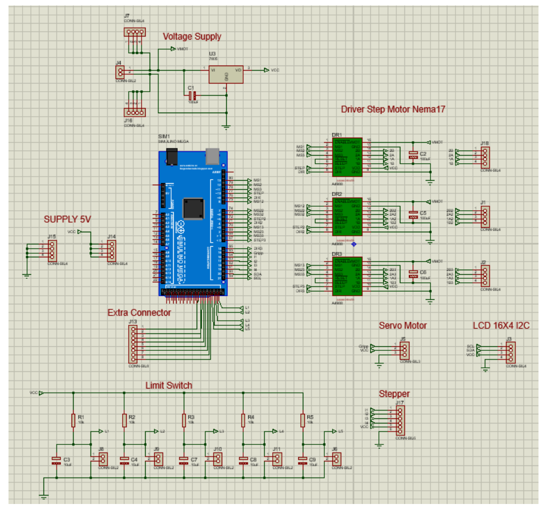


Fig. 16: Circuit designed in Proteus

To supply the components of this project, a 12V, 10A power supply has been used. The selection of this power supply was done by calculating the maximum current that the circuit may draw at any point. The calculation of the current can be seen in the Table II.

TABLE II: Current Consumption of the Project

Total Current Consumption of Choco-Bot			
Component	**Unit Current**	**Quantity**	**Total Current**
Arduino MEGA	73.19 mA	1	73.19 mA
NEMA17, A4988	1.2 A	4	4.8 A
SG-90 Servo	800 mA	1	800 mA
4010 Fan	100 mA	3	300 mA
LCD 20x4	75 mA	1	75 mA
Others	—	—	1 A
Total	—	—	**7.481 A**

To keep the project organized, a two-side PCB (Printed Circuit Board), shown in Fig. 17, has been designed in the Proteus software as well. To save even more space with the circuit, the PCB was designed in a way that it fits in the Arduino MEGA as a shield.

C. Implementation Phase

For the manufacturing of the mechanical parts that compose the robot, the best alternative was 3D printing, because of the

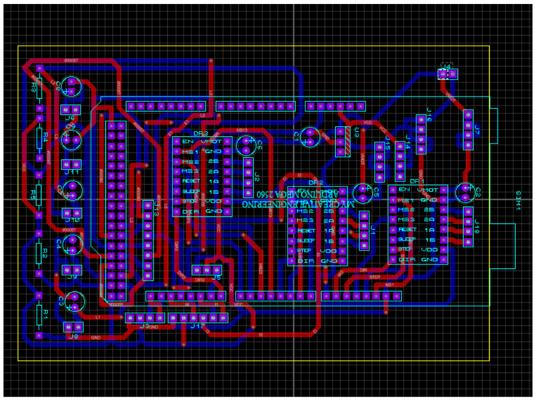


Fig. 17: PCB designed in Proteus

complex geometry of the pieces. The selected material for the manufacturing of most of the pieces was PLA filament, and the 3D printer used to manufacture the pieces was an Ender 3 Pro with a tolerance of ±0.2mm. The only piece that requires a different type of filament is the base socket piece (see Fig. 18), which needs PETg filament that is more resistant to higher temperatures that come from the base motor.

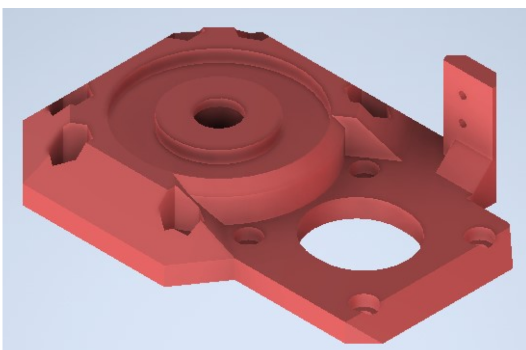


Fig. 18: Base socket piece

A comparison between both materials, PLA and PETg can be seen in Table III.

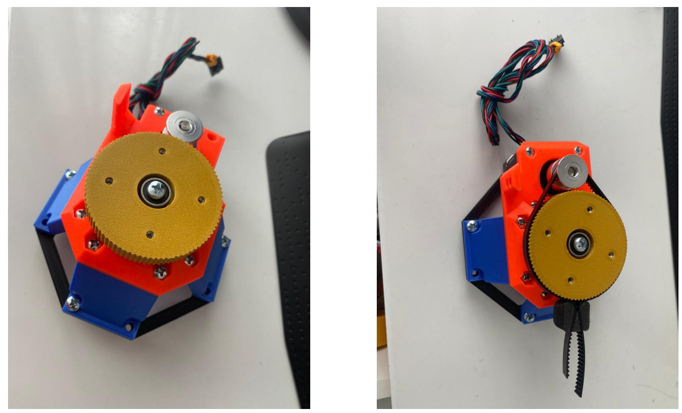
TABLE III: PLA and PETg Mechanical Properties Comparison

Property	**PLA**	**PETg**
Heat Deflection Temperature	55ºC	68ºC
Modulus of Elasticity	2.3 GPa	2.1 GPa
Max. Melting Temperature	145-160ºC	220-245ºC
Glass Transition Temperature	55-60ºC	80-82ºC
Elongation	2%	14%
Final Traction Force	26.4 MPa	45.8 MPa

1) Assembly of the structure of the robot: The first thing that was implemented, was the structure of the robotic arm, and the first step in here was the assembly of the base. For this, the support pieces, the base socket, the base gear and the stepper motor of the base were needed to build the lower part of the robot. To provide rotation in the base an axial bearing was positioned in the base socket. This step is represented in Fig. 19

After this, the gear has to be put over the bearing so it can rotate freely. Also, a small gear for the shaft of the motor has to be set, and a GT2 belt has to be used to communicate this two gears as it is shown in Fig. 20.

Fig. 19: Lower part of the arm assembly



(a) (b)

Fig. 20: Establishment of the communication between the base gear (a) and the shaft (b)

To finish with the base assembly, the main body part has to be coupled in the base gear so it can rotate with no problem, as shown in Fig. 21. This piece is the one that will hold the remaining motors for the arm and the forearm.

The next step in the structure assembly is the construction of the mid-part of the robot. This part consists in the arm of the robot, which contains the gears for the other motors to communicate with the joints.

To start, the gears have to be coupled and adjusted to their respective pieces, as it is indicated in Fig. 22. By doing this step, the gears can be communicated with their respective stepper motors with GT2 belts, just as it was done for the base.

To provide a better rotation on the pieces, some bearings have to be put in the corresponding parts as shown in Fig. 23.

After that, the resultant two pieces are assembled together to obtain a complete mid-body part for the robot. This obtained piece is the one shown in Fig. 24.

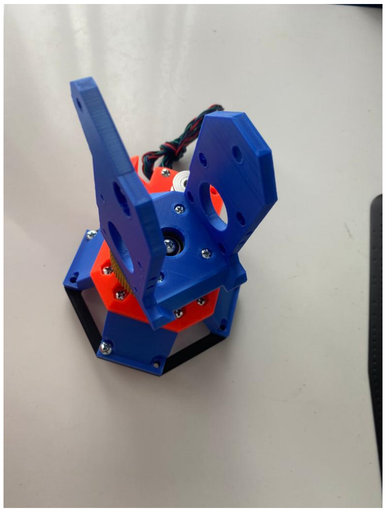


Fig. 21: Coupling of the main body piece

Fig. 22: Motor gears positioning

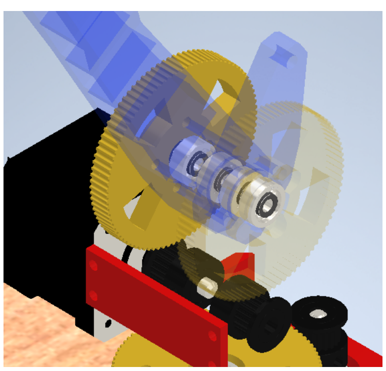


Fig. 23: Bearings including to the mid-body part of the robot

The final step to obtain the whole structure of the robot is to build the linkage mechanism that contains the wrist, elbow and forearm of the robot. For this, the forearm has to be assembled to the mid-body part with the help of the elbow, as shown in Fig. 25. Also, the wrist has to be set in the tip of the forearm, because the wrist is the piece that will hold the end-effector.

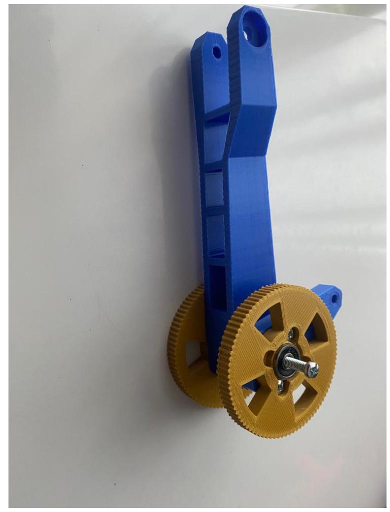


Fig. 24: Mid-body part subassembly

Fig. 25: Forearm assembly

After this, both linkage bars have to be put in the arms structure, to communicate the joints. This process can be seen in Fig. 26. This linkage bars are not connected directly to the joints, they are connected to the elbow piece to move the arm and the forearm.

After obtaining this subassembly, it has to be placed in the lower part of the robot, more precisely, in the main body part. This is done by also putting the remaining stepper motors on the sides of the main body part, with the gear in the shaft so the gears can be communicated with the motors with the GT2 toothed belts. When this is done, the result is the structure of the robot, that is shown in Fig. 27.

2) Assembly of the end-effector of the robot: After the structure of the robot has been completed, the end-effector of the robot has to be implemented. For this, the first step is to assembly the base part of the gripper with the wrist part in the structure, so the gripper can be assembled directly in the robot. This process is shown in Fig. 28.

The next thing to do is to attach the sliders to the base of the gripper as shown in Fig. 29. This pieces once inserted,

(a) (b)

Fig. 26: Positioning of the linkage bars

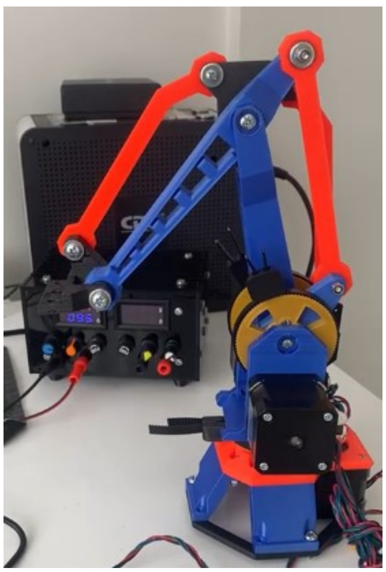


Fig. 27: Robot structure assembly

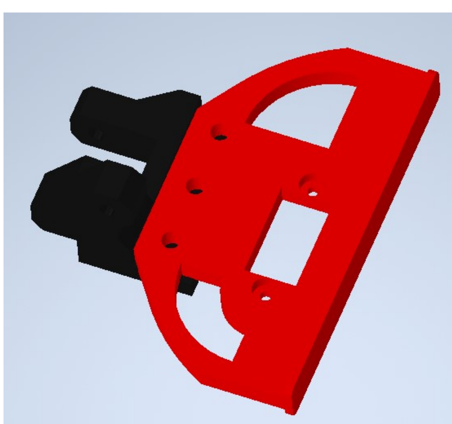


Fig. 28: Attachment of the gripper base to the wrist in the structure

will be the initial point for the opening-closing system of the gripper.

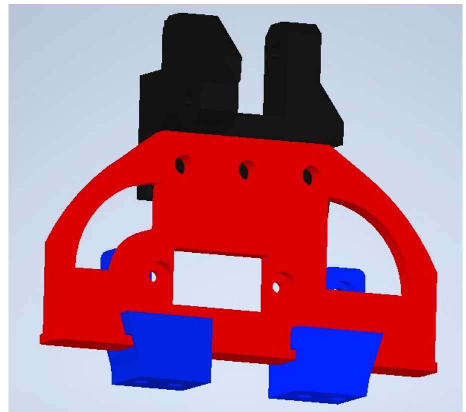


Fig. 29: Insertion of the sliders in the base of the gripper

After this, the 360º servomotor has to be inserted in the socket located in the gripper's base as shown in Fig. 30. The purpose of inserting the servomotor there is to provide a rotational movement that will open or close the gripper with the help of a mechanism.

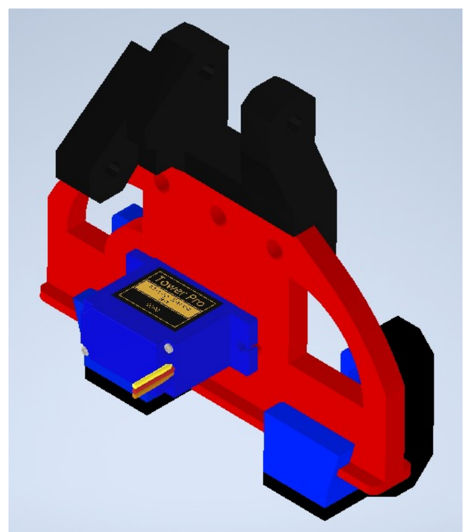


Fig. 30: Assembly of the servomotor and the gripper's base

Next on, the finger pieces have to be attached to the assembly. This pieces are the ones that will constraint the mechanism of opening and closing of the gripper. The process for this step is shown in Fig. 31.

After setting all the previous pieces, the mechanism can be assembled in the end-effector. This mechanism can be seen in Fig. 32 and contains two racks and one pinion. When the servomotor that was previously attached rotates, the pinion also rotates, and because of the racks, the rotational movement becomes linear movement, providing a motion of opening and closing for the gripper.

In order to finish the assembly of the end-effector, the nail pieces have to be set. This pieces are the one that grip the chocolate. Both of them have to be located in the fingers as shown in Fig. 33.

To achieve a better gripping effect, a rubber exercise band

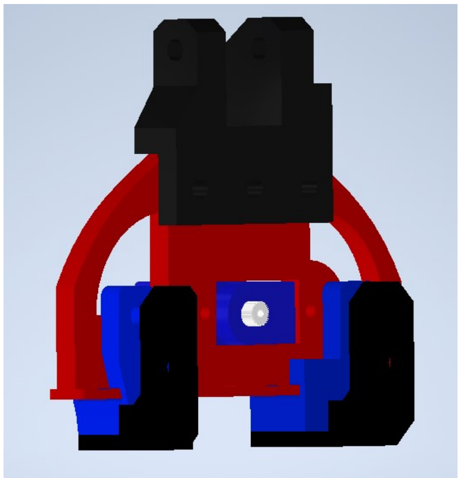


Fig. 31: Attachment of the finger pieces

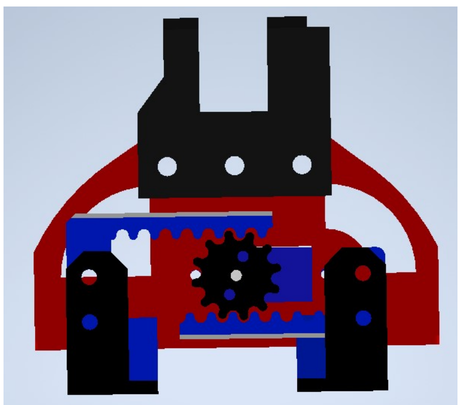


Fig. 32: Gripper mechanism insertion

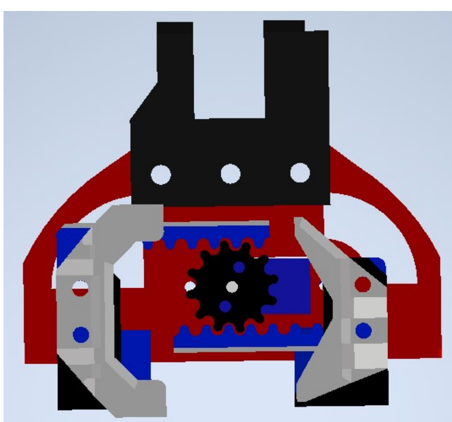


Fig. 33: Insertion of the gripper nails

was attached to one of the nails, as shown in the Fig. 34. The purpose of this is for the elastic band to acquire the shape of the chocolate, providing tension to it, to grip it and to mold the gripper to the chocolate shape. This means that it doesn't matter what is the shape of the chocolate, because of the elastic band, the gripper will mold to the shape of the chocolate that has to be moved.

Also, to secure the chocolate in place, without damaging it,

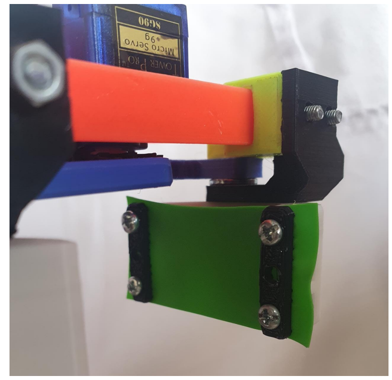


Fig. 34: Insertion of the elastic band in the nail

two small pieces of foam have been added to the opposite nail to emulate a bumping effect, as shown in Fig. 35.

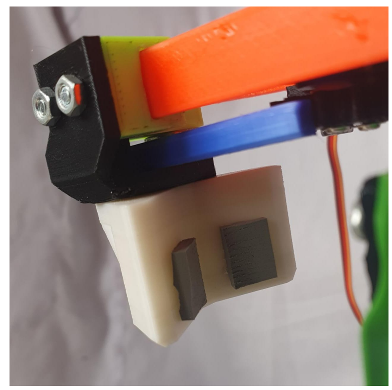


Fig. 35: Implementation of foam bumpers in the opposite nail

After the assembly of the gripper is completed, the complete assembly of the robot is finalized as well. The result of the assembly between the structure and the gripper is shown in Fig. 36.

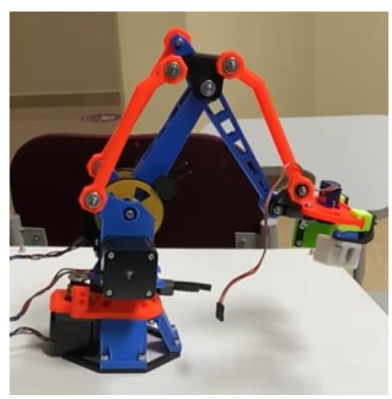


Fig. 36: Whole assembly of the robot

3) Electronic implementation: After the robot is completely assembled, the focus has to be set in the electronic part of the project. As stated before on the Section III-B6, a PCB circuit has been designed for the project. Also it is stated that an enclosure for the electronic components has been designed.

This PCB was manufactured in a double-sided bakelite, in a way that in one of the sides, the only component that will be inserted is the microcontroller, making the PCB to be similar to a shield. In the opposite side of the PCB, all of the other control components have been placed, like the A4988 drivers, the IR diode, and the connections for the motors. This PCB was placed inside the electronic enclosure, along with the power supply, as shown in the Fig. 37.

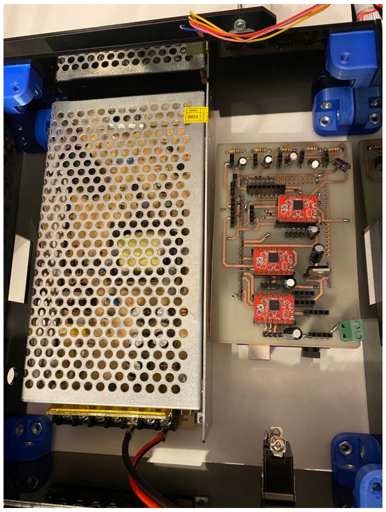


Fig. 37: Implementation of the PCB in the enclosure

After making the PCB in its entirety, an important step had to be done to secure the correct functioning of the project, and that was the calibration of the A4988 drivers. In this case, a calculus to select the correct reference voltage had to be done using the Equation 9.

$$V_{ref} = 8 \times I_{limit} \times R_{ref} \tag{9}$$

For the project, the limit current had to be set as 1.2A because of the NEMA-17 steppers, and the reference resistance is 0.1Ω. By applying the Equation 9, the reference voltage gives a value of 0.96V.

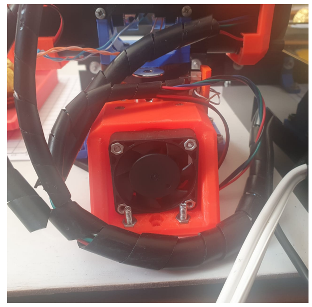
Another important thing that was added to the project was the refrigeration of the stepper motors by the means of small fans. This fans were attached to the motors using special sockets as shown in Fig. 38, and with a supply of 12V.

4) Conveyor Belt: The construction of the conveyor belt is based on 6 cylinders. 5 of the cylinders are used to help with the movement of the conveyor. The remaining cylinder is controlled by an stepper motor to provided the movement to the conveyor.

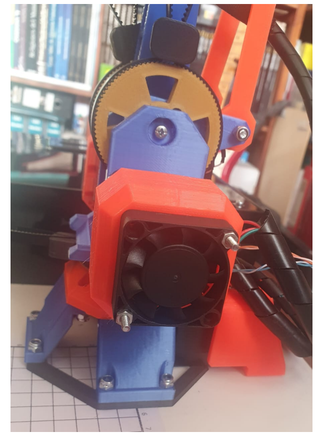
The constitution of the conveyor is shown in the Fig 39, where the cylinder in the left is the one that provides the movement.

The band used for the conveyor belt, is a special designed polyurethane band. This band helps the chocolates not to slip in the conveyor, and also the friction of the band is not high in order to not delay the movement with the cylinder.

The final result of the assembly of the conveyor is shown in Fig 40.



(a)



(b)

Fig. 38: (a) Fan to cool the base motor, (b) Fan to cool down the arm motor

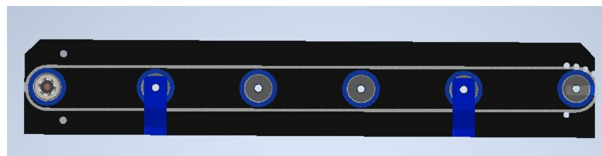


Fig. 39: Cylinders on the Conveyor

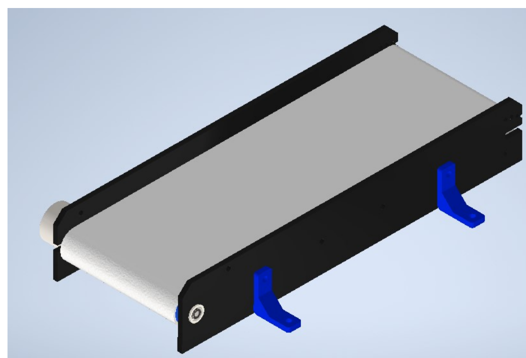


Fig. 40: Final Design of the Conveyor

D. Visual Interface

The visual interface has been improved during the whole proyect, in this term by final presentation the interface´s version is the v7.4.

Choco-Bot v7.4 has a developed interface so the user can control the robot with total freedom without any complication. This interface divides the control of Choco-Bot v7.4 in two different ways: automatic control and manual control. The user can select whichever of the control options by pressing the corresponding button in the remote control.

*1) **Mode Selection**:* To select automatic control the user has to press button (1). The same principle applies in all of Choco-Bot v7.4 menus. Fig. 41

When the user selects an option (either automatic or man-

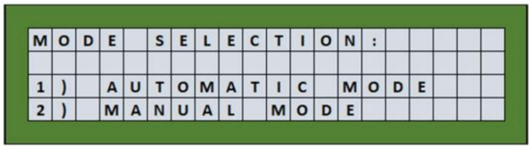


Fig. 41: Selection Menu

ual), it is allowed to return to the mode selection menu by pressing the PREV button in the remote control.

*2) **Automatic Mode**:* In this mode, the user has 3 principal options: Fig. 42

1) Start the routine
2) Actual session stats checking
3) Auto-home function

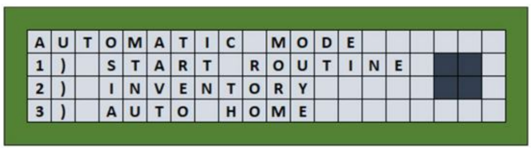


Fig. 42: Automatic Mode

- **Start Routine:** If the user selects this option, Choco-Bot v7.4 will start its purpose: pack a chocolate box. The chocolate is deposited in the conveyor belt, so it can be moved automatically to a specific position so the robot can pick it up. The robot will deliver the chocolate to a preset position and repeat the same process for the remaining chocolates, until the 2x2 box is fully completed. Before starting this process, the robot will automatically perform an auto-homing function. Also, the user can visualize which chocolate has been placed by Choco-Bot v7.4 in the screen (inFig. 43, dark gray squares). After the box is completely filled, the robot will return to the home position to be ready for the next chocolates to be packaged.
- **Inventory:** In this screen the user can see how many chocolates have been packaged and how many boxes have been completed.

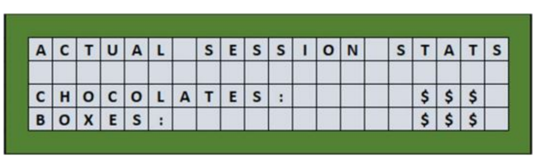


Fig. 43: Inventory

- **Auto Home:** Auto-home option will open the Choco-Bot v7.4 gripper and also will set all the angles (base, arm, forearm) to 90 degrees.

*3) **Manual Mode**:* In this mode the user can do three things: Fig.44

1) Move and control the arm.
2) Open and close the gripper.
3) Move the conveyor motor.

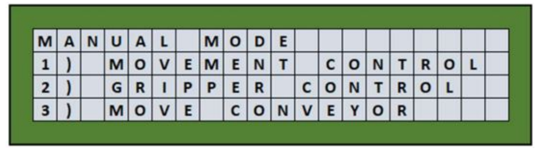


Fig. 44: Manual Mode

- **Movement Control:** Choco-Bot v7.4 offers the user to control the robot by using two different types of inputs:

  - **Angle input (Forward kinematics)**
  - **Coordinate input (Inverse kinematics)**

  By default, when Choco-Bot v7.4 starts, it will show the user the angle input. The user can change to the coordinate input by pressing the CH+ button in the remote and return to the angle input by pressing the CH- button.

  - **Angle Input Control:** This is the mode that is set by default in Choco-Bot v7.4 programming, so it will be shown any time the machine is turned on. This mode allows the user to control the angles of the links in the robot, by changing the values to move the base, arm, and forearm motors. Fig 45

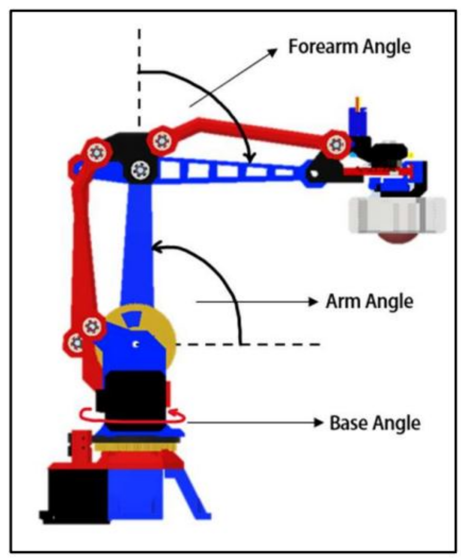


Fig. 45: Angles in Choco-Bot v7.4

In order to control these angles, the user has to select one of the options (1-3) to change the angle value. Fig 46

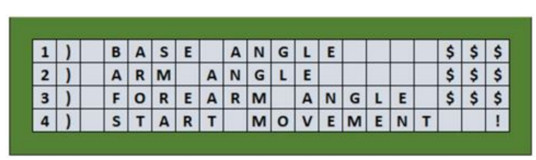


Fig. 46: Angle input control (forward kinematics method)

To change the value of the selected angle, the user has to press the + or – buttons, either to increase or decrease the angle value in one degree. When an option is selected in the bottom right corner a ! will be printed. To confirm the value, the user has to press the PLAY/PAUSE button on the remote control, and Choco-Bot v7.4 will save the desired value. When all the values are the ones desired, the user can press the button 4 on the remote to start the movement of the arm.

– **Coordinate Input Control:** Choco-Bot v7.4 also allows the user to introduce coordinates as inputs to control the movement of the robot. The principle to input the coordinates is the same as in the angle input, but with some slight differences. Fig 47

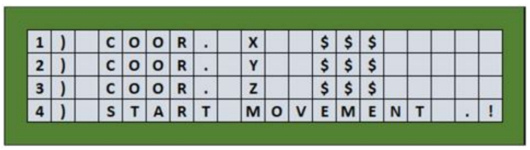


Fig. 47: Coordinate input control (inverse kinematics method)

Coordinate input will affect all of the angles in the robot, so when the user confirms a coordinate set, Choco-Bot v7.4 will move every motor to achieve the desired position. This method is a little bit more precise than the previous one, as it has two different options for steps. The user can modify the coordinate either by 1 or 0.01 to obtain decimal values.
The default step is of 1, but the user can change to the 0.01 step by pressing the CH button on the remote control. To return to the step of 1, the user has to press the EQ button on the control. An indicator (.) in the bottom right cornet will appear if the selected step is of 0.01. It is important to state the direction of the positive X, Y and Z axes of Choco-Bot v7.4 so the user may input the correct values. The directions of the axes are shown in Fig. 48, when the robot is at home position:

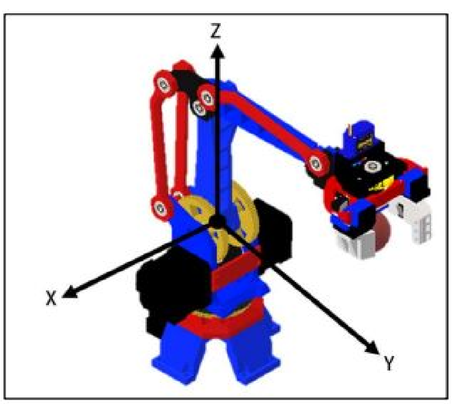


Fig. 48: Coordinate axis of Choco-Bot v7.4 when is set in home position

To make an example, if the user wants to move the gripper to the packaging box, the coordinate in X must be negative, and in Y positive because of the positions. The Z coordinate works in a slightly different way. It refers to the position of the gripper respect to the reference point, which is the joint of the base and the arm. This means that, for example to pick a chocolate, the user has to input negative values in Z. If Choco-Bot v7.4 detects an unreachable position and the user wants to move the gripper to that point, the message in Fig. 49 will be displayed in the interface for 3 seconds.

Fig. 49: Unreachable point warning message for coordinate input

By this mean, Choco-Bot v7.4 is protected from going to unwanted positions and damage itself in the process.

– **Kinematics Information:** In any type of movement input (either angles or coordinates), if the user presses the button 5, the information about the exact position of the robot will be displayed on the screen as shown in Fig. 50 $ represents coordinates and % angles.

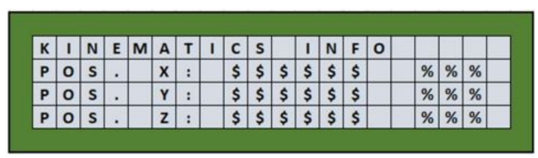


Fig. 50: Kinematic info screen. $ represents coordinates and % angles

By looking at this information the user can check the position of the robot in coordinates (left values), and also know the values of each of the angles in that specific position (right values).

- **Gripper Control:** The control of the gripper is quite simple. It has just two options, to open the gripper and to close it (check Fig. 51).

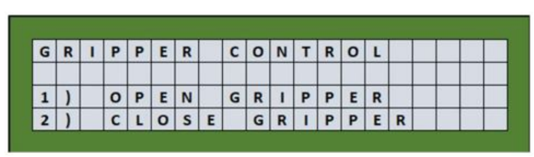


Fig. 51: Gripper control menu

It is important to note that in order to close the gripper, there must be an object underneath the gripper itself, otherwise, Choco-Bot v7.4 will fail, because of its design.

- **Conveyor Movement:** The option of the conveyor movement is the simplest among all the options in the interface. Whenever the user presses the button to activate the conveyor movement it will just rotate the conveyor motor to translate the chocolate underneath the gripper.

E. Test and Validation Phase

In order to determine whether the robot is correctly programmed and analyzed, movement tests were applied for both angle and coordinate input methods.

The first one to be tested was the forward kinematics method. The input values are the ones shown in Fig. 52. For the inverse kinematics, the input values are set as in Fig. 53.

Fig. 52: Test of forward kinematics of the robot

Fig. 53: Test of inverse kinematics of the robot

After making the robot move by pressing the action button, the kinematics info menu was checked to compare the result obtained by the microcontroller and the MATLAB calculus. These results are shown in Fig. 54.

As it can be seen, the values for either the forward kinematics and inverse kinematics doesn't vary in a large quantity compared to the values obtained in MATLAB.

IV. Tests & Results

In order to perform the tests, several movements have provided to the inverse kinematics method, which is the one that may tend to fail. Also, the routine was tested in order to determine the behavior of the project.

A. Inverse Kinematics

1) I.K Geometric Method: A total of 20 tests were performed to obtain the accuracy of the geometric analysis algorithm for inverse kinematics.

The tests carried out were based on entering a specific coordinate within the workspace of the robot arm. And then compare the result of the coordinate with the desired values. Fig 55

According to the tests carried out, an error rate of 2.85% was reached, which indicates that the method used has a reliability of 97.15%.

Real				Obtained			
	X	Y	Z		X	Y	Z
Desired	-155,18	130	-29,28	Desired	-154,2	132,3	-29,5

Real				Obtained			
	X	Y	Z		X	Y	Z
Desired	80	110	70	Desired	80	110	70

Real				Obtained			
	X	Y	Z		X	Y	Z
Desired	0	200	-65	Desired	0,8	201,5	-65,8

Fig. 55: Examples Test of Geometric Method

2) I.K Iterative Method: Similarly, for this method, 20 tests were carried out where the desired coordinate was entered, and the analyzed coordinate was obtained. This method gave a 5.014% error, but only if the times in which the iterative algorithm found an approximation is taken into account, because the algorithm has a maximum limit of 1000 iterations, there were occasions in which it was not reached. find an acceptable solution. Therefore, this method becomes a method that is not completely reliable and has a higher percentage of error than the geometric method.

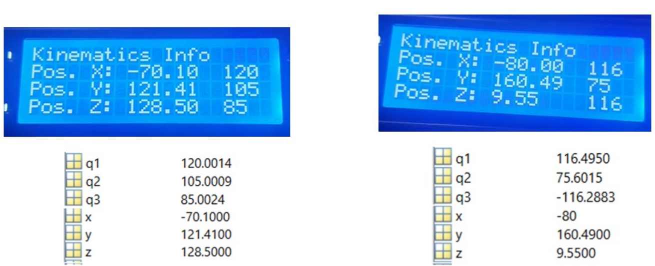
Real				Obtained			
	X	Y	Z		X	Y	Z
Desired	-155,18	130	-29,28	Desired	-150,5	129,6	-26,7

Real				Obtained			
	X	Y	Z		X	Y	Z
Desired	80	110	70	Desired	76,5	112,23	70,64

Real				Obtained			
	X	Y	Z		X	Y	Z
Desired	0	200	-65	Desired	2,37	203,41	-62,89

Fig. 56: Examples Test of Iterative Method

3) Routine Test Results: To test the project, 8 automatic routine tests have been done, in which the robot was supposed to deliver all 4 chocolates to the box. The results can be seen in Table IV.



(a) Forward kinematics results (b) Inverse kinematics results

Fig. 54: Results of the movements of the robot

TABLE IV: Results of Tests for Automatic Routine

Routine Tests Results			
Test	Completion (/4)	Routine Completed	Cause of Failure
1	4/4	Y	———————
2	4/4	Y	———————
3	1/4	N	Chocolate not gripped
4	4/4	Y	———————
5	4/4	Y	———————
6	4/4	Y	———————
7	3/4	N	Chocolate not gripped
8	4/4	Y	———————

		Features					
	Element	Tipo	Diameter	Lenght	Uni Price.	Quantity	Total Price
Screw	Screw M4	estrella	4 mm	50 mm	$ 0,05	2	$ 0,10
				40 mm	$ 0,05	4	$ 0,20
				16 mm	$ 0,05	10	$ 0,50
				10 mm	$ 0,05	8	$ 0,40
	Nut M4 Autoblock		4 mm		$ 0,06	24	$ 1,44
	Screw M3	Estrella	3 mm	6 mm	$ 0,05	38	$ 1,90
	Nut M3		3 mm		$ 0,03	20	$ 0,60
	Screw M2	Estrella	2 mm	10 mm	$ 0,05	10	$ 0,50
	Nut M2		2 mm		$ -	10	$ -
	Arandelas	Planas	4 mm		$ 0,02	20	$ 0,40
					Subtotal N°1:	**98**	**$ 6,04**
Mechanical Elements	GT2 Belt timing	6 mm		1,2 m	$ 6,00	1	$ 6,00
	Bearing	Lineal	4 mm	13 mm	$ 2,00	18	$ 36,00
	Bearing	Axial	25 mm	50 mm	$ 5,00	1	$ 5,00
	Conveyor	-	-	300 mm	$ 22,50	1	$ 22,50
					Subtotal N°2:	**20**	**$ 69,50**
Control Coponents	LCD 20X4 I2C				$ 14,00	1	$ 14,00
	Infrared Sensor and Control 1838				$ 9,50	1	$ 9,50
	Minor Electronic Components				$ 7,45	1	$ 7,45
	Motor Nema17				$ 19,45	4	$ 77,80
	Driver A4988 Motor				$ 3,50	4	$ 14,00
	Servomotor Sg90				$ 6,00	1	$ 6,00
	Laser Cut and Acrilic				$ 8,00	1	$ 8,00
	Power Supply 12V 10A				$ 9,00	1	$ 9,00
	PCB 2 LADOS				$ 18,00	1	$ 18,00
					Subtotal N°2:	**15**	**$ 163,75**
					Total:	**118**	**$ 239,29**

Fig. 57: Final Budget

By looking at these results, it can be concluded that most of the time the robot work with total normality. The times where the routine failed, was when the gripper mechanism malfunctioned. This happens because of the design of the gripper, so in the future an improvement about this gripper system can be done.

V. Conclusions

- The manipulative robots have been designed in order to eliminate the efforts caused by the lifting of the merchandise and are intended for various and diversified fields.
- With our design, you can carry out simple and concise explanations about step motors, electronic components, parts of an arm, as well as the definition of link, degrees of freedom.
- Above all, it can be concluded that the process carried out within this investigation determines the fulfillment of the main objective of packing chocolates with spherical dimensions of 2 cm in diameter.
- It was possible to design and incorporate a gripper with an adaptable point to adapt to any surface of the chocolate.
- It was possible to create and incorporate a user interface capable of interacting with all the functionalities of the robot arm, as well as for: Forward Kinematics, Inverse Kinematics, Homing routine, Chocolate routine and also being able to observe the number of packaged chocolates.

Acknowledgments

Thank you to Professor Patricio Cruz for helping us to develop this project in a fantastic way. The work done and the memories shared with this project are simply marvelous. Also thanks to Professor Luis Andrade for helping us with tips about the construction of the prototype itself.

Appendix A
Total Costs

The final budget of the project is shown in Fig. 57.

Appendix B
Mechanical Diagrams

The following link leads to the files of the mechanical plans: Link: https://bit.ly/3yngCbo

Appendix C
Electrical Diagrams

The following link leads to the files of the electrical plans: Link: https://bit.ly/3ygBKjt

Appendix D
User Manual

This link guide to the user manual: Link: https://bit.ly/3yngCbo

REFERENCES

[1] Fortaleciendo el sector de cacao en Ecuador — Rikolto en Latinoamérica. (2022). Retrieved 4 July 2022, from https://latinoamerica.rikolto.org/es/project/fortaleciendo-el-sector-de-cacao-en-ecuador

[2] (2022). Retrieved 4 July 2022, from https://blogs.cedia.org.ec/obest/wp-content/uploads/sites/7/2020/07/An%C3%Alisis-de-cacao-24-de-junio-2020-7.pdf

[3] Robots de paletizado de KUKA — KUKA AG. (2022). Retrieved 4 July 2022, from https://www.kuka.com/es-es/productos-servicios/sistemas-de-robot/robot-industrial/robot-de-paletizado

[4] Balasubramanian, R. (2011). The Denavit Hartenberg Convention. USA: Robotics Insitute Carnegie Mellon University.

[5] Lens, T., Kunz, J., Stryk, O. V. (2010, November). Dynamic modeling of the 4 DoF BioRob series elastic robot arm for simulation and control. In International Conference on Simulation, Modeling, and Programming for Autonomous Robots (pp. 411-422). Springer, Berlin, Heidelberg.

[6] Ayodeji, S Owoyemi, Joshua Eng, M Martins, O.O. & Adekunle, Adefemi. (2016). Dynamic Simulation of a 4 Degree of Freedom (4DOF) Robotic Arm for Small and Medium Scale Industry Packaging.

[7] Cuadrado, Javier Naya, Miguel Ceccarelli, Marco & Carbone, Giuseppe. (2002). AN OPTIMUM DESIGN PROCEDURE FOR TWO-FINGER GRIPPERS: A CASE OF STUDY.

[8] Lanni, Chiara Ceccarelli, Marco. (2009). An Optimization Problem Algorithm for Kinematic Design of Mechanisms for Two-Finger Grippers. The Open Mechanical Engineering Journal. 3. 10.2174/1874155X00903010049.

[9] Ballentine, G. (2022, February 24). SOLIDWORKS Rack and Pinion Mate Tutorial — GoEngineer. https://www.goengineer.com/blog/solidworks-rack-and-pinion-mate

[10] Craig, J., (2005). Introduction to Robotics, Mechanics and Control. 3rd Ed. Pearson Education International, pp.62-76.

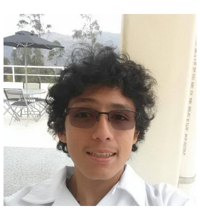
[11] In-Depth: Control Stepper Motor with A4988 Driver Module & Arduino. (n.d.)., from https://lastminuteengineers.com/a4988-stepper-motor-driver-arduino-tutorial/

[12] Castaño S. (2021, October). Como usar un Servomotor con Arduino (180° y 360°). https://controlautomaticoeducacion.com/arduino/servomotor/

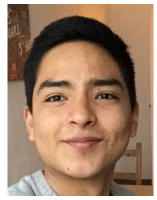
[13] Naga, K., Ananth, S., Rakesh, V., & Visweswarao, P. K. (n.d.). DESIGN AND SELECTING THE PROPER CONVEYOR-BELT. International Journal of Advanced Engineering Technology.

[14] Constantin, Daniel Lupoae, Marin Baciu, Catalin & Ilie, Buliga. (2015). Forward Kinematic Analysis of an Industrial Robot.

BIOGRAPHY SECTION



Francisco García: Student of Mechatronics Engineering in Universidad Internacional del Ecuador. Actually is coursing the seventh semester of that carrer. He made his previous studies in Colegio Cardenal Spellman. He has participated in numerous courses and webinars about different themes that involve robotics, electronics and control systems.



Ricardo Yepez: Student of Mechatronics Engineering in Universidad Internacional del Ecuador. Has made previous studies in physics in Germany for 2 years before enrolling in mechatronics engineering. In the actual time he is coursing the seventh semester of the career in Universidad Internacional del Ecuador.