UNDERGRADUATE THESIS:

Industrial Electronics and Automation Engineering (163)

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DESIGN AND DEVELOPMENT OF A MANIPULATOR ROBOTIC PROTOTYPE

Author: Sánchez Clemente, Sergio Andrés

Director: Armesto Ángel, Leopoldo

Codirector: Girbés Juan, Vicent

School: Escuela Técnica Superior de Ingeniería del diseño (ETSID), june 2021

Escuela Técnica Superior de Ingeniería del Diseño

“I dedicate this work to a couple of beautiful people that couldn’t see this project finished and to another beautiful person that would have enjoyed watching it started.”

Content table

Figures' index

Tables' index

Definitions

Robotics

It is a multidisciplinary field in charge of building robots: (re)programmable mechatronic devices able to perform different tasks with some (or none) level of intelligence and (or not) reacting to sensor information to actuate over the surrounding environment.

Sensor

When applied to electronics, device able to detect a physical magnitude and translate it to an electrical signal that can be later processed and used in order to produce an output.

Motor

Electromechanical transducer that transforms incoming electrical energy into mechanical energy (normally in the form of a rotating shaft). There are many kinds of motors, either with incorporated control circuits or without them.

µC (Microcontroller)

Programmable IC (Integrated Circuit) with a set of modules that perform a series of tasks individually to globally realize specific functions written on an uploaded software. They are used as a brain for different 'intelligent' projects having sensor monitoring and/or actuator writing objectives.

Mechanism

A mechanism, inside engineering semantic field, is a set of mechanical pieces that work together in order to produce a determinate mechanical output.

1. MEMORY REPORT

1.1. Introduction

1.1.1. PROJECT'S SCOPE

The main goal of this project is the design and practical development of a robot arm capable of interacting with the environment.

Project's completion shall be considered when, after design and construction stages, such device is able to move smoothly on multiple axes under the command of different sources of input and is able to grasp different objects inside its design boundaries.

1.1.2. PROJECT'S MOTIVATION

The main purpose of this device is to serve as a robotic platform with multiple sources of input able to realize, after modifications on the effector, different general or specific object manipulation tasks, with a price as low as possible.

The need for this type of platform comes from analyzing the different options available on the market and understanding that some of them have low functionality-price ratios and a very closed system, being difficult to modify them outside manufacturer's parameters.

1.1.3. PROJECT'S METHODOLOGY & GOALS

Using a correct methodology when designing any kind of project is vital, since good organization is key for success and good comprehensibility of the process.

This project's development process starts with the definition of the generalities for the arm, focusing on structural overall definition without initially going into particularities. Followingly, after a general design is generated, calculations shall be done for the overall links and robot's structure (and the motors involved in its movement). Then, a more detailed design shall be made by means of a 3D modelling software. Finally, construction shall begin.

Structural definition shall be done by means of functionality and simplicity. First, overall link diagrams are made in order to clarify what the parts of the robot shall be. This shall allow posterior calculations for both dynamics and kinematics to be done more easily. It has to be a good initial point for detailed design.

The next step, as stated before, is the structural calculations for the several links and joints. This is a way to estimate the torque needed by each joint's motor per unit of length and mass and also how robot's kinematic and dynamic models can be obtained.

According to the general design decided for the robot and according to the parameters estimated on the calculations section, the use of a 3D modelling software shall be necessary to create a more specific model, containing all mechanisms in detail. This is the design that shall be used for device's manufacturing.

After the manufacturing of the device and after the necessary mechanical and electronical corrections, the robot shall be tested in several ways electronically and mechanically to ensure every subsystem works according to plan, and it shall finally be put to test with its full functionality.

1.2. State of art

1.2.1. INTRODUCTION

Robotics is a multidisciplinary field of knowledge that combines other disciplines such as Mechanics, Electronics, Computer Science, Physics, etc. Due to this, there is a wide range of mathematics involved in performing the necessary operations in the development of a robot.

Robots were first introduced as an idea in the satiric novel *Rossum's Universal Robots* (Karel Capek, 1921). The word *"robota"* is the origin of the term, being considered to descend either from Slavic languages meaning "slave" or from Czech meaning "labor".

Robots are nowadays considered to be mechatronic devices able to be reprogrammed and reconfigured in order to perform a variety of tasks. They are involved in increasingly complex tasks, either aiding or even substituting humans in performing them.

In this section, the mathematical tools that shall be used for the development of the several sections are explained. Then, some examples of already existing robots shall be presented. Finally, the regulations on the topic shall be described.

1.2.2. THEORY

This subsection aims at explaining the different tools (equations, ways of proceeding, etc.) that are needed for the mathematical modelling of this robotic device in a cartesian 3D space. This modelling includes kinematic (speeds and positions) and dynamic (torque) limitations as well as the procedure to make the robot "understand" its position in the space and how to get to different configurations by itself.

Joints and links

Joints are considered to be the articulated parts of a robot that enable it to move with a certain degree of freedom. There are many types of such parts, depending on their mobility and type of movement, but for the sake of simplicity on this project only the basic dichotomy "revolute" and "prismatic" joints shall be presented.

Revolute joints are 1 DoF (Degree of Freedom) joints which allow the part to which they are connected to orbit around it. Prismatic joints are 1 DoF linear joints that do not rotate around their movement axes.

The sum of many different joints of the several kinds allows the machine to have greater degrees of freedom. Commonly, these devices combine articulations in such a way that allows them to be positioned at any point in a 3D space (within system's reach) and oriented at any angle (within system's capabilities), providing a total of 6 DoF (3 DoF for space positioning and 3 DoF for space orientation). Some of these robots have more than 6 DoF, resulting this in redundant capabilities, which is, the ability to position and orientate its end-effector tool in many different configurations and still be able to reach the goal target.

Links are the parts of a machine connected and enabled to handle movement and forces on such system. They can have an infinite number of shapes and sizes and can be assumed to have or not mass and thus a center of gravity (it shall depend on the analysis to be performed at each moment and such study's necessities).

Movement transmission

The movement generated by an actuator device on a determinate input can be translated into other types of motion by means of a determinate mechanism. This means that a rotary motion can be translated into a linear one and vice versa. Such motion type translation is regulated by the following equation:

$$\omega = \frac{v}{r}$$

This equation presents three variables, being "ω" the angular speed of the system, "r" the radius of the rotating point respect to its center of rotation and "v" the linear speed that results from the other two factors.

The torque provided by an actuator shows the capability to transmit a force given a certain radius of transmission. This relationship is provided as a vectorial product between such distance and its related force when taking into account a 3D space:

$$T = r \times F$$

In this expression, "T" stands for torque, "r" for the transmission radius and "F" for the force transmitted.

When considering the output of a mechanical system given a certain input provided by an actuator, the characteristics of such can be amplified or diminished in certain ways according to the needs of the system. The speed of the system and the torque that can be applied can be modified by using reductive transmissions. These transmissions can be made by means of gears in direct contact or with belts.

If it is desired that the output of a system is faster than the input, such input needs to be attached to a smaller wheel. However, this shall in turn cause that the torque provided is smaller, since both angular speed and torque on these transmissions are inversely proportional. In the opposite case, the resulting angular speed shall be lower and therefore the torque shall be greater. This behavior is described by the following basic relationship equation:

$$R = \frac{\omega_{IN}}{\omega_{OUT}} = \frac{d_{OUT}}{d_{IN}} = \frac{T_{OUT}}{T_{IN}}$$

On this equation, "ω" represents the angular speed of the driver ("IN") or the driven gear ("OUT"), "d" represents the diameter of the pitch circle (contact circular surface in a gear) of the driver and driven gears, and "T" represents the torque provided by the driver gear or outputted by the driven gear. It can be seen how such relationship is inversely proportional, as previously explained.

Robot basic kinematics

A point at a 3D space can be described by 3 cartesian coordinates referring to the distance in a certain unit to the center of the world's origin reference frame. Such point has a structure like this:

$$P = (xo, yo, zo)$$

These points can also be the origin of other reference frames. These reference frames can be placed and oriented respect to original ones by means of applying several different transformations.

Rotation matrices

A rotation matrix is a 3x3 matrix that allows, given an input vector, its rotation in 3D space. Rotations around X, Y and Z axes take different forms:

$$R_X(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

$$R_Y(\alpha) = \begin{pmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix}$$

$$R_Z(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Every time these rotation matrices are multiplied by a vector (or reference frame), a mobile vector (or reference frame) is generated. When it is desired to concatenate these operations, the order of the product shall determine if the successive rotations are applied respect to the original reference frame (OXYZ) or respect to the recently created new one (OUVW). Such order applied to the resulting rotation matrix when applying successive rotations can be seen below:

$$R_{i+1} = R_{basic} \cdot R_i$$

$$R_{i+1} = R_i \cdot R_{basic}$$

The first one represents successive rotations respect to OXYZ and the second one represents successive rotations respect to OUVW.

Translation matrices

These are matrices that allow the translation of a point in 3D space respect to an original reference frame in each of its axes (X, Y and Z):

$$t = \begin{pmatrix} t_X \\ t_Y \\ t_Z \end{pmatrix}$$

Each of the components represents the translation on each axis.

Transformation matrices

When it is desired to translate and rotate a reference frame, a transformation matrix (T(R,t)) is applied. These are composed after the initial translation of the origin point of such frame and its posterior rotation and also include perspective and scale factors useful when calculating computer graphics transformations:

$$T(R,t) = \begin{pmatrix} Rotation_{3X3} & Translation_{3X1} \\ Perspective_{1X3} & Scaling_{1X1} \end{pmatrix}$$

As in this matter the perspective and scaling components are not used, the first one shall have value 0 and the second one shall have value 1. This can be seen below:

$$T(R,t) = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

$$T(R,t) = \begin{pmatrix} r11 & r12 & r13 & tx \\ r21 & r22 & r23 & ty \\ r31 & r32 & r33 & tz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Such transformations follow the same order rules applied to rotation matrices when concatenating them:

$$T_{i+1} = T_{basic} \cdot T_i$$

$$T_{i+1} = T_i \cdot T_{basic}$$

On the previous equations, the first one represents transformations respect to the OXYZ reference frame while the second one stands for transformations related to the OUVW mobile reference frame.

Denavit-Hartenberg convention and parameters

A robot is constituted by links and joints of several kinds. Each of them can be mathematically reached from an initial reference frame located at the base by means of successive transformations.

The Denavit-Hartenberg convention establishes a series of rules by which reference frames must be placed on each of robot's joints such that they can be reached one respect to the previous one by means of: one rotation respect to the Z axis, one translation on the Z axis, one translation on the X axis and one rotation respect to the X axis. The composition of these transformations provides the Denavit-Hartenberg transformation matrices between successive reference frames:

$$ {}^{i-1}_{i}A(qi) = T\big(Rz(\theta_i)\big) \cdot T\big(tz(d_i)\big) \cdot T\big(tx(a_i)\big) \cdot T(Rx(\alpha_i)) $$

Each of the parameters for every "i" transformation is known as a DH parameter and allows to quantify each of the rotations and displacements for each of the reference frame transformations.

The composition of such DH transformation matrices allows the placement of each of the reference frames from the original ones up to the end-effector:

$$ {}^{i}_{n}A(q1, q2, \dots, qn) = {}^{0}_{1}A(q1) \cdot {}^{1}_{2}A(q2) \cdot \dots \cdot {}^{n-1}_{n}A(qn) $$

These transformations are useful when modelling robots via software, so the DH parameters shall be provided to the user in case they needed them.

Direct kinematics and inverse kinematics basics

Direct kinematics are a way in which, given a series of joint values, the configuration at which a robot is placed and oriented can be found. Inverse kinematics, on the other hand, is the way in which, given a certain configuration, it can be computed what the joint values should be for the robot to reach that target.

The fastest computational way in which this can be done is by means of trigonometry. Given a non-rectangle triangle with vertices A, B and C, and sides a, b and c, being A opposite to a, B opposite to b and C opposite to c, the following theorems are presented:

- Sine theorem:

$$\frac{a}{\sin(A)} = \frac{b}{\sin(B)} = \frac{c}{\sin(C)}$$

- Cosine theorem:

$$a^2 = b^2 + c^2 - 2bc \cdot \cos(A)$$
$$b^2 = a^2 + c^2 - 2ac \cdot \cos(B)$$
$$c^2 = a^2 + b^2 - 2ab \cdot \cos(C)$$

By means of these equations several relationships between links and joints can be found to develop the several kinematic models.

IK model strategy

The main proceeding way to develop the Inverse Kinematic model is to compute, given the end-effector position and orientation, the point at which the wrist is and then, to compute the rest of the angles.

Coordinated movements

When performing the different movements, all of robot's joints must be able to perform at the same time. Besides, each of joint's movements must be planned in such a way that avoids sudden movements that, due to inertia, could potentially damage or destroy the device. For this, cubic trajectories are defined for each of the joints and are executed at the same time on all of them. Such trajectories have the following format:

$$qtraj(i) = a(i) \cdot t^3 + b(i) \cdot t^2 + c(i) \cdot t + d(i)$$

The equations that allow to calculate each of the parameters (a, b, c and d) for each of the joints (i) and for each of the instants (t) can be seen below:

$$a(i) = -2 \cdot \left(\frac{qT(i) - q0(i)}{T^3}\right)$$

$$b(i) = 3 \cdot \left(\frac{qT(i) - q0(i)}{T^2}\right)$$

$$c(i) = 0$$

$$d(i) = q0(i)$$

On the previous equations qT(i) is the final joint value on a movement for each of the joints (i) and q0(i) represents the initial joint value. The variable T stands for the time along which the movement shall last.

1.2.3. SIMILAR EXISTING DEVICES

Regarding robotic devices aimed at a common consumer market niche, there exist a variety of options on the affordable price range.

Arduino Robot Tinkerkit Braccio T050000

This is a 5 DoF Arduino microcontroller-based robot arm. It is a relatively small device with an operating reach of 80 cm and a maximum load capacity at a distance of 32 cm of 150 grams.

It is powered through a shield, which draws 1.1 A from motor 1 to motor 4 connectors and 750 mA from motor 5 to motor 6 connectors. It has an operating level of 5 V.

This robot is prepared for education and free experimentation, with the addition of sensors and actuators with no soldering required.

Sainsmart 6-axis desktop robotic arm

This is a 6 DoF easily programmable robot arm. It has a palletizing robot structure, having a parallel mechanism on its back that can be adjusted via a motor, keeping the upper structure on the same angle respect to ground unless specifically modified.

Motors are MG966R servomotors for the main joints (11 kg·cm torque) and SG90 (1.6 kg·cm) servomotors for the wrist's 2-axis rotation capabilities. The controller to be used can be determined at consumer's please, since these motors are compatible with any of them that may have PWM signal generation capabilities.

This robot's main use is STEM (Science, Technology, Engineering and Mathematics) education, allowing to learn more about this kind of systems, their coding and their overall behaviour.

Its price ranges on the 115-150 € interval.

Niryo one 6-axis collaborative robot arm

This device is a collaborative 6 DoF robot arm. It is compatible with many different systems, including ROS, Python, C++, Matlab, etc.

It has a relatively low size and a short operating area (44 cm reach), useful load capacity of 300 g and good repeatability (1 mm deviation). It also has Ethernet, Wi-Fi, Bluetooth and USB connection capabilities. It is built in aluminum and 3D printed PLA.

It is considered to be a collaborative robot because it is able to detect collisions via a magnetic sensor placed inside the motor. Due to this, it can work with humans safely in an industrial environment and can also be used for STEM education.

Its price is found on the 2200-2300 € range.

1.2.4. STANDARDS AND REGULATIONS ON THE TOPIC

This subsection shall analyze the regulations that there exist in Spanish law system and which the project shall need to meet.

Object

These regulations explained below shall establish a framework that electrical installations connected to power supplies in the range of low voltage shall need to comply with. The purpose of this is to preserve security for people and goods, ensure the proper working of such installations, and contribute to their technical reliability and economic efficiency.

This law shall apply to new installations and their modifications and ampliations. Excluded are specific equipment for mining, traction material, automobiles, ships, airships, communication systems and military systems and other devices subjected to specific regulation.

Definitions

It is understood as an electrical installation every set of devices and associated circuits aimed at production, conversion, transformation, transmission, distribution or utilization of electrical energy.

This device shall enter in the range of very low tension for being a DC and lower than 75 V product.

Generalities

In case the installation may produce perturbations over telecommunications, energy distribution network or receptors, it shall be provided with the appropriate protection devices.

Equipment used in the installation shall be used in the way and with the finality for which it was manufactured. Technical requirements of these devices shall be verified by competent organisms of the Autonomous Communities.

Low voltage cables used are regulated on UNE 21027-9 and UNE-EN 50525-2-31.

Motor and receptors installation

Power connectors feeding only one motor must be rated for an intensity of at least 125 % the intensity through the motor when at full load.

Power connectors feeding several motors must be rated for an intensity not inferior to 125 % the one of the most powerful motor at full load plus the intensity at full load of the rest.

Power connectors feeding motors and other receptors must be rated for the overall intensity of all receptors plus the one required by the motors, calculated as it has been stated in the previous paragraph.

Motors must have an automatic supply-cutting device that acts when there is a lack of tension. The chance of an incident in case a spontaneous turning on of the motor occurs must be forced excluded.

1.3. Needs' study

1.3.1. INTRODUCTION

This section aims at explaining device's main features according to what a user can expect and how the designer shall apply those expectations, as well as a general specifications review.

Several points need to be considered as key general fields to take into account when having a general idea of the design, being those: functionality (speed, torque, range, accuracy and repeatability, operation comfort and maintenance ease), safety (mechanical safety, electrical safety and robot safety), economy (budget expectations) and cosmetics (general theme of the robot).

1.3.2. USER SPECIFICATIONS REVIEW

This subsection studies what features a hypothetical user would be keen to have at their disposal when using the device regarding different topics:

1. Functionality:

   a. Regarding speed:

      A1. User should be able to use the device for a wide range of alternative tasks, for which they shall need it to be versatile regarding the speed of operation, that is, the speed at which the robot receives and treats the info as well as the speed at which it is able to transform that data into a tangible action.

      A2. For some functions, the robot can be required to emulate animations proper to living beings, so a rapid response to change moving direction is necessary.

b. Regarding torque:

B1. Torque output of the robot needs to be as high as possible without compromising speed of operation.

B2. Torque output must be able to produce enough end-effector force to hold at least 0.2 kg when fully extended.

c. Regarding range of motion:

C1. Robot's working space shall be sufficiently large to be able to fulfill significantly useful pick & place operations not interfering on non-interesting objects accidentally.

C2. Robot's working space shall include decent heights to be able to exchange objects between various levels.

C3. Robot shall have at least 5 DoF, including at least 1 exclusively for the end-effector.

d. Regarding accuracy and repeatability:

D1. Robot's end-effector accuracy must be in the range of 2-3 mm.

D2. Robot's repeatability shall be high enough to repeat end-effector positions with a max deviation of 5 mm from the first to the last repetition.

e. Regarding operation comfort:

E1. Robot shall have different operation modes including computer direct control and position recording (with the mouse), radio control, mobile Bluetooth app interface, external µC control (emulating a production line) and own motion planning actions.

E2. Robot's program modification shall be eased by means of creating specific functions for its movement, including inverse kinematics, trajectories, etc.

E3. At least one of the communication modules shall be exchangeable (Bluetooth exchangeable with BLE).

f. Regarding maintenance ease:

F1. In case of piece failure by wear or excess of working stress (payload out of robot design parameters), it shall be able to be replaced easily and as modularly as possible.

F2. In case of motor failure by excess of working stress (payload out of robot design parameters), they shall be able to be extracted and replaced easily and rapidly.

2. Safety:

a. Regarding mechanical safety:

A1. Robot parts shall not have enough mass or acceleration to damage the user severely.

A2. In the case of uploading bad software and the robot going out of control, the most dangerous (powerful) motors shall be disabled quickly by means of a mechanical switch that the user shall be in range of pressing to avoid personal or material damage.

A3. The user shall be able to include other ways to measure the danger externally and then send a signal either manually or automatically that allows the robot to self-deactivate.

b. Regarding electrical safety:

B1. The user shall always be separated from electrical connections.

B2. All connections the user may want to realize to the robot shall always be done by means of plugs, never by means of direct wiring. The robot shall be prepared for these connections to be done.

c. Regarding robot safety:

C1. A mechanical switch shall be placed easily accessible for the user in case of something going wrong so that motors are physically separated from their power source and robot cannot self-damage.

C2. Robot's library to be included in the program shall have into consideration impossible configurations in order to not push its boundaries out of their limit, therefore breaking its pieces or damaging its motors.

3. Economy:

   a. Regarding budget expectations:

      A1. The price for developing a prototype shall be, taking into account only manufacturing, as below as possible to 550 € in cost. Overall price must be lower than 650 € (including VAT).

      A2. Manufacturing methods shall be adapted to obtaining the most quality result out of the least monetary investment.

      A3. In case of a future development of a business plan including such device manufacturing techniques previously mentioned shall be updated to fit with the new requirements and expectations.

4. Cosmetics:

   a. Regarding robot's thematic appearance:

      A1. The general appearance of the robot shall be as 'futuristic' and 'stylish' as possible. This can be considered appealing by a potential customer.

      A2. Thematic to be used on the robot shall be similar to some science fiction arms used in popular culture resources (movies, videogames, etc.).

      A3. Device's cosmetic appearance parts should not interfere with its performance requirements specified on this section in any case.

1.3.3. DESIGNER SPECIFICATIONS REVIEW

In order to perform accordingly to user expectations, some specifications must be made by the designer in order to simplify further designs and specify certain topics more deeply. The same 4 points shall be considered:

1. Functionality:

    a. In order to reach a versatile behavior, motor selection shall be key, keeping a high speed and torque while maintaining a low weight and gear wear.

    b. The chosen motors need to have a communication interface as simple as possible. If possible, this shall be PWM communication with a µC.

    c. Motors shall have as high rotation angle span as possible.

    d. Global robot's weight shall not exceed 5 kg.

    e. End-effector shall be able to hold different tools and adapt to new attachments. This feature shall serve to take bigger-sized objects by changing this part.

    f. Every mechanical subsystem shall be divided into different easily replaceable pieces so that in case of failure, the monetary active used to fabricate the substitution for the broken piece is the minimum possible.

    g. A frontal light shall be designed and placed to improve lighting conditions in case of possible computer vision related projects using the arm (with external hardware via one of the communication channels).

2. Safety:

    a. Wires with one of their tips open shall not be exposed, so that accidental ground connection with the user results impossible.

    b. Power lines shall be plugged into their corresponding sockets, therefore not touching any wire connection, but only the insulated end cap.

3. Economy:

    a. When selecting the different materials and components to use, there shall be an evaluation considering 3 main factors: functionality, price and simplicity of use. The ponderation of each factor shall depend on each case.

4. Cosmetics:

    a. The robot shall be painted on the manufacturing process with a color scheme decided during design phase.

1.3.4. GENERAL SPECIFICATIONS REVIEW

Final considerations regarding the general aspects of the robot shall include 2 main points, these being:

1. Safety guidelines:

    a. Beware the arm is an unaware machine and can produce damage to a physical person. Therefore, the user must stay conscious at all times about robot's position at any given time during its operation.

b. During manual control of the arm while on operation via any of its modes, the user must be aware that they are handling a physical machine that can harm people or materials nearby.

c. During manual control of the arm and although motion is restricted to avoid self-destruction for certain ranges of movements, the user must be aware not to damage the machine and use soft short-range displacements for each axis.

d. When changing a communication module, the user must be careful not to drop any liquid into any gap there may exist to allow this exchange.

2. Usage requirements and capabilities:

a. In order to use the full functionality of the robot, the user shall need to have at their disposal a smartphone (for app control), a computer (to program the robot) and/or devices with radio control to interface with the device.

b. Further development upon robot's capabilities shall be possible and the device is specifically thought for this purpose. It shall be achieved by either developing new algorithms for the machine, using external devices for wireless communication or using external µCs to transmit sensor or algorithm results' data.

1.4. Alternative solutions

1.4.1. INTRODUCTION

In this section, the general decisions made on the different modules and components that shall compose the robot shall be developed following this methodology: firstly, a description of what is needed shall be presented; then, a series of options shall be presented and finally, the best solution shall be presented with its selection duly justified.

The structure for the different sections shall follow the logical design methodology explained in section 1.1.3. ("PROJECT'S METHODOLOGY"). To start, a breakdown of the robot's general structure with components decisions to be made can be seen below:

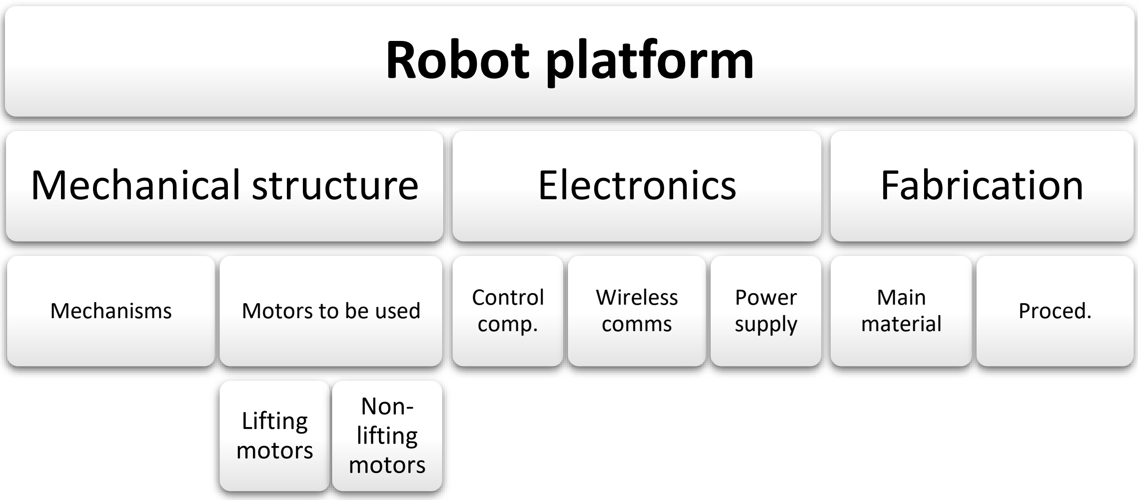


Figure 1. *Robot platform block diagram structure with general subsystems*

Source: Own

All sections above are interdependent, so decisions on them cannot be taken independently. However, prioritization has been made according to what is stablished on section 1.1.3. ("PROJECT'S METHODOLOGY"), starting by defining what the overall shape of the robot shall be, then continuing with the mechanisms to be used, and, according to these, the motors to be used, the supply they shall need, the circuit for them, etc.

1.4.2. MECHANICAL STRUCTURE

The **mechanical structure** is formed by robot's body and is responsible for transforming actuator-applied mechanical energy into its own movement. It comprises both the overall structure as well as the various mechanisms in charge of moving each of the joints.

1.4.2.1. Overall shape

The first step when deciding what the components for the robot shall be is determining how the robot shall be in general terms, that is, its type, overall structure and applications. Later, the specific mechanisms and motors to be used shall be decided.

Mobility of the robot

When considering options on the mobility of the robot to build, it is necessary to establish several points as **selection criteria**:

a. Operation area needs to be as big as possible.
b. Functionality-cost ratio has to be as high as possible.
c. The robot must be able to work not interfering with non-interesting objects on a desktop.

The first criterion is deduced from the user specification 1-C1 on section 1.3.2. and obeys the need to create the most versatile robot possible. The second criterion comes from user specification 3-A2 on section 1.3.2. and complies with 3-A1 on that same section, with the purpose of saving costs during development. The third and last criterion satisfies section 1-C1 on section 1.3.2 and constrains the range of operation to fit on a desktop and not interfere with non-targeted objects the position of which should not be modified.

The available **options** for how the robot can be regarding its mobility are 3:

a. A fixed robot.
b. A linearly displacing robot.

c. A completely free-moving robot.

The first option would consist on the robot having a fixed base, therefore not able to move around the environment, having as workspace area every point reachable with a certain configuration by only rotating its base.

The second option would consist on the robot being attached to a moving base that could slide on rails along a determined path. This path would require a structure surrounding it in order to hold the rails and the needed traction provided by a motor.

The third and last option would be to make the robot completely mobile by using some sort of mechanism (common wheels, castor wheels, etc.). Displacement space would be a 2D plane, along which the robot would be able to freely position and rotate itself.

Analyzing the three different options by using the selection criteria, it can be observed for every option that:

- The first option is the most economical since it only requires the rotating motor and the base to work (regarding mobility only). However, it has a very small operating area compared to the other two and would not drag objects accidentally as easily as it may happen with the other two options.

- The second option is not either the cheapest or the most expensive out of the three, since it does not need a full automobile base design with all of the costs that it requires, but it is not as simple as just a rotating base. It has the most balanced displacement area, since it could displace, unlike first option, but only in 1D (along a linear path). Given an aerial structure, this would not interfere on non-targeted objects in the surroundings.

- The third option is the most expensive one, since it requires many more subsystems to design, control and feed. However, it would give the robot the most freedom out of the three, although it would be very difficult to sensitize up to a necessary degree to not collide with objects by chance.

A table **summarizing** the points made above can be appreciated below:

	CRITERIA		
OPTION	Operation area	Functionality/cost	Adaption to surface
Fixed base	LOW	MEDIUM	MEDIUM
Linear disp. base	MEDIUM	MEDIUM	HIGH
2D-free mobile base	HIGH	MEDIUM	LOW

Table 1. *Mobility options comparison*

Source: Own

Given the comparison, the realization is that the second option is the most versatile and fulfills best the selected criteria. Therefore, the adopted solution shall be adapting a **linearly displacement robot that shall follow an aerial path structure** to not interfere with undesired objects and an inverted structure to reach ground level.

Type of robot

Once it has been defined that the robot shall move along a railed path, the next step is defining which kind of robot shall be used according to what is stated on the specifications and to several selection criteria specified below. The selection of the robot is one of the most important decisions, since it affects all of its posteriorly defined characteristics, including its overall dynamics and mechanisms, functionality, programming, price, etc.

In order to select a robot type that best fits what is described on section 1.3. ("Needs'study"), the following **selection criteria** have been sorted out:

a. The robot shall have at least 5 DoF.
b. The robot shall have at least 1 for the end-effector.
c. Functionality-cost ratio shall be as high as possible.
d. Manufacturing simplicity.

The first criterion and second criteria come from the user specification 1-C3 on section 1.3.2., and it provides a restriction with the goal of making object manipulation versatile and making the robot able to change not only an object's position in 3-D space, but also its orientation in 2 different planes. The third criterion is considered for the same reasons as it has been considered when deciding what the mobility of the robot had to be. The fourth criterion is taken into account in order to reduce costs and manufacturing time, since a project of this kind needs to be the simplest possible accomplishing all of the conditions.

When evaluating possible **options** for the type of robot, out of the many hybrids there exist on the market, the most general ones need to be considered:

a. SCARA robot.
b. Cartesian robot.
c. Anthropomorphic robot.
d. Delta robot.
e. Collaborative robot.

A SCARA robot is a 4 DoF robot (2 DoF for XY movement, 1 DoF for prismatic Z movement and 1 DoF for end-effector orientation) with an open kinematic chain which is commonly used for component insertion.

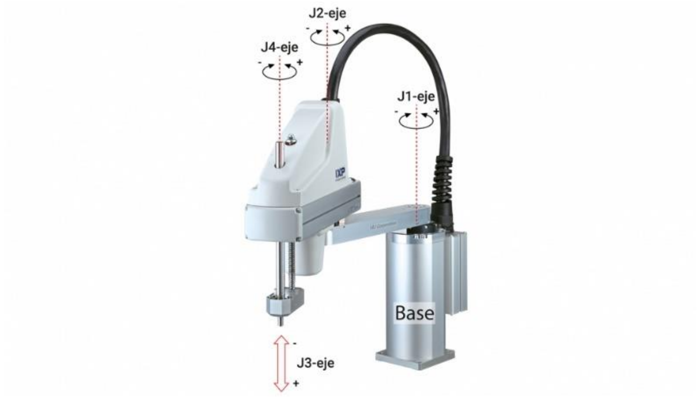


Figure 2. *SCARA Robot*

Source: https://img.interempresas.net/fotos/2251366.jpeg

A cartesian robot is a 3 DoF robot (one DoF for each XYZ individual axis) with an open kinematic chain and typically built with 3 prismatic joints. An example for this are common 3D printers.

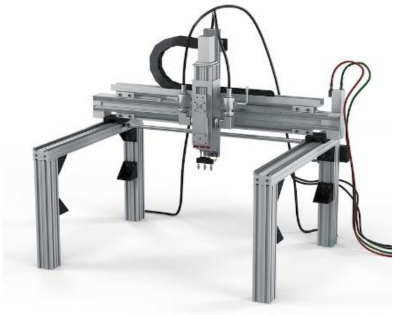


Figure 3. *Cartesian Robot*

Source: https://roboticsandautomationnews.com/wp-content/uploads/2020/04/cartesian-robot-image-copy.jpg

An anthropomorphic robot arm is a generally 6 DoF (3 DoF for end-effector position and 3 DoF for end-effector orientation) with an open kinematic chain used in industry for many different applications due to its versatility.

Figure 4. *Anthropomorphic Robot*

Source:
https://www.comau.com/SiteCollectionImages/our_competences/robotics/robot_team/selector/racer3.jpg

A delta robot is a kind of parallel robot with 3 DoF (one for each XYZ individual axis position of the end-effector) with 3 parallelogram links connected to 3 revolute joints.

Figure 5. *Delta robot*

Source: https://res.cloudinary.com/engineering-com/image/upload/w_350,c_limit,q_auto,f_auto/YF003N_s_nn8toi.jpg

Collaborative robots can be a type of anthropomorphic robot arms, with the difference that they incorporate many more sensors that make them more reactive to the environment, being a suitable option in case of a self-contained device for safe human interaction.

Figure 6. *Collaborative robot*

Source: https://www.canonicalrobots.com/images/virtuemart/product/Elfin5.19.jpg

Analyzing each of the options according to the selected criteria, the evaluation results for each of the options can be seen below:

- The first option (SCARA) does not directly comply with the first criterion on its basic form, having 4 DoF instead of 5 DoF but when included the rail displacement, the condition is accomplished. It also accomplishes the second criterion, having a specific DoF for the end-effector. It also has a good functionality/cost ratio, since this robot does not require very expensive mechanisms, and has a very simple-to-manufacture structure.

- The second option (Cartesian) does not comply with the 5 DoF condition and the use of a displacement rail would be redundant with the movement of one of its axes. Besides, it does not have 1 DoF for the end-effector, for which it would need it to be an additional mechanism. Due to the last two points and even though it is not an expensive to manufacture robot, the functionality/cost ratio is relatively low. However, it is a very simple device to manufacture.

- The third option (Anthropomorphic) complies exceedingly the first and the second criteria. Therefore, it has a great functionality factor, but also a little higher cost, so the functionality/cost ratio is lightly above 1. Regarding the last point, it is important to state that it is a more complex robot to manufacture.

- The fourth option (Delta) has 1 DoF less than what is required when adding the rails, and has no freedom regarding end-effector orientation, so to be able to adapt it to requirements, the same operation applied for the cartesian robot (external mechanism added) would be needed to comply. Due to this, the functionality/cost factor is low. It is a very easy-to-assemble robot.

- The fifth option (Collaborative) responds to first and second criteria as well as an anthropomorphic robot. It also has a very good functionality/cost factor. However, it is very complex to build, since it requires many sensors to increase awareness over the environment.

The following table **summarizes** the points made about each of the options:

	CRITERIA			
OPTION	5 DoF	1 DoF end ef.	Fun/cost ratio	Simplicity
SCARA	Adapt.	YES	MEDIUM/HIGH	MEDIUM/HIGH
Cartesian	NO	Adapt.	LOW	HIGH
Anthropomorphic	YES +	YES +	MEDIUM/HIGH	LOW/MEDIUM
Delta	NO	Adapt.	LOW	HIGH
Collab.	YES +	YES +	MEDIUM/HIGH	LOW

Table 2. *Type of robot options comparison*

Source: Own

NOTE: The tag *Adaptable* references in the case of the 5 DoF condition that the device alone added to the rails does accomplish the condition. Applied to the 1 DoF end-effector condition it means that the device would need a motorized tool or adaption to fulfill the requirement.

NOTE: The tag *YES +* implies that the particular option that it refers to complies exceedingly with the condition.

The type of robot that complies the best with all of the conditions is the **Anthropomorphic robot**, since it has the best freedom of movement (commonly 6 DoF) and a very good functionality/cost factor for this particular project's objective, which is quite an open range of tasks. In order to increase the manufacturing simplicity of the robot and still comply with the specifications, the overall arm shall be made (including the rails) to match the 5 DoF required (not 6), so that the end-effector complexity is significantly decreased.

1.4.2.2. Mechanisms

The overall motorized joints' structure, in order to accomplish with the parameters given, shall be distributed in the following way: 1 prismatic joint for displacement along the linearly railed path, 1 revolution joint for arm rotation, 1 revolution joint for arm's link equivalent to a human shoulder, 1 revolution joint for arm's equivalent to human's elbow and 1 revolution joint for end-effector's orientation (equivalent to human's wrist up and down movement).

In order to perform the motion accordingly to this description, it is necessary to define how motor's movement shall be trespassed to the different joints. Each joint has different weightlifting, speed and precision requirements, so the way motion is transferred to every part and therefore the different mechanisms for each of the cases need to be studied separately.

Joint 1: Lateral displacement joint

This movement, as explained before, needs to be performed linearly, for which a prismatic action is needed. In order to constrain this motion, certain **selection criteria**, with points similar to others expressed before, need to be considered:

a. Action smoothness and speed
b. Cost
c. Simplicity

The first criterion attends the fact that motion on the robot shall have the least peaks possible to position it as accurately as possible and without wearing any of the parts involved due to continuous or sudden violent inertia moment changes. It is also important for certain movements to be as smooth as possible in order to not interact by accident with the environment.

The second criterion is necessary to restrain the plausible selection of very powerful actuation mechanisms that could be excessive for system's weight and general scope.

The third criterion is a very general one and has been established in order to keep posterior manufacturing at the lowest costs possible.

The several **options** that could be applied are the following ones:

a. Hydraulic actuator
b. Pneumatic actuator
c. Belt transmission with rotary motor
d. Linear electric actuator

Hydraulic actuators are widely used in industry for a very different set of functions, including powering construction heavy machines, actuating on elevators, etc. They consist of an empty metallic cylinder with a sliding component that, when hydraulic fluid flows in one direction, gets extended and when it flows in the opposite direction, it gets retracted. They require a powerful enough pump (it depends on the application), a hydraulic circuit with tubes, hydraulic liquid, and a set of tools to work on them.

Pneumatic actuators are very similar in their design to hydraulic actuators but work with air flow instead. They have faster motion speeds but less precision and holding force.

Using a transmission belt together with a motor that originates the motion is very common for linearly actuated joints. A correctly tightened belt is key for the proper working of the system, since otherwise slips can occur, resulting in positioning errors that could be fatal for the robot depending on the situation.

An electric linear actuator consists of a rotating motor with a mechanism that transforms this motion into a linearly sliding one. They are very powerful depending on the reduction applied in the transformation mechanism.

Followingly, an **analysis** on how each option fits into the different criteria is performed:

- Hydraulic actuators can be considered to have a relatively smooth action, but they are slow compared to other types of actuators. Their cost is very high, due to the amount of components required for their proper working (pump, hydraulic system, etc.). Therefore, their simplicity is very low.

- Pneumatic actuators are very fast but not so smooth and they are very difficult to control due to the compressibility of the air they contain. Their cost suffers from the same problems a hydraulic system has, and although their simplicity is higher than that of the first option, it is nowhere near the one from other options.

- Belt transmissions offer a smooth action, while having a speed that depends on the actuator's speed and reduction used. Their cost is very low, since transmission belts are relatively cheap and depending on the motor chosen it can also be very affordable (relating such limit to the boundaries of this project, which can be seen on 3 a. on section 1.3.2.). Their simplicity also depends on the implemented system.

- Linear electric actuators have a very smooth action since, for every unit of distance covered, they have to go through a series of continuous points. Their speed depends on the reduction used, since it is inversely proportional to the torque produced. Regarding the cost, it can widely oscillate from a few euros to hundreds of them. They are the simplest option to apply since they are a self-contained device.

A **summary** of the points previously made can be seen followingly:

	CRITERIA			
OPTION	Smoothness	Speed	Cost	Simplicity
Hydraulic	MEDIUM/HIGH	LOW	HIGH	LOW
Pneumatic	LOW	HIGH	HIGH	LOW
Belt transmission	HIGH	HIGH	LOW	HIGH
Linear actuator	HIGH	MEDIUM	LOW/MEDIUM	HIGH

Table 3. *Joint 1 mechanism options comparison*

Source: Own

Out of the direct comparison between the different characteristics each option provides, the first two options are rapidly discarded since they are actuator mechanisms intended for clearly larger devices, with extremely high prices and too complicated systems for the scope of this project.

Between the resting two options, when comparing their smoothness, they can be both considered as almost equal. Their speed, however, is their first differentiating factor, since linear actuators can have speeds of some mm/s, but even the fastest ones are slower than belt transmissions with the right motor selection.

When considering the cost for both options, it is important to remark that although both of them can be made specifically for the project designing the correct gears and picking the right motors, belt transmission would still be slightly cheaper, since they require lesser moving pieces outside belt, tractor gear and a passive and opposite gear.

Regarding the simplicity of both actuators, it is necessary to remark that although in principle it is equal, in practicality, adjusting the belt to the system is faintly more difficult than just building or installing the correct linear actuator.

All in all, taking into consideration the several conclusions about the actuators, it has been decided that the adjusting complexity of the **belt transmission** shall be taken as a challenge since this system provides commonly higher speeds to linear actuators, while keeping all the other factors similar.

Joint 2: Robot rotation

This motorized joint is in charge of rotating the whole robot respect to its sliding base. This is a key movement as much as the sliding one is, since out of these two motions it depends that the whole robot is well positioned and does not have vibration issues.

Having considered the importance of this joint, the following **selection criteria** shall be established:

a. Reliability
b. Speed
c. Cost

The three main **options** considered for this drive are:

a. Direct drive
b. Gear drive
c. Belt drive

The first option consists on attaching the motor directly to the joint and controlling torque applied, having position measured by a sensor (encoders, potentiometers, etc.). This avoids friction energy losses as eliminating the impeding action of the gears.

The second option consists on having several gears attached between the motor drive and the joint itself.

The last alternative is to use a belt connecting the motor and the joint.

After alternatives and selection criteria presentation, an **analysis** can be done:

- Regarding reliability, direct drive offers as much of this quality as the proper motor can ensure, since no other factor or component intervenes between such actuator and the joint. This kind of drives are as fast as their motor, for the same reasons explained before, which is commonly relatively high. The cost of implementing this option is usually high, due to the electronics involved as well as the kind of motors used to obtain a good behavior.

- Gear drives have a good reliability if gears are well manufactured, since due to normal forces between the teeth, slipping results very unlikely. Speed for this kind of mechanisms oscillates according to the reduction implemented between the input and the output shafts when operating the joint, as well as it depends on the kind of motor used. Cost for these systems ranges from very affordable values depending on the material used even a few centimes, up to hundreds of euros if the material used has a higher quality.

- Belt drives are moderately reliable if their belt is well tuned since, in case of slip occurring, this could have fatal effects on the system. This joint’s speed can be as high as motor’s if reduction is

1 on 1. The cost of these transmissions is very low, since belts this application can require are really not expensive.

The points considered above can be **summarized** in the following table:

	CRITERIA		
OPTION	Reliability	Speed	Cost
Direct drive	HIGH	HIGH	HIGH
Gear drive	HIGH	MEDIUM/HIGH	LOW/MEDIUM
Belt drive	MEDIUM	MEDIUM/HIGH	LOW

Table 4. *Joint 2 mechanism options comparison*

Source: Own

Given the comparison above, it is important to remark that direct drive is part of an incipient field of development in robot dynamics, improving robot performance immensely, reaching important milestones further beyond common industrial robots' capabilities, referring to torque, speed and acceleration. Direct-drive motors are very expensive and would by themselves exceed this project's budget, so they are not an option.

Gear and belt drives are both very used mechanisms, with similar characteristics between them, excluding limitations such as belt breakdown tension or belt slipping. Final decision has been taken having in mind the similar possibilities of both systems, which are not intended to move great masses in any plane different to horizontal plane, as they shall only need to exceed inertia's effect.

The decided mechanism to be used at this joint is the **gear drive**, since even though gears of any kind offer friction and therefore wear issues, belt drives can offer more problems if not well adjusted or if submitted to great efforts during a period of time.

Joint 3: Robot's shoulder

This joint is in charge for carrying the most of torque, since it needs to be able to rotate the whole body of the device, for what it is necessary that, in full extension, the arm can be turned into any direction with no excessive speed decrease.

Further calculations and considerations into the torque topic regarding the different joints shall be studied later on the memory report in order to calculate the load motors present on this joint are able to withstand when the arm is fully extended.

To have the highest torque possible out of the same motors that shall be used for the majority of the joints (the same type of motor shall be used for as many joints as possible to keep costs as low as they can be), there shall be 2 motors involved in this shoulder's movement.

As what the type of mechanism should be, its characterization shall depend on the following **selection criteria**:

a. Reliability
b. Speed/Torque
c. Cost

These criteria are almost the same they were for the previous section. The first criterion regards the accuracy and repetitiveness that can be extracted out of each option, as well as the amount of wear that they shall undergo.

The second criterion aims at explaining which option is the fastest respect to the amount of torque they are able to output, considering an initial speed output of the motor used.

The third one complies with the same principle expressed on economics in section 1.3.2., which has been cornerstone for development of the different sections up until this one.

Regarding the **options**, the main three ones are:

a. Direct drive
b. Gear drive
c. Belt drive

All these three options have been discussed in the previous joint section.

Applying the different considered choosing directives, the same **analysis** can be made for each of the options.

It is also necessary to add that the direct drive is a very balanced option, since it allows high speeds while keeping high torques. A gear drive is the option capable of outputting the maximum torque out of the options and with a speed as high as the gear reduction set allows it to be. For the belt drive, it can be said that it may be the fastest type of mechanism depending on the kind of motor chosen, and if no belt reduction set is built, the one with the lowest torque.

A table with the analysis results can be appreciated below:

	CRITERIA			
OPTION	Reliability	Speed	Torque	Cost
Direct drive	HIGH	HIGH	MEDIUM/HIGH	HIGH
Gear drive	HIGH	MEDIUM/HIGH	HIGH	LOW/MEDIUM
Belt drive	MEDIUM	MEDIUM/HIGH	LOW	LOW

Table 5. *Joint 3 mechanism options comparison*

Source: Own

Direct drive is automatically excluded by means of price, since, as explained in the previous section, is well over the budget destinated for this device's creation.

Between belt drive and gear drive, the chosen option is gear drive, because it offers higher torque with greater reliability. This gear drive can be applied in several ways, being one of those with a separate motor and gear set relationship and being the other a servomotor by itself.

The main advantage of a gear drive set is that it can be fully designed and adapted for the exact reduction needed (or for a higher one in case of some minimum acceleration needs). In case of a broken piece, it can be easily replaced, since the mechanism shall be accessible. The disadvantage that comes directly from this premise is that it needs to be designed, taking some time to develop and therefore adding some cost to the project.

The most important advantage of using a servomotor is that it is a self-contained unit that already contains motor, gear reduction set and power/signal regulators for the system. This is very practical from the point of view of modular design (1-F1 on section 1.3.2.) since the whole servomotor can be just added to the joint and replaced if it is necessary. The main disadvantage this option offers is the fact that servomotors have a maximum torque that, as a proper unit with its own gears, cannot be modified. In the case of a need to modify their maximum torque, they would also require external gears, which would accomplish a redundant function (excessive costs).

Having considered the different advantages and disadvantages between these two options for the shoulder mechanism, it can be stated that leaning on the use of **servomotors** is a better alternative. This is mainly due to the fact that for every joint, the selected motor can be chosen to have the amount of needed torque plus a margin to have some weightlifting and acceleration capabilities, so no extra gears are needed. Servomotors also have an already-included control unit that make them able to reach the position (or speed, depending on the kind of motor) with no external sensors needed. All in all, they are a better option for the application they are intended to have on this device.

Joint 4: Robot's elbow

This fourth motorized joint is responsible for the movement of the forearm respect to the previous parts of the robot. It is the second motorized joint that needs to hold most of the weight, after the shoulder. This is due to the fact that it has to overcome the torque generated by both forearm's weight, wrist's weight and carried object's weight.

The **deciding factors** to intervene in the decision for this joint are:

a. Reliability
b. Speed
c. Kinematic ease
d. Cost

These factors have been considered multiple times for each joint's mechanism selection.

The first criterion responds to avoiding problems during operation, with undesirable actions or lack of desirable ones when needed. These problems could also cause, due to a sudden, fast or unpredictable move damage either the user, the environment or the proper robot.

The second criterion responds to ensuring the maximum speed while keeping a good torque and being able to replicate smooth actions during operation. This corresponds to 1-A2 on section 1.3.2..

The third criterion seeks the simplicity when posteriorly operating the robot both directly and by means of preprogrammed actions. It is meant to simplify the movements of the robot.

The final criterion serves a similar function as it does when appearing on the rest of the links: keeping the price under specified margin (3-A1 on section 1.3.2.).

In order to maximize the capabilities of this joint, the weight of the forearm needs to be as little as possible, for which it has been decided to place the motor that moves this articulated part out of the forearm. Having in mind the delocalized actuation to be done, the following two **options** have been considered logical:

a. Belt transmission
b. 4-link parallel mechanism

The first one consists of having a belt drive attaching the motor (from its delocalized position) and the joint, potentially having some reductive effect on the gears to which the belt is attached.

The second one consists on designing a mechanism with 4 different links: the first one is in charge of joining the axis of the motor with the axis of the joint on one of the links; then a second bar with the same length as this previous one shall be attached at a distance offset on one direction from the first one; then a third and fourth bars with the same length shall be placed parallelly between them joining the other two and forming a rectangle shape.

The **analysis** inferred based on the past 4 factors can be seen below:

- Regarding reliability, belt drives have slipping issues in various conditions, such us overspeed or lack of tension. However, in normal conditions and if well adjusted, they are strongly reliable.

Speeds on the belt depend on speeds the motor is able to output and the reduction applied, so, aside from maximum strains on the belt, theoretically, no limitations can be found. This direct transmission is kinematically very easy to describe, although a movement on the shoulder shall affect the angle of the forearm indirectly, which the transmission shall need to counteract in order to fulfill kinematic models. The cost for this transmission is very low.

- Four-link mechanisms have no reliability issues, since movement comes directly transmitted by means of a rigid piece from the motor used. Their only limitation is an excessive effort which would, depending on some factors either break the connection between the links or break the proper linking pieces. Their speed is the same transmitted by the motor, since transmission of the motion is direct and no other factor intervenes halfway. Their kinematic properties that keep the attached link on the same direction if rotated makes it easier to control the arm, since no indirect rotations shall happen between shoulder-elbow and elbow-wrist links. Its cost shall depend on the material used for the mechanism, but as very few pieces are involved (4 links together with unions), it can be kept low.

Next, a **summary table** can be seen regarding the characteristic's comparison:

	CRITERIA			
OPTION	Reliability	Speed	Kinematic ease	Cost
Belt transmission	MEDIUM/HIGH	HIGH	MEDIUM/HIGH	LOW
4-link mech.	HIGH	HIGH	HIGH	LOW/MEDIUM

Table 6. *Joint 4 mechanism options comparison*

Source: Own

With all points considered, the taken option is the **4-link parallel mechanism**. This is mainly due to two factors: the first one is its reliability, which, with a good machining of its pieces, should be nearly perfect, and the second one is its kinematic ease, since when the elbow is not being operated, it shall remain with the same angle respect to the shoulder.

In order to keep costs as low as possible, part of robot's structure can be used as part of the 4-link parallel mechanism. This is possible with a good placement of elbow's motor: if the center of the motor axis is placed at the same center of shoulder's motors' axles, the joining link between them can be omitted because the structure of the robot does the part of such union. The same can be said about the lower and shorter link parallel to the one with which the motor transmits the movement to the forearm, since it can be assumed to be part of that piece it controls.

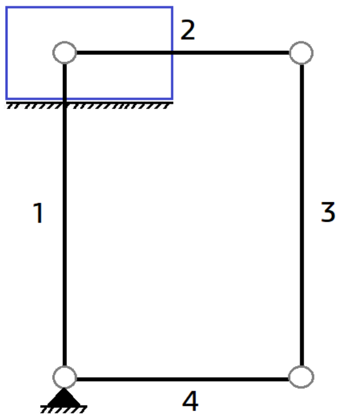


Figure 7. *4-link parallel mechanism as originally presented*

Source: Own

On *fig. 7* the mechanism is presented as it is originally thought. Applying the previously mentioned subtraction of components to the system is equivalent to extracting links 1 and 4 by means of replacing link 1 by the union link between shoulder and forearm and link 4 by the proper forearm, setting the joint between 3 and 4 to be part of the forearm's structure.

In the end, costs can be saved by means of manufacturing only two of the pieces of the mechanism and by designing the structure of the robot to take their role.

Joint 5: Robot's wrist

The final main motorized joint corresponds to the degree of freedom of the end-effector. This shall provide one of the two angles with which the end-effector shall be able to be oriented.

The movement of the wrist needs to be of at least 180 degrees. An axis parallel to the horizontal plane shall be coincident with the joint's orientation.

The motor in charge of this joint shall be placed on the forearm, as early placed as possible in order to reduce the amount of torque it creates for the elbow motor.

To choose among the different options, the chosen criteria are the following:

a. Reliability
b. Weight
c. Cost

Reliability is a common criterion for all these different subsections, and it regards the trust on the joint's behavior when on movement.

As this is the last weight-carrying part of the robot, the mechanism attached to the motor needs to be as lightweight as possible since otherwise it shall diminish the capacity of the robot to pick heavier objects, decreasing the useful load.

The cost needs to be as low as possible, as with the rest of the past sections.

The two main options considered for the wrist are:

a. Belt transmission
b. 4-link parallel transmission

The explanation for these two mechanisms has been made previously for other joints. The only peculiarities are that this belt transmission shall involve a motor placed in the proper forearm whereas the 4-link mechanism shall involve 3 links, having the link equivalent to number 1 on *fig. 7* removed.

It needs to be said that both these two options have been tried experimentally, and with these experiments, it was found that:

- Belt transmission is reliable if well adjusted, but when tested to its limits, it slips from the gears. Due to this, it is necessary to add an external sensor connected to the proper end-effector's attachment so that in case of slip the system does not generate any offset. Considering this, it can be said that even being a very lightweight system, it needs the additional sensor, which has some weight to it. Finally, the cost for this system is very low.

- A 4-link structure is as reliable as the material its links are formed of is. Due to these links and to the mechanical joints between them, its weight is slightly greater than the whole belt transmission system plus the sensor. The cost for this structure is very low.

Below, a table with the **summarized** characteristics of both systems is presented:

	CRITERIA		
OPTION	Reliability	Weight	Cost
Belt transmission	MEDIUM/HIGH	LOW	LOW/MEDIUM
4-link mech.	HIGH	MEDIUM	LOW

Table 7. *Joint 5 mechanism options comparison*

Source: Own

In order to choose the most adequate option, it has been established that the weight factor is essential, since the maximization of the robot's lifting capabilities is key for a correct set of abilities for further developments. Due to this, the chosen option is the **belt transmission**, which, as explained before, shall require the inclusion of a position sensor for a better position control.

End-effector

The end-effector is the part of the robot in charge of being in contact with the objects to be manipulated. There are a great number of different end-effector kinds depending on the function they must accomplish: grippers, suction cups, hand-like manipulators, etc.

For this project, as there is a need for picking, positioning and orientating objects, the decided alternative is the use of a gripper. Given this, a selection of the kind of mechanism used for actuating the gripper must be performed according to some **criteria**:

a. Functionality
b. Cost

The first criterion implies the different uses the gripper can be given and the different number of shapes it can handle.

The second one implies no overcosts due to kinematic redundancies or unnecessary movements.

There are several **options** from which the type of mechanism can be chosen:

a. Linear sliding gripper
b. Four links with four motors
c. Four-link mechanism with 2 motors

A fourth option could be a robotic hand, but this was easily discarded at the beginning of the thought process due to the number of resources it would take to design and produce a perfectly functional human-like hand gripper. There are even entire investigation lines aimed at doing this correctly.

The first option consists of the actuation of a linear joint in order to close the gripper symmetrically regarding its center. This could be achieved by several means, including a rack and pinion mechanism. A diagram with a simplified version of this can be on *fig. 8*.

The second option consists of 4 links attached two on two parallelly and with the possibility of performing different actions individually by using one motor for each of them. This can be appreciated on *fig. 9.*

The third option is composed by 4 links with a parallel mechanism between them due to which by the actuation of two of them the rest remain parallel one to the other (*fig. 10).*

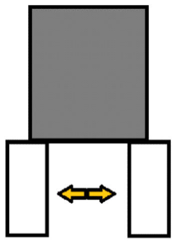


Figure 8. *Linear sliding gripper diagram*

Source: Own

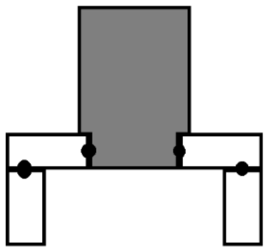


Figure 9. *Four links with four motors gripper diagram*

Source: Own

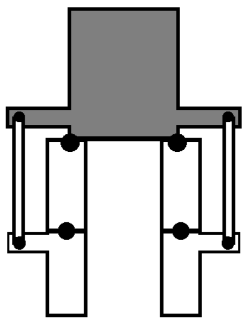


Figure 10. *Four-link mechanism with 2 motors*

Source: Own

An **analysis** of the three different options according to the selected criteria is performed followingly:

- Linear sliding grippers are very used for simple object grasping. However, for certain more complex contour shapes they require higher pressure to picking them due to geometry of their links. Therefore, their functionality is not bad, but it is not the most versatile option. Their cost depends on the material they are made of, and the motor used for the sliding mechanism.

- The second option is the most versatile one, being able to adapt the shape the links form between them individually in order to pick different objects. However, even though it is the most functional option, it is also the most expensive since it demands the use of four motors.

- The third one offers the chance to pick better rounded-shaped objects due to its several grip points while keeping the front links always perpendicular to the end-effector's rear side. It is affordable to produce.

A **summary** of these features can be seen in the table below:

	CRITERIA	
OPTION	Functionality	Cost
Linear sliding	LOW	LOW
4 links, 4 motors	HIGH	HIGH
4 links, 2 motors	MEDIUM/HIGH	MEDIUM

Table 8. *Joint 6 mechanism options comparison*

Source: Own

Taking into account what has been summarized above, it is easy to see that the most balanced option is the one referring to the **4-link mechanism with 2 motors**. This is the chosen option, since although it does not provide freedom for second links' positioning, it allows to rotate each of the sides individually, which allows for different configurations; besides, it can wrap rounded objects due to its geometry or take linear-sided objects.

Several links with different shapes can be manufactured in order to be modularly exchanged to improve the whole gripper's functionality.

1.4.2.3. Motors to be used

The election of motors for each of the joints is a key part of the alternative solution election, since on it rests the accomplishment of the necessary parameters for the development of the project, especially regarding movement, weight and cost requirements.

There shall be a division between the types of motors, according to if they are needed to lift weight or generate at least a certain torque in order for their associated joint to work or if they just need to pull/push their joint horizontally. Each of the two divisions shall have a different study, and for each of these sections there shall be a specific reasoning behind the election of the type of motor.

LIFTING MOTORS

This section oversees representing the thought process behind the election of the motors responsible for joints in charge of lifting the heaviest weight out of them all, that is, joints **3** and **4**. Rotation joint (joint **2**) shall also be included on this list.

Motors chosen in this section shall be the same between them to reduce costs and simplify the designing process. They can be used in different quantities for each joint, depending on the amount of torque required by each of them, which shall be studied more deeply on further sections.

To choose the motors used for this particular functionality, the following **selection criteria** have been established:

a. Maximum torque and size

b. Speed

c. Rotation angle

d. Control
e. Cost

In this section of the report, no torque data shall be provided since detailed calculations shall be performed on further sections. However, a relationship between maximum torque in general terms related to each option's size shall be established, since it does not make any sense from the point of view of efficiency the use of very powerful but very large motors in a project like this, which requires a maximum weight of 5kg.

Speed of the motors is key for the requirements (as much their smoothness as their raw angular speed).

The third criterion establishes the maximum range of motion given by a specific type of motor.

It is very important to determine which kind of control the motor incorporates by itself (none, speed or position), since an already-incorporated speed or position control shall simplify greatly the electronics and computer runtime needed for the normal operation of the most basic motion functions of the robot.

Having in consideration the cost of the motors is necessary to keep the project grounded and inside the boundaries specified in 3-A1 on section 1.3.2..

The available most realistic **options** to choose from have been considered to be the following ones:

a. Stepper motors
b. Speed-controlled servomotors
c. Position-controlled servomotors
d. DC motor with reduction

An **analysis** of these different options for each of the criteria can be seen followingly:

- Stepper motors have high torques for high sizes, having a low torque-size ratio. They can have speeds up to 1000+ rpm. However, it is important to remark that their torque diminishes as their speed is increased. They can rotate 360+ degrees. Their position control is based on the combination of coils being activated. Depending on the chosen model, their price can range from reasonable to unaffordable.

- Speed-controlled servos with sizes and weights able to fit this project's span usually have lower torques than position-controlled ones, ranging up to 6 kg·cm. Their speed can be up to 170+ rpm. They are usually called continuous rotation (CR) servomotors because they have a 360+ degrees span. These motors are very low cost.

- Position-controlled servos can have up to really high torques with very small sizes. These torques, for servos reasonably inside this project's scope range from a few kg·cm up to 60-70 kg·cm. Their speeds normally range between up to 40-50 rpm. Their rotation angle goes from 180 to 270 degrees commonly, with some cases up to 280 degrees. They have automatic position control, since they have intrinsic circuitry in their package with a PID capable of reaching the given reference position (inside servomotor's capabilities). Their cost is usually low.

- DC motors can be electronically operated for torque control, and with a reduction gear set can produce quite a lot of it on the output with not an especially big size. Their speeds can be relatively high taking into account a low reduction ratio. Their rotation angle span is 360+ degrees. Their speed control can be easily made by means of the voltage applied to it, but they need a closed loop control to be able to position the shaft precisely. They are not especially expensive on the sizes able to fit this project.

Below, a table with a **summary** of the mentioned characteristics is presented:

	CRITERIA				
OPTION	Torque/size	Speed	R. angle	Control	Cost
Stepper	LOW	MEDIUM/HIGH	360+	Position (driver)	MEDIUM
Speed servos	MEDIUM/HIGH	MEDIUM	360+	Speed (internal)	LOW
Position servos	HIGH	MEDIUM	270	Position (internal)	LOW
DC	MEDIUM	HIGH	360+	Speed (open-loop)	LOW

Table 9. *Lifting motors comparison*

Source: Own

Given the previous analysis of every option's characteristics, it can be stated that DC motors with gear sets can be discarded due to their lack of self-contained control, for which they require some external sensorization, which would increase the complexity of each joint's specific design.

Regarding speed servos, while it is true that they have self-contained speed control with a PID that is pushing to get the same speed every sample time, all lifting joints need to be positioned accurately. For this, continuous rotation servos cannot be positioned by themselves, but they need external sensorization too, since an estimation of the position given a speed supposition would not provide accurate results.

The final decision needs to be made between the two options that have position control by themselves: stepper motors and position servos.

Stepper motors are widely used in robotics, and they have a smooth, fast, and easy-to-control action. They can also rotate infinitely, which makes them perfect for linear actuation in CNC-type machines, robotic joints with belt transmission, etc. However, when taking into account their direct competitor (position servos), they lose on some factors, these being cost (they are normally more expensive for the same amount of maximum torque) and control (they require an external driver, while position servos have their own control unit inside and only need supply power and signal).

The chosen option is to use **position servos**, since they fit perfectly on size, cost, speed and torque (there exist many models due to which different torques can be chosen). They do not allow full rotation though, but this is not specifically required by any joint inside this subsection.

NON-LIFTING MOTORS

These are the motors in charge of moving two types of joint: those with a linear belt transmission that do not lift any weight and those on the end-effector. For each type of joint, the same kind of motor shall be used to save costs.

Belt transmission joints

Belt transmission joints are, in this robot, number **1** and **5**. These joints shall need to be able to rotate indefinitely in each of the senses and be position-sensed from the outside by an external position sensor. This is mainly due to plausible belt slips, in order to correct them with no damage for any of the links of the robot.

The main **selection criteria** taken into account when considering the election of the type of motors responsible for this movement are:

a. Easy control
b. Weight
c. Cost

The first of the criteria refers to the ability the proper motor has for its self-contained control, and how easy it is to interface with it via electronics or computer software.

The second criterion is very important for the proper working of the different joints and to decrease robot's overall weight. For joint 2, it shall be key to choose the lightest one possible, since otherwise the maximum force able to be exerted on the tip of that link can be significantly reduced.

The third criterion, as in other sections, corresponds to 3-A1 on section 1.3.2..

Followingly, the considered as logical motor **elections** are listed below:

a. Stepper motors

b. Speed-controlled servomotors

c. DC motors

These options have been covered previously. No especial considerations are needed to be explained.

According to the different criteria, the following **analysis** can be made regarding the different options:

- Stepper motors are relatively easy to use given the proper library and given the adequate electronic driver (they need one driver for each of the motors). As they commonly have a metal casing, they are moderately heavy. Their cost can vary depending on the model used but, for medium range applications regarding precision, strength and speed, they are quite more expensive than other type of motors.

- Speed-controlled servomotors are very easy to control regarding speed, since they can self-regulate and require a simple PWM signal to "understand" their reference parameter. They are lightweight in general, being in the range of a few grams. Their cost is nicely affordable too.

- DC motors are not so easy to control regarding speed, since they are controlled open loop with the voltage applied. Depending on the model chosen they can be either lightweight or heavy, but in the ranges of this project's weight there are plenty of options to choose from. They are really cheap depending on the model.

A **summary** table with the analysis performed can be seen below:

	CRITERIA		
OPTION	Easy control	Weight	Cost
Stepper	MEDIUM	MEDIUM/HIGH	MEDIUM
Speed servos	HIGH	LOW	LOW/MEDIUM
DC	LOW/MEDIUM	LOW	LOW

Table 10. *Belt transmission motors' comparison*

Source: Own

NOTE: Control, weight and cost are valued related to one another option, not in general terms.

After this comparison, it can be appreciated that the most balanced option is the **speed-controlled servomotors** since they are cheap, lightweight and have a very easy protocol to interface with. They have a good torque/size ratio too, having commonly 5 kg·cm the cheapest models. They shall however, as it has been previously covered, external position sensors.

End-effector motors

These 2 motors shall be chosen having in mind the 4-link mechanism used and elected on section 1.4.2.2. ("Mechanisms") on subsection "End-effector". For this, the election **criteria** are the following:

a. Functionality
b. Control

The first one is key to maximize useful weight and overall picking capabilities, while the second one aims at making it easy to position end-effector links to pick objects.

Two main **options** have been considered for this purpose:

a. Position-controlled servomotors

b. Small linear actuators

The first ones have already been developed in earlier sections. The second ones are small versions of the linear actuators presented on theory, having a length of few cm and therefore a short range of action.

Analyzing the previously presented options there can be found that:

- Position-controlled servomotors offer a good functionality, providing a reliable and accurate motion when a simple PWM command signal is given. Their control is very functional (they have internal PIDs that make them reach their end goal).

- Linear actuators of the size required for this project's scope regarding end-effector have a good functionality, making it easy to avoid the design and manufacture of several motion transforming pieces, even though their range of motion is limited to the length they can displace during actuation. They need external sensing in order to measure position.

Summarizing the previously mentioned features, the following table can be presented:

	CRITERIA	
OPTION	Functionality	Control
Position servos	HIGH	HIGH
Linear actuators	HIGH	NONE

Table 11. *End-effector's motors comparison*

Source: Own

After considering the different features of each of the options, the election of **position servomotors** over linear actuators has become evident due to its ease of control given not much torque shall be needed (they shall reach their goal position with not especially high mechanical impedances).

1.4.3. ELECTRONICS

This section is aimed at explaining the different considerations made regarding the different electronic subsystems present on the robot arm. These subsystems include the control component, the communication subsystem, and the power supply.

1.4.3.1. Control component

The control component is responsible for the coordination of the different electronics subsystems, including external communications, motor signal coordination and, all in all, the correct execution of the software written in it. It is also responsible for robot's intelligent behavior, making it able to react against changes with the correct sensing.

The election of the controller also implies the election of a determinate IDE with which it shall be programmed. Due to this, the ease in the use of this IDE shall be taken into account when considering the controller to be implemented.

The applied **criteria** in order to select the correct controller are the following:

a. Software and inner hardware capabilities
b. External hardware capabilities
c. Board needs / programming ease
d. Performance / cost ratio

The first criterion refers to everything related with the software and the hardware the controller has that can improve robot's overall abilities such as the amount of memory it has, the speed of its clock, its sensing capabilities, etc.

The external hardware capabilities refer to the treatment the controller is able to do to incoming or outcoming signals, the number of pins it has, etc.

This third criterion relates to what the controller shall need in order to work properly, regarding power supply, IDE and community feedback or anything necessary on that regard.

The last criterion to take into account means to relate how the overall performance of the previous sections relates to board's price. High performance with low costs shall give a high ratio, whereas the opposite case shall retrieve a low ratio. Medium or high performance with high costs shall be considered a low ratio too, since costs are important regardless of functionality for extreme cases.

The **options** that have been considered for the development of this subsystem are:

a. STM32F4 Discovery
b. Arduino Mega 2560
c. Raspberry Pi 4 2-8 GB

The main difference between these options is that the first two are µCs (microcontrollers), whereas the third one is a PC.

The STM32F4 Discovery series from STMicroelectronics are a kind of 32-bit microcontroller boards with built-in Analog I/O, Digital I/O pins, sound port, USB, 3-axes accelerometer and microphone. They incorporate ARM Cortex nuclei which have very fast computing capabilities (84 MHz clock speed), 512 KB of flash memory and 90 KB RAM memory. They can be programmed by means of Keil µVision IDE.

The Arduino boards are a series of ATMEL-based microcontrollers with integrated Analog I/O, Digital I/O and communication port pins. The Arduino Mega 2560 is a slightly old Arduino with the highest number of pins, having 13 PWM-enabled pins, 69 digital pins, 16 analog pins, 4 serial ports, an I2C port, 2 ICSP ports and an ISP port. It has an 8-bit AVR-based ATMega2560 microcontroller at the board's core, with 256 KB flash memory, 8 KB Static RAM memory and 4 KB EEPROM memory.

The Raspberry Pi family is a series of small computer boards with built-in peripheral ports which, depending on the model, can include USB, microHDMI, audio connectors, Ethernet ports and even some wireless connectivity options like Bluetooth and Wi-Fi. One of the latest additions to this family is the Raspberry Pi 4 model B, which comes with different RAM memory capacity election (from 1 to 8 GB). Its processor is a quad-core Cortex A72 with a 64-bit architecture and a 1.5 GHz clock speed. It has 2 2.0 USB ports, 2 3.0 USB ports, several audio and video ports, an Ethernet port, Wi-Fi connectivity, Bluetooth 5.0 BLE, 17 Input / Output pins and a HAT ID bus. It can consume up to 3 Amps.

A **summary table** with the key comparative characteristics can be found below:

	CRITERIA					
OPTION	Architecture	Flash memory	Clock speed	"RAM"	I / O pins	Comms and extras
STM32F4 Discovery	32-bit	512 KB	84 MHz	90 KB	16 under control	USB, sound port, 3-axes accel., microphone
Arduino Mega 2560	8-bit	256 KB	16 MHz	8 KB	69 DP, 16 AP, 13 PWM	4 serial ports, 1 I2C port, 2 ICSP ports, 1 ISP port
Raspberry Pi 4	64-bit	microSD card	1.5 GHz	up to 8GB	17	2x 2.0 USB, 2x 3.0 USB, audio and video ports, Ethernet, Wi-Fi connectivity, BLE 5.0, HAT ID bus

Table 12. *Controller options characteristics*

Source: Own

From *Table 12*, it becomes evident that the difference between the µCs and the PC presented is the raw computing power. The 2 µCs are more hardware-oriented, whereas the PC is aimed at providing high processor speed, high RAM memory and program memory and many communication capabilities with different devices such as monitors. This would be the most effective option for the project if it were not for the amount of electronic connections needed to be added to the project regarding motors, sensing, digital I/O or analog I/O and if it were not for the performance/cost ratio:

while it is true that this option offers the highest computer performance out of the three, it in fact exceeds the necessities of the robot, having also a cost that is elevated higher than the previous ones, in some cases being almost the entirety of this project's budget. The decision must be made between the 2 µCs.

Regarding computing "power", it needs to be stated that the STM32F4 family has a more powerful architecture, clock speed and RAM than the Arduino. It also has better capabilities related to the flash memory. However, when it comes to hardware, the Arduino has many more I/O pins, and although the STM32F4 has different extras, the Mega 2560 still has more communication ports, which is a good feature for 1-E1 on section 1.3.2.. Another good trait of the Arduino Mega 2560 is that it has great community support, which shall ease programming for the user, with detailed but simple to use libraries. Finally, it must be stated that its price is affordable for the scope of this project.

The chosen option, therefore, is an **Arduino Mega 2560**.

1.4.3.2. Wireless communications

The project requires 4 external communication modes on 1-E1 on section 1.3.2., which include 2 wired communications and 2 wireless ones. The 2 wired communications shall be achieved by means of 2 out of the 4 serial ports the Arduino Mega 2560. The 2 wireless communications shall be done by using 2 different modules, one per each kind. This shall be discussed on this section.

<u>Radio communications</u>

This wireless channel shall be obtained by means of a radio module, which shall provide a fast communication through a software protocol, in order to plausible gesture following applications as smooth as possible. The decided **criteria** for this election are:

a. Functionality / use
b. Range of transmission

The first criterion establishes that the elected module needs to be able to be used for as many applications as possible and must be able to be used without any conflicts with other subsystems and as easily as possible.

The second criterion aims at obtaining the module that has the best range regarding the area it can cover for reliable communication.

There are two main **options** which, due to the experience of the designer, work reliably for different types of projects:

a. RF 433 MHz module pair

b. HC-12 radio transceiver

The first of the options consists of a pair of modules formed by an emitter and a receiver working at a voltage of 5 V (2 m range) or at 12 V (300 m range with a 16.5 cm long antenna). They are really cheap modules which can only exert one function each (one is always an emitter, and the other is always a receiver). They need the *VirtualWire.h* library.

The second option is a transceiver (emitter and receiver) module working at a voltage of 5 V, with a very economical price. It operates at frequencies between 433.4 MHz and 473 MHz. It has a range of up to 1 km with no additional hardware or extra voltage to apply. This module communicates with the µC by means of a serial port.

To **summarize** these characteristics, the following table is presented:

	CRITERIA	
OPTION	Functionality	Range
RF 433 MHZ	MEDIUM	up to 300 m with especial conditions
HC-12 transceiver	HIGH	up to 1 km

Table 13. *Radio modules comparison*

Source: Own

Given the different characteristics exposed, the election of the **HC-12 module** is considered to be the most efficient one. One of the reasons for this is that the RF 433 has only one functionality per module, which limits the range of options for posterior bidirectional communications, and its range is very short compared to the HC-12. Another one is that, from experience, previous use of the *VirtualWire.h* library has shown that it is incompatible with the traditional *Servo.h* library used by the Arduino IDE, for which the adaption of the RF 433 to this project would be more complicated, with the need of creating proper timing functions to overwrite the PWM signal generation with external timers to the one used by the proper method.

The HC-12 module shall be connected to one of the 2 serial ports that are left unused on the board.

Bluetooth communication

The user specifications review specifies that both the use of Bluetooth and BLE is possible for this kind of wireless communication. The module needs to be exchangeable by another similar one the user might wish to add, for which the connection of the elected module must be done with header sockets.

Even though the use of both common Bluetooth and BLE must be decided at each moment by the user, the election of the default type and its representative module that shall be used in the robot must be done in this section. For this, the following **criteria** have been established:

a. Range
b. Ease of use

The first criterion aims at choosing an option that has a good communication range, since the user might want to be placed at some distance from the robot when interacting with it via the mobile app.

As a base module to use and as it is easily exchangeable with other modules if that is user's will, the used election shall be the one that has the easiest handling regarding software.

For each of the two available systems, for familiarity, availability, use and cost, the following **options** are considered:

a. HC-05 (Bluetooth 2.0)
b. HM-10 (BLE 4.0)

The first of the modules is a very common and cheap Bluetooth module that can communicate through a serial port with an external µC. Its operating voltage is 5 V and it can be configured on either master or slave mode via AT commands on the serial terminal. It has a range of 10 m and a transmission speed of up to 1.3 Mbps.

The second module is also cheap and common one like the HC-05 which works on 5 V level and has, theoretically, a range of up to 100 m (from experience, this is far from truth). BLE (*Bluetooth Low Energy*) is a type of Bluetooth communication aimed at having as low consumption as possible. It works by sending info to a service, which hosts a specific kind of variable, which can be a number, a character, etc. These posts are updated when new info is available, and therefore they can be read. Their identifiers can (or not) comply with the *GATT Services* standard.

Here, a table with the **summarized** characteristics is presented:

	CRITERIA	
OPTION	Range	Ease of use
HC-05	10 m	HIGH
HM-10	< 100 m	LOW

Table 14. *Bluetooth modules comparison*

Source: Own

The HM-10 module has the highest of the ranges. However, for any user, even advanced ones, it is not the easiest one to use, since it requires good understanding in how the BLE technology works. The HC-05 on the other hand, even though it is not the one with the biggest range, is the easiest one to use, since it allows the transmission of relatively large numbers, strings, etc. at the cost of very few lines of code, with a wide variety of applications due to this.

The elected module is the **HC-05**. Its range is not necessarily a problem, since for most applications, a range greater than 10 m is not needed. What is more, the user shall be able to exchange it with no problems whenever they desire to.

The HC-05 module shall be connected to the only serial port left unused on the board.

1.4.3.3. Power supply

For the power supply of the whole robot, several things shall be needed to be considered, one of them being the fact that the several subsystems shall need different levels of voltage to be supplied with. For this, a voltage regulator shall be used for each variant of the power source's output level.

The kind of power source to be used for the whole robot shall be decided on this section. For that purpose, the following **criteria** shall be taken into account:

a. Functionality
b. Cost

These two considerations have the same explanation provided in other sections, which are, essentially, providing the best capabilities for the least cost.

The several **options** that have been considered are:

a. Batteries
b. Standard power source
c. AC to DC transformer

The first option consists of a series of rechargeable batteries which, after every use, should be recharged by means of connecting them (individually or the whole pack, depending on the type used) to the grid in some way.

The second option is using a standard regulated power supply.

The third option is to use a common transformer with rectification for AC to DC conversion included in the circuit. This would be plugged to the electrical grid.

When **analyzing** the given options according to the criteria, it can be found that:

- Batteries offer a way to feed the system without it being connected physically to a socket. This can be useful when operating it in an environment with no near sockets. Their cost is relatively low, depending on their capacity and format.

- A standard power source offers a reliable way to feed a system, since it provides the whole conditioning for the input, heat dissipation and voltage / current limitation. Their cost ranges from low to very high prices depending on a series of factors.

- Transformers with rectification circuit may not offer the smoothest of the supplies, but they are very affordable.

A **summary** with the considerations expressed before can be seen below:

	CRITERIA	
OPTION	Functionality	Cost
Batteries	LOW	LOW/MEDIUM
Power source	HIGH	MEDIUM/HIGH
Transformer	MEDIUM/HIGH	LOW

Table 15. *Power supply options comparison*

Source: Own

Given what has been explained, batteries are the first option discarded. The robot aims at being on a flat surface working, and even though batteries would allow it to move with no cables attached, it would not be able to remain operative for long periods of times, which is something the user could intend to do.

Between power sources and transformers, knowing that a precise level of voltage is not necessary and, given the price difference in some cases, **transformers** shall be chosen as the option to implement.

The number of transformers used should, in any case, be the least possible and in the design stage, different subsystems supplies shall be organized such that they provide the correct amount of power for every component and are minimized in their use.

1.4.4. FABRICATION

Regarding **fabrication**, depending on the method used and the materials involved functionality and costs shall vary.

The correct study of the materials to use is key for the proper working of the robot since an incorrect conclusion on this section could be the cause of potential failures or malfunctions.

The correct election of the fabrication method shall allow simplicity and cost reduction for prototype's manufacture as well as in case of future mass production.

1.4.4.1. Main material

The election of the correct material is key for developing a product with the correct behavior regarding stiffness, weight, durability, vibrations, etc. It must be able to function as designed and feel as good as possible while doing so. The biggest amount mass on the robot shall probably come from its structure, which shall be made of the given material.

The chosen material shall **need** to be:

a. Stiff
b. Lightweight
c. Cheap

Stiffness is necessary when designing a robot as they are traditionally conceived. It is true that some investigation fields tend to work on flexible robots based on air flow or even origami, but for this project's scope (especially because of budget limitations), it shall be kept as simple as possible, that is, with a rigid structure.

The device needs to be as lightweight as possible in order for it to work with the smallest and least motors possible. It must also be able to comply with 1 d. on section 1.3.3..

Since a great part of robot's mass shall be made of the elected material, its cost needs to be the minimum possible.

The materials elected as **options** are:

a. Steel
b. Plastic
c. Wood

Analyzing these materials according to material's needs, the following statements can be found:

- Steel is a very stiff and durable material, with strong forces and bending moments needed for it to flex. However, it is not lightweight, and its price is not the lowest.

- Plastic, depending on its density (with some manufacturing techniques such as 3D printing this can be chosen by the grade of infill used), it can be either very stiff or very flexible. It is also a very lightweight material (depending on the density used) and is currently very cheap.

- Wood is more or less stiff depending on the kind of wood, specifically according to its cut size and its density (which commonly depends on the kind of wood used): plywood has a 400 to 700 kg/m^3 density, but balsa wood has 100 to 170 kg/m^3. Its weight also depends on their cut size and density. It is a very cheap material.

Summarizing these properties:

	CRITERIA		
OPTION	Density	Stiffness	Cost
Steel	8050 kg/m^3	HIGH	HIGH

Plastic	1240 kg/m^3 (PLA)	MEDIUM/HIGH	LOW
Wood	100 kg/m^3 (balsa) to 1330 kg/m^3 (ebony)	LOW/MEDIUM	LOW

Table 16. *Material options comparison*

Source: Own

As it can be seen on the table, steel is the densest material and therefore, for the same amount of volume, it shall be the heaviest one out of the three. Steel is the stiffest material too. If it were not because of its weight and price, which exceed this project's budget alone if a prototype were fabricated, steel would be the election.

Between plastic and wood, the election is **plastic**. This is mainly due its difference on stiffness and the finish it shall provide to the pieces. Besides, plastic's density can be modified by means of its inner fill to have determinate properties, what makes it very versatile.

This shall be robot's main material, which shall be responsible for making the different pieces. Those parts shall be attached between them by means of metal axis and screws so that they are easily dismountable and as modular as possible (1-F1 on section 1.3.2.).

1.4.4.2. Procedure

In order to use the main material to create the various pieces, a fabrication procedure needs to be elected. This fabrication technique shall be effective enough for the prototype creation, given limited resources. In a plausible future, this procedure may be changed in order to fulfill greater demands.

The manufacturing procedure should follow the following **directives**:

a. Manufacturing speed

b. Cost per piece

The first one is key for prototyping as fast as possible in order to try different piece designs that could need experimental testing in order for them to be modified and improved.

The cost for each of the pieces is key for fast prototyping and permanent piece manufacturing so that it can accomplish budget limitations.

The two **options** considered for this procedure are:

a. Mold plastic injection
b. 3-D printing

Mold plastic injection consists on the fabrication of pieces which, when plastic is injected inside them, generate the shape of the parts that are willed to be fabricated.

3-D printing is a very popular technique that allows the fabrication of nearly any shape by means of a cartesian robot that has, as an end-effector, a hot plastic injector.

An **analysis** of this options shows that:

- Mold plastic injection is a fast technique. However, if taking into account the fabrication of the necessary molds, it becomes evident that it is not an immediate technique. The cost of this procedure is very high, especially due to the fabrication of such molds, with prices ranging from 75 $ up to 20000 $.

- 3-D printing is also a pretty fast technique with relatively strong results regarding the pieces fabricated (by infill regulation, plastic part's density and therefore stiffness and weight can be "designed"). The cost for 3-D printing, as well as the cost of its material are very low.

A **summary** table is presented below:

	CRITERIA	
OPTION	Manufacturing time	Cost per piece
Plastic injection	LOW	HIGH
3-D printing	MEDIUM/HIGH	LOW

Table 17. *Manufacturing options comparison*

Source: Own

It must be noted that, for large quantities of pieces, plastic injection can be the fastest option. However, for prototyping, **3-D printing** is the most suitable option for the project, since it is the fastest, the cheapest and the most versatile technique.

In the future, if new product production numbers were necessary, this technique could be changed.

1.4.5. ALTERNATIVES' SUMMARY

Once all subsystems' main features have been decided and, before starting with the detailed design, it is considered useful to summarize all the decisions made on this section:

	CRITERIA		
OPTION	DECISIONS MADE		
MECHANICAL STRUCTURE	**Overall shape**	Mobility of the robot	Robot shall displace linearly and follow an aerial path structure
		Type of robot	Anthropomorphic robot (with 5 DoF)

	Mechanisms	Joint 1: Lateral displacement	Belt transmission	
		Joint 2: Robot rotation	Gear drive	
		Joint 3: Robot's shoulder	Servomotor directly attached to joint	
		Joint 4: Robot's elbow	4-link parallel mechanism	
		Joint 5: Robot's wrist	Belt transmission	
		End-effector	4-link parallel mechanism with 2 motors	
	Motors	Lifting motors	Position high torque servomotors	
		Non-lifting motors	Belt transmission motors	Speed-controlled servomotors
			End-effector motors	Position servomotors
ELECTRONICS	**Control component**	Arduino Mega 2560		

FABRICATION	**Wireless communications**	Radio communications	HC-12 module
		Bluetooth communications	HC-05 module
	Power supply	Transformers with AC to DC rectification	
	Main material	Plastic	
	Procedure	3-D printing (prototyping)	

Table 18. *Summary of section 1.4.*

Source: Own

1.5. Detailed design

1.5.1. INTRODUCTION

The robot shall be able to perform different types of tasks in a sequential way, by user interaction or by reacting to the environment. The first two options are already hardware-added to the basic robot. The following one depends on a user-implemented set of sensors and electronics communicated to the arm by means of its installed communication ports.

When turned on, if motor-enabling switch is on, the robot shall execute a series of preprogrammed initial moves that shall allow the user to check that every single subsystem in the robot operates correctly.

Once the robot has been initialized, it shall respond to several inputs or execute a given set of movements (whatever the user has programmed it to do) when receiving a specific signal which can be either a word, a direction command, a position and an orientation given a point, etc.

The robot shall move its displacement joint firstly, then, when at its correct z coordinate, it shall move the rest of main joints to their corresponding position simultaneously. While performing these movements, the program stops and cannot execute other instructions except for interrupts, in the case there were. Once the goal position has been reached, the rest of the program shall keep running, either receiving new signals or performing new movements (depending on what the user has programmed it for).

Additional functionalities the robot has include a frontal light that shall provide a bright area ahead of it, enabling either the user or the proper robot (in case of computer vision applications) to be able to see where the end-effector is pointing at. Besides, its communication capabilities enable it to act as a signal repeater if configured for doing so, which can be useful for home automation applications.

This section of the memory report aims at explaining with detail each of the different subsystems of the robot, regarding how they are and how they work.

The work breakdown structure shows how robot’s different subsystems are divided.

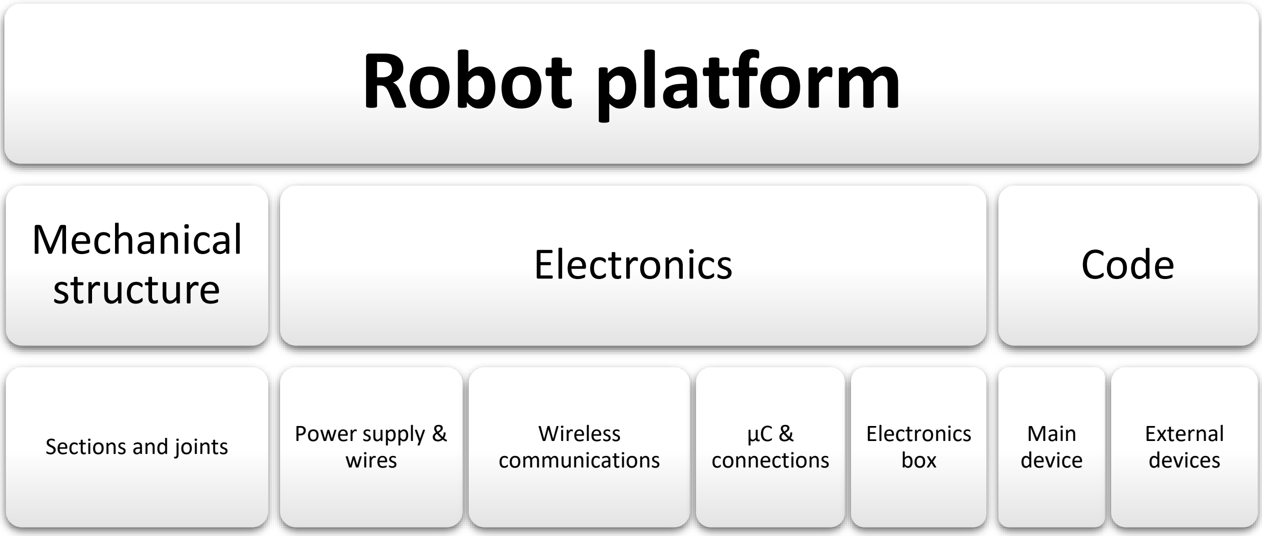


Figure 11. *Work breakdown structure diagram*

Source: Own

1.5.2. MECHANICAL STRUCTURE

The mechanical structure of the robot comprises two different meanings: the different links and the different mechanism parts.

This section aims at explaining how each section of the mechanical assembly works breaking it down by the different joints and sections of the arm.

All dimensions are presented on section 2 ("PLANS & DRAWINGS"), only a few relevant ones are presented on this section. References to such plans shall be made for each of the components.

Rail structure

This structure (*plan 1*) is the one that holds the rails of joint 1 and therefore the robot in the air. It shall have overall dimensions 800 x 413 x 390 mm in "I" shape. It shall be made with 20x20 mm aluminum extrusions. There shall be two 800 mm profiles, two 240 mm profiles, two 413 mm profiles and two 350 mm profiles. Each extrusion with a free exit shall be blocked with a plastic peg, having a total of 12 of these pieces.

The result is shown in the following figure:

Figure 12. *"I"-shaped robot-lifting aluminum structure*

Source: Own, Autodesk Fusion 360

Joint 1 section

Joint 1 section regards the displacement joint and all the different pieces involved in it.

The first point to describe is the placement of the rails along which the robot shall displace and on which it shall be supported. They shall be placed symmetrically respect to the length of the I-shaped structure at a distance of 200 mm one respect to the other.

On one side of the structure there shall be the motor in charge of moving the belt transmission. This motor shall be a 5.5 kg·cm DS04-NFC servomotor from manufacturer *Luxorparts* (*fig. 13*), which has a maximum speed of 4.71 rad/s. On its shaft, there shall be a 30 mm wide cogwheel with 52 teeth (*plan 2*) with a belt fixing expansion bigger than the proper wheel, with two holes for motor fixing.

Figure 13. *DS04-NFC servomotor*

Source: https://images.jumpseller.com/store/mactornica/5514405/1_1.jpg?1623785245

Figure 14. *52-tooth cogwheel for belt drive*

Source: Own, Autodesk Fusion 360 (GF Gear Generator)

The motor shall be caged by means of a piece resembling a box with its right side open and holes for servomotor attachment (*plan 3*).

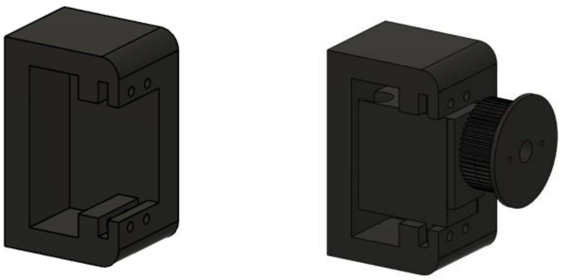


Figure 15. *Motor bracing and motor with bracing and cogwheel attached*

Source: Own, Autodesk Fusion 360

A 50 x 50 x 15 mm brass square shall be centered on beam's height and at 10 mm off cage's superior edge. This can be seen on *fig. 16*.

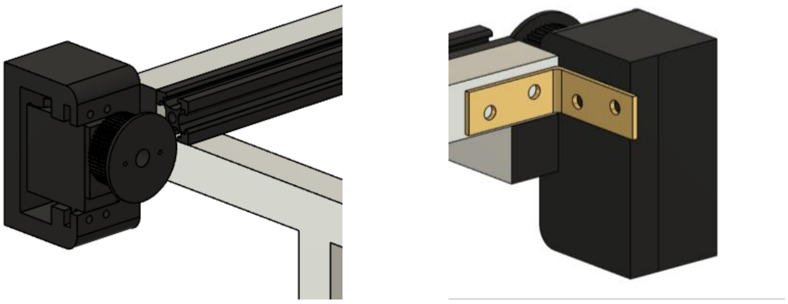


Figure 16. *Motor side in place with frame*

Source: Own, Autodesk Fusion 360

On the other side of the structure there shall be a GT2 pulley (*fig. 17*) inside a casing (*plan 4*), trespassed by a 5 mm diameter cylindrical aluminum shaft with a length of 50 mm. The casing shall be attached to an adapting piece (*plan 5*)), which shall be in direct contact with the square, which shall connect it to the structure. This square shall be placed on half the height of the lateral beam to which it is attached, having its perpendicular other side levelled with beam's outer edge.

Figure 17. *GT2 20 tooth 5 mm bore timing belt pulley for 3D printer*

Source: https://cf3.s3.souqcdn.com/item/2019/02/21/45/93/53/93/item_L_45935393_e4a00158f23cd.jpg

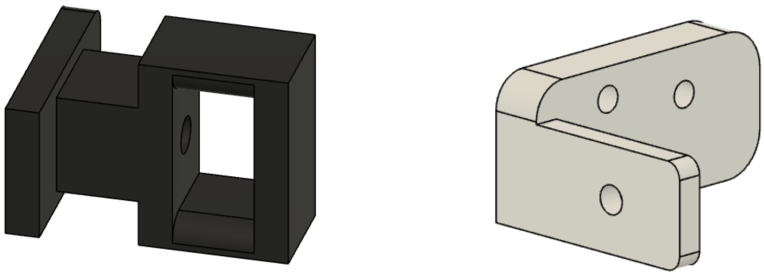


Figure 18. *Casing for the pulley (left) and adapting piece (right)*

Source: Own, Autodesk Fusion 360

The casing for the pulley consists of a hollowed box with a cylindrical aperture on its back side through which the metal shaft of the assembly is inputted. It also has a base that is directly attached to the adapting piece.

The adapting piece resembles the proper square, but is, like the case, also made of plastic. It has two holes for square's screws plus another one for the shaft's entry.

On the back of the adapting piece, passed through the shaft, a stop piece (*plan* 6) is placed.

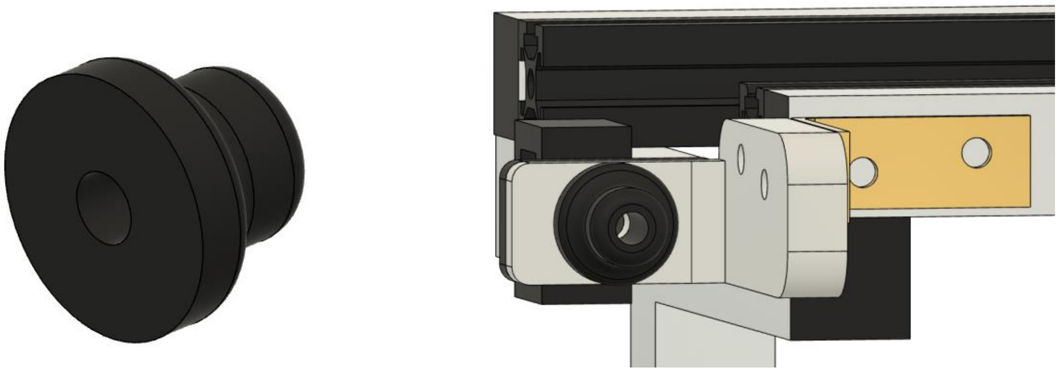


Figure 19. *Shaft's stop piece alone (left) and in place (right)*

Source: Own, Autodesk Fusion 360

The result of assembling this whole left side of the belt transmission mechanism can be seen on *fig.20*.

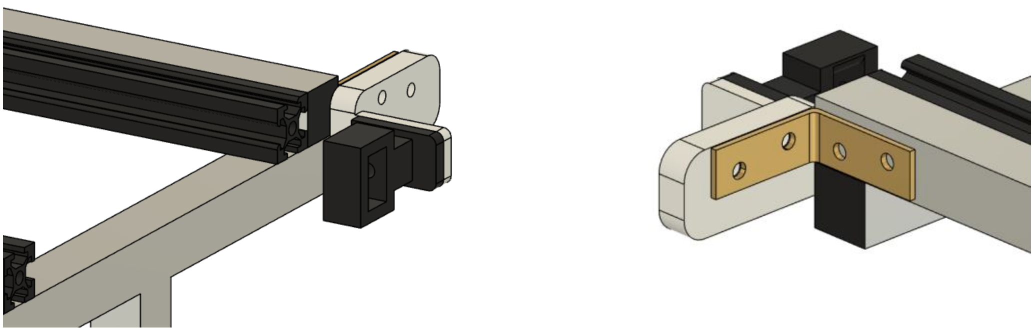


Figure 20. *Pulley side in place with frame*

Source: Own, Autodesk Fusion 360

The whole belt transmission mechanism by itself (not taking into account the proper robot adaption) can be seen on the following figure:

Figure 21. *Belt transmission mechanism attached to frame*

Source: Own, Autodesk Fusion 360

The union between the belt mechanism and the robot is connected to robot's base. This mobile base is responsible for several functions, among which there is holding the entirety of robot's weight, holding the rotation mechanism, attaching the robot to the sliding joint and supporting the box for electronics, which is placed directly above this piece.

Robot's mobile base is composed by several pieces: two opposing sides with a holding for the shafts responsible for arm's sliding capabilities, the other two opposing sides have ventilation grilles with one of them having an aperture for the laser distance sensor and the bottom part holds all of them in one. Their assembly can be seen in the following figures:

Figure 22. *Robot's base 3D view*

Source: Own, Autodesk Fusion 360

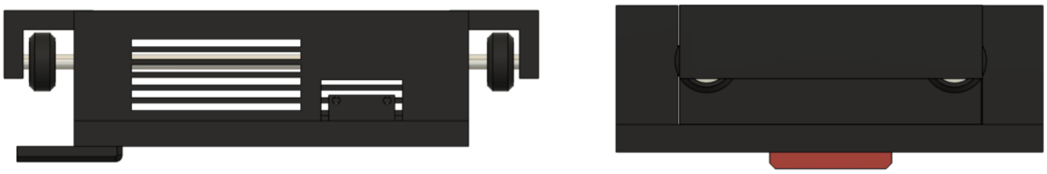


Figure 23. *Robot's base right and back views*

Source: Own, Autodesk Fusion 360

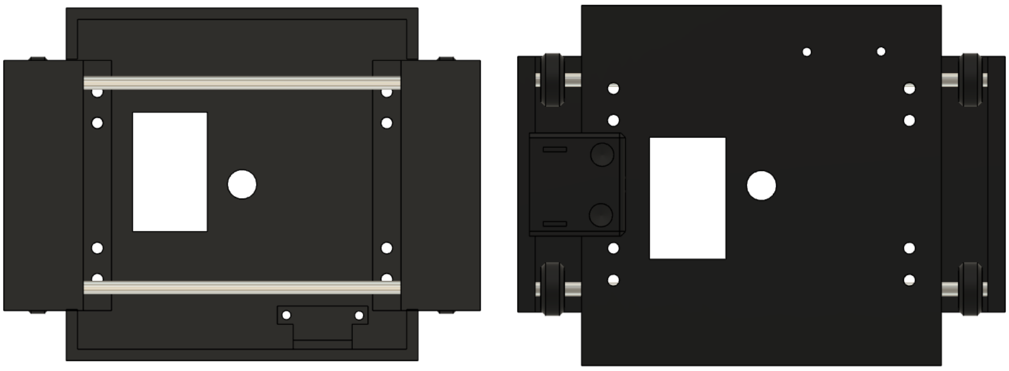


Figure 24. *Robot's base top and bottom views*

Source: Own, Autodesk Fusion 360

The base is attached to the rails by means of four 3D printer wheels 24 mm in external diameter (*fig. 25*). These wheels are attached to the robot by means of 5 mm cylindrical aluminum rods with a length of 200 mm each at a reason of two of them per pair of wheels. These rods are connected to the robot by trespassing through the holes found on the supporting identical pieces in the base (*plan 7*).

These pieces have two passing holes on their side that opposes them and two halfway holes on the cavity where the wheels are located. These holes lock the shafts in place. The four bottom holes are used to screw these two parts to the bottom piece in the base.

Figure 25. *3D printer wheel*

Source: https://images.autods.com/ebay_images/21312-Creality-3D-Printer-POM-Wheel-Plastic-Pulley-Linear-Bearing-for-Ender-3Ender-3-ProCR7CR8CR10CR10SS4S5-Pack-of-10-b0bab412-ab77-4882-ae28-ddbe783f05f3

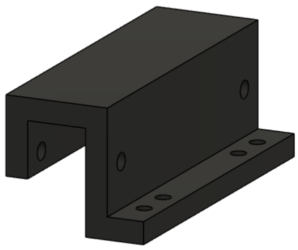


Figure 26. *Robot's base shaft and wheel holding part on the base*

Source: Own, Autodesk Fusion 360

There are two other pieces conforming the lateral surroundings of robot's base (*plan 8* for the left lateral panel and *plan 9* for the right lateral panel). They do not accomplish any structural functionality, but they are meant to protect the interior of the base and at the same time allow the air to flow inside from the outside.

These parts are different among them, the one on the left side of the arm having larger grilles than the other, and the other one (on robot's right side) having a gap through which the distance measurement laser shall trespass. They shall be glued to the bottom piece of the base.

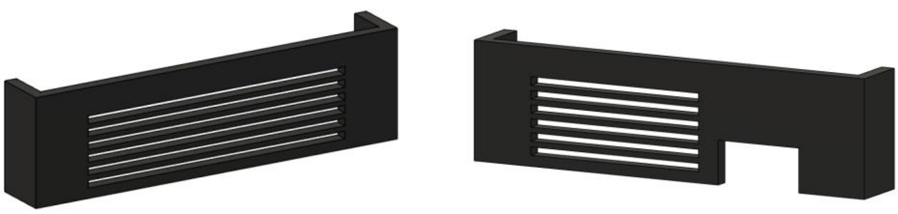


Figure 27. *Robot's base external panels*

Source: Own, Autodesk Fusion 360

The four pieces previously mentioned are attached to the bottom piece of the base (*plan 10*). This square piece holds robot's weight and encapsulates the rotation mechanism. It has eight holes useful to screw the different components that are placed above, one central hole for robot's rotation axle and a square cavity for rotation motor. It is the only structural custom-made piece (except for structure beams) to be made in a material other than plastic, being made of plywood, and being cut by means of a keyhole saw and a boring machine.

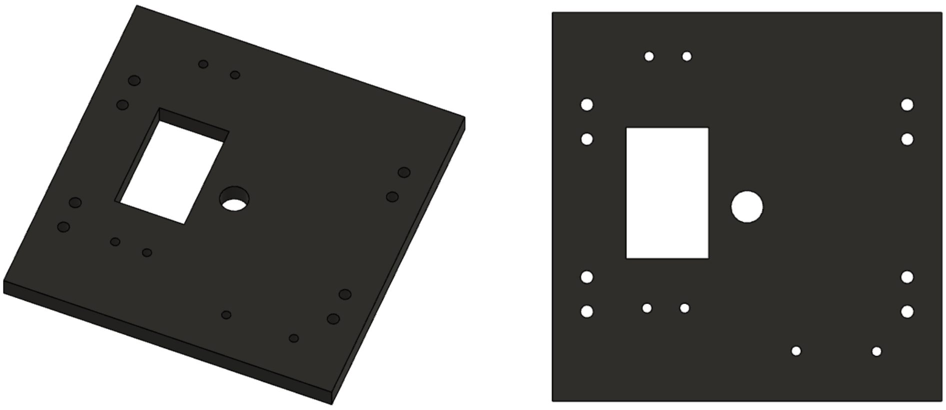


Figure 28. *Robot's base bottom square piece*

Source: Own, Autodesk Fusion 360

The belt to be used is a 3D printing GT2 timing belt 6 mm thick. It shall be attached to the robot through the base, to which it shall be connected by means of a piece with two square cavities through which the belt shall pass (*plan 11*). The belt shall pass through these cavities and shall be screwed to this piece. At the same time, this piece shall be screwed to robot's mobile base (to its bottom piece).

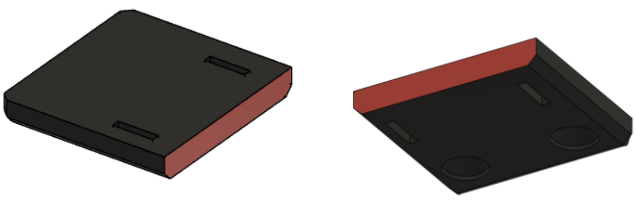


Figure 29. *Belt attachment piece*

Source: Own, Autodesk Fusion 360

Additionally, the laser supporting piece (*plan 12*) shall be screwed to the bottom part of robot's mobile base facing outwards in such way that, when the laser module is attached, the lateral wall of the box does not interfere with laser's ray.

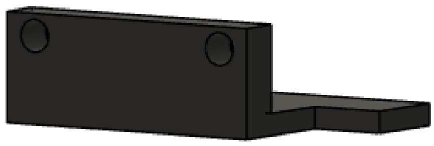


Figure 30. *Laser holding piece*

Source: Own, Autodesk Fusion 360

In order for the sensor to be able to detect a physical limit in the structure, another piece must be placed (*plan 13*), but on the structure, with its biggest face 20 mm away from the external limit of its corresponding beam. This piece has a square-type constitution, with a larger smooth surface that serves for reflecting laser's ray.

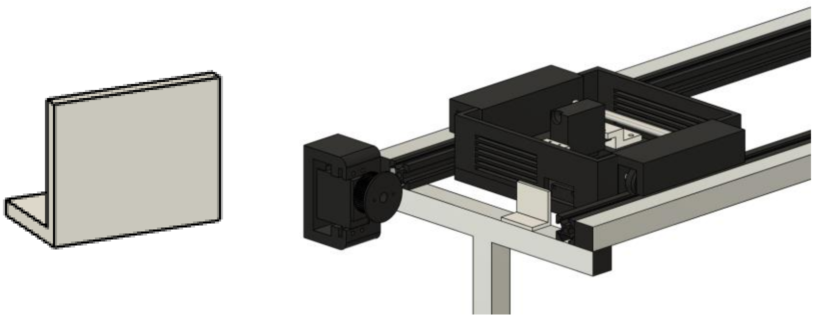


Figure 31. *Laser reflecting surface piece alone (left) and in place (right)*

Source: Own, Autodesk Fusion 360

The whole base in place with the structure can be seen on *fig. 32*:

Figure 32. *Robot base in place with the structure*

Source: Own, Autodesk Fusion 360

The whole sliding mechanism shall have an approximate theoretical maximum speed of 0.071 m/s (71 mm/s). More detail on this calculation on Annex I.

Joint 2 section

This joint aims at rotating the whole robot respect to its moving base. For this, a 25 kg·cm DS3225 servomotor shall be used (*fig. 33*). Its power output shall be translated to a gear set that shall be discussed on section 1.5.4. ("ELECTRONICS").

Figure 33. *DS3225 servomotor*

Source: https://img.tttcdn.com/product/original/p/gu1/R/8/RM10768/RM10768-1-fa74-zjbO.jpg

The motor shall be attached to the robot base by means of two identical 3D printed supports (*plan 14*). These pieces are screwed on each side of the motor and directly to the base. They hold the motor that rotates the gears responsible for joint 2 rotation.

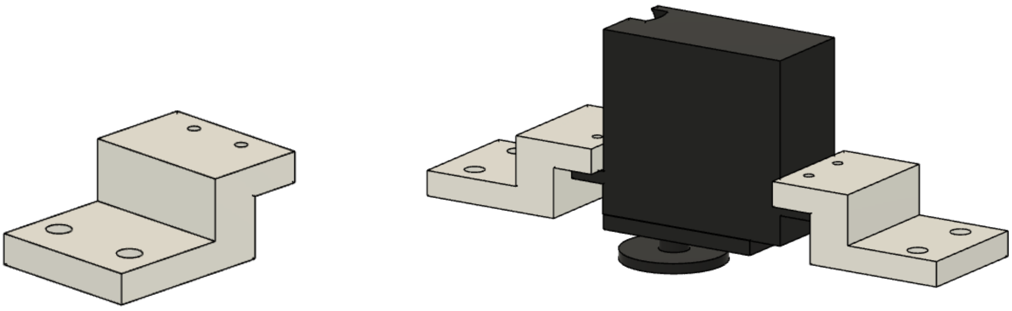


Figure 34. *Rotation motor holding (left) and two of them positioned (right)*

Source: Own, Autodesk Fusion 360

These parts, together with the motor, can be seen in position on *fig. 35*:

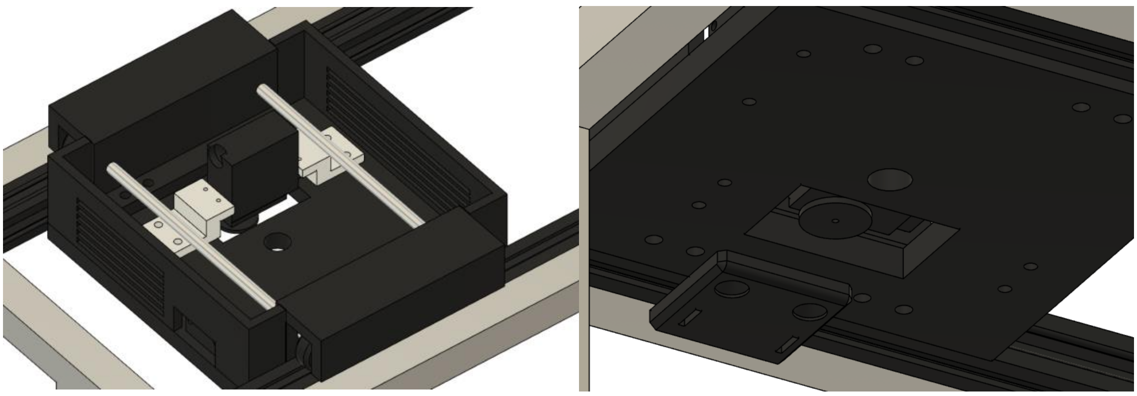


Figure 35. Rotation motor with holdings in place

Source: Own, Autodesk Fusion 360

In order to explain how the robot is attached to its base, the first piece that rotates and which pulls the rest of the robot with it needs to be discussed (*plan 15*).

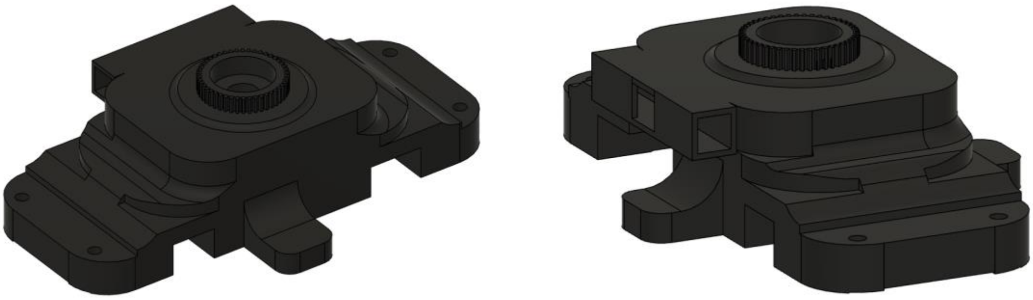


Figure 36. *Shoulder 3D views 1*

Source: Own, Autodesk Fusion 360

This piece is responsible for casing the shoulder joint and, therefore, it holds the rest of the robot underneath. It holds in its interior a cavity (*fig.* 37) especially prepared to host an M12 carriage bolt with length 100 mm, which acts as the rotation axle. Bolt's square section allows to fix the bolt with the piece so that it does not rotate freely regarding the shoulder part.

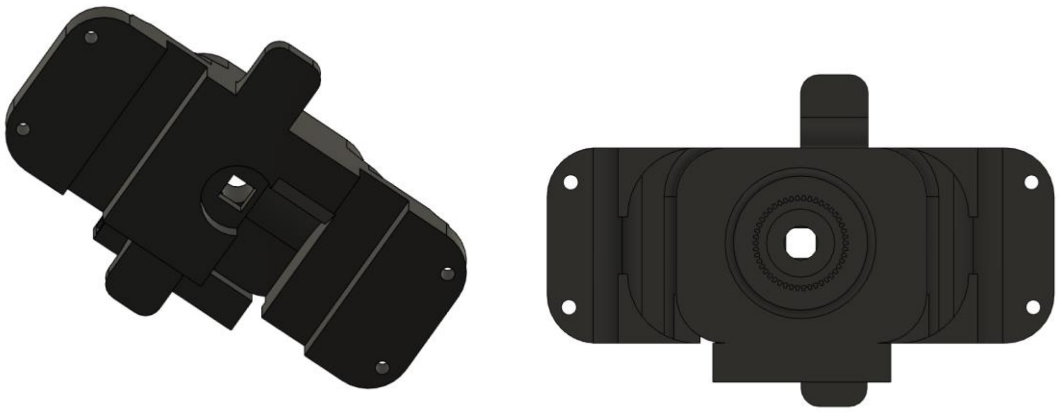


Figure 37. *Shoulder 3D views 2*

Source: Own, Autodesk Fusion 360

On the top part of the shoulder there shall be an empty cogwheel with characteristics given on *plan 15*. This cogwheel shall hold inside its cavity an axial ball bearing with 28 mm external radius, 12 mm internal radius and 11 mm in height (*fig. 38*).

Figure 38. *Axial ball bearing*

Source: https://www.kugellager-express.de/media/image/product/6280/lg/axial-deep-groove-ball-bearing-51208-40x68x19-mm~2.jpg.webp

When assembling this shoulder piece to the mobile base, the flat rounded surface of the bearing shall touch the bottom piece of the base (*fig.* 28) and, due to its surface area, it shall keep the whole robot stable (no lateral vibrations) when rotating while it keeps the rotation action smooth.

The inner axle (carriage bolt) shall trespass the central hole of the bottom piece of the base and exit on the top of it with an extra height of 4 cm above it, inside the mobile base. At this stage, the shoulder piece is in contact from beneath the base and with it, but in order to attach it, an extra axial ball bearing of the same kind of the one presented before shall be placed with its center coincident with the center of rotation, with its lower side touching the surface of the bottom piece of the base; this shall be combined with a cylindrical plastic separator 20 mm external diameter, 12 mm internal diameter and 10 mm height (*plan 16*) that shall separate two M12 nuts from the bearing and which, due to nuts' pressure, shall hold tight the shoulder piece (and therefore the rest of the robot) to the base while allowing a smooth operation when rotating.

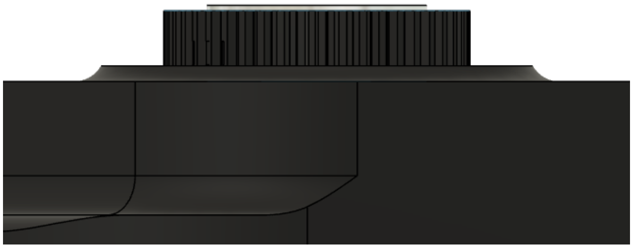


Figure 39. *Bearing visible over cogwheel's maximum height*

Source: Own, Autodesk Fusion 360

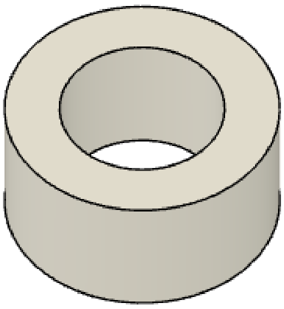


Figure 40. *Cylindrical plastic separator*

Source: Own, Autodesk Fusion 360

Finally, the remaining piece to be added to complete this mechanism is the rotation motor spur gear wheel (*plan 17*), which engages directly with shoulder's wheel. The size and number of teeth of this gear wheel shall be the same as that from the shoulder, which corresponds to a reduction relationship of 1:1 (this means that the angle inputted by the servomotor, its speed and torque shall be directly applied to robot's center of rotation but with opposite sense).

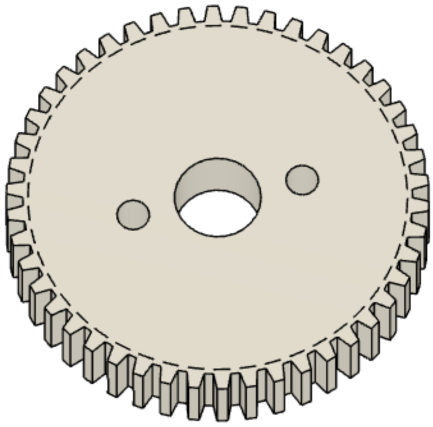


Figure 41. *Rotation motor cogwheel*

Source: Own, Autodesk Fusion 360 (GF Gear Generator)

The two passing holes present on the cogwheel correspond to the ones used for screwing it directly to the servomotor. With this and the shoulder piece, the resulting rotation gear set can be seen in the following figure:

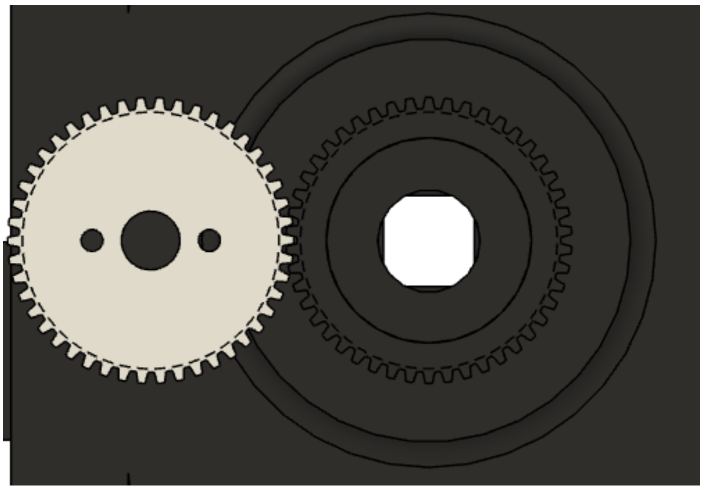


Figure 42. *Rotation gear set*

Source: Own, Autodesk Fusion 360 (GF Gear Generator)

When placing these pieces, the whole mechanism placed in the mobile base can be seen as it results on the following figure:

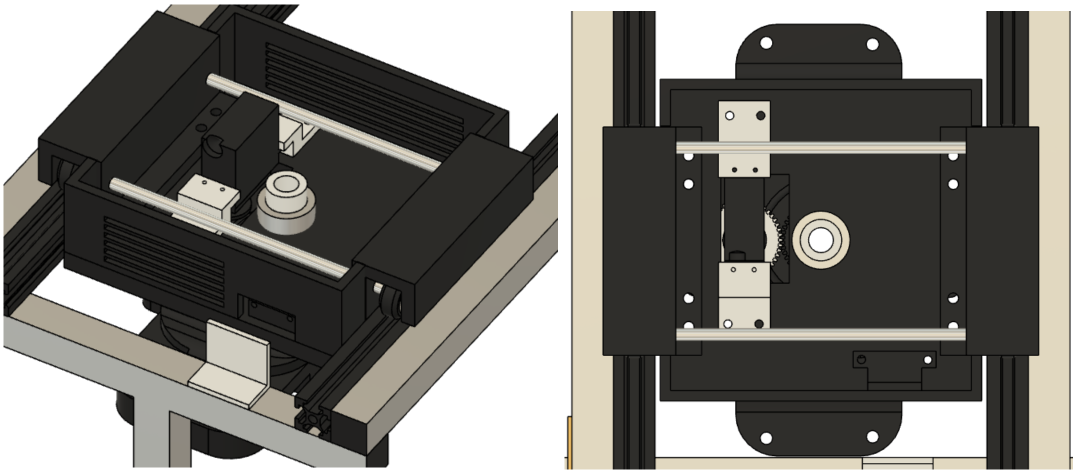


Figure 43. *Different views of the rotation section*

Source: Own, Autodesk Fusion 360

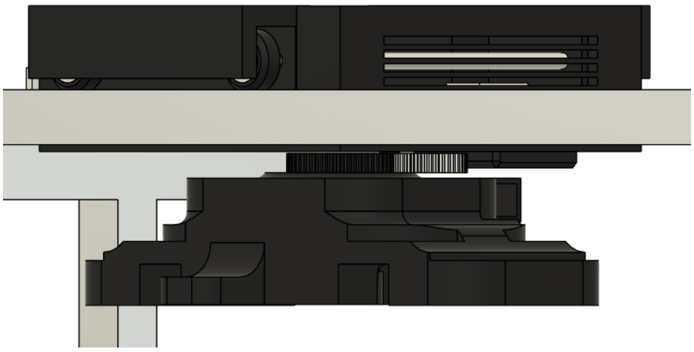


Figure 44. *Side 3D view of the rotation section*

Source: Own, Autodesk Fusion 360

The whole rotation mechanism shall have an approximate theoretical maximum speed of 8.1 rad/s. More detail on this calculation on Annex I.

Joint 3 section

Joint 3 is in charge of displacing the robot forwards and backwards at a certain height, being able to be related to human's "shoulder". For this, the use of two 25 kg·cm DS3225 servomotors shall be necessary (*fig. 33*).

The two servomotors used for this section shall be hosted by the shoulder piece presented on *fig. 36* and *fig. 37*. The motors shall be placed each one inside a casing which shall serve as an attachment to the shoulder (*plan 18*).

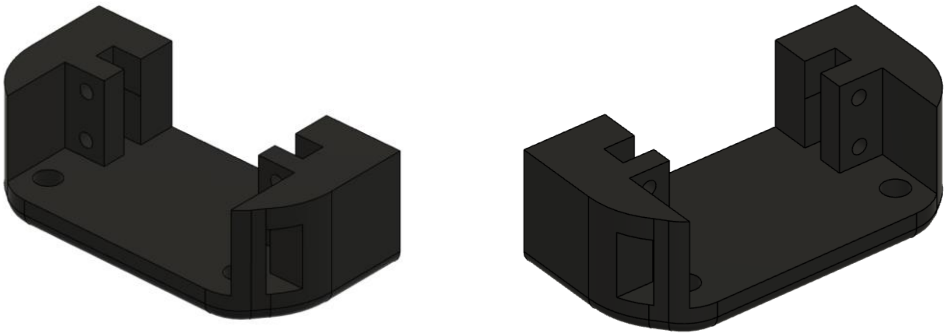


Figure 45. *Casing for left motor (left) and right motor (right)*

Source: Own, Autodesk Fusion 360

These supports for the servomotors have already-made holes that allow those actuators to be mounted and dismounted easily with screws. They are also attached to the shoulder joint by means of bolts that can be inserted or removed very easily. Each one has a square hole on one of their laterals so that the wires for the motors can be passed through easily. These wires are later passed through the square tube on shoulder's piece (*fig. 46*).

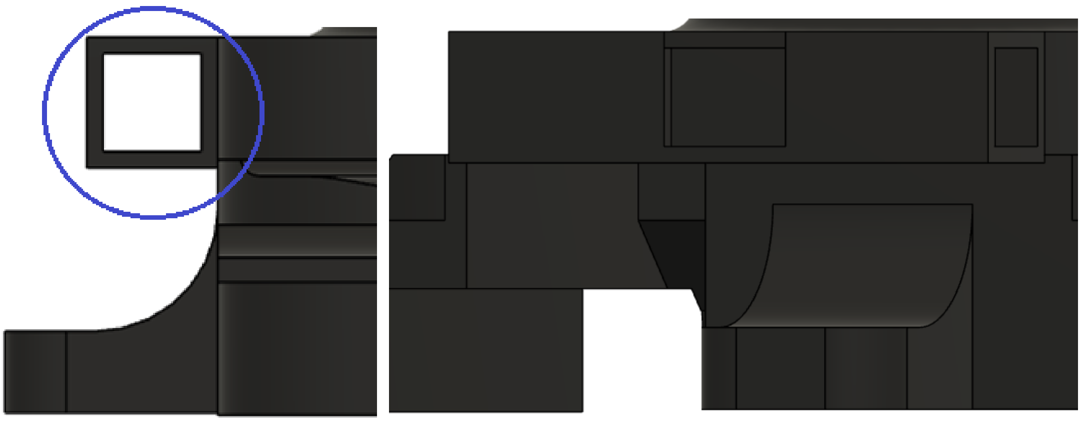


Figure 46. *Conduct for wires on shoulder's piece*

Source: Own, Autodesk Fusion 360

The motors held inside the shoulder joint are attached directly to the arm section. This arm section is divided in two parallel pieces (*plan* 19) each one opposing to the other. Each one consists of a rounded rectangular piece with a cylinder attached to the inner side. At the center axis of this cylinder there is a passing hole from the outer side of the piece. The shaft corresponding to the elbow joint (5 mm cylindrical aluminum rod with a length of 116 mm) is inserted on these cavities on the pieces.

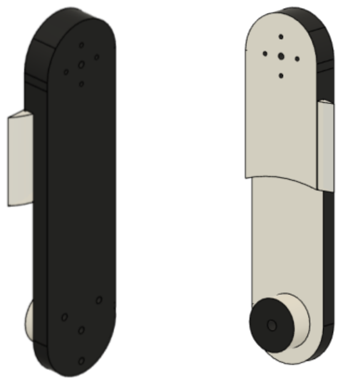


Figure 47. *Arm section pieces*

Source: Own, Autodesk Fusion 360

There are 5 different holes on the upper side of these pieces: four peripheral ones are used to screw it to motor's end, in order to fix it, whereas the central one is used in case a fast disassemble were necessary, as it serves to enter the screw that holds motor's end to motor, so that this way the whole piece can be extracted from the motor together with its end.

On the outer side of this last piece a rounded stop (*plan 20*) is screwed to the three holes that are peripheral to the central cavity reserved for the shaft. This mentioned stop can be seen on *fig. 48*.

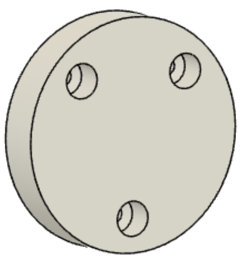


Figure 48. *Elbow shaft stopping piece*

Source: Own, Autodesk Fusion 360

When joining this mechanism pieces together, the result can be seen below:

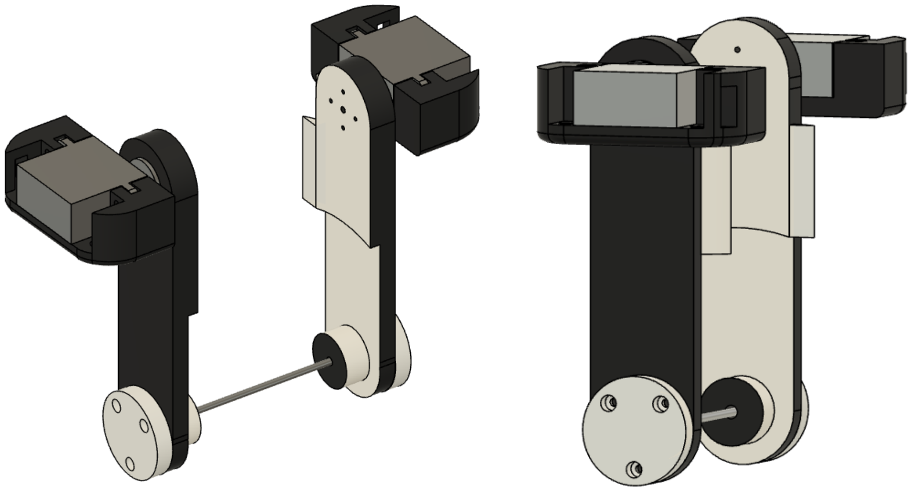


Figure 49. *3D views of the shoulder mechanism*

Source: Own, Autodesk Fusion 360

The previous mechanism is attached to the overall robot construction by means of bolts that pass through the horizontal surface of the supports for the motors (*fig. 45*) and up to the upper holes located on each side of the shoulder.

The resulting assembly can be seen below:

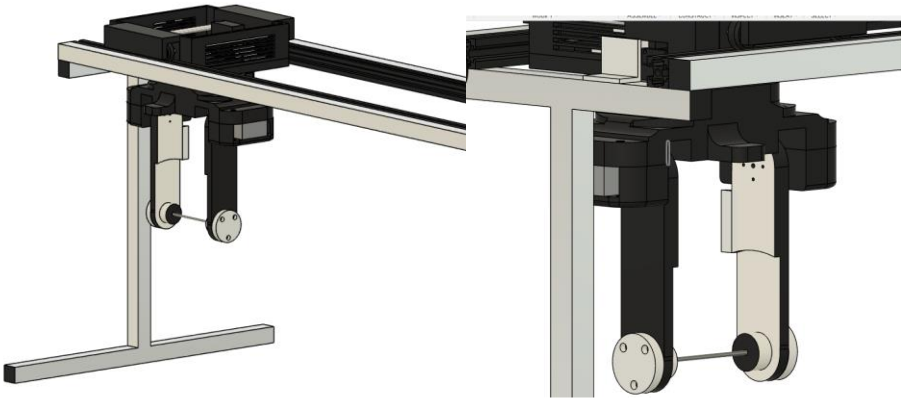


Figure 50. *3D views of the shoulder joint mechanism*

Source: Own, Autodesk Fusion 360

For this joint control, the two motors shall work in opposite ways one to the other.

This whole rotation mechanism has, like joint 2's mechanism, an approximate theoretical maximum speed of 8.1 rad/s. More detail on this calculation on Annex I.

Joint 4 section

This joint is in charge of lifting the forearm together with the wrist, the end-effector and the load. For this, the use of a 25 kg·cm DS3225 servomotor shall be necessary. It shall be located on the shoulder piece, with an additional part that shall be a support for this motor. Such piece (*plan 21*) can be seen on the following figure:

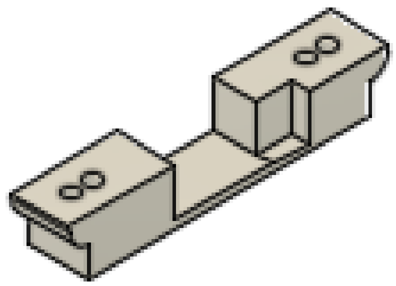


Figure 51. *Joint 4 motor holding piece*

Source: Own, Autodesk Fusion 360

This piece has four holes on one of its lateral sides in charge of fixing the servomotor so that, when this is placed, both parts can be screwed together (*fig. 52*). This part has four other passing holes from its top to its bottom that make it possible to fix it to the shoulder. Finally, it also has a cavity on the back of the motor which serves as extra space for the wires to exit this assemble.

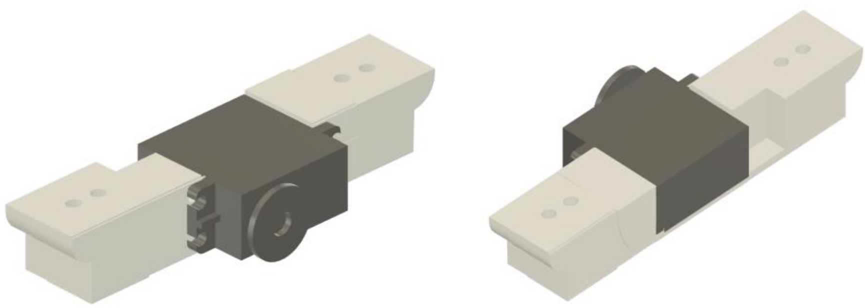


Figure 52. *Servomotor fixed in place with the piece*

Source: Own, Autodesk Fusion 360

In order to be assembled in place, a set composed by a bolt (from below the piece and through the passing holes), a washer (between the piece and its shoulder mounting), another washer (over the surface of the attachment section in the shoulder) and a nut (closing the set and attached to the bolt) are used for each of the holes. This assemble can be seen on *fig. 53*.

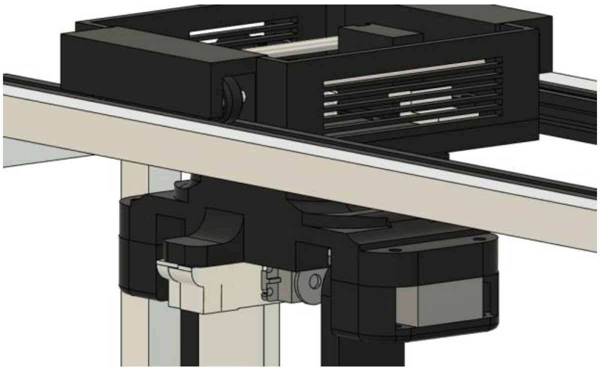


Figure 53. *Forearm lifting motor in place*

Source: Own, Autodesk Fusion 360

Another piece to be presented is what in anthropomorphic terms can be considered to be the "forearm" of the robot (*plan 22*). It is a piece holding the wrist and end-effector mechanisms as well as wrist's corresponding actuators. It is also the piece to which the servomotor and holding pieces pair (*fig. 52*) actuate, for which it has two shafts, being one the rotating axle and the other the actuated one that moves this part of the robot up and down.

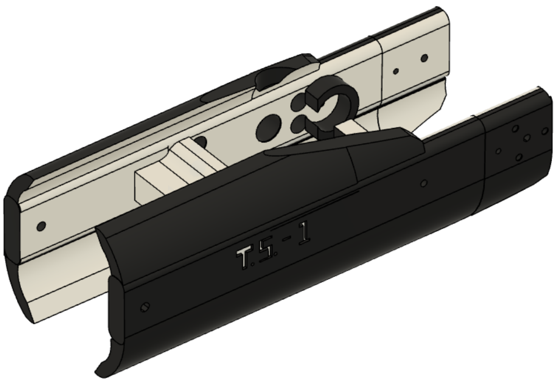


Figure 54. *Forearm raw structure*

Source: Own, Autodesk Fusion 360

The forearm is an elongated piece with two almost symmetrical pieces united by a central bridge. Next to the bridge, there is a casing for the wrist motor and four opposite holes which allow an easy screwing of such actuator to its place.

This piece has space for three different shafts: two dedicated for the vertical lifting movement and one last one for the wrist. In this part the first two ones shall be covered. Both these two shafts shall be cylindrical 5 mm aluminum rods (the same ones used for previous axes). The axle for the 4-link parallel mechanism receives the force of the motor and moves the forearm. They both have a total length of 70 mm.

Regarding the rotation axis, it needs to be said that two cylindrical cavities are attached to the inner side of each lateral of the piece. These cavities house ball bearings for 5 mm shafts with an external radius of 16 mm. The axle shall pass through the bearings and therefore through the aligned passing hole.

The assembled result can be seen on *fig. 55*:

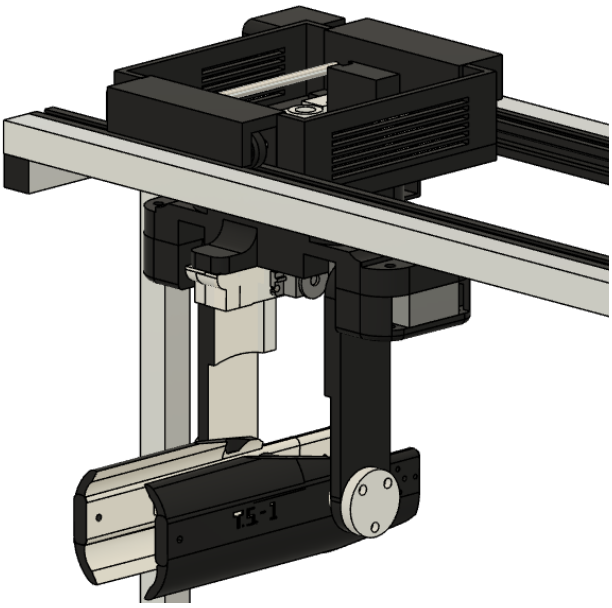


Figure 55. *Forearm in place*

Source: Own, Autodesk Fusion 360

Stopping the shafts at each side from exiting their cavity requires stops like those seen on *plan 27*.

Figure 56. *Stop (left) and stop in place (right)*

Source: Own, Autodesk Fusion 360

The previously mentioned 4-link mechanism is formed by two pieces: the first one is an adaptor for the servomotor (*plan 23*) and the second one is a bar-like link which transfers motion to the back shaft of the forearm (*plan 24*).

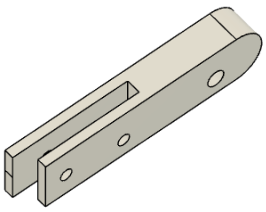


Figure 57. *Adaptor piece for the servomotor*

Source: Own, Autodesk Fusion 360

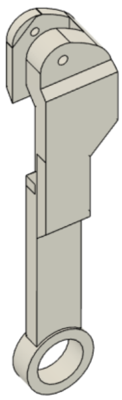


Figure 58. *Motor to shaft transmission link*

Source: Own, Autodesk Fusion 360

The piece on *fig. 57* has two holes passing through its central cavity that allow it to be attached to the servomotor plus another passing hole on its rear that, when placing a socket head shoulder screw, enables the other 4-link piece to rotate respect it.

The piece on *fig. 58* connects the adaptor from *fig. 57* with the rear shaft present at the forearm. For it to be connected with the adaptor via the bolt previously mentioned, it presents a hole that goes through its left side and until the half of the other side. In order to be connected with forearm's rear shaft, it presents a cavity on its lower side that enables ball bearing placement (the same type as those from the elbow rotation axis). This bearing is pressed against the wall on the inner forearm.

Two final kinds of pieces are going to be located on this section of the forearm. The first one is a 24 mm in diameter cylindrical plastic part (*plan 25*) with an inner passing hole of 5.5 mm in diameter which shall be placed on the rear shaft of the forearm and shall be able to freely rotate around it to make wires coming from the front of the robot roll when movement occurs without unnecessary tensions that can slow the robot down. The second one is a set of two antisymmetric pieces that elongate the robot on the back of the forearm and accomplish an esthetic function (*plan 26*).

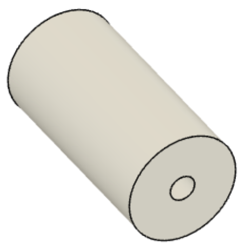


Figure 59. *Cylinder roll for wires*

Source: Own, Autodesk Fusion 360

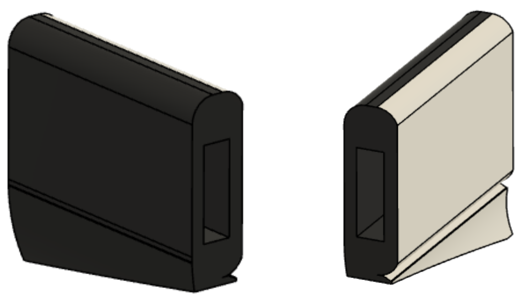


Figure 60. *Cosmetic pieces attached at the rear of the forearm*

Source: Own, Autodesk Fusion 360

After assembling the 4-link mechanism with the rolling cylinder and the esthetic pieces, the bearing from inside the motor to shaft transmission link (*fig. 58*) gets pressed against the inner side of the forearm and against the rolling cylinder.

The resulting assemble can be seen in the following figure:

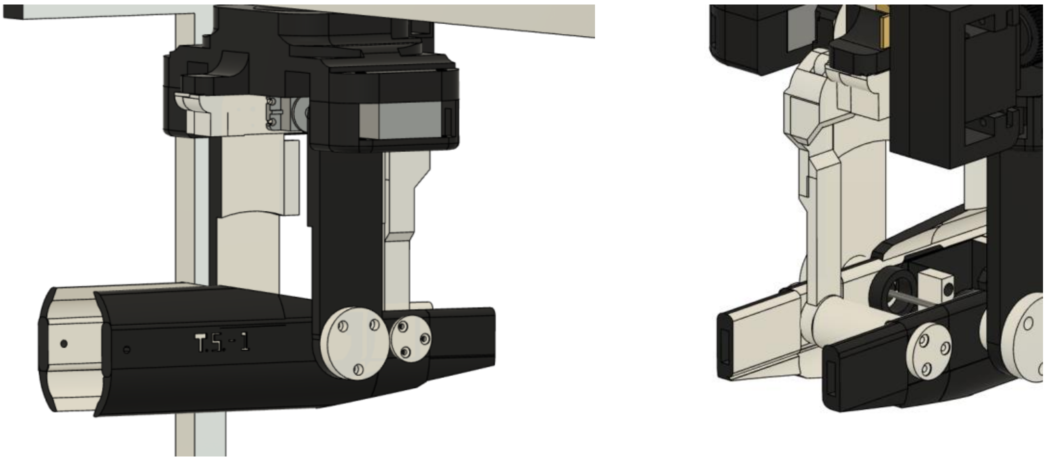


Figure 61. *Side views with 4-link mechanism and rolling cylinder in place*

Source: Own, Autodesk Fusion 360

The elevation mechanism has an approximate theoretical maximum speed of 8.1 rad/s. Calculations on this topic can be found on Annex I.

<u>Joint 5 section</u>

This section is in charge of moving the end-effector up and down from the wrist joint as well as holding the pieces in charge for this function (motors and mechanism parts).

The forearm has a cavity near the bridge responsible for holding the continuous rotation 5.5 kg·cm DS04-NFC servomotor. Such motor drives the belt transmission mechanism that drives the wrist joint up and down. The belt to be used is a 3D printing GT2 timing belt 6 mm thick, the same used for joint 1 transmission.

The belt transmission mechanism has two gears enabling its movement. The first one is a 52-tooth cogwheel with 30 mm of inner circle diameter and 8 mm passing hole diameter (*plan 28*). It is designed to fit the GT2 belt.

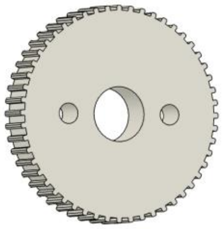


Figure 62. *Cogwheel designed for GT2 belt transmission*

Source: Own, Autodesk Fusion 360

This gear wheel is placed screwed directly to the servomotor accommodation piece.

Wrist's joint is formed by a 5 mm diameter cylindrical aluminum shaft with length 116 mm. This is the same type of shaft used for the rest of cases of equal application. The shaft at the wrist shall be tapped on both sides by means of the same pieces as the one seen on *fig. 56*.

The other side of the belt transmission, which rotates around previously mentioned shaft, consists of a piece that acts as the end-effector's base (*plan 29*), where the gripper shall be located. This whole piece has two gear wheels: one corresponds to the belt's transmission second half and the other one is also part of the base and transmits end-effector's movement to the position sensor.

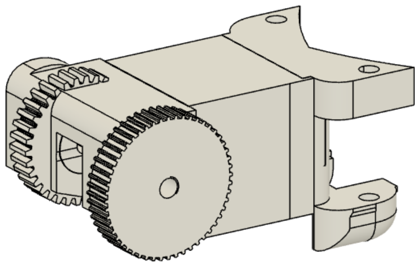


Figure 63. *End-effector's base piece*

Source: Own, Autodesk Fusion 360

The position sensor is attached to the bridge by means of the piece on *fig. 64* (*plan 30*), and on its central rotation piece a cogwheel exactly the same as the one embedded on the end-effector base is placed (*fig. 65*).

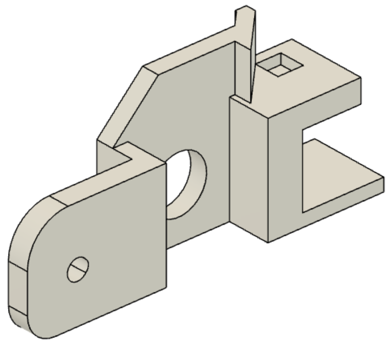


Figure 64. *Attachment of the position sensor*

Source: Own, Autodesk Fusion 360

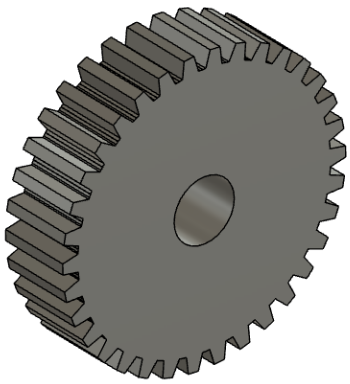


Figure 65. *Position sensor gear wheel*

Source: Own, Autodesk Fusion 360

The cogwheel on *fig. 65* (*plan 31*) has 34 teeth, an external circle diameter of 50 mm and a passing hole with diameter of 6 mm. The assembly of such gear wheel with the attachment of the position sensor placed on forearm's bridge can be seen on *fig. 66*.

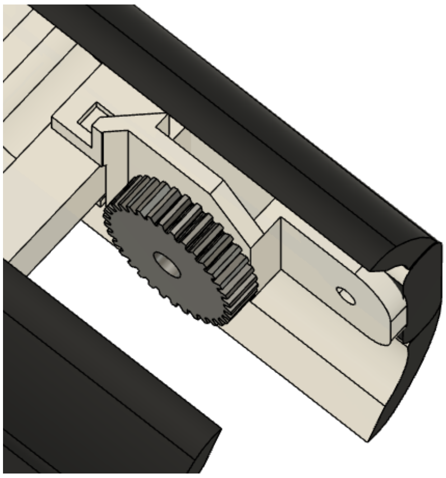


Figure 66. *Position sensor attachment with position sensor gear wheel in place*

Source: Own, Autodesk Fusion 360

Wrist's position sensor shall be caged inside the cavity formed by *fig. 64*'s piece with the forearm.

The whole wrist mechanism assemble can be seen on *fig. 67* (No belt represented).

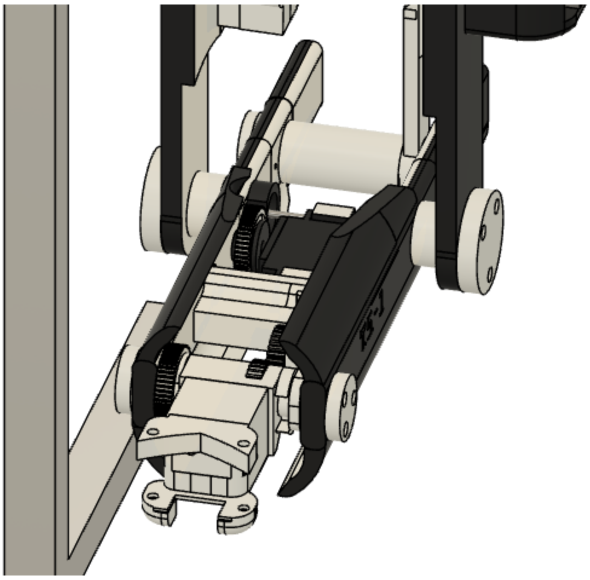


Figure 67. *Wrist transmission mechanism in place*

Source: Own, Autodesk Fusion 360

End-effector section

This section includes the description of the already-included end-effector of the device. It is, as described on section 1.4 (“Alternative solutions”), a mechanical gripper with two position servomotors and two 4-link parallel mechanism. This allows the gripper to open and close parallelly and therefore be able to directly modify its aperture by means of writing angles for each of the motors.

The used motors shall be SG90 position servomotors from manufacturer *TowerPro*. These motors shall be placed underneath the end-effector base, screwed each one inside their corresponding cavity (*fig. 68*).

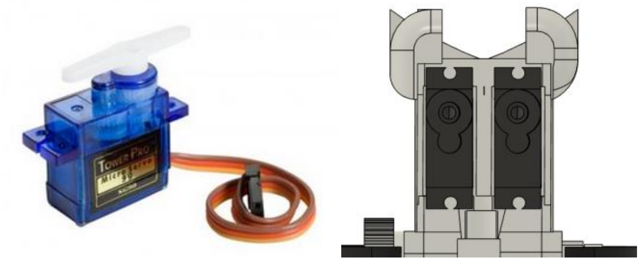


Figure 68. *SG90 position servomotors (left) and servomotors in place (right)*

Sources: https://tienda.bricogeek.com/3972-large_default/micro-servo-miniatura-sg90.jpg (left)

Own, Autodesk Fusion 360 (right)

They shall be using 13 mm bars they include when bought, which shall be the direct attachment between servomotors and the first stage of the mechanism.

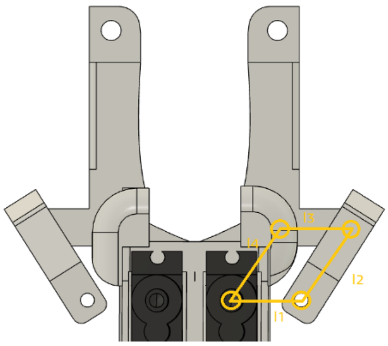


Figure 69. *First stage of the end-effector mechanism*

Source: Own, Autodesk Fusion 360

As seen on *fig. 69*, the first stage of the whole end-effector mechanism is already a 4-link parallel mechanism connecting the motors, the base of the end-effector and the upper gripper links. It has the first articulated point on the center of the motor, the second one at a distance l1 from this first one, the third point at a distance l2 from the second one and the fourth point at a distance l3 from the third. Distances l1 and l3 are the same and distances l2 and l4 are also the same. The lines connecting articulated points are parallel such that l1 and l3 are parallel as well as l2 and l4.

The first stage of the mechanism involves the use of two new pieces, being the first one a connector (*plan 32*) between the already-included bar that comes with the motor and gripper's upper link (*plan 33*), which is the second new piece. The first piece can be seen on *fig. 70*, whereas the second one can be seen on *fig. 71*.

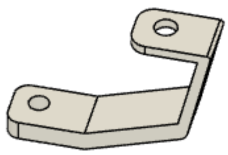


Figure 70. *Motor to gripper's upper link connecting piece*

Source: Own, Autodesk Fusion 360

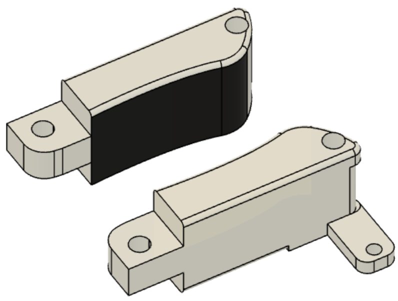


Figure 71. *Gripper's upper link pieces*

Source: Own, Autodesk Fusion 360

The piece on *fig. 70* uses its lower side to connect itself to the servomotor and its upper side to connect itself to gripper's upper link. To attach this connector to the motor, one screw shall be used on each side.

The piece on *fig. 71* corresponds to one of the two links used on the gripper. Each one can be considered to be the upper side of each "finger", what to humans it corresponds to metacarpals. They are attached to the base by means of M4 cylindrical aluminum shafts with length 35 mm and to the link in *fig. 70* by means of M3 countersunk machine screws with a length of 10 mm and M3 nuts.

The following stage of the end-effector is a second 4-link parallel mechanism (*fig. 72*).

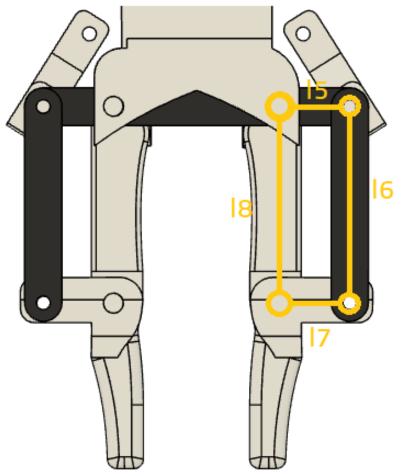


Figure 72. *Second stage of the end-effector mechanism*

Source: Own, Autodesk Fusion 360

This second stage of the mechanism also includes 4 articulated points. The first one is placed sharing space with the fourth articulated point of the first stage of the mechanism, the second one is placed at a distance l5 from the first one, the third one is placed at a distance l6 from the second one and the fourth one is placed at a distance l7 from the third one. Distances l5 and l7 are the same and distances l6 and l8 are also the same. The lines connecting articulated points are parallel in a way such that l5 and l7 are parallel between them and l6 and l8 are also parallel between them.

The second stage of the end-effector mechanism is composed, at each side, by three pieces: the first one has already been discussed and it corresponds to the piece on *fig. 71*, the second one is part of the second 4-link parallel mechanism connecting base with last link (*plan 35*) and the third one is the last piece of reach in the end-effector, corresponding to human's phalange (*plan 34*). There is also a central bonding piece between each side and the base (*plan 36*).

Figure 73. *Lateral link connector*

Source: Own, Autodesk Fusion 360

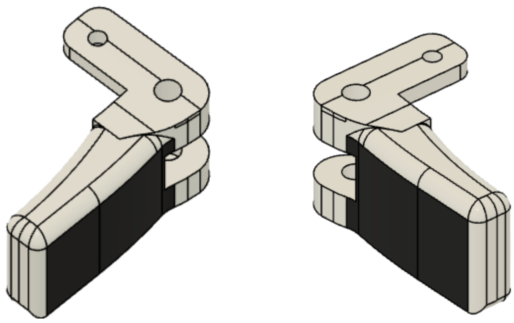


Figure 74. *Gripper's lower link pieces*

Source: Own, Autodesk Fusion 360

Figure 75. *Central bonding piece*

Source: Own, Autodesk Fusion 360

The central bonding piece is attached to the end-effector base due to the M4 cylindrical aluminum shafts with length 35 mm used previously to attach the piece from *fig.* 71.

Finally, the pieces on *fig. 74* are united to gripper's upper link pieces by means of M4 cylindrical aluminum shafts with length 15 mm.

This end-effector configuration can pick objects up to **88.34 mm** in width and has exchangeable gripper links (tip of the fingers).

On the upper side of end-effector's base there is a casing which holds the frontal light inside (*plan 37*). The arrow above indicates the direction in which the frontal light points.

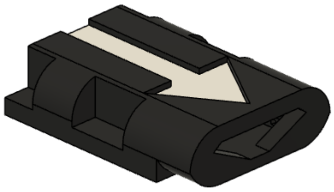


Figure 76. *Frontal light casing*

Source: Own, Autodesk Fusion 360

Finally, going from the end-effector back until where the servomotor controlling the wrist is there are two pieces, attached to the bridge by means of M3 chipboard screws with length 10 mm, which intend to receive the wires from end-effector's servomotors in order to be conducted to the rear side of the robot.

The lower piece out of them two (*plan 38*) shall be screwed to the bridge. The upper covering piece (*plan 39*) shall be screwed to that same bridge.

The wires on the rear side shall be left pending by gravity and shall be conducted to the area where the mobile base is, where they shall be connected to the electronics box. They shall be covered by 500 mm of black polypropylene spiral cable cover with inside diameter of 30 mm.

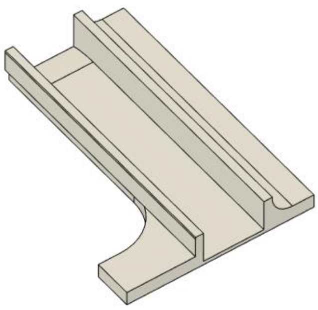


Figure 77. *Lower wire covering piece*

Source: Own, Autodesk Fusion 360

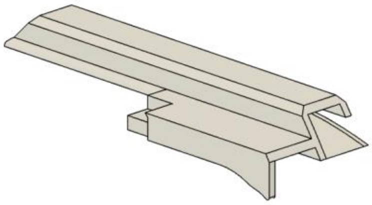


Figure 78. *Upper wire covering piece*

Source: Own, Autodesk Fusion 360

The upper conduit on *fig. 78* is used for the wires from the frontal light installed on the end-effector base, whereas the lower one is used for the wires coming from the servomotors, which pass underneath it and enter the main conduit on the lower wire covering piece from *fig. 77*.

The whole wrist and end-effector sections can be seen together below:

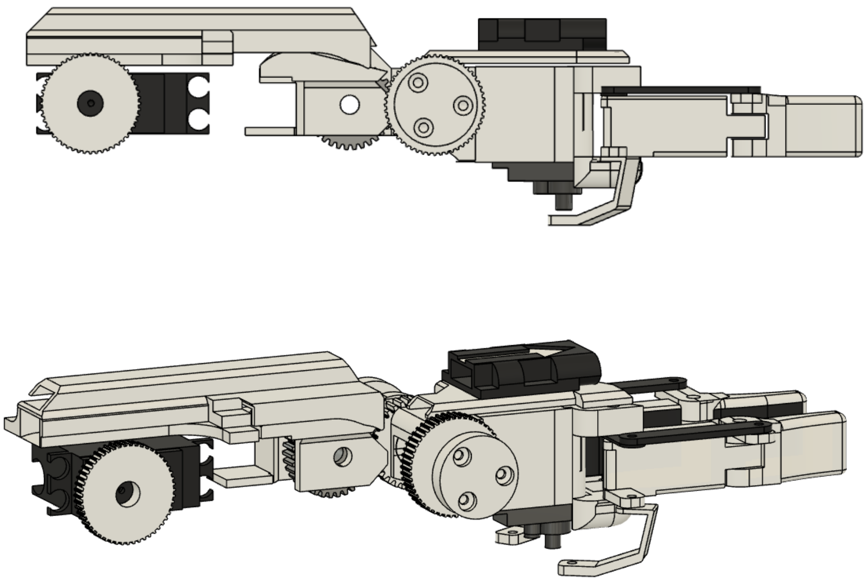


Figure 79. *Assembly of the wrist and end-effector sections*

Source: Own, Autodesk Fusion 360

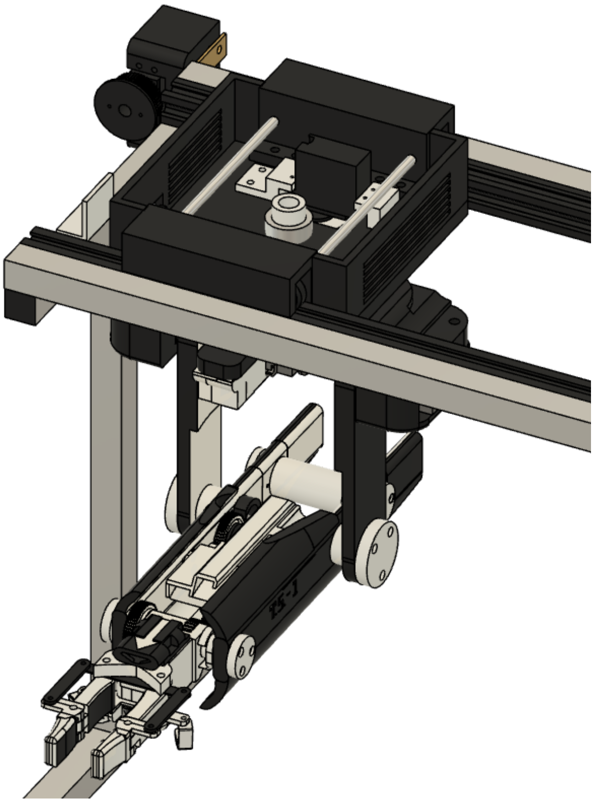


Figure 80. *Mechanical assembly of the robot with no electronics box*

Source: Own, Autodesk Fusion 360

1.5.3. MECHANICAL MATHEMATICAL MODELS

These models comprise the kinematic and dynamic models a user could find useful when using or programming the robot for different tasks they shall define.

The inverse kinematic model shall be an already built-in function that the user shall be able to use.

1.5.3.1. Kinematic models

These models come from the study of pure kinematics, regardless of the dynamics of the loaded or unloaded system. They are useful for the user so that they can calculate the reach of the robot for different tasks, be able to simulate with different simulation software programs (*Matlab*, *CoppeliaSim*, etc.) by placing mathematical reference frames, etc.

Kinematic joint limitations

The different types of motors used have different ranges in what to maximum angles it concerns. This becomes a limitation when controlling them since surpassing these limits on a direct order to the robot (no filtering stages left on interface) can result in disaster both for the machine and potentially for the user and their environment. The correct definition of these maximum and minimum angles is necessary to avoid undesired effects on operation.

Joints 1 and 5 use continuous rotation servomotors. These motors have 360 + degrees of rotation available. However, the joints they actuate on do not. Depending on which joint is being treated, they shall have different final points they can reach.

Position servomotors used have a rotation range of 270°. This is the base rotation limit for joints 2, 3 and 4, which use this type of motors. However, each of them has specific limitations due to their mechanisms.

The frame along which joint 1 translates has a length of 800 mm. However, as the robot has some real width (it is 280 mm wide from shoulder to shoulder on their outside), this is not the absolute limit. Due to this width, half of it on each side shall not be available when displacing laterally for the previously mentioned safety measures. Besides, structure's width on each side also takes away 20 mm from each lateral. Therefore, θ_1 shall remain on its lower side equal or over 160 mm to be able to be reached by the robot and on its higher side it must remain equal or under 640 mm. A greater operation area can be obtained by means of arriving to one of the limits and rotating so that the end-effector can get outside this limitation.

Joint 2 has a wide angle of actuation, since no mechanical limitations outside its own position servomotor limitations can be found. Therefore, the reach for this joint is θ_2 existing between 0° and 270°, both included.

Joint 3 has position servomotors directly attached to it. Their range is the same as for joint2. However, when to the mechanism it concerns, it introduces restrictions to the movement, being this the superior shoulder piece. Range of movement has been limited by servomotor placement and shoulder constrictions to be from 0 to 180°, being the 90° level straight-pointing downwards perpendicularly to the ground horizontal plane.

Joint 4 has a four-link bar mechanism attached to position servomotors. They introduce a limitation when colliding the bar attached to the motor on the roof of shoulder's inner cavity. This upper collision is put together with forearm's collision with vertical motor attachment piece and restricts the angle of operation from 40° (that is 50° from the horizontal position of the forearm parallel to the ground plane, position, which is considered to be 90°) to 175°.

Joint 5 has a continuous rotation servomotor at its motion command. It allows 360 + degrees of non-stop rotation. However, due to collisions with the wire holding parts on its upper part it has been decided to put the 0° reference 90° up from its horizontal position (therefore, perpendicular to the plane ground). The lower limitation has been considered to be 200° due to possible collisions of the end-effector links with the sides of the robot.

Below, a table summarizing joint limitations can be seen:

Joint	Limitations
θ_1	$\theta_1 \in$ [160, 640] mm
Θ_2	$\Theta_2 \in$ [0, 270] °
Θ_3	$\Theta_3 \in$ [0, 180] °
Θ_4	$\Theta_4 \in$ [40, 175] °
Θ_5	$\Theta_5 \in$ [0, 200] °

Table 19. *Joint limitations*

Source: Own

Denavit-Hartenberg parameters

The DH parameters represent a way in which several reference frames can be placed along the different axis relating each one to their immediate previous and, in the end, allow transformations to be made between the first reference frame and end-effector's reference frame for direct kinematics.

DH calculations as well as each of the reference frames placed on the several joints can be checked on Annex I on A1.3, where detailed emphasis on each step of the procedure is presented. The table where these parameters are presented in meters and radians units can be seen below (*table 20*):

Joint	Type	θ	d	a	α
1	**P**	0	θ_1	0	$-\frac{\pi}{2}$
2	**R**	θ_2	-0.0785	0	$-\frac{\pi}{2}$
3	**R**	$\theta_3 + \frac{\pi}{2}$	0	0.124	0
4	**R**	$\theta_4 - \frac{\pi}{2}$	0	0.134	0
5	**R**	θ_5	0	0.1	π

Table 20. *Denavit-Hartenberg parameters*

Source: Own

Direct kinematics model

This model aims at obtaining the final cartesian coordinates and orientation for the end-effector (x, y, z, α, β) given each motor's orientation (θ_1, θ_2, θ_3, θ_4, θ_5). The α variable represents end-effector's y-axis rotation, whereas β is the orientation of the end-effector when exerting a z-axis rotation.

When considering the implementation of this mathematical model as an algorithmic function with outputs given specific inputs, there shall be two different approaches, one being more sequential, as a series of steps with intermediate variables that avoid repetition and saving computation time while the second one is more direct, saving memory from unnecessary variable creation.

On the table below, the general equations for the model are given so that the different coordinates and orientations can be calculated directly from the angles. However, a more optimized approach can be made starting from further developed equations on Annex I, A1.3, where these results come from.

Variable	Value
α	$\alpha = \theta_2 - 45°$
β	$\beta = \theta_5 + \theta_4 + \theta_3 - 180°$
x	$x = (d1 \cdot \cos(\theta_3) + d2 \cdot \sin(\theta_4 - 90 + \theta_3) + lw \cdot \cos(\theta_5 + \theta_4 + \theta_3 - 180°)) \cdot \sin(\alpha)$
y	$y = (h1 - d1 \cdot \sin(\theta_3) + d2 \cdot \cos(\theta_4 + \theta_3 - 90°)) - lw \cdot \sin(\theta_5 + \theta_4 + \theta_3 - 180°)$
z	$z = \theta_1 + (d1 \cdot \cos(\theta_3) + d2 \cdot \sin(\theta_4 - 90 + \theta_3) + lw \cdot \cos(\theta_5 + \theta_4 + \theta_3 - 180°)) \cdot cos(\alpha)$

Table 21. *Direct kinematics model equations*

Source: Own

Inverse kinematics model

This model obtains the different angles each motor shall take when trying to reach a position. It shall be developed as a mathematical model and later translated into C++ code in order to be implemented.

This robot has 5 DoF, which are distributed into 3 space coordinates (cartesian coordinates) and 2 angular coordinates for base and wrist desired orientation. The device is capable of therefore reaching a determinate object in two-axis rotations to accommodate interaction.

The developed mathematical I.K. model for the robot given a 5-variable input (x, y, z, α, β) and a 5-variable output (θ_1, θ_2, θ_3, θ_4, θ_5) with from right to left each angle corresponding to a different joint in order (global linear displacement, global rotation, shoulder rotation, elbow rotation, wrist rotation) can be seen below, with the specific mathematical development performed to reach the solution in this section presented can be seen on Annex I, section A1.3.1.

Joint	Value
θ_1	$\theta_1 = 780\,mm - (z - x \cdot tg(135° - \alpha))$
θ_2	$\theta_2 = \alpha + 45°$
θ_3	$\theta_3 = \arccos\left(\frac{d2^2 - \sqrt{\left(\frac{x}{\cos(135° - \alpha)} - lw \cdot \sin(180 - \beta)\right)^2 + (y + lw \cdot \cos(180° - \beta) - h1)^2} - d1^2}{-2 \cdot d1 \cdot \sqrt{\left(\frac{x}{\cos(135° - \alpha)} - lw \cdot \sin(180 - \beta)\right)^2 + (y + lw \cdot \cos(180° - \beta) - h1)^2}}\right) + \arccos\left(\frac{\frac{x}{\cos(135° - \alpha)} - lw \cdot \sin(180 - \beta)}{\sqrt{(\frac{x}{\cos(135° - \alpha)} - lw \cdot \sin(180° - \beta))^2 + (y + lw \cdot \cos(180° - \beta) - h1)^2}}\right)$
θ_4	$\theta_4 = 90° - \arcsin\left(\frac{y + lw \cdot \cos(180° - \beta) - (h1 - d1 \cdot \sin(\theta_3))}{d2}\right)$
θ_5	$\theta_5 = \beta + \theta_4$

Table 22. *Inverse kinematics model equations*

Source: Own

1.5.3.2. Dynamic models

These models illustrate what the maximum weights are that the robot can hold for certain positions. In order to calculate these weights, the worst-case scenario configurations have been taken into account from each joint to be applied. These joints to be analyzed shall be shoulder joint, elbow joint and wrist joint since the rest of them have no effect on lifting capabilities.

The analysis has been made from each joint up to a mass placed at the end of the end-effector. Calculations performed can be checked on Annex I, section A1.3.2.

The final weight that can be lifted by the robot shall depend on the minimum of the weights able to be carried from the joint with lesser power, since this shall be a restrictive factor.

The results can be seen on the following table:

From joint	Maximum weights to be hold by each joint
From joint 3 (shoulder)	1.038 kg
From joint 4 (elbow)	0.795 kg
From joint 5 (wrist)	0.334 kg

Table 23. *Maximum weights the device can hold straight*

Source: Own

As seen on *table 23* and with everything stated earlier on this section, the most restrictive joint regarding holding weight can be said to be the wrist joint (joint 5), with a maximum lifting weight of **0.334 kg**, which shall be the maximum the robot can hold.

However, with a specific situation in which the wrist is locked in place, the maximum weight the robot could hold can increase up to **0.795 kg**.

Both of these weights comply with user specification 1-B2 on section 1.3.2.

1.5.4. ELECTRONICS

This section presents the solution implemented for the whole electronics of the robot, in charge of the power supply, the communication modules and the main control unit (responsible for coordinating and ruling the rest of subsystems).

The whole electronics diagram can be seen on *plan 49*.

1.5.4.1. Power supply & wires

The power supply is the source of power (voltage and current) of electronic systems. This device needs to feed three types of subsystems: the µC, the high torque servomotors and finally the rest of modules and smaller servos.

The three global electronic subsystems are fed by means of a 36 W (12 V and 3 A output) transformer with reference KDTR1236W for each of them (3 in total).

Figure 81. *KDTR1236W transformer*

Source: https://greenice.com/19739/transformador-led-230vac-12vdc-36w-3a.jpg

The supply for the µC shall be direct, connecting the male connector of the transformer to the female connector of the Arduino Mega 2560. The supply for the high torque servos and for the other modules plus the smaller servos shall be provided through a relay circuit controlled by the µC.

Relay power control circuit

This circuit is in charge of safely disconnecting the supply, enabling the proper robot to self-unplug from the grid its motors and several other subsystems in case of the user willing this or automatically if certain safety concerning condition has been reached.

The input for this control circuit shall be performed by means of a DC jack female connector (5.5 mm x 2.1 mm) at a reason of 1 for each of these circuits (2 in total).

Figure 82. *DC jack female connector*

Source: https://www.tiendatec.es/4140-large_default/conector-dc-jack-hembra-55x21mm-soldar.jpg

This circuit is composed by three different pieces: a G5LE-1 DC5 relay from manufacturer *Omron Electronics Inc.* with a nominal coil tension of 5 V and a maximum current flow of 10 A (*fig. 83*), a 2N7000 N-channel MOSFET from manufacturer *On Semiconductor* (*fig. 84*) and a 1N4001 diode from manufacturer *Diode Incorporated* (*fig. 85*).

Figure 83. *Omron Electronics G5LE-1 DC5 relay*

Source: https://www.digikey.es/product-detail/es/omron-electronics-inc-emc-div/G5LE-1%2520DC5/Z1014-ND/280368

Figure 84. *On Semiconductor 2N7000 N-channel MOSFET*

Source: https://media.digikey.com/Renders/~~Pkg.Case%20or%20Series/TO-92-3(StandardBody),TO-226_straightlead.jpg

Figure 85. *Diodes Incorporated 1N4001 diode*

Source: https://media.digikey.com/Photos/Diodes%20Photos/261-DO-41.jpg

The output of connectors from *fig. 82* shall be connected to the proper control circuit, with the positive polarity entering the normally open terminal (one of the commutable inputs) and the ground polarity entering directly to the next stage.

The diode shall be connected in antiparallel to the coil of the relay. The output of the coil connected to the anode of the diode shall be connected to the drain of the MOSFET. The input of the coil shall be connected to its 5 V supply (provided by the 5 V output of the Arduino Mega 2560 board). The gate of the MOSFET shall also be connected to the Arduino Mega 2560, providing this the pulsing signal to activate or deactivate the relay whenever it is needed. The source of the MOSFET shall be connected to the GND pin of the Arduino board, closing the circuit.

The resulting circuit can be seen on a Proteus simulation on *fig. 86*:

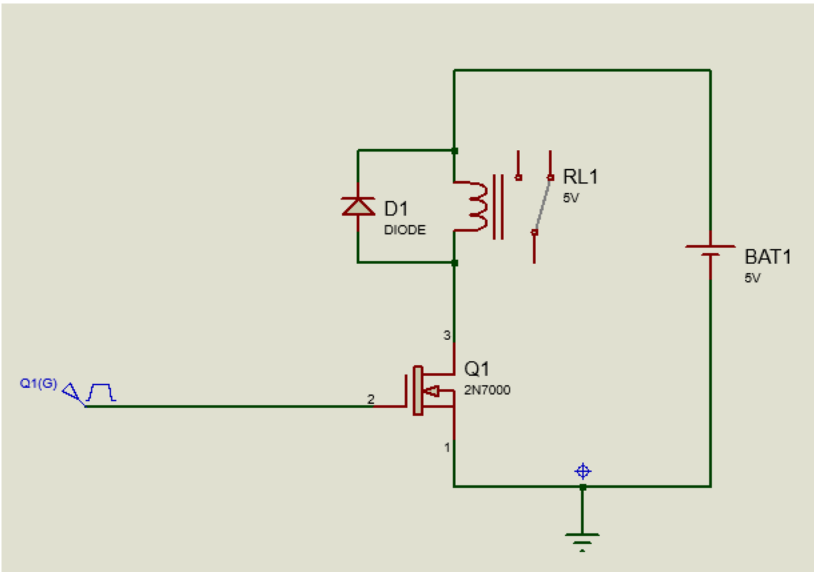


Figure 86. *Relay power control circuit*

Source: Own, Proteus 8 Demonstration

When a HIGH (+5V) pulse is applied to the gate of the MOSFET, the relay is switched to the normally open terminal where the positive polarity of the supply is connected, allowing the flow of current to the common terminal. Otherwise, when a GND (0V) level is applied to the same terminal of the MOSFET, the normally closed terminal of the relay, left floating, is connected to the common terminal, not allowing any current flow since the circuit with the power source is not closed.

<u>**Safety power switch**</u>

This power switch is only used for the high torque servomotors, and it allows or impedes the flow of current inside this subsystem. It is placed on the top of the electronics box (which shall be developed later on section 1.5.4.5 ("Electronics box")).

Such switch shall be a RA1113112R power switch from *E-switch* manufacturer (*fig. 87*):

Figure 87. *E-switch RA1113112R power switch*

Source: https://media.digikey.com/Photos/E-Switch%20Photos/RA1113112R.jpg

This switch has one of its terminals connected to the common terminal of the relay from the power control circuit corresponding to the high torque servomotors, with its other terminal going to the voltage level adaption circuit. This allows to allow or block current from entering the inner stages of the system.

<u>**Voltage level adaption**</u>

Each subsystem shall have an adapted level of voltage to suit its necessities. The level needed for the µC shall be regulated by the proper Arduino board, having no problem in doing so when supplied with the 12 V from the KDTR1236W transformer (inside Arduino Mega's input supply voltage boundaries). The voltage that the high torque servomotors shall need is 6 V, so the input voltage needs to be regulated (50 % of the supply voltage). The tension level for the rest of the modules and the lesser powerful servomotors shall be 5 V (41.67 % of the supply voltage).

In order to perform the voltage regulation, LM317 linear voltage regulator modules shall be used (*fig. 88*):

Figure 88. *LM317 linear voltage regulator module*

Source: https://media.cablematic.com/__sized__/images_1000/ak00600-01-thumbnail-1080x1080-70.jpg

The voltage supply shall be adapted to each of the two power rail levels needed by means of the included potentiometer. One of these regulators shall be adapted to 6 V and the other one shall be adapted to 5 V.

The 6 V power rail shall result from the output of the LM317 regulator, with its input connected, regarding the positive polarity, directly to the safety power switch and regarding its ground level to the ground of the DC transformer (previous to the relay triggering circuit).

The 5 V power rail shall be outputted by the LM317 regulator, with its input directly connected to one of the outputs of the relay power control circuit.

End-effector servomotors activation circuit

End-effector position servomotors shall be supplied by the 5 V power rail. They shall be switched ON and OFF by means of another relay power control circuit, which shall be activated by means of the software uploaded to the µC. This shall allow to dampen the power consumption when the motors are not needed.

Wires used on the device

There shall be two types of wires on the device regarding their functionality: higher power cables and lower power cables.

The first of the types aforementioned shall be used for more power-demanding subsystems, connecting several circuits among them, being especially dedicated to connecting the power rails with the acceptance circuit (power control circuit and voltage regulation) and to connect the servomotors to power. These wires shall be 2 x 0.5 mm^2 plane electric wires with two lines (one for each polarity) with maximum tension withstanding of 300 V and maximum current allowed of 5 A (*fig. 89*):

Figure 89. *2 x 0.5 mm2 plane electric wire*

Source: https://cdn1.efectoled.com/541816-thickbox_default/cable-electrico-manguera-2-x-05mm-para-tiras-led-12v-dc-2p.jpg

The red color shall be used for the positive polarity and the black color shall be used for GND. This shall be particularly useful to understand when explaining connections on section 1.5.4.4. ("µC & connections").

The second type of cables to be used need to withstand less power: they shall be used for signal transmission and lower power systems' supply. The elected wires are the *TUOFENG 1007 22 AWG* wires (*fig. 90*):

Figure 90. TUOFENG 1007 22 AWG wires

Source: https://images-na.ssl-images-amazon.com/images/G/30/apparel/rcxgs/tile._CB483369954_.gif

1.5.4.2. Wireless communications

The wireless communications of the robot are an important part of its functionality. There shall be two types of communication modules on the device: a Bluetooth communication module and a radio communication module.

Bluetooth communication module

As developed on section 1.4.3.2. ("Wireless communications"), the chosen module for this functionality shall be the HC-05 Bluetooth module (*fig. 91*):

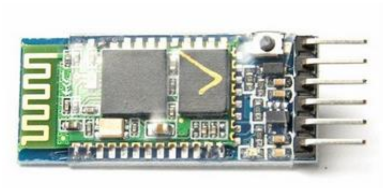


Figure 91. *HC-05 Bluetooth module*

Source: https://solectroshop.com/1805-small_default/modulo-inalambrico-arduino-hc-05-bluetooth.jpg

This module has two supply pins for the positive polarity (it works from 3.3 V to 6 V) and the other one for GND level. There are two pins, RXD and TXD, which work as serial port input and output respectively. The KEY and STATE pins of the module shall not be used for this project.

Radio communication module

The chosen module for this functionality, as explained on section 1.4.3.2. ("Wireless communications"), is the HC-12 module (*fig. 92*):

Figure 92. *HC-12 radio module*

Source: https://www.electromaker.io/uploads/media-library/825472/113990039_SPL.jpg

This radio module works at a frequency of 433 MHz and has two supply pins (+5 V and GND), two serial port pins (RXD and TXD, with the same functionality as the ones for the HC-05 module). It has a range of up to 1 km.

1.5.4.3. Sensors

There are two types of sensors on this device: a laser distance-measuring sensor, responsible for measuring joint 1 displacement, and a potentiometer, which measures joint 5's wrist angle.

Laser distance sensor

The distance sensor on joint 1 shall be a laser distance sensor in charge of measuring the distance existing between itself and a lateral barrier placed on the structure as a stop. The laser module used is a VL53L0X sensor (*fig. 93*):

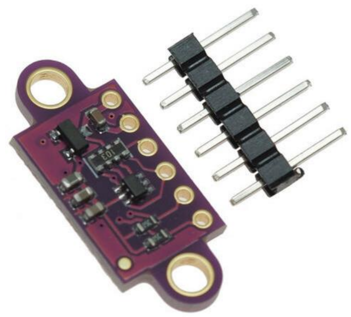


Figure 93. *VL53L0X laser distance sensor*

Source: https://q.ebaystatic.com/aw/pics/cmp/ui/imgzoommask_50x50.png

This sensor has 6 pins: two of them are used for the power supply (+5 V and GND), two other pins are used for communications with the µC through its I2C port (SDA and CLK) and the final two pins are used for generating an interruption (GPIO1) and for shutdown (XSHUT, with reverse logic, LOW to activate).

Potentiometer as a sensor

A potentiometer is a variable resistance that changes its resistivity value according to its shaft orientation, which makes it possible to measure angles by measuring the different resistivities as a voltage divider.

The potentiometer used in the robot is a linear 10 kΩ potentiometer (*fig. 94*):

Figure 94. *10 kΩ linear potentiometer*

Source: https://www.cetronic.es/sqlcommerce/ficheros/dk_93/productos/451047005-1.jpg

It has 3 pins, one for each side's polarity and the third one behaving as the output.

1.5.4.4. µC & connections

The chosen microcontroller shall be an *ATmega2560* in the form of an Arduino Mega 2560 board, as explained on section 1.4.3.1. ("Control component"). This board can be seen on its physical form on *fig. 95,* whereas its pinout, useful for later connections to the rest of the components, can be seen on *fig. 96*:

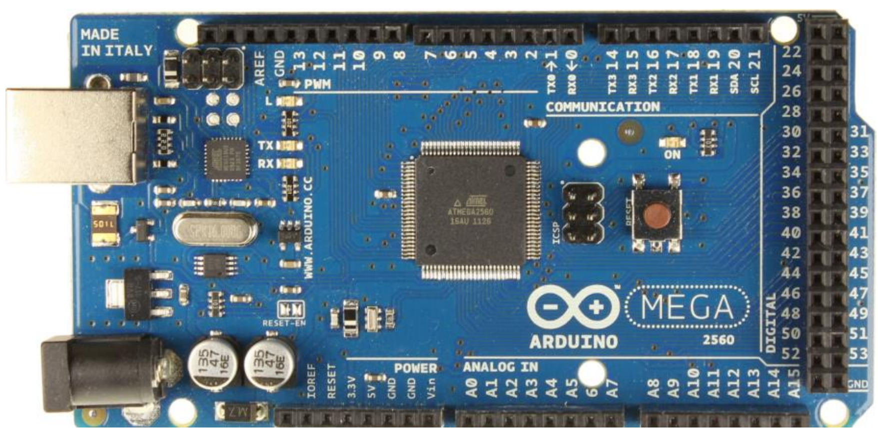


Figure 95. *Arduino Mega 2560 board*

Source: https://manuais.iessanclemente.net/images/8/80/ArduinoMega.jpg

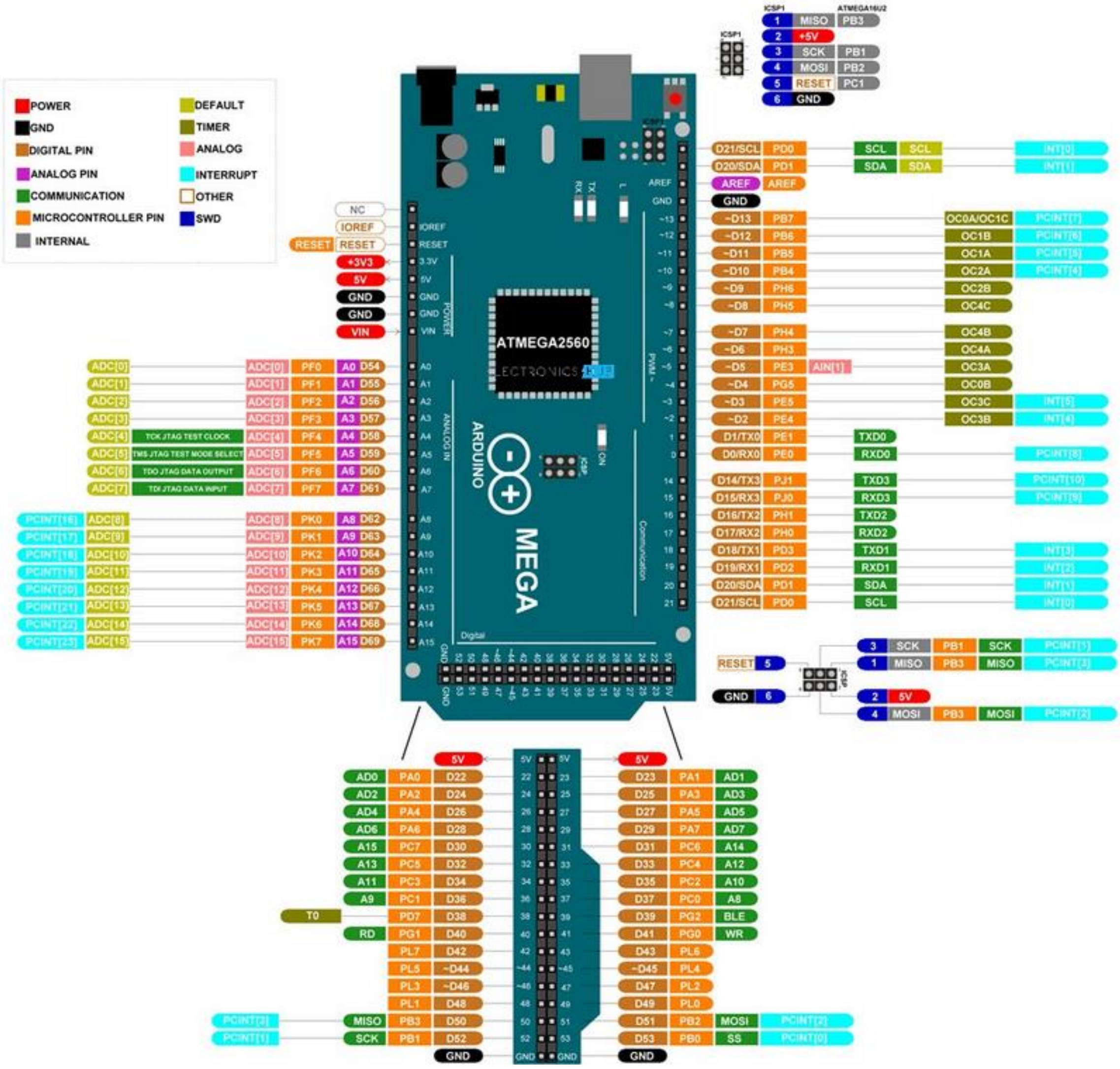


Figure 96. *Arduino Mega 2560 pinout*

Source: https://www.electronicshub.org/wp-content/uploads/2021/01/Arduino-Mega-Pinout.jpg

This past figure shall be taken as a reference when presenting the different connections made regarding the µC and the rest of peripherals, and the name for each of the ports to be referred shall always be directly extracted from that image.

The different pins to be found on the board are coded both by means of colors and by means of the starting letters of each socket, having a "D" for digital pins, an "A" for analog pins, etc. They also have their corresponding port name and pin referred directly to the controller, although this shall not be used in the following explanations.

Nature of the connections

When it concerns to the conditions on which the connections have been made, several rules have been established:

1. Every end of soldered electrical connections shall be covered by means of heat shrink sleeve independently of the kind of wire used.

2. Given a situation in which the connection needs to be made to a header socket, that end of wire shall be soldered to a header pin and present that solder covered with a heat shrink sleeve.

3. Every connection shall be made in such way that there is little more wire used than the strictly necessary to ensure good maneuverability when working on this area.

4. Connections needed to be done by means of soldering shall necessarily be covered by heat shrink sleeve.

Once these rules have been established regarding the nature of the connections, it is necessary to include what the header pins and sockets to be used are. The heat shrink sleeve to be used can be seen on *fig. 97*, it is a 6.4 mm of inner diameter *BPX0064* heat shrink sleeve colored in black. The header pins used from *TE Connectivity* manufacturer can be seen on *fig. 98*. The header sockets from *Harwin* manufacturer can be seen on *fig. 99*.

Figure 97. *6.4 mm inner diameter BPX0064 heat shrink sleeve*

Source: https://www.cetronic.es/sqlcommerce/ficheros/dk_93/productos/999112240-2.jpg

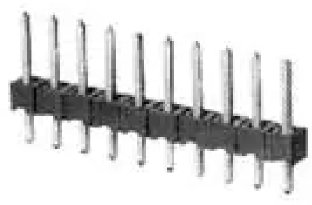


Figure 98. *Header pins to be used*

Source: https://www.mouser.es/images/tycoelectronics/images/102972-1_SPL.jpg

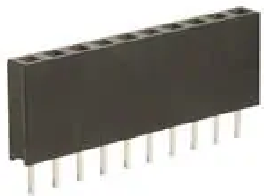


Figure 99. *Header sockets to be used*

Source: https://www.mouser.es/images/harwin/images/M20-782_SPL.jpg

Power supply and end-effector activation connections

The relay power control circuit shall be assembled on solder boards (2 circuits per each board). The 6 V rail power control circuit and the 5 V rail power control circuit shall be soldered to the same board. The end-effector power control circuit shall be attached on a separate solder board. The used solder boards shall be 30 x 70 mm tinned universal PCB boards (*fig. 100*).

Figure 100. *Tinned universal PCB boards*

Source: https://img.dxcdn.com/productimages/sku_490213_1.jpg

Power supply connections from the relay power control circuit are composed by thick wires (*fig. 89*) with header pins at each end that shall connect the different unconnected tracks on the solder board. Signals from the µC to the circuit shall be transmitted through thin wires (*fig. 90*).

The relay power control circuit shall be connected to the µC by means of the gate pin on the MOSFET transistor. The circuit that controls the 6 V power rail shall be connected to digital pin D30 on the Arduino Board. The circuit that controls the 5 V power rail shall be connected to digital pin D32. The circuit that controls the end-effector supply shall be connected to digital pin D31 on the Arduino.

The GND power rail from both input supplies shall be connected together and to the Arduino Board by means of one of its GND pins.

The rest of specific connections between the components of each relay power control circuit with the voltage regulation have already been developed on section 1.5.4.1. ("Power supply & wires").

Wireless communication modules' connections

The HC-05 module shall be connected to 4 header sockets mechanically (by pressing its on-board pins into the cavities of the sockets). These sockets shall be soldered to the same solder board that the end-effector power control circuit is assembled onto.

The HC-12 module shall be soldered to the same solder board to which the HC-05 header sockets are soldered.

The signal connections for both modules shall be done by using thin wires, which shall be soldered to the board where these two modules are.

The power supply for the HC-05 and HC-12 modules shall be connected to the 5 V power rail by means of thin wires.

The RX pin from the HC-05 module shall be connected to the TX2 pin (D16) and its TX pin shall be connected to the RX2 pin (D17). This module uses Serial port 2 (*Serial2*).

The RX pin from the HC-12 module shall be connected to the TX1 pin (D18) and its TX pin shall be connected to the RX1 pin (D19). This module uses Serial port 1 (*Serial1*).

Sensor connections

The laser distance sensor shall be attached to the 5 V power supply rail. Its XSHUT pin shall be attached to pin D12 of the Arduino board. Its SDA pin shall be attached to the SDA pin of the Arduino board (D20) and the SCL pin shall be attached to the SCL pin of the Arduino board (D21). All of these connections shall be done by means of thin wires.

The potentiometer shall be connected to the Arduino 5V supply, with the signal output connected to analog input A1 (D55).

There shall be a series of physical ports that shall be used in order to ease connections: 1 DE-9 port, 1 DE15HD (VGA) port and 1 DA-15 port, all of which can be seen on *fig. 101*.

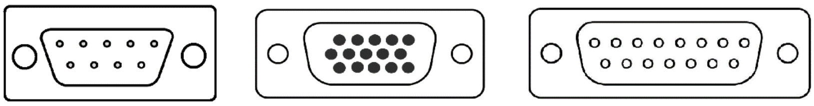


Figure 101. *From left to right: DE-9, DE15HD and DA-15 ports*

Source: Own

The potentiometer shall be connected to supply and signal by means of thin wires in two stages: the first stage shall go from the 5 V power rail and signal input to DE-9 male port, and the second one shall go from DE-9 female port to the proper sensor. All of this and other connections regarding DE-9 port can be seen on *fig. 103*.

The female connector shall be external and mechanically connected to this male connector.

Frontal light connection

This light shall be composed by a white LED (3.5 V level, 20 mA typical current value) and a 75 Ω ¼ W resistor in series with this light-emitting diode.

Figure 102. *High brightness white LED and 75 Ω ¼ W resistor*

Sources: https://www.cetronic.es/sqlcommerce/ficheros/dk_93/productos/84-439-1.jpg

https://m.media-amazon.com/images/I/415HKYg9dxL._AC_UY218_.jpg

The frontal light shall be connected to GND power rail and to digital pin 25 (D25), which shall activate such light, it shall use thin wires in two stages: the first one from the pins and to the male DE-9 port and the second one from the female DE-9 port to the proper frontal light.

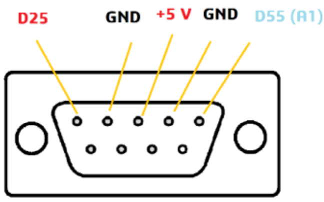


Figure 103. *DE-9 port with potentiometer and frontal light connections*

Source: Own

This connection from *fig. 103* is made to the male port being seen from the outside of the surface where it is screwed. The female port shall be the complementary to these connections.

End-effector servomotors connections

The robot has two end-effector servomotors, even though it is prepared to be able to hold more of them (on its basic form this has been limited to 2).

These position servomotors shall be connected to 5 V power rail and GND rail (red cable and black cable respectively on each motor) and to signal pins D6 and D5 (right and left finger motors respectively) by means of thin wires in a two-stage process: the first stage shall go from the proper connections to a DA-15 male port and the second stage shall go from a mechanically attached DA-15 female port to the proper servomotors.

The connection result can be seen on *fig. 104*:

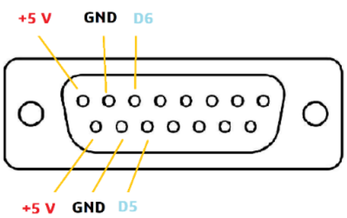


Figure 104. *DA-15 port with end-effector servomotors connections*

Source: Own

The connection presented on *fig. 104* is made to the male port being seen from the outside of the surface where it is screwed. The female port shall be the complementary to these connections.

Physical communication port connections

The device has a physical communication port that uses Serial port 3 (*Serial3*) to interact with external devices. This port shall use pins RX3 (D16) and TX3 (D15) and a GND. The connection of this port shall be done by means of thin wires connected to the previously mentioned pins and to the GND power rail. Such attachment shall be done up to a physical DE15HD male port, with the complementary DE15HD female port being added attached to the desired device. This connection can be seen on *fig. 105*:

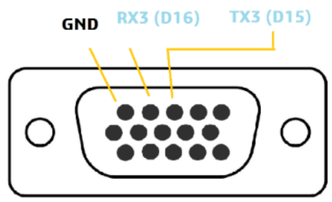


Figure 105. *DE15 HD port with external physical communications port*

Source: Own

High-torque servomotors and continuous rotation servomotors connections

Both high-torque position servomotors and continuous rotation servomotors have three wires for their connection: 2 of them constitute their power supply and the third one is for their signal. In the case of the position servomotors, their supply shall be connected, regarding the positive polarity, to the 6 V power rail. In the case of the continuous rotation servomotors, they shall be powered by means of the 5 V power rail. Both types of servomotors shall be connected to the common GND power rail regarding their negative terminal. Both types of motors shall be fed by means of thick wires.

The connection of the three wires for each motor shall be performed by means of thick wires. There shall be a solder board dedicated for the assembly of the connections. On the edge of this board, the use of pluggable blocks from manufacturers *Wurth Electronik* (males) and *Phoenix Contact* (females) (*fig.* 106) shall be necessary and they shall be soldered to it. These connections shall have two stages: wires shall go from their corresponding rails and signal pins to the board where the female plugs shall be soldered and from the male plugs the wires shall go until the motors. Male and female plugs shall be attached mechanically.

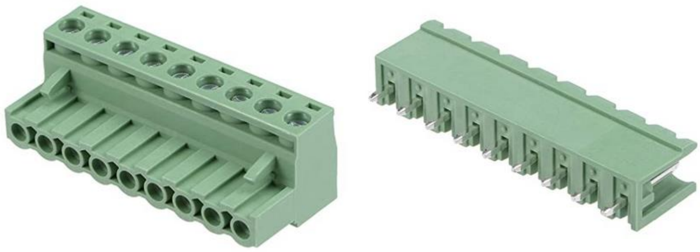


Figure 106. *Pluggable male (left) and female (right) blocks*

Sources: https://images-na.ssl-images-amazon.com/images/G/01/apparel/rcxgs/tile._CB483369110_.gif

https://images-na.ssl-images-amazon.com/images/G/01/apparel/rcxgs/tile._CB483369110_.gif

Each three terminals shall be connected such that, from the point of view of female connectors seen from the front (looking at the inner pins), the positive polarity is on the left, the GND pin is on the center, and the signal is placed on the right side.

A summary table also presenting the signal pin for each of the motors can be seen on the following table:

Joint/type of motor	Power supply	Signal pin
θ_1 / Cont. rotation servo	5 V	D9
Θ_2 / Position servo	6 V	D2
Θ_3 / 2 position servos	6 V	D3 (right), D4 (left)
Θ_4 / Position servo	6 V	D8
Θ_5 / Cont. rotation servo	5 V	D7

Table 24. *Signal pins for each joint's motors*

Source: Own

Note that all of these pins used for servomotor control correspond to PWM pins on the Arduino board diagram (*fig. 96*).

External wire protection

The wires coming from female connectors DE-9 and DA-15 and from the pluggable blocks shall exit the tangible inner limits of robot's structure and travel left floating in the air. These cables shall be covered by polypropylene spiral 30-34 mm cable protector from manufacturer Hellermann Tyton (*fig. 107*) cut at a length of 550 mm:

Figure 107. *Polypropylene spiral 30-34 mm cable protector*

Source: rs-online.com, code: 817-9870

1.5.4.5. Electronics box

This is the section of the robot where all the electronics of the robot (excepting sensors and motors) are located. It is composed by two main pieces: the box holding the electronics (*fig. 108*) and the plug for motors, sensors and frontal light (*fig. 109*).

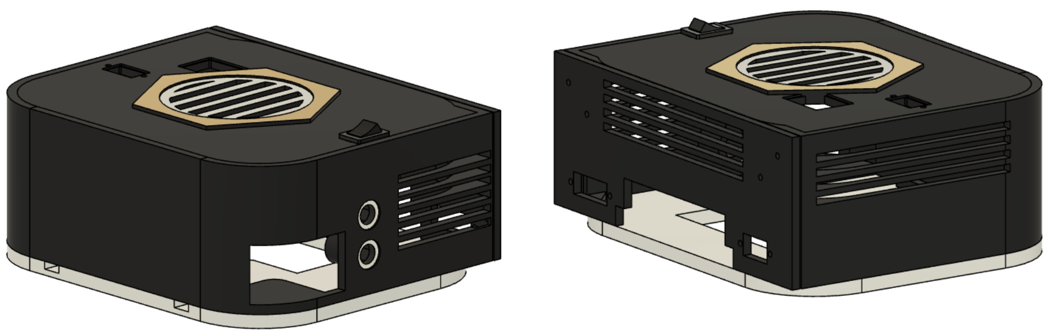


Figure 108. *Box for electronics*

Source: Own, Autodesk Fusion 360

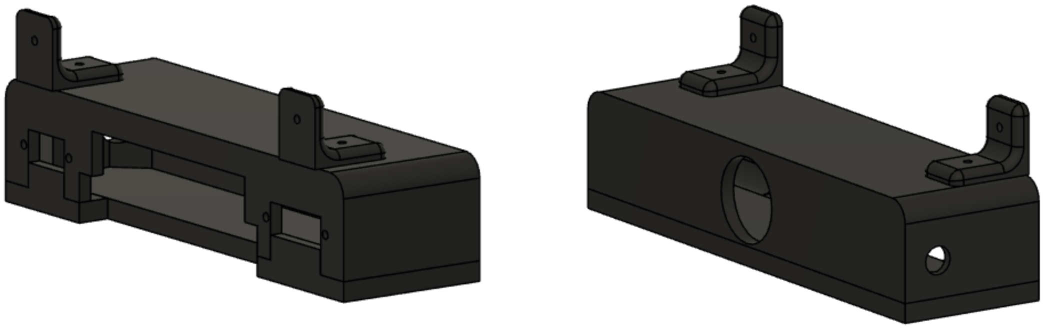


Figure 109. *Wire plug for the box for electronics*

Source: Own, Autodesk Fusion 360

The box for electronics shall be attached to the mobile base of the robot, whereas the wire plug shall be attached to the robot by being attached to the box for electronics.

<u>Box for electronics</u>

This box is made by 3D-printed pieces and composed by a base, four walls and a ceiling.

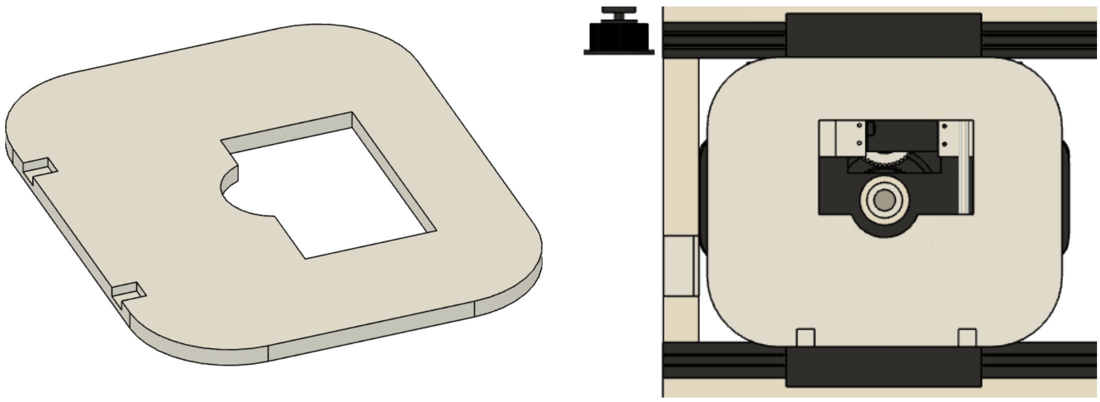


Figure 110. *Electronics box base and base mounted on the whole assembly*

Source: Own, Autodesk Fusion 360

This base (*plan 43*) is attached to the robot assembly by means of screws placed on the two square cavities on the front side to the pieces seen on *fig. 26*. It is placed such that the center of rotation of the robot (and therefore where the central rotation shaft is) is coincident to the center of the circumference present as a whole on this proper piece.

It is in charge of holding the electronics on its upper surface. Each solder board shall be attached by means of 2 M3 threaded rods with length 50 mm, one at each side, secured by 4 M3 nuts each, and leaving a gap with the base when adjusting the nuts of about 30 mm at least. The 2 voltage regulators shall be screwed through the holes the proper regulator board has. The Arduino Mega 2560 board shall be attached by 3 M3 threaded rods with length 40 mm secured by 4 M3 nuts each, leaving a gap of at least 20 mm with the base respect to the lower surface of the board. The wires from the laser distance sensor and from the rotation motor (placed on the surface below) shall pass through the central cavity of the base and be connected to their proper sockets on the electronics placed on the surface of this base.

Figure 111. *Left lateral panel of the electronics box*

Source: Own, Autodesk Fusion 360

This lateral panel (*plan 44*) has a major cavity through which the supply wire shall be provided for the Arduino board and through which the programming cable can be connected in order to upload new software or communicate via Serial port 0 (*Serial*) with the computer. The other two cavities next to the ventilation grid (14.874 cm^2) are where the DC female jack connectors (*fig. 82*) shall rest and to which the *KDTR1236W* transformer terminal shall be plugged.

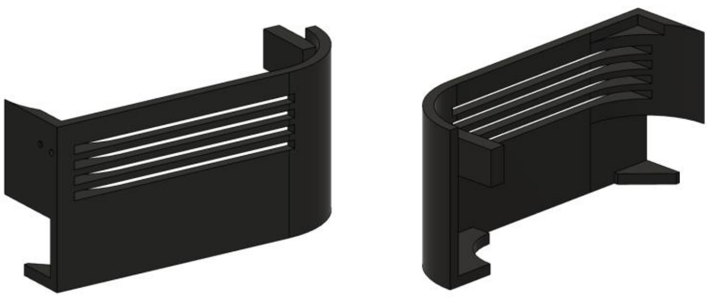


Figure 112. *Right lateral panel of the electronics box*

Source: Own, Autodesk Fusion 360

This right lateral panel (*plan 45*) has a ventilation grid with a greater surface than the one on the left (17.92 cm^2).

The following panel to be examined is the rear cover of the box (*plan 46*):

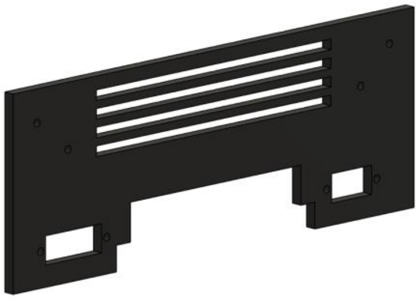


Figure 113. *Back lateral panel of the electronics box*

Source: Own, Autodesk Fusion 360

This cover has a ventilation grid (18.72 cm^2) above three cavities: the right cavity is where the DE-9 port shall be screwed by means of 2 M3 countersunk machine screws with a length of 10 mm and an M3 nut each; the central cavity is where the male plugs (*fig. 106*) shall join with female plugs for bigger servomotor connections and, finally, the left cavity is where the DA-15 port shall be screwed using 2 M3 countersunk machine screws with a length of 10 mm and an M3 nut each. There are 4 holes next to the ventilation grid that shall allow posterior attachment to the two other lateral panels (*fig. 111* and *fig. 112*).

The last panel (*plan 47*) that forms one of the lateral panels of the box has only the functionality of closing the gap left by the other four, and can be seen on *fig. 114*:

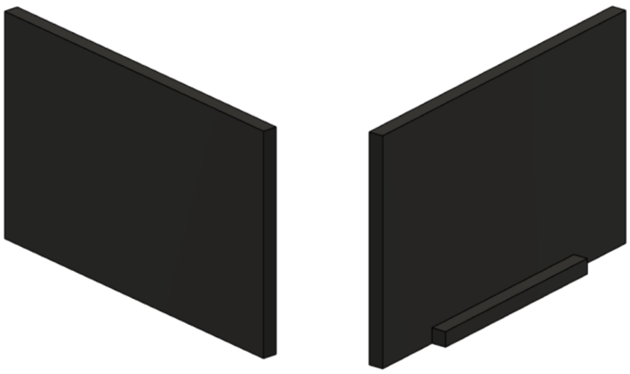


Figure 114. *Frontal panel of the electronics box*

Source: Own, Autodesk Fusion 360

Both panels from *fig. 111* and *fig. 112* are attached to the base of the box by means of *Pattex Extreme Pro* glue, the rear panel is attached to these two other panels by means of screws through the aforementioned four holes and the frontal panel is glued with *Pattex Extreme Pro* to the right and left panels. The result can be seen followingly:

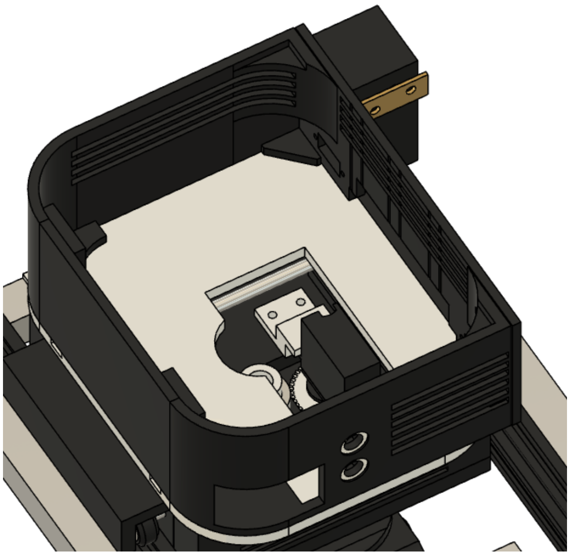


Figure 115. *Lateral panels together with the base of the electronics box*

Source: Own, Autodesk Fusion 360

The last piece of this box is its ceiling (*plan 48*), which is in charge of holding the safety power switch on its lateral left-side cavity, allowing Bluetooth module exchange through its right-side cavity, holding in position the external communications port and closing the electronics box. It also has a ventilation grid on its center.

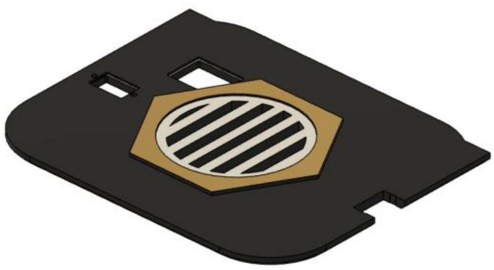


Figure 116. *Upper panel of the electronics box*

Source: Own, Autodesk Fusion 360

This previous piece is held in place by pressure. The electronics box result without the plug can be seen on *fig. 117*:

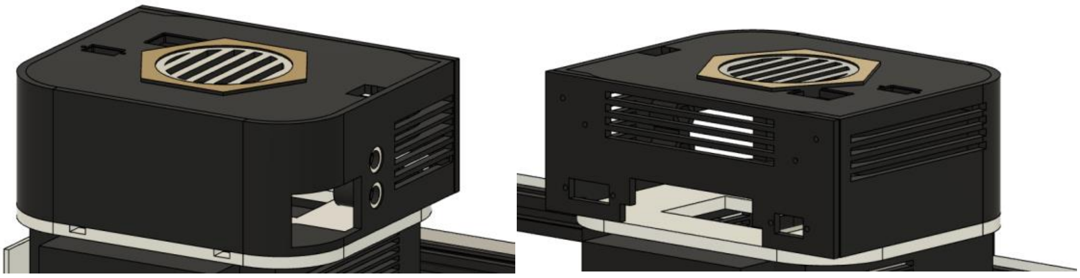


Figure 117. *Box for electronics assembly view (no plug)*

Source: Own, Autodesk Fusion 360

<u>Wire plug for connections</u>

This set is formed by three different pieces. The first of these pieces can be seen on *fig. 118*:

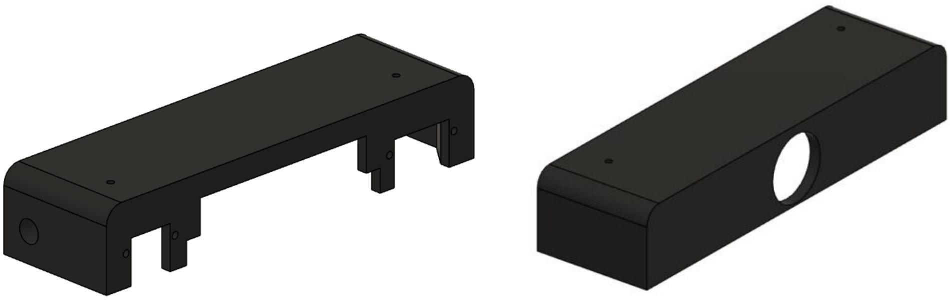


Figure 118. *Upper piece of the wire plug*

Source: Own, Autodesk Fusion 360

This piece (*plan 40*) holds female connectors DE-9 and DA-15 in place by means of two M3 countersunk machine screws with a length of 10 mm and two M3 nuts each of them in order to allow them to be mechanically plugged to their male counterparts on the box for electronics. It also allows the wires with the male plugs (*fig. 106*) to be attached to their female counterparts on the rear of the box for electronics and provides a rear hole for the wires to travel from the box for electronics to their motors, sensors and light in their corresponding places, as well as a lateral right-side hole to allow the wires that power and control the displacement servomotor, placed out of robot, on the rail structure, to travel up until this motor.

The next piece to be considered is the base of the plug (*plan 41*), which closes its volume, protecting the inner connections (*fig. 119*). This piece is glued with *Pattex Extreme Pro* to the upper part.

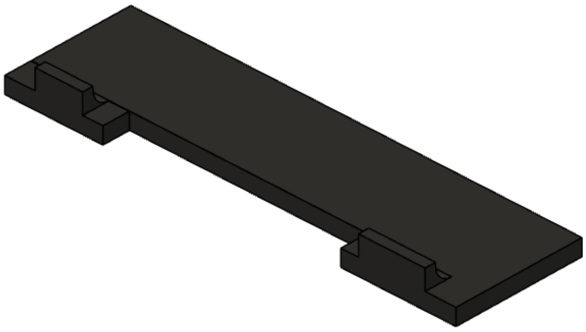


Figure 119. *Base piece of the wire plug*

Source: Own, Autodesk Fusion 360

The final piece to be considered is an attachment (*plan 42*) that holds the plug in place to avoid mechanical looseness when the robot displaces.

Figure 120. *Attachment for the wire plug*

Source: Own, Autodesk Fusion 360

The final assembly of this box for electronics joining the box and the wire plug can be seen below:

Figure 121. *Box for electronics and wire plug assembly*

Source: Own, Autodesk Fusion 360

The final assembly of the whole robot can be seen on *fig. 122*:

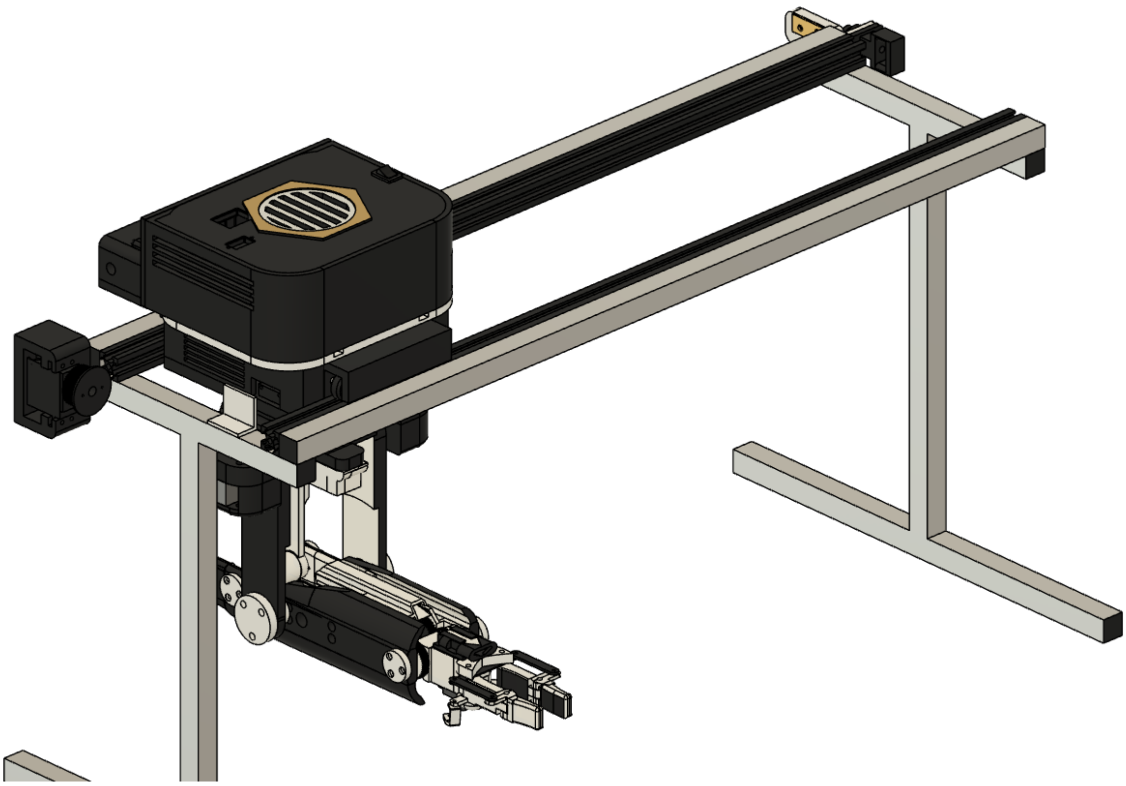


Figure 122. *Final assembly overview*

Source: Own, Autodesk Fusion 360

1.5.5. CODE

This section studies the software that is uploaded to the robot and to the devices that control it so that the hardware can be used as it is intended to and in order to be as efficient as possible regarding ease of use for the different functions, computation time and memory.

1.5.5.1. Main device

The main device shall be uploaded both with basic movement and functionality pieces of code. The specific way in which these functions can be used by a user are explained on the user manual (Annex IV, on A4.3). Each referenced table is placed on Annex III ("Code").

<u>Libraries, variables, macros, classes and parameter declarations</u>

The first thing to define in the program is the several libraries that shall be needed to be included. These libraries shall necessarily include the *Servo.h* library to move the motors via PWM signal generation that shall indicate the angle to be moved to each of them, the *SoftwareSerial.h* library, which shall be in charge of enabling serial port communications, the *Wire.h* library to enable I2C communication and the *VL53L0X.h* that enables an easy way to access the data from the laser distance sensor (*table 25*).

The following step is declarating the movement parameters to be used. These variables, macros and classes include: *dT* (step size to be used between each position of the robot on complex polynomic trajectories), *T* (the time the robot shall spend between initial and final position), *N* (the number of intermediate angles on polynomic trajectories as the second variable divided by the first one), the global *GJOINTS* macro (indicates the amount of joints the device has), *servos[10]* from the class Servo (creating an array with the number of servomotor objects that shall be later used), the global *6V_rail* macro (easing the code to be later implemented as it declarates the pin for 6V power rail's MOSFET's gate), the global *5V_rail* macro (with the pin used for the 5V power rail's MOSFET's gate), the global *Effector_relay* macro (with the pin for the relay in charge of enabling the end-effector motors), the *servos_5V_activated* (to enable or disable wrist and displacement servomotors), the global *ACTIVATE_MOTORS* and *DEACTIVATE_MOTORS* macros (enabling or disabling the 6V and 5V power rails at the same time), the global *ACTIVATE_EFFECTOR* and *DEACTIVATE_EFFECTOR* macros (activating or deactivating effector's MOSFET's gate at its relay power control circuit), the global *Wrist_rotatio*n macro (reads wrist position sensor value and escales it) and a data structure for defining servos named *TS1_Servo* (with its pin, offset angle and limit positions, for which section 1.5.3.1. ("Kinematic models") can be seen). This piece of code can be seen on *table 26*.

The next step is defining the structures for the different servos, as well as defining their several offsets (positional deviations in the design process respect to the real 0° and 90° levels on the different servomotors). This piece of code can be seen on *table 27*.

Followingly, the different parameters related to communications are declared, including macros that shall ease the use of the different communication ports. This can be inspected on *table 28*.

The next step is the declaration of the rest of variables and parameters, relating to ranges on laser measurements including the halfway mark, laser object declaration, enabling/disabling Text-To-Speech Engine on mobile app, kinematic registering variables, kinematic model parameters, HOME position, etc. This is presented on *table 29*.

Function declarations

In order to keep the code ordered, the first way in which functions have been presented is by their definition. This includes the setting of conversion, speaking, servomotor handling, communications handling and complex kinematic model actuation functions. The specific code for this section is on *table 30*.

The declaration of these functions can be seen on *table 31*. Here there is a summary:

- The 4.1. functions include a factor conversion for floats and a Speaking function (for Text-To-Speech Engine on mobile app).

- The 4.2. functions include attaching and detaching functions.

- The 4.3. functions include moving functions for each of the individual joints to go to a certain angle target from current position and with no intermediate positions. In the case of the *void writeAdvServo(int angle)* function, it allows basic coordinated movement between the shoulder servomotors to reach a position. In the case of the *void openGripper(int percentage_right, int percentage_left)* function, it allows to open the gripper at the end-effector at each side (each finger) a certain percentage.

- The 4.4. functions aim at developing basic information treating protocols for incoming UART, Bluetooth and radio communications and retrieves such data in a series of globally declared variables.

- The 4.5. functions work to allow calculations on trajectory generation to be made, allow complex 3D polynomic (3^{rd} degree) interpolation movements with different coordinated joints to be performed, 3D inverse kinematics to be applied for the robot (providing 3D coordinates, and two rotation angles on the end-effector) and a testing function for the gripper.

On *table 32*, the initialization sequence for each of robot subsystems, including testing gripper movements is performed. This must be executed each time the device is turned on.

Program basics

The program must always include a *void setup()* and a *void loop()* functions. The proposed structure for this can be seen on *table 33*.

The *setup()* function is executed only once and must always contain the *START_ROBOT()* initialization sequence in order for the robot to be able to work adequately.

The *loop()* function executes continuously unless a *break* is applied. It can contain the *Check_comms()* function, being this function in charge of "listening" to external communications (if the program only requires repetitive task movements for instance this function would not be necessary). The *ROBOT_PROGRAM()* function can be named otherwise and it includes the several instructions for the robot depending or not on the communication mode it finds itself in, reacting or not to external instructions.

1.5.5.2. External device example

An example code to illustrate how an external device can communicate with the robot via, in this case, the radio channel, can be seen on *table 34*.

The device used for this purpose is made by an external and wireless Arduino board with an MPU6050 accelerometer and gyroscope module with an HC-12 module attached which, when attached to a mobile limb of the human body, registers and sends information on such member's movement to control the robotic arm.

After library and variable initialization, the information corresponding to movement is obtained and stored for each of the axis of each of the services, then constrained to a value of -99 to 99 (losing resolution and increasing transmission speed and reliability). Then, those values are added with an initial and ending character to allow the robot to detect when a piece of information is complete, and the result is sent every 50 ms.

1.6. Detailed justifications & results

1.6.1. INTRODUCTION

This section aims at justifying design elections developed on section 1.5. ("Detailed design") concerning mechanical and electronic specific traits of the system as well as presenting the results of fabricating the prototype.

1.6.2. MECHANICAL STRUCTURE JUSTIFICATIONS

These justifications comprise the mechanical, most physical side of the device. It includes justifications related to the different mechanisms and tangible pieces of the robot.

Rail structure

The robot could be placed on rails the same type as the ones used with any length, with greater structures. However, as the intention is to make a prototype, conservative dimensions for the structure have been considered in order to be able to work better.

The rails have been chosen after several considerations, when getting inspired by the finesse working, smoothness and simplicity of 3D printer sliding mechanisms.

The specific length has been chosen as two times the length of a 20 x 20 mm aluminum extrusions (each one is 400 mm long). The width of the structure has been chosen arbitrarily in order to have enough separation for the robot to fit underneath, but not an excessive one that could cause bending in the parts that conform the mobile base. The height of the structure has also been chosen arbitrarily to leave enough space for the robot to operate underneath but also being able to reach objects present on the floor.

Protecting pegs have been added to prevent plausible undesired interactions with structure's edge that could cause damage to the user.

The rails have been placed in such a way that they allow maximum displacement of the robot taking profit of the structure's shape.

Joint 1 section: displacement

The DS04-NFC servomotor used for this joint has the main functionality of breaking stop inertia of the device and then keeping a steady displacement regardless of friction (reduced by the use of the 3D printer wheels), for which 5.5 kg·cm of torque is enough. The calculated speed of 71 mm/s for this joint is more than enough to comply with smoothness and speed requirements from section 1.3.2 ("USER SPECIFICATIONS REVIEW").

The bracing for the servomotor used on this joint (*fig. 15*) shall have enough space for it to fit in but with an extra 0.5 mm on each side of the bracing in order to prevent more than likely 3D printing heating errors. This casing also has thick walls to allow screws to attach it more easily to the square and therefore to the structure. The right side of the bracing is open so that wires from the motor can be outputted easily in order to travel up to the robot arm.

The use of brass squares to attach this joint's motor to the structure is the most logical, since they are really cheap pieces much stronger than PLA and they avoid the manufacture of other parts that accomplish the same objective.

The use of the GT2 belt has been decided because it is very widely used and reliable.

The diameter of the wheel from *fig. 14* has been decided in order to maximize linear speed for this joint while taking into account that it must not be too big in order to not collide with the structure or to not force the motor to be farther away from the rails (longer wires and belt).

The casing for the GT2 pulley (*fig. 18*, left part) has been designed with its right-side edge rounded to allow better flow of the belt and with its square profile limiting the movement of the pulley inside the piece.

The attachment between the pulley casing and the brass squares (*fig. 18*, right part) has been designed as 90 degrees of the 180 degrees necessary to attach the pulley to the back side of the structure.

The stop stabilizes the shaft of the pulley and is designed to allow this piece to slide in and out if necessary.

The base of the robot needs to be able to hold the whole weight of the robot and let it displace along the rails. In order to hold the weight of the robot, the base needs to be more bend-resistant and durable than PLA, for which it has been decided that this piece is made of 10 mm thick plywood. To let it displace, 3D printer wheels have been used (*fig. 25*), trespassed by 5 mm cylindrical aluminum rods with a length of 200 mm which accomplish three main functions: holding the wheels, supporting robot's weight, and being an extra help for the base in order to not allowing the holding pieces to bend inwards due to robot’s weight pulling downwards while wheels exerting an opposite force (*fig. 126*).

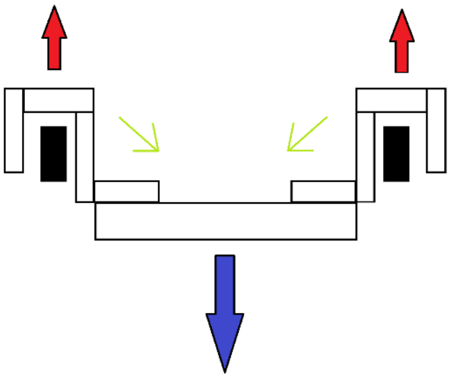


Figure 123. *Weight (blue), wheels’ normal force (red) and bending (green)*

Source: Own

On this previous drawing, the blue arrow represents the direction by means of a sliding vector in which the weight of the robot pulls, the red arrows represent the direction in which the normal force of the wheels is exerted opposite to the rails that support them and the green arrows represent the bending that tends to occur on this structure had the shafts not been in place. When introducing the shafts, the bending moment is compensated by means of the normal force exerted by these cylinders on the horizontal walls of the base.

These supports for the wheels that attach the base to the rails cover two of the four lateral sides of the base structure, being the other two (on the right and on the left, *fig. 27*) not related to structural resistance. These other walls needed to have some function besides just protecting the inside of the robot from entering objects that could potentially damage the motor or the other parts, so it was decided that they would have ventilation grids. This was mainly due to the fact that, as the device can displace laterally, the air flows in this direction and impacts the lateral surfaces of the robot so, placing hollowed surfaces enables the structure to take profit of the movement to create natural ventilation flow due to the movement.

The right lateral wall also has an opening that allows the laser distance measurement to be made. For this sensor to be placed a piece had to be made and screwed to the base aligned with the hole (*fig. 30*). On the structure, a reflecting piece needed to be mounted for the laser to bounce on structure’s limit (20 mm inner to the outer face of the structure).

The mobile base needed to be attached to the belt system, for which the piece from *fig. 29* was designed.

Joint 2 section: rotation

In order to make the robot rotate around its mobile base's center, a relatively powerful motor was needed in order to overcome the static inertia of the robot with different loads and configurations and be able to accelerate and decelerate quickly. For this, 25 kg·cm DS3225 servomotors were thought to be the best option.

These motors needed to be attached to the mobile base, for which the simplest way was to design geometrically simple holdings that would be attached to the base (*fig. 34*). Once the position of this holdings respect to the motor, the mechanism and the proper base had been established, its holes were used to signal on the base where they should be placed so that when manufacturing, such piece would have no correlation problems with the holding.

As explained on section 1.4.2.2. ("Mechanisms"), the chosen option was to use gear transmission. This gear reduction needed to have as few cogwheels as possible, for which the chosen number was 2: one would be attached to the motor and the other one to the shoulder piece. The safest way to ensure the correct attachment of this second cogwheel to the robot arm was to blend it with the shoulder piece; in other words, making the shoulder have the cogwheel embedded (*fig. 36* and *fig. 37*).

With the shoulder piece having the cogwheel in charge of its rotation embedded, as explained in the previous paragraph, the need to look for a way to connect this part with the mobile base was obvious. In order to do this, several options were considered, ranging from basic axle holding to the final one, which was making the axle trespass both the shoulder and the mobile base through the rotation center by means of a hole. In order to reduce friction between the parts, the height of the cogwheel was reduced and an axial bearing with its two contact surfaces being able to rotate was placed inside the cogwheel at the shoulder.

The several plausible ways to develop the rotation axis mechanically was using a non-threaded cylindrical shaft, a threaded cylindrical shaft and a bolt. The latter option was chosen for the sake of simplicity, so that it could be attached by nuts on the other side that allowed to hold the weight of the robot arm. Then, another bearing in contact with the separator was placed inside the mobile base box, on its lower surface and centered on the center of rotation to make both separator and nuts be able to rotate freely with no friction with the base. The separator was placed in order to elevate the nuts to a height where the thread of the carriage bolt used is. Two nuts were used in order to ensure that none of them would get loose.

Joint 3 section: shoulder joint

Shoulder position servomotors (25 kg·cm DS3225 servomotors) had to be placed on this piece, so some type of support had to be designed to ensure correct fixation (no vibrations, correct alignment…).

These supports for the servomotors were thought to fit the servomotors so that they could be mounted and dismounted easily by means of simple screwing. Their wires needed to be outputted by their side facing the rear of the robot. These wires would go up to a square-hole tunnel in the shoulder and from this to the back connector of the electronics box.

The links that the motors at the shoulder would actuate needed to be the correct length, for which the standard height of the robot while standing still was considered (*fig. 47*). These pieces would be attached to a shaft so that they would be able to move the part they are attached to but so that this part is also able to rotate around it. The shaft needed to be placed with no possibility to being loose or to falling from the robot, for which two stops, one for each side (*fig. 48* and *fig. 49*), were designed.

Joint 4 section: elbow joint

The motor used to actuate this joint (25 kg·cm DS3225 servomotors) was decided to be placed out of location, where the mass it carried did not include its own (more or less 70 grams). For this, the closest and simplest place to locate it was on the shoulder part. In order to acquire this, a strapping piece was designed (*fig. 51*) to hold this motor in place (*fig. 52* and *fig. 53*).

The part to which the elbow joint is attached is the forearm structure (*fig. 54*). The inner shape of this piece was designed to be able to hold the necessary bearings, mechanisms (for the wrist), motor holdings and shafts. The outer shape was designed to protect its interior by means of its extended circular walls and its length decided to be proportional to the rest of the arm and not excessive for its related motors to hold or move it.

The links between the elbow motor and its actuated joint form, as stated in section 1.4.2.2 (“Mechanisms”), a four-link mechanism, which was a direct, simple and reliable approach to controlling this set of articulated pieces. The shaft to which the mechanism is attached on the forearm is tapped by both sides so that it does not move and so that it does not fall on any of these sides by means of the pieces in *fig. 56* (*on the left*). To fix the shaft that attaches to such axis so that it does not move laterally, a cylinder roll separator that can freely rotate was added (this also would help the wires not to get tangled).

Final pieces on this section were cosmetic and were meant to increase cosmetic value of the device (according to section 1.3.2. (“USER SPECIFICATIONS REVIEW”), point 4, A1 to A3).

Joint 5 section: wrist joint

The design of this joint was decided to oscillate around a belt transmission system section 1.4.2.2. (“Mechanisms”). A continuous rotation servomotor was decided to be used instead of position servomotors because, although this made control more difficult (a proportional P controller with feedback from the potentiometer used as a position sensor), in case of belt slips, this would not cause any effect on the final actuation on the motor, isolating the perturbance in time. For this, the position sensor was decided not to be connected to the belt, but to the proper wrist shown on *fig. 63* by means of the gear wheel (*fig. 65*) since this auxiliary gear system would not be permuted by belt slips on the traction side of the belt transmission, measuring the actual value of position, not the one that allegedly is written by the motor to the system.

The piece that holds the position sensor in place had to leave an open cavity on the upper side to let the wires flow to the main wire casing placed on the bridge of the piece.

End-effector section

Regarding the end-effector, whereas the standard mechanism had already been decided on section 1.4.2.2. (“Mechanisms”), several other tool designs were considered as possible extensions of such in the future, including origami soft gripper prototypes developed at Boston University (these proved to be very fragile on early prototypes, for which not much development time was spent on this aspect of the section).

The size of the gripper has been chosen in order to ensure a good opening range. The span this end-effector gripper is 44.17 mm on each side (calculated on the 3D model), which enables it to pick objects up to 88.34 mm in width with no inner mechanism collisions.

The shape used for the fingers was chosen to be slightly rounded, so that they were able to pick rounded objects but also having a straight tip that allowed picking the rest of objects.

As this project is open for developments and such improvements are intended to be performed, thinking of possible Artificial Vision applications, a light was considered to be added at the end-effector base in order to light objects for helping with lighting conditions in this kind of projects. Such light had to be bright but with a low power consumption, for which an LED was thought to be the best option. This Light-Emitting Diode was thought to be added inside a casing that both protected it and indicated the user in which direction it must be directed.

The whole end-effector mechanism is thought to be added and replaced easily by means of extracting the axes of each of the pieces of the fingers and removing the screws from their corresponding nuts.

1.6.3. ELECTRONICS JUSTIFICATIONS

These justifications comprise the electronics side of the device, regarding power supply and wires, wireless communications, sensors, µC & connections and electronics box.

Power supply & wires

The power supply provided to the prototype had to be as cheap and powerful as possible. The idea was to buy the most common device possible that could fulfill these characteristics. That is why a common transformer with a rectified DC input for each of the electronic subsystems was enough. The different subsystems divided by their power requirements are: µC, modules and lesser powerful servomotors, and high-torque servomotors. The last two subsystems would require voltage regulation (5 V and 6 V respectively) while the Arduino board chosen on section 1.4.3.1. ("Control component") has built-in regulators.

The system needed to be as efficient as possible, so, in order to avoid unnecessary current use, a subsystem to control this needed to be implemented so that the brain of the device was able to activate/deactivate some of the branches when needed and in a safe and isolated manner. This circuit is developed in section 1.5.4.1. ("Power supply & wires"), and its simulation can be seen on *fig. 124* (OFF state) and on *fig. 125* (ON state).

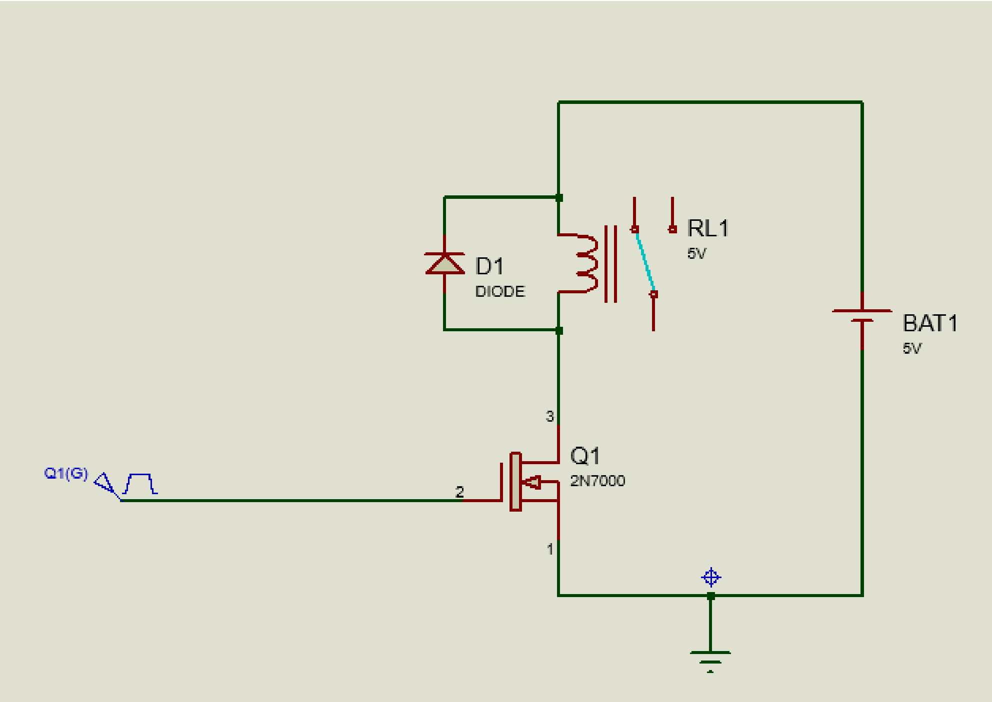


Figure 124. *Relay power control circuit triggered on OFF region (MOSFET)*

Source: Own, Proteus 8 Demonstration

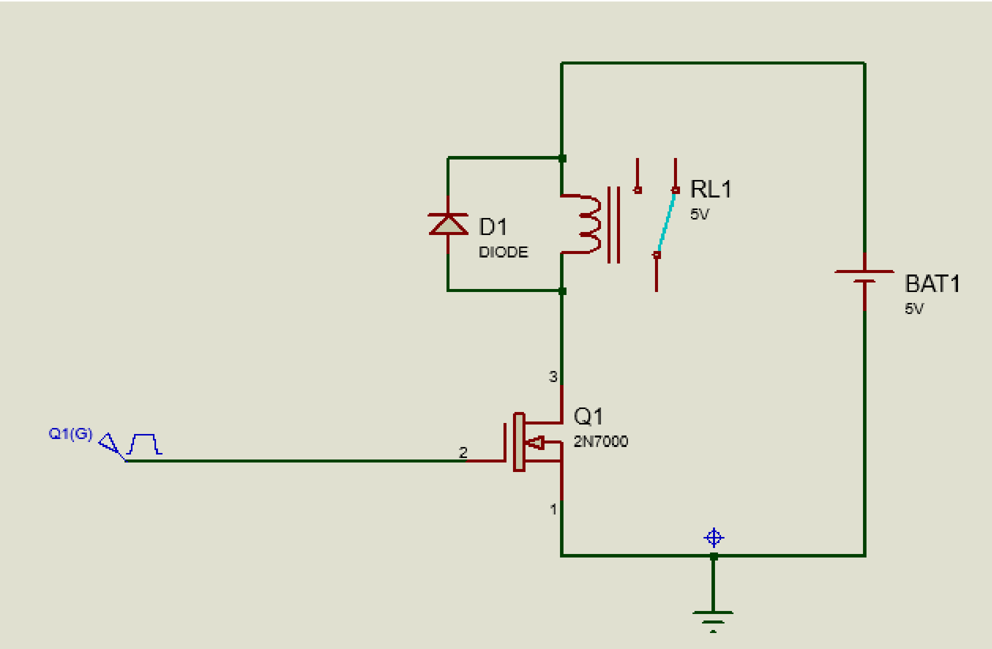


Figure 125. *Relay power control circuit triggered on ON region (MOSFET)*

Source: Own, Proteus 8 Demonstration

To isolate the control subsystem from power, the element that connects power to each of the rails is a relay. This component fits the isolating side of the desired solution and, since this feature is not intended to be activated continuously, speed of operation is considered adequate.

When considering the election of components, polarizing current needed by the coil of the relay needed to be withstood by the chosen transistor. Since the coil needs 79,4 mA consumption in order to be polarized, a Microchip's 2N7000 N-channel MOSFET was a good election, since this operating current is easily inside its boundaries.

The previously presented circuit acts as a control circuit for an inductor supply (relay's coil). Because of this, certain effects of the different operation stages have to be taken into account. The addition of a flywheel (also called flyback) diode provides a way for the current to flow backwards in certain polarization stages, avoiding potentially self-destructing voltages.

The addition of a safety power switch was key for fast high-torque motor disabling, so that the user can avoid potential damages to themselves, to the environment or to the robot itself.

In order to get the appropriate level at the power rails, the use of a voltage regulator was needed. These chips are a better option compared to other technologies, since they are far more efficient and have a very practical use. As the input voltage is 12 V and in two of the three cases (except for the self-regulation capabilities of the Arduino board) the needed tension is lesser than this quantity, the needed IC was a step-down (or Buck) converter. The chosen option was an LM317 regulator, since it can handle voltages from 4.2 V up to a 40 V input and an output from 1.2 V to 37 V, it has a 2.2 A typical current output drain and a medium working range temperature (-55 °C to 155 °C). On *plan 49*, LM317 board has been presented as a set of equivalent components, since *Kicad* software has not the exact board used, so in reality no capacitors or resistors present on that representation have been used.

The wires to be used needed to vary in characteristics depending on the voltage and current necessities. For this, more powerful connections required bigger cables able to withstand greater currents than those with information transmission or lower currents. Chosen elements can be inspected on *fig. 89* and *fig. 90*.

Wireless communications

The specific reasons behind the election of each of the two wireless communication modules has been developed on section 1.4.3.2. ("Wireless communications").

The motivation of electing modules of this kind was to incorporate cheap yet reliable and functional communications with different control elements by which the device could be operated, providing the user with the widest range of options not compromising system's stability with exaggeratedly big or power consuming devices that could not fit in robot arm's space.

When considering the Bluetooth module, the HC-05 is a very common and functional module when concerning Arduino BT communications. Other variations like the HC-06 and other more complex modules were considered. However, this was the most functional and simple approach with which the user could be get used to easily.

When considering the Radio module, the HC-12 is a very simple yet complete option, with more functionality than the RF433 modules and cheaper than other more complex modules.

Sensors

The use of a laser sensor was a debate at the beginning of the designing process, considering the alternative of using encoders, widely used in robotics. However, from the point of view of experience, belt tensioning can be badly performed if some manufactured piece has a defect. So, this was an option that allowed for real distance measurement (excluding the proper intrinsic errors the sensor has).

When applied to this system, the sensor had to be placed on one of the sides. Considering that this position is at the exterior of the robot at some distance from the maximum width the shoulder piece can take and at some distance from the center of the robot, it can be understood that in order for the center to be positioned, some offsets shall be needed to be applied in the code.

Due to library requirements, XSHUT pin shall be necessarily attached to pin D12 of the Arduino board.

The used potentiometer shall be able to measure up to 180° turns by means of a ratio between 0 V and 5 V read by the analog input of the Arduino board. The circuitry for this device can be seen below:

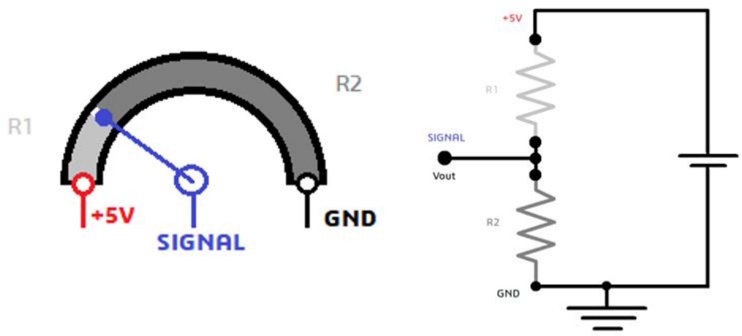


Figure 126. *Potentiometer circuit and equivalent circuit*

Source: Own

By means of the equivalent circuit seen on *fig. 126*, taking into account that no current flows into the voltage reading pin of the Arduino, the following modelling equations can be obtained for the circuit:

$$\frac{5 - Vout}{R1} = \frac{Vout}{R2} \rightarrow \frac{R1}{R2} \cdot Vout + Vout = 5 \rightarrow \left(\frac{R1}{R2} + 1\right) \cdot Vout = 5$$

$$Vout = 5 \cdot \frac{R2}{R1 + R2}$$

Taking into account the 0° to 180° span of rotation and knowing that R1 and R2 can be expressed as a function of the whole respect to the other, a transfer function can be deduced given an entered amount (*Input*) of degrees:

$$R2 = 10000\ \Omega - R1, \qquad R1 = \frac{10000\ \Omega}{180°} \cdot Input \rightarrow$$

$$R2 = 10000 - 55.56 \cdot Input, \quad R1 = \ 55.56 \cdot Input$$

$$Vout = 5 \cdot \frac{10000 - 55.56 \cdot Input}{10000 - 55.56 \cdot Input + 55.56 \cdot Input} \rightarrow$$

$$Vout = 5 \cdot (1 - 0.005556 \cdot Input)$$

The result of this development is the following transfer function, on which it can be proven that, given a 0° input the output is 5 V and given a 180° input the output is 0 V:

$$\boxed{Vout = 5 - 0.02778 \cdot Input}$$

<u>µC & connections</u>

The wires coming from the power admission subsystem to the power rails and from these to high-torque servomotors are performed by means of thick wires from *fig. 89*, having a wide section that allows for more current flow without overheating. The connections between signal terminals and 5V-supplied subsystems are performed by means of the cables from *fig. 90* since no overheating problems should occur due to low power transmission.

The frontal light shall be composed by a white 3.5 V LED. This kind of diode have a 20-mA typical current value. It has been decided that the current shall be limited to this value, which shall not damage or fuse the LED. The resistance value to be used can be calculated by first knowing the voltage drop on such component due to LED's polarizing tension and given the maximum current specified before:

$$I = \frac{V}{R} \rightarrow R = \frac{V}{I} = \frac{Vsupply - VLED}{I}$$

$$R = \frac{5\,V - 3.5\,V}{20\,mA} = 75\,\Omega$$

The value obtained is standard, so it can be exactly the used one. Next, the power dissipation must be obtained:

$$P = I^2 \cdot R = 0.03\,W$$

This power dissipation value indicates that a ¼ W resistor is enough for this circuit.

When considering the use of the DE-9 port, it must be said that there are more pins than needed, but this did not a problem compared to the advantages this format brings with ordered connections and easily mountable case. It also allows for future versions to include more light indicators.

The extra pins on the DA-15 port serve as a way to, without changing the physical design of the electronics box, incorporating new communication functionalities with plausible extra wires. With this port, a future ampliation of the robot could be made incorporating up to 5 servomotors working at the end-effector or adding an extra degree of freedom. The pins have been elected in such a way that in case of an error when soldering, same polarity pins are touching, avoiding undesired and potentially dangerous short circuits.

The same can be said about the DE15 port, which holds more than enough pin to allow for future versions to be ampliated regarding their functionality.

The use of pluggable terminal blocks was determined due to their ease when plugging connections, which allows the robot to be unplugged and extracted from the rails to be put in another structure more easily. These pluggable terminal blocks are able to withstand up to 300 V, which holds them no problem when considering the values this device is dealing with when feeding its motors.

External wire protection was necessary to avoid undesired tangles or collisions due to their looseness.

Electronics box

This box is intended to: contain the electronics in a same place and allow for the connections to be output to other parts of the device. The easiest shape for this structure was a literal box with no extravagant external geometries.

Each of the physical ports had to be placed in the most useful location around the 3D exterior shape of the box. Given that the biggest number of cables that travel to the robot arm are positioned at the back of the robot to provide the least resistance to motion, the connectors for such cables (DE-9, DA-15 and pluggable terminal blocks, not including the DE15 port) were placed also at the back of the device.

The distribution of the different cable outputs had a reflex on the shape of the back panel of the box, being developed to hold in place the different ports and having a big central cavity for the pluggable terminal blocks.

Lateral panels were designed to allow air flow to cool the inside of the box when moving allowing heat exchange and generating a situation of forced convection that used the motion of the robot.

The frontal panel had no other function than just closing the whole box.

The upper panel had to be designed in such a way that allowed easy dismount to access any possible problems in wiring. It needed to have a square hole that allowed the user to pick the HC-05 module and replace it by another one inserting it in the same cavity. It was also the position of the box where the safety power switch had to be placed for easy access. Finally, an extra ventilation grid was placed at the center of it.

A wire plug for connections was designed in order to allow easy plug and holding that protected the back side of the connectors (*fig. 109*).

1.6.4. FABRICATION RESULTS

When the design is completely finished and some final considerations regarding planning have been made, fabrication shall begin. When fabrication has concluded, the software is inputted as a way to test device's capabilities. The results from fabrication and demo coding are here presented.

Performance results

The performance specifications established on the needs' study shall be presented and analyzed on this subsection:

- Theoretical **maximum load** at the wrist has been checked, as it is able to lift **0.33 kg** soft drinks. This complies exceedingly with section 1.3.2, point 1 and B2.

- The device is **accurate** at a level of a 2 mm range and deviates a maximum of 2 mm from one **repetition** to the following one. This complies with section 1.3.2, point 1 D1 and D2.

- **Speed** and **movement smoothness** has also been checked to be **pretty good**. It is able to reach different heights without colliding with the environment. This complies with section 1.3.2, point 1 A1, A2, C1 and C2.

- The final **mass** of the robot is **4.3 kg** (not counting rail structure section, which is not part of the moving device but a support). This complies with section 1.3.2, point 2 A1 and section 1.3.3, point 1 d).

- Regarding working temperatures, **no overheating** has been observed on any of the motors on stall mode. When operating with an under maximum load (0.15 kg on the end-effector), a small increase in elbow and wrist motor's temperatures has been observed after 30 minutes of operation.

To sum up, the manufactured prototype complies with all the necessities studied on section 1.3. ("Needs' study"), with its weak points being accuracy and servomotor heating.

Robot fabrication result

Followingly, some pictures taken on the real prototype are presented:

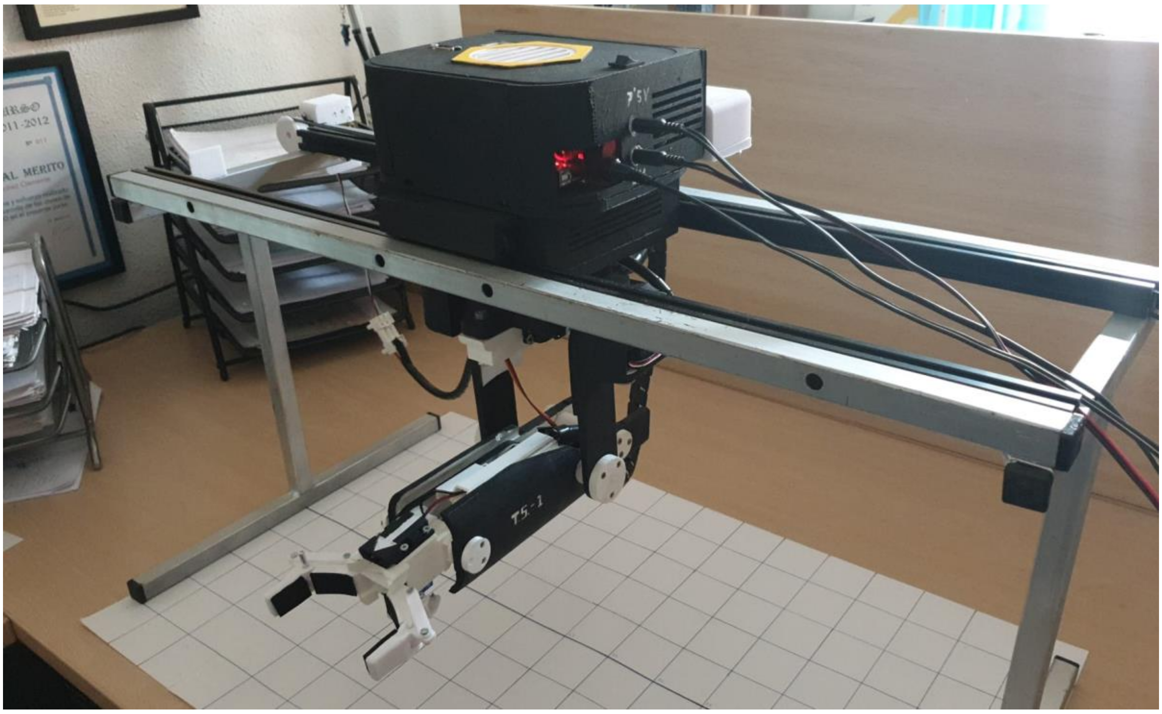


Figure 127. *Left lateral view of TS1 prototype*

Source: Own

Figure 128. *Right lateral view of TS1 prototype*

Source: Own

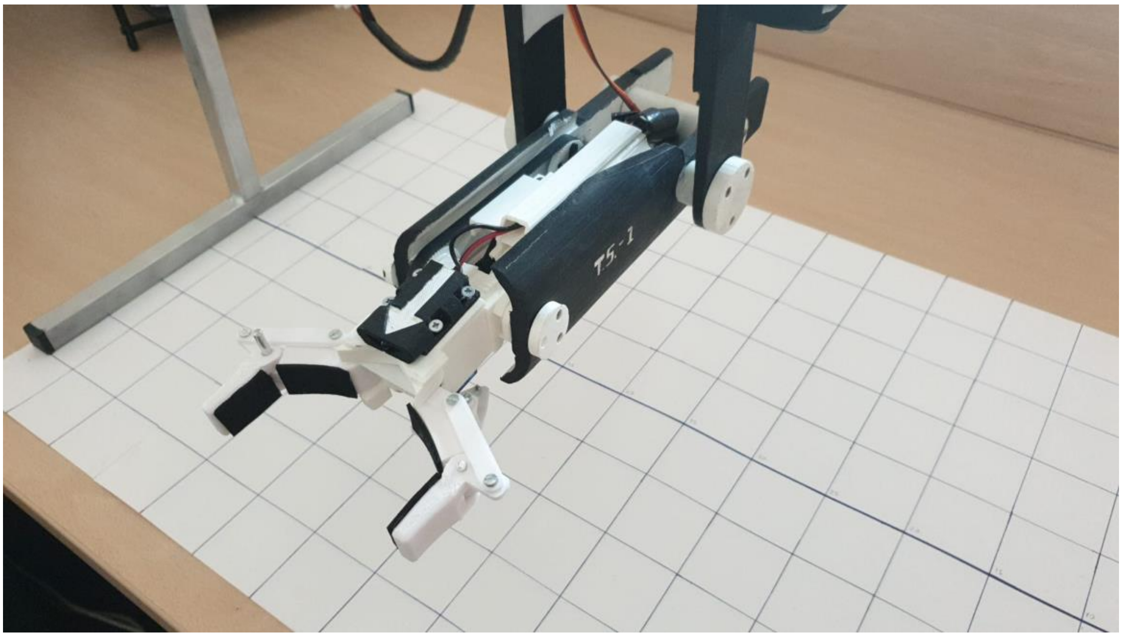


Figure 129. *Central section close-up*

Source: Own

Annex I: Calculations

A1.1. INTRODUCTION

This section aims at explaining with full detail each important calculation performed on this project so that, with this part of the report especially dedicated to it, the adequate document space is used for maximum comprehensibility.

A1.2. MAXIMUM JOINT SPEEDS CALCULATION

In section 1.5. ("Detailed design"), the results of calculating the maximum speeds for each of the joints were given, but no further detail was provided. This section aims at presenting such detail on the calculations that have been performed on this regard.

Calculations performed on this section take into account only the kinematics of the mechanism, not its inertia, since these are maximum theoretical speed calculations for each joint given ideal no mass conditions.

Joint 1 maximum speed

This is a belt transmission, which transforms rotatory motion into linear one. In order to find out which is the linear one (final one), first it is necessary to find the angular one.

Angular speed shall depend on two factors for a belt transmission of this kind: servomotor's maximum angular speed and gear belt contact area radius. The first can be found on the datasheet of the DS04-NFC servomotor (0.22 s/60° ≈ 45 rpm), whereas the second one is proper to the design (0.015 m radius).

First, a necessary unit conversion to rad/s is necessary, so the following transformation factor is used:

$$1\, revolution = 2\pi\, radians, 1\, minute = 60\, s$$

$$\omega = 45\, \frac{revolution}{minute} \cdot 2\pi \frac{radians}{revolution} \cdot 1 \frac{minute}{60\, s} = \, 4.71 \frac{rad}{s}$$

Once the transformation has been performed, given the wheel radius, a simple angular to linear speed conversion can be performed:

$$v = \omega \cdot R$$

$$v = 4.71 \frac{rad}{s} \cdot 0.015\,m = 0.071 \frac{m}{s}$$

The conclusion is that the system can travel linearly at a maximum speed of **0.071 m/s** or, what is the same, at 71 mm/s.

Joints 2, 3 and 4 maximum speeds

These are various systems with different mechanisms, being joint 2 a gear reduction set, joint 3 a direct coupling by the servomotors and joint 4 a 4-link parallel mechanism. However, despite having various forms, they all have a 1:1 transmission and therefore the speed calculated out of the servomotors shall be the same for the joint (assuming ideal conditions and no mass, that is, purely kinematically conditions).

This calculation is pretty simple, since it only takes a transformation in the units given by manufacturer's datasheet for the motor (which is the same for those 3 joints: DS3225 position servomotor). The maximum speed reported for this motor for maximum torque election is 0.13 s/60°.

The unit conversion can be performed easily:

$$1\,revolution = 2\pi\,radians = 360° = 6 \cdot 60°, 1\,minute = 60\,s$$

$$\omega = \frac{60°}{0.13\,s} \cdot \frac{2\pi\,radians}{6 \cdot 60°} = 8.1 \frac{rad}{s}$$

The resulting maximum rotational speed for maximum torque is **8.1 rad/s**, or what it is the same, 464 °/s.

Joint 5 maximum speed

This, like the first joint, is a belt transmission. However, it differs from the other joint in that it does not transfer the angular motion into a linear one but keeps it on its basis form. This way and knowing from design that belt reduction relationship is 1:1, calculations are much eased.

The motor used for this joint is the same as the one used for joint 1, a DS04-NFC servomotor with a maximum frequency of 0.22 s/60° ≈ 45 rpm. Given that the transmission is performed from one gear to another, the calculations to be made are the same as the first ones given for joint 1, with no need to transform the kind of motion:

$$1\, revolution = 2\pi\, radians, 1\, minute = 60\, s$$

$$\omega = 45\, \frac{revolution}{minute} \cdot 2\pi\, \frac{radians}{revolution} \cdot 1\, \frac{minute}{60\, s} = \; 4.71\, \frac{rad}{s}$$

Therefore, it can be said that the maximum angular speed for joint 5 is **4.71 rad/s**, or what it is the same, 269.86 °/s.

End-effector maximum closing/opening speed

The end-effector presents a 4-link parallel mechanism. This does not transform angular speed from its source to the effected link and therefore the angular speed coming out of the servomotor shall be the one transmitted to the joint.

The motor used is an SG-90 microservo from manufacturer *TowerPro*. According to its datasheet, at 5 V it provides a maximum angular speed of 0.1 s/60°.

A simple unit conversion is needed:

$$1\, revolution = 2\pi\, radians = 360° = 6 \cdot 60°, 1\, minute = 60\, s$$

$$\frac{1}{\omega} = 0.1\, \frac{s}{60°} \cdot 6 \cdot \frac{60°}{2\pi\, radians} = \; 0.095\, \frac{s}{rad}$$

$$\omega = \frac{1}{0.095\, \frac{s}{rad}} = \; 10.47\, \frac{rad}{s}$$

The maximum angular speed able to be reached by the end-effector is **10.47 rad/s**, or what it is the same, 600 °/s.

A1.3. MECHANICAL MATHEMATICAL MODELS

This section aims at determining the different mathematical models regarding different kinematic and dynamic aspects of the whole robotic device, including its Denavit-Hartenberg (DH) parameters, Jacobian matrix determination, Direct Kinematics model and Inverse Kinematics model.

A1.3.1. Kinematics calculations

This section specifies the calculations performed on a kinematic level, not taking into account dynamics.

DH parameters

The Denavit-Hartenberg parameters are a way of representing successive reference frame translations and rotations from one link to its consecutive one on a robot in order to visualize, given the different structural characteristics of the device, how to transform each intermediate reference frame to reference frame "n".

In order to perform the analysis, the first step is defining where each reference frame for each of the joints must be placed from the original one on, for which the 3D model of the robot needs to be taken into account:

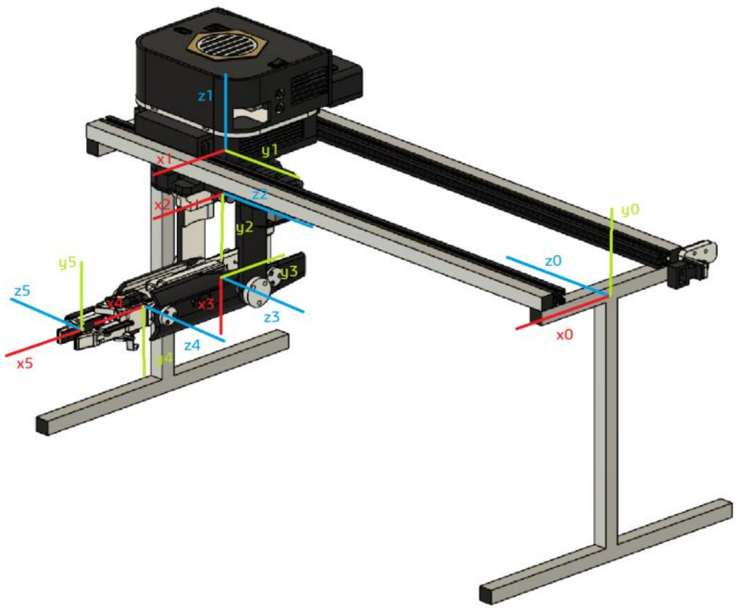


Figure 130. *Robot 3D view with reference frames for each joint in place*

Source: Own, Autodesk Fusion 360

When applying the different transformations from one reference frame to the other in the order specified by the DH convention (rotation in z -> translation in z -> translation in x -> rotation in x), the steps to follow are these following:

- From reference frame 0 to reference frame 1 there is a prismatic joint. There is a z displacement equal to joint 1's value and an x rotation equal to $\alpha = -\frac{\pi}{2}$ radians.

- From reference frame 1 to reference frame 2 there is a rotational joint. There is a z rotation equivalent to joint 2's value, a z displacement with value d=-0.0785 meters and an x rotation of $\alpha = -\frac{\pi}{2}$ radians.

- From reference frame 2 to reference frame 3 there is a rotational joint. There is a z rotation equivalent to joint 3's value plus $\frac{\pi}{2}$ radians and an x rotation of α=0.124 meters.

- From reference frame 3 to reference frame 4 there is a rotational joint. There is a z rotation with value joint 4's value minus $\frac{\pi}{2}$ radians, followed by an x displacement of a=0.134 meters.

- From reference frame 4 to reference frame 5 there is a rotational joint. There is a z rotation equivalent to joint 5's value, an x displacement of 0.1 meters and an x rotation of α = π radians.

<u>Direct kinematics model</u>

There are several ways to compute the direct kinematic (D.K.) model: one of them is by calculating the different transformations using the DH parameters in as many reference frame translations as joints are in the robot and the other one is by taking a geometric approach. Either way is viable for "on paper" calculations. However, when taking into account a µC application, the fastest option needs to prevail.

Calculating successive transformations usually takes more computational time than performing simple geometric calculations. Due to this, the development of a trigonometric D.K. model has been considered necessary.

For this model, given a five-angle input (θ_1, θ_2, θ_3, θ_4, θ_5), an output (x, y, z, α, β) shall be directly obtained. This result is composed by a cartesian coordinate point (x, y, z) and Y and Z rotations of the end-effector reference frame (α, β). Equations shall be presented in terms of meters and degrees so that this way the user finds it easier to use these models.

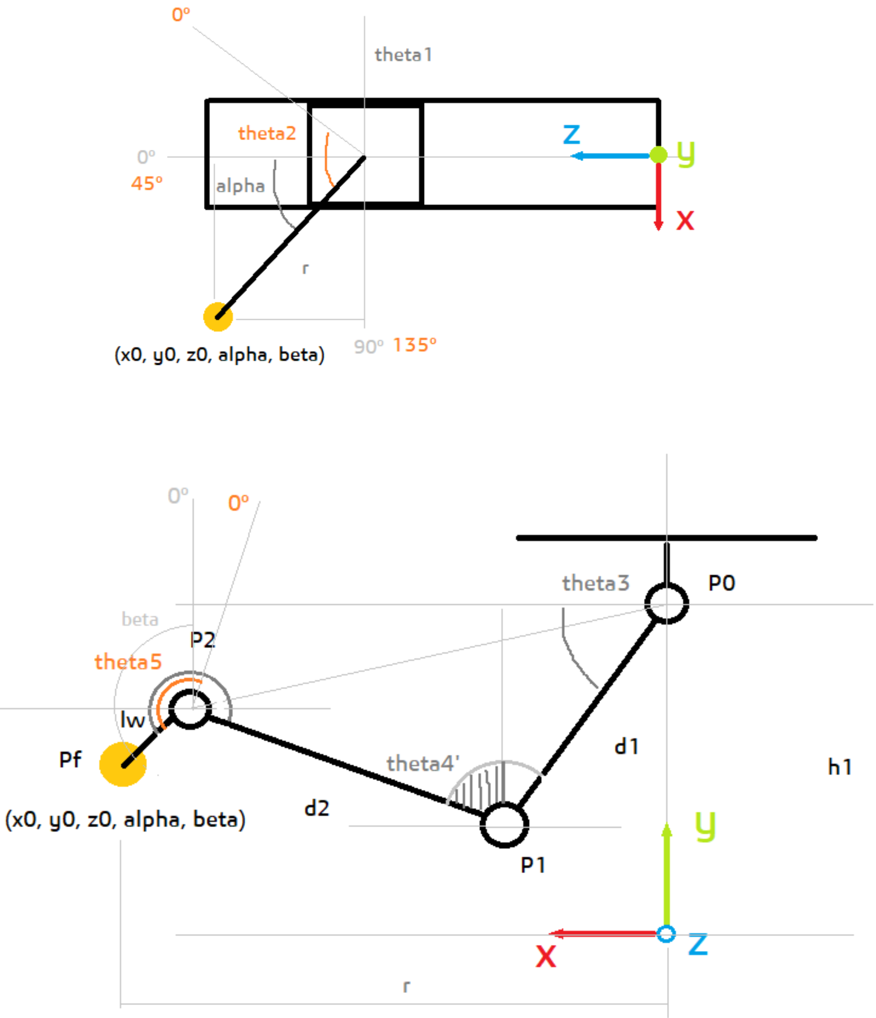


Figure 131. *Simple drawing with auxiliary variables in place for D.K. model*

Source: Own

The first thing to notice is that angle α comes directly from realizing the rotation of the robot and taking into account that since rotation servomotor has 270° of range it has been placed with its 0° angle at -45° respect to the mathematical 0°. Therefore, it can be seen that:

$$\theta_2 = \alpha + 45°$$

$$\boxed{\alpha = \theta_2 - 45°}$$

In order to compute the value of x and z it is necessary to obtain the projection of the extended robot arm on XZ plane, which is represented with variable name "r". For this, the use of a lateral 2D view is considered as useful (*fig. 131*, lower picture).

The procedure to calculating the "r" distance is going to be based on adding the projection distance on "r" that points P0, P1, P2 and Pf have. For this, the first step is finding what the projection of P1 is respect to P0 (located on 0 distance respect to the start of "r"):

$$P1_r = P0_r + d1 \cdot \cos(\theta_3)$$

$$P1_r = d1 \cdot \cos(\theta_3)$$

Then, the next step is to calculate θ_4', necessary for future equations:

$$\theta_4' = \theta_4 - (90° - \theta_3) \rightarrow \theta_4' = \theta_4 + \theta_3 - 90°$$

Followingly, the projection of P2 can be computed:

$$P2_r = P1_r + d2 \cdot \sin(\theta_4')$$

In order to define the last equation for "r" axis length definition, the β angle is needed. To obtain it, the last composed angle needs to be observed. 0° axis for robot's wrist motor rotation is located at 90° respect to d2 link. β angle can be determined to be a portion of θ_5 angle. In order to quantify what amount of it is corresponded the following equations are developed by summing angles from the d2 bar until lw link (corresponding to the distance between end-effector's base and the start of gripper's last link) covering the whole range:

$$\theta_5 = 90° - \left(90° - (90° - \theta_4')\right) + \beta$$

By simplifying this expression, it can be found that:

$$\theta_5 = \ 90° - \theta_4' + \beta$$

$$\boxed{\beta = \theta_5 + \theta_4' - 90°}$$

Then, the final calculation for finding "r" can be performed:

$$Pf_r = r = P2_r + lw \cdot \cos{(\beta)}$$

The value for x and z cartesian coordinates value can directly be found when knowing "r":

$$\boxed{x = r \cdot \sin{(\alpha)}}$$

$$\boxed{z = \theta_1 + r \cdot \cos{(\alpha)}}$$

Only the value of y remains to be found. A second analysis on the second picture on *fig. 131* needs to be performed. Similarly, to the previous procedure, the final height shall be the addition of the heights of each consecutive serial points. First, the y coordinate for P1 has to be calculated:

$$P1_y = P0_y - d1 \cdot \sin{(\theta_3)}$$

$$P1_y = h1 - d1 \cdot \sin{(\theta_3)}$$

Then, second point's height can be obtained:

$$P2_y = P1_y + d2 \cdot \cos{(\theta'_4)}$$

Finally, the y variable can be cleared:

$$Pf_y = y = P2_y$$

$$\boxed{y = -lw \cdot \sin{(\beta)}}$$

When inserted this calculation procedure into robot's program, two approaches can be considered, being one the direct calculation of these parameters and being the second one a more stepped approximation by using intermediate variables that shall consume more memory but can potentially avoid the repetition of some calculations.

<u>Inverse kinematics model</u>

This model explains how, given a final position and end-effector two-axis orientation, the angles the different motors need to cover are. For that, an input of the kind (x, y, z, α, β) where (x, y, z) represent cartesian coordinates and (α, β) represent Y and Z orientation of the end-effector shall correspond to a 5-variable output (θ_1, θ_2, θ_3, θ_4, θ_5), with from the left to the right being the angles of global linear displacement, global rotation, shoulder rotation, elbow rotation and wrist rotation.

In order to do the calculations, leaning on *fig. 132* shall be necessary to define new intermediate variables that ease the calculations and provide a comprehensible graphical approach:

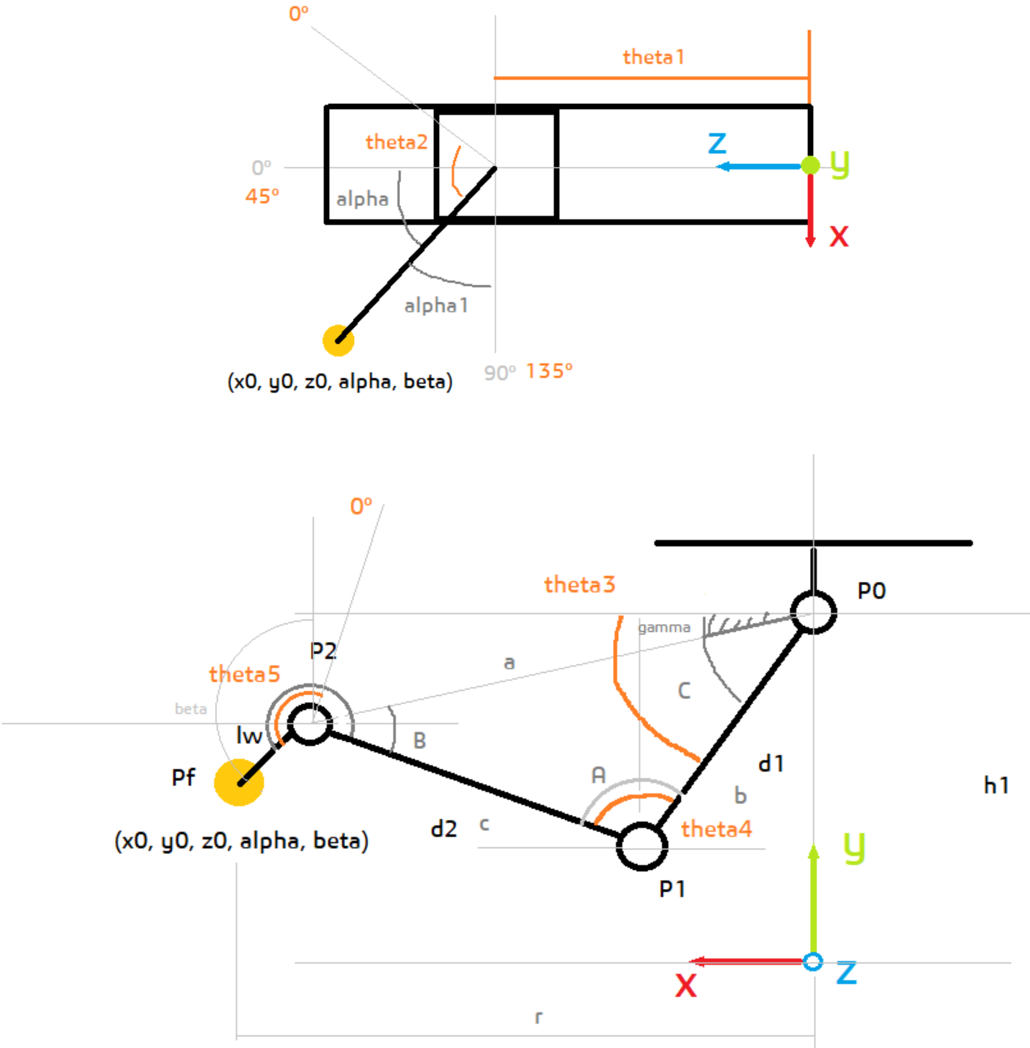


Figure 132. *Reference frame placement on robot's working space*

Source: Own

NOTE: The lw distance is considered to be the distance from the wrist joint to the half of the end-effector's gripper. It is considered the region with the higher grip.

The first step is clearing the expression for α_1, which is an auxiliary variable:

$$\alpha_1 = 135° - \alpha$$

This value is due to the fact that the 90° angle regarding rotation corresponds to 135° of the position servomotors when centered. The next logical step is calculating "r", which corresponds to the reach of the arm when extended in the final configuration when the arm translates to the determined position. In order to do this, the following development is considered logical:

$$x = r \cdot \cos(\alpha_1) \rightarrow r = \frac{x}{\cos(\alpha_1)}$$

It is necessary to now define what shall be the point where the base shall be located (regarding displacement) such that with the given rotation angle α the point reached is the given point (x, y, z). This displacement shall be called z':

$$z' = z - r \cdot \sin(\alpha_1)$$

Given the definition for "r", z', which corresponds to the theoretical value of θ_1 shall be:

$$z' = \theta'_1 = z - \frac{x}{\cos(\alpha_1)} \cdot \sin(\alpha_1)$$

$$\theta'_1 = z - x \cdot tg(\alpha_1)$$

This value of θ'_1 is not the definitive angle to be written by the joint, but it needs a certain treatment in order to adapt it to the characteristics of such displacement joint. Since the measurement of distance is done respect to the complementary space to the desired displacement (measuring right space while displacement is considered to be the distance respect to the left global reference axe), this shall be taken into account and the equation modified to fit this real situation. The real joint displacement shall therefore be the right beam border (800 – 20 mm) minus the theoretical distance:

$$\boxed{\theta_1 = 780\ mm - \theta_1'}$$

Regarding θ_2, it needs to be said that it can be obtained directly from the upper drawing on *fig. 132*, having an offset of 45° with respect to α:

$$\theta_2 = \alpha + 45°$$

The next step is to calculate the inverse kinematics of the wrist point such that the wrist is oriented in the desired angle β respect to the vertical plane. For this, a new reference point must be computed starting from the final point following the triangle of *fig. 133*:

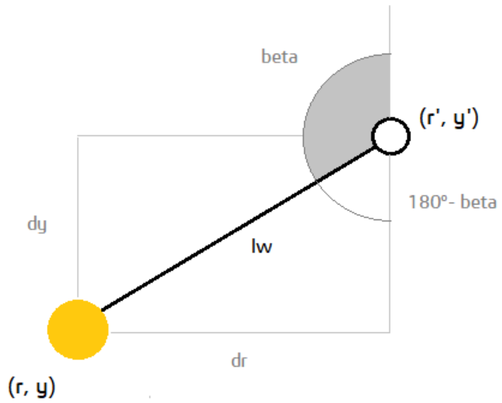


Figure 133. *End-effector and wrist triangle*

Source: Own

Given the initial final point (r,y), which is the projection of the point on arm's deployment length "r", the wrist point can be calculated by means of simple trigonometry. First, the two displacements "dr" and "dy" have to be calculated:

$$dr = lw \cdot \sin(180 - \beta)$$

$$dy = lw \cdot \cos(180 - \beta)$$

Once the displacements have been obtained, the wrist point can be calculated by subtracting or adding the displacements (depends on which displacement it is being considered) to the original projection point. The result can be seen below:

$$r' = r - dx \rightarrow r' = r - lw \cdot \sin(180 - \beta)$$

$$y' = y + dy \rightarrow y' = y + lw \cdot \cos(180 - \beta)$$

This newly obtained coordinates shall allow further calculations on arm deployment angles. In order to continue with the procedure, the cosine theorem needs to be applied to the triangle P0P1P2 seen on *fig. 133*. According to this theorem (applied to the mentioned triangle):

$$c^2 = a^2 + b^2 - 2ab \cdot \cos(C)$$

This equation, when changing the different sides of the triangle by the name they really have for this robot, results in the following:

$$d2^2 = a^2 + d1^2 - 2 \cdot a \cdot d1 \cdot \cos(C)$$

On this equation, “a” is the distance between P0 and P2; it is therefore the module of the vector that joins these two points:

$$a = \sqrt{r'^2 + (y' - h1)^2}$$

This module is added to the previous equation and cleared respect to C, resulting in:

$$C = \arccos\left(\frac{d2^2 - \sqrt{r'^2 + (y' - h1)^2} - d1^2}{-2 \cdot d1 \cdot \sqrt{r'^2 + (y' - h1)^2}}\right)$$

Now the triangle between shoulder’s attachment to the mobile base and wrist joint has been deduced. However, there could exist an infinite number of triangles of these characteristics if the angle at which it is oriented respect to the base is not given. Therefore, the next step consists on the calculation of this triangle’s inclination, for which the angle ϒ shall be computed:

$$\Upsilon = \arccos\left(\frac{r'}{a}\right) \rightarrow \Upsilon = \arccos\left(\frac{r'}{\sqrt{r'^2 + (y' - h1)^2}}\right)$$

The third angle θ_3 can now be computed directly from C and ϒ:

$$\boxed{\theta_3 = C + \Upsilon}$$

Now, relative to P2 from P0, only the angle θ_4 is left to be computed. For this, firstly the height of P1 needs to be computed:

$$P1_y = h1 - d1 \cdot \sin(\theta_3)$$

With this, θ_4 can be defined as:

$$\theta_4 = 90° - \arcsin\left(\frac{y' - P1_y}{d2}\right)$$

$$\boxed{\theta_4 = 90° - \arcsin\left(\frac{y' - (h1 - d1 \cdot \sin(\theta_3))}{d2}\right)}$$

Finally, θ_5 is directly obtained as:

$$\boxed{\theta_5 = \beta + \theta_4}$$

A1.3.2. Dynamics calculations

The dynamics model aims at calculating the maximum mass the robot shall be able to carry in each worst-case position from several joints, starting by the wrist joint, continuing with the elbow joint and finishing with the shoulder joint. The study of previous joints in the kinematic chain shall not be performed since they do not have an effect on holding capabilities.

These different calculations aim at explaining what the maximum holding torque of the motors is capable of doing to hold a suspended object on the middle of the end-effector.

In order to simplify calculations, links shall be assumed to have a uniformly distributed mass and their center of gravity, where the weight of the pieces shall be applied, shall be considered to be placed right at the middle of each segment. The mass for each link has resulted from slicing the different pieces with the *Ultimaker Cura* program, which prepares pieces to be 3D printed and retrieves the mass of such parts when made with the given material (in this case PLA). The addition of motors weights and other mechanical additions has also been considered.

Calculations performed shall be done considering 2D conditions on XY plane independently of Y rotation and therefore independently of the Z coordinate, since this angle has no effect on holding torque.

The result providing the minimum amount of mass able to be hold shall be the limiting factor independently of the results of other joints.

Wrist dynamics

Wrist joint is the one closer to the mass, therefore, it is the holding position where the distance shall have a lower effect and the mass a bigger effect.

Firstly, it is necessary to rely on a diagram to continue with the calculations:

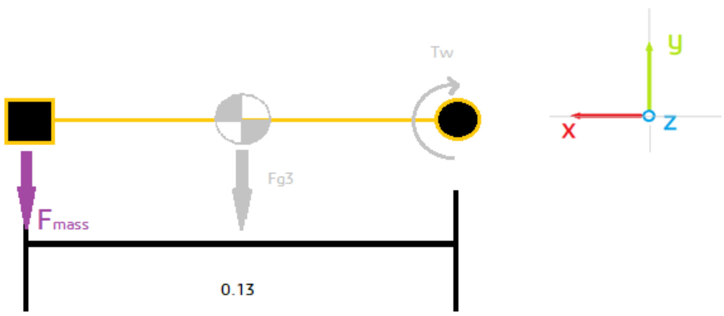


Figure 134. *Wrist weights and torques diagram*

Source: Own

The position seen on *fig. 134* has been chosen because it is the configuration at which the loads are maximum and therefore the motor shall need to push more. This reasoning is supported by the following equation:

$$T = F \cdot l \cdot \sin(\theta)$$

From this equation it can be deduced that, given "l" distance and "F" forces applied by the loads and which cannot be modified by the chosen configuration, the torque exerted by the loads "T" shall be maximum when sin(θ) is maximum, and that is for a 90° angle from the force respect to the link.

As it is desired to calculate the maximum mass that can be held given a situation of equilibrium it is necessary to determine what this means. Equilibrium shall be considered when, presented the different torques of the diagram the sum of them equals 0, meaning no angular acceleration shall occur, which, applied to this case, is equivalent to stating that the torque exerted by the motor is equal to the torque exerted by the different loads for a maximum object mass. This can be represented by the following equations:

$$Tw - \sum_{j=1}^{n} Tj = 0 \rightarrow Tw = \sum_{j=1}^{n} Tj \rightarrow Tw = \sum_{j=1}^{n} (Fj \cdot lj \cdot \sin(90°)) \rightarrow Tw = \sum_{j=1}^{n} (Fj \cdot lj)$$

Developing this last expression respect to the diagram on *fig. 134*:

$$Tw = \sum_{j=1}^{n} (Fj \cdot lj) \rightarrow Tw = (0.13 \cdot Fmass + 0.065 \cdot Fg3)$$

The first step to be performed is obtaining the maximum torque able to be exerted by the servomotor in charge of wrist's movement (Tw), which is model DS04-NFC continuous rotation servomotor. This number, according to the datasheet, is 5.5 kg·cm, which needs to be converted into N·m units:

$$Tw = \frac{5.5\,kg \cdot 9.8067\,\frac{m}{s^2}}{\frac{100\,cm}{1\,m}} \cdot cm = 0.539\,N \cdot m$$

The next step is the calculation of the different forces exerted. They are gravitational forces, so their value shall be the mass of the respecting object multiplied by the gravity. The mass for link 3 has been calculated as seen on the introduction to this section.

$$Fg3 = m3 \cdot g = \ 0.1\,kg \cdot 9.8067\,\frac{m}{s^2} = 0.98067\,N$$

$$Fmass = mass \cdot g = mass \cdot 9.8067\,N$$

When applying these two results to the torque equilibrium equation, the result is the following:

$$Tw = (0.13 \cdot Fmass + 0.065 \cdot Fg3)$$

$$0.539 = (0.13 \cdot mass \cdot 9.8067 \ + 0.065 \cdot 0.98067)$$

$$mass = \frac{0.539 - 0.065 \cdot 0.98067}{0.13 \cdot 9.8067} = 0.373\,kg$$

The wrist is able to hold straight a maximum mass of **0.373 kg** or, what it is the same, 373 grams.

Elbow dynamics

This section aims at calculating the ability of the elbow to hold mass placed at the end-effector given the worst-case scenario, which shall be set when the forearm piece is parallel to the XZ plane from *fig. 135,* position at which the holding torque shall be maximum. These calculations shall not take into account the ability of the wrist to hold the weight, since this has been covered on the last section.

The following diagram showcases the different forces applied to the aforementioned forearm link.

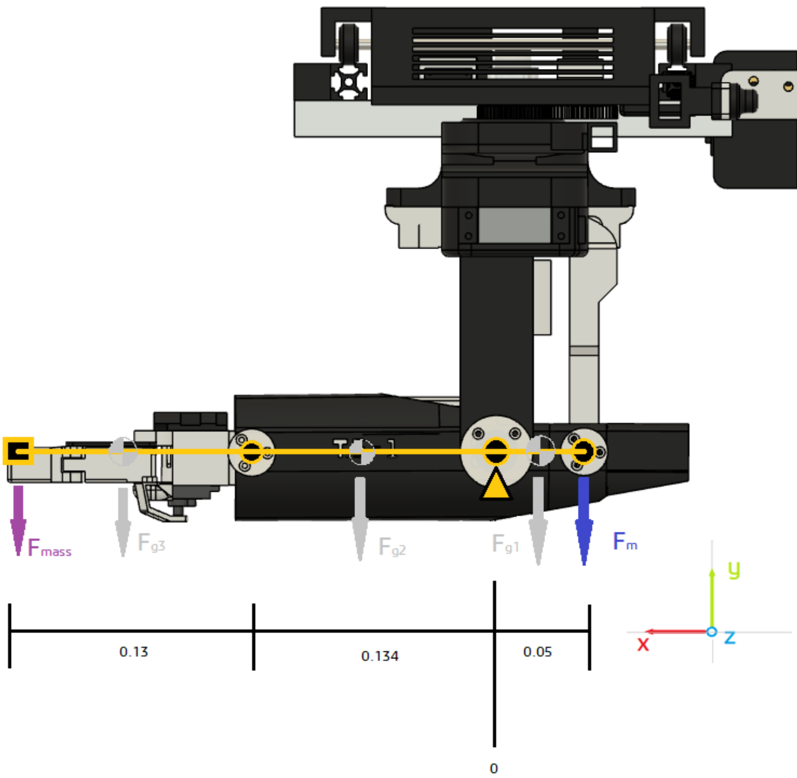


Figure 135. *Elbow weights and torques diagram*

Source: Own, Autodesk Fusion 360

The equilibrium equations for this particular case shall be posed related to the elbow joint, which shall be the 0 point.

$$Telbow + Fg1 \cdot l1 = \sum_{j=1}^{n} (Fj \cdot lj)$$

$$Telbow + Fg1 \cdot l1 = Fg2 \cdot l2 + Fg3 \cdot l3 + Fmass \cdot lmass$$

$$Telbow + Fg1 \cdot 0.025 = Fg2 \cdot 0.067 + Fg3 \cdot 0.2 + Fmass \cdot 0.264$$

The only unknown that must be left on the previous equation is the mass of the object. Therefore, the rest of variables must be corresponded to a value, being Telbow the torque exerted by elbow's 25 kg·cm DS3225 servomotor, Fg1 the weight exerted by the rear of the forearm, Fg2 the weight exerted by the forearm on its frontal side, Fg3 the weight of wrist's link and Fmass the weight of the object.

The first step to clear these variables is translating servomotor's maximum holding torque to standard units (N·m):

$$Telbow = \frac{25\ kg \cdot 9.8067 \frac{m}{s^2}}{\frac{100\ cm}{1\ m}} \cdot cm = 2.45195\ N \cdot m$$

Then, each of the gravitational pulls on the different link and object masses need to be calculated:

$$Fg1 = m1 \cdot g = \ 0.008\ kg \cdot 9.8067 \frac{m}{s^2} = 0.078\ N$$

$$Fg2 = m2 \cdot g = \ 0.3\ kg \cdot 9.8067 \frac{m}{s^2} = 2.942\ N$$

$$Fg3 = m3 \cdot g = \ 0.1\ kg \cdot 9.8067 \frac{m}{s^2} = 0.98067\ N$$

$$Fmass = mass \cdot g = mass \cdot 9.8067\ N$$

Then, each of the resultant forces are added to the equilibrium equation:

$$2.45 + 0.078 \cdot 0.025 = 2.942 \cdot 0.067 + 0.98067 \cdot 0.2 + mass \cdot 9.8067 \cdot 0.264$$

By clearing this equation respect to mass, it can be found its value for this maximum situation:

$$mass = \frac{2.45195 - 0.197114 - 0.196134}{2.589} = 0.795\ kg$$

The maximum mass that the elbow is able to hold straight is **0.795 kg**, or what it is the same, 795 grams.

Shoulder dynamics

This section aims at studying maximum mass to be hold by the arm related to shoulder servomotors' capacity. This shall be studied by means of studying a worst-case scenario at which shoulder to elbow joint is holding the whole arm horizontally to the XZ plane.

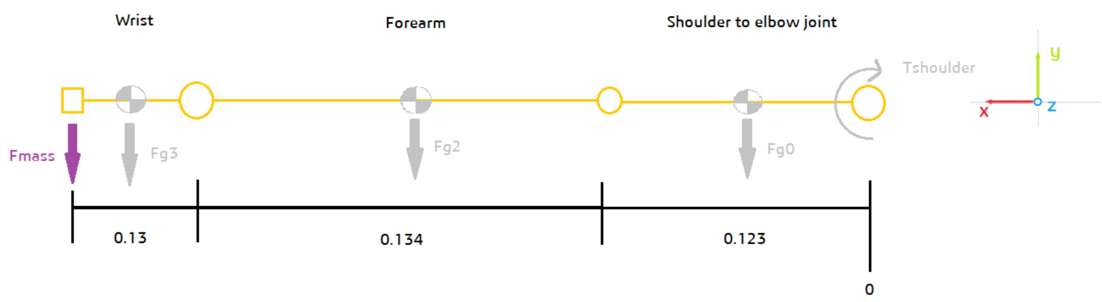


Figure 136. *Shoulder weights and torques diagram*

Source: Own

On *fig. 136*, the mass on the rear side of the forearm (Fg1) has been neglected since it is small enough to not cause appreciable changes on calculations.

The first step is to propose the torque equilibrium equation for this particular case:

$$Tshoulder = \sum_{j=1}^{n} (Fj \cdot lj)$$

$$Tshoulder = Fg0 \cdot l0 + Fg2 \cdot l2 + Fg3 \cdot l3 + Fgmass \cdot lmass$$

$$Tshoulder = Fg0 \cdot 0.0615 + Fg2 \cdot 0.19 + Fg3 \cdot 0.322 + Fgmass \cdot 0.387$$

The torque exerted by the motors shall be the double that any one of them can produce, therefore being 50 kg·cm which, translated to N·m is the double of the value calculated for Telbow, that is, 4.9039 N·m.

The value for Fg2 and Fg3 shall be the same calculated for the elbow section, so it is not necessary to repeat this process. Regarding Fg0, it must be said that the mass of both parallel pieces (as observed in *fig. 47* and *fig. 49*) shall be added one to the other in order to make their whole weight for these calculations realistic, therefore having a mass of 0.15 kg.

$$Fg0 = m0 \cdot g = \ 0.15\ kg \cdot 9.8067 \frac{m}{s^2} = 1.471005\ N$$

When adding the values to the initial equilibrium equation, the following result can be appreciated:

$$Tshoulder = 1.471 \cdot 0.0615 + 2.942 \cdot 0.19 + 0.981 \cdot 0.322 + Fgmass \cdot 0.387$$

$$4.9039 = 1.471 \cdot 0.0615 + 2.942 \cdot 0.19 + 0.981 \cdot 0.322 + mass \cdot 9.807 \cdot 0.387$$

$$mass = \frac{4.9039 - 0.0905 - 0.559 - 0.3159}{3.795} = 1.038\ kg$$

According to these calculations, from the shoulder joint, the arm is able to hold in its worst-case scenario a weight of **1.038 kg** or, what it is the same, 1038 grams.

Annex II: Economic study

A2.1. INTRODUCTION

This Annex serves the purpose of studying the rentability of this project considering a plausible future commercialization. All necessary quantities that are not hypothetical have been calculated on section 4. (“BUDGET”).

This study shall focus on the economical method, which balances incomes versus expenses.

A2.2. ECONOMIC STUDY

The first step is identifying the variables that shall influence the selling price and, once this has been stablished, checking the profitability of the project and if this is able to cover the expenses it has generated.

Determining the target niche, the volume of sales and the selling price

This product targets at people that want to initiate in robotics by learning the basics and operating a robot, as much as to people that already have a background on this field but want to go deeper with program development and more specific applications including artificial vision, position sensing, human-machine interfaces and robotic interaction.

Similar products studied on section 1.2.3. (“SIMILAR EXISTING DEVICES”) are well known to have good sales numbers in fields like education and research, since they have the perfect size and set of capabilities to fit the necessities these clients may have. So, it can be expected that this niche is equally interested in a product like the one presented in these pages.

The volume of sales shall depend on a series of factors, including the popularity of the product (which depends on the marketing campaign developed) which, shall grow in time (it is very well known that people spread their opinion regarding products they like, for which a great user experience is necessary), and the amount of money the client may want to pay for the product (for which a well-determined selling price is key). This number has to be kept conservative and has been hypothesized to be **1000 units** the first year.

The selling price needs to be a low as possible and superior to the sum of the manufacturing price (+VAT), with a price of 653.23 € (see section 4. (“BUDGET”)), plus the variable costs, with a price of 12.12 € per unit (see section 4. (“BUDGET”)), taking also the fixed costs into account. It is established as a **10 % superior** to the total price (665.35 €), which is an amount of **731.89 €**.

Determining total income and total costs

With the expected volume of sales, which has been determined on previous paragraphs (1000 units), the total income is considered to be the selling price of each product (731.89 €) times the volume of sales. The number obtained is **731890 €**.

The total costs can be defined as the sum of the fixed costs, the total manufacturing costs (volume of sales times the price it costs to manufacture each product (653.23 €)) and the total variable costs (volume of sales times the variable cost for each unit). This number is **678818 €**.

Determining the gross profit, the rate of return and the sales balance

The gross profit (B) is defined as:

$$B = Total\ Income - Total\ Costs$$

Therefore, the gross profit for the project in this hypothetical case is 731890 € minus 678818 €, which results **53072 €**.

The rate of return (r) represents the amount of value invested that returns as means of benefit:

$$r = \frac{B}{investment} \cdot 100$$

For this project, the amount invested can be considered to be equal to the fixed costs (13468 €), since these costs pose a bet that may or may not be recovered later and no machines have been bought for the project. The rate of return for the project is a **394.06 %**.

The sales balance (n_{eq}) represents the number of units to be sold such that the income covers the fixed costs totally and starts producing benefits. It is calculated as:

$$n_{eq} = \frac{Fixed\ costs}{Unit\ price - Unit\ variable\ costs}$$

This value, for the project, represents a total of **18.71 units**, which means that when selling 19 units, the project could start being profitable.

A2.3. CONCLUSIONS

This project is an academic exercise, which means that no economical or financial operations have been made apart from the purchase of the necessary materials. However, as it is proven by the numbers given above, it could be a profitable project with an adjusted benefit margin of only a 10 %. It is very possible that at least 19 units could be sold given the appropriate marketing; it is not a far-fetched assumption.

The first year when initiating an enterprise project is very unstable regarding many aspects of the demand: sales usually increase as years go by even though this depends on many other factors. Knowing 19 units are the only demand from which benefits can be made is encouraging.

On the other hand, it is true that this is not the most affordable didactic robot in the market, but it is not the most expensive one either. What is more, potential clients could be universities and educational institutions that may not be so constrained as an individual client regarding budget.

All in all, this could be a viable project if marketing is well done and production items can be assembled to work together fluidly, generating benefit reasonably early. However, if it is desired to generate a massive amount of benefit, the selling price would need to be increased always taking into account the effects that this could cause on demand.

Annex III: Code

A3.1. INTRODUCTION

This Annex includes the code that has been referenced on section 1.5.5. ("Code") organized in different tables, enabling better readability of such section of the report.

A3.2. MAIN DEVICE CODE

These pieces of code reference the software uploaded to the proper robot arm. They can be seen below and are organized in tables according to how they have been referenced.

```
// Sketch designed for an Arduino Mega 2560 development board

#include <Servo.h> // 1. MOVEMENT PARAMETERS AND DEFINITIONS ----
#include <SoftwareSerial.h> // 2. COMMS PARAMETERS AND DEFINITIONS ----
#include <Wire.h>
#include <VL53L0X.h> // Laser lateral distance measurements
```

Table 25. *Library declaration*

Source: Own

```
// 1. MOVEMENT PARAMETERS AND DEFINITIONS ---------------------------------------------

const int dT = 20; // step size parameter, must be greater than 20 (ms)
const int T = 600; // movement period between positions
const int N = T / dT; // number of steps
```

```
#define GJOINTS 5 // Global joints until end-effector
#define 6V_rail 30 // 6V relay power control circuit MOSFET's gate pin
#define 5V_rail 32 // 5V relay power control circuit MOSFET's gate pin
#define Effector_relay 31 // Effector's relay power control circuit MOSFET's gate pin

boolean servos_5V_activated = false;

#define ACTIVATE_MOTORS digitalWrite(6V_rail, HIGH);digitalWrite(5V_rail, HIGH);
#define DEACTIVATE_MOTORS digitalWrite(6V_rail, LOW);digitalWrite(5V_rail, LOW);
#define ACTIVATE_EFFECTOR digitalWrite(Effector_relay, HIGH); servos_5V_activated = true
#define DEACTIVATE_EFFECTOR digitalWrite(Effector_relay, LOW); servos_5V_activated = false
#define Wrist_rotation {map(analogRead(A1), 240, 615, 35, 115)-28} // Angle reading

Servo servos[10]; // Number of servo motors in TS1

typedef struct {
  uint8_t pin;
  int offset;
  int min_pos;
  int max_pos;
} TS1_Servo;
```

Table 26. *Movement parameters and macros*

Source: Own

```
// Servo offsets
int rot_offset = -10, vert_offset = -38, horiz_offset = 36, adv_offset_right = -5, adv_offset_left = 0, d_inf_offset = 0, d_sup_offset = 0, i_inf_offset = 0, i_sup_offset = 8, wrist_offset = -1;
int limit_shoulder_max = 160, limit_shoulder_min = 60;
int wrist_margin = 4, disp_margin = 5, laser_margin = 0, pot_margin = 0;

// Limits on each joint's servos
TS1_Servo lateral_disp = {9, 0, 0, 360};
TS1_Servo rot = {2, rot_offset, 90, 180}; // [0, 270] boundaries in simulation, 45 & 225
TS1_Servo advance_right = {3, adv_offset_right, 60, 180};
TS1_Servo advance_left = {4, adv_offset_left, 0, 180};
TS1_Servo vertical = {8, vert_offset, 25, 150};
TS1_Servo wrist = {7, wrist_offset, 50, 115};
TS1_Servo grp_d_inf = {6, d_inf_offset, 0, 180};
TS1_Servo grp_d_sup = {13, d_sup_offset, 0, 180};
TS1_Servo grp_i_inf = {5, i_inf_offset, 0, 180};
TS1_Servo grp_i_sup = {11, i_sup_offset, 0, 180};
```

Table 27. *Servo declarations*

Source: Own

```
// 2. COMMS PARAMETERS AND DEFINITIONS --------------------------------------------------

#define COM Serial
#define HC12 Serial1
```

```
#define BT Serial2
#define USER Serial3
#define lights_on digitalWrite(25, HIGH)
#define lights_off digitalWrite(25, LOW)
#define NOTIFICATION lights_on; delay(50); lights_off;delay(50); lights_on;delay(50);lights_off;

// 0: Computer serial, 1: Radio, 2: BT/BLE, 3: External comms at 9600 bds predefined, 4: Pick & place
int communication_mode = 2;  boolean received = false;
```

Table 28. *Different communication parameters*

Source: Own

```
// 3. OTHER PIN DECLARATIONS AND VARIABLES ---------------------------------------------

VL53L0X laser;
#define Lights 25 // Lights placed on robot's gripper
String hello = ""; // Global info variable
boolean Speak_Now = true; // Enables robot to speak/reply through the app

// Kinematic generals
float Glob_displacement = 300, Glob_rot = 135, Glob_horizontal = 90, Glob_vertical = 90, Glob_wrist = 90;
float Glob_sup = 60, Glob_inf = 60;
float offset_horizontal = 38, offset_vertical = 32;
float h1 = 33, d1 = 12.5, d2 = 13.5, lw = 9; // in cm
const double min_distance = 35.0; // in mm
```

```
const double max_distance = 570.0; // in mm

const double midway = 380.0;

const double HOME_POSITION[GJOINTS] = {midway, 135.0, 90.0, 90.0, 90.0};

double currentQ[GJOINTS] = {Glob_displacement, Glob_rot, Glob_horizontal, Glob_vertical, Glob_wrist};
```

Table 29. *Kinematic generals and other declarations*

Source: Own

```
// 4. FUNCTION DEFINITIONS --------------------------------------------------------------------

float mapFloat(float var, float in_min, float in_max, float out_min, float out_max);

void SPEAK(String output);

void attachServo(TS1_Servo &servo);

void detachServo(TS1_Servo &servo);

void writeServo(TS1_Servo &servo, int angle1);

void writeAdvServo(int angle);

boolean rotWrist(int angle);

boolean displaceRobot(int target_distance);

void openGripper(int percentage_right, int percentage_left);

void Check_comms();

void ROBOT_PROGRAM();

void TEST_EFFECTOR();

void computeCubicTrajectory(const double q0[GJOINTS], const double qT[GJOINTS], const double T, double *ai, double *bi, double *ci, double *di);

void evalCubicTrajectory(const double ai[GJOINTS], const double bi[GJOINTS], const double ci[GJOINTS], const double di[GJOINTS], double t, double *qtraj);

void moveAbsJ(const double q0[GJOINTS], const double qT[GJOINTS], const double
```

```
T);
void moveAbsJ0(const double qT[GJOINTS], const double T);
void goToPoint(float x, float y, float z, float alpha, float beta);
void START_ROBOT();
```

Table 30. *Function definitions*

Source: Own

```
// 4.1. GENERAL FUNCTIONS DECLARATION
float mapFloat(float var, float in_min, float in_max, float out_min, float out_max) {
  return (var - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
void SPEAK(String output) {
  if (Speak_Now) Serial2.println(output);
}

// 4.2. SERVO FUNCTIONS DECLARATIONS
void attachServo(TS1_Servo &servo) {
  servos[servo.pin].attach(servo.pin);
}
void detachServo(TS1_Servo &servo) {
  servos[servo.pin].detach();
}

// 4.3. BASIC MOVEMENT FUNCTIONS DECLARATION
void writeServo(TS1_Servo &servo, int angle1) {
  if ((servo.pin == 7 || servo.pin == 12) && !servos_5V_activated) {
```

```
    return 0;
  } else {
    attachServo(servo);
    angle1 = constrain(angle1 + servo.offset, servo.min_pos, servo.max_pos);
    servos[servo.pin].write(angle1);
  }
  if (servo.pin == 8) Glob_vertical = angle1 - vertical.offset;
  else if (servo.pin == 2) Glob_rot = angle1 - rot.offset;
}


void writeAdvServo(int angle) {
  attachServo(advance_right);
  attachServo(advance_left);
  angle += horiz_offset;
  if (angle >= limit_shoulder_min && angle <= limit_shoulder_max) {
    int dom_angle = constrain(angle + adv_offset_right, advance_right.min_pos,
advance_right.max_pos);
    int beta_angle = mapFloat(angle, advance_right.max_pos, advance_right.min_pos,
advance_left.min_pos, advance_left.max_pos);
    servos[advance_right.pin].write(angle);
    servos[advance_left.pin].write(beta_angle);
  } else if (angle < limit_shoulder_min) {
    angle = limit_shoulder_min;
    int beta_angle = mapFloat(angle, advance_right.max_pos,
advance_right.min_pos, advance_left.min_pos, advance_left.max_pos);
    servos[advance_right.pin].write(angle);
    servos[advance_left.pin].write(beta_angle);
  } else if (angle > limit_shoulder_max) {
    angle = limit_shoulder_max;
    int beta_angle = mapFloat(angle, advance_right.max_pos,
advance_right.min_pos, advance_left.min_pos, advance_left.max_pos);
```

```
    servos[advance_right.pin].write(angle);
    servos[advance_left.pin].write(beta_angle);
  }
  Glob_horizontal = angle;
}

boolean rotWrist(int angle) {
  attachServo(wrist);
  int current = Wrist_rotation;
  current += pot_margin;
  float K = 1.5; // 1.5 is fine
  if (abs(angle - current) >= 5) {
    int U = -K * (angle - current);
    if (U > 90) U = 90;
    else if (U < -90) U = -90;
    servos[wrist.pin].write(90 - U);
  } else if (abs(current - angle) <= wrist_margin && abs(angle - current) < 5) {
    if (angle < current) servos[wrist.pin].write(0);
    else if (angle > current) servos[wrist.pin].write(180);
  }
  if (abs(current - angle) <= wrist_margin) {
    servos[wrist.pin].write(90);
    detachServo(wrist);
    return true;
  } else return false;
  detachServo(wrist);
}

boolean displaceRobot(int target_distance) {
```

```
  attachServo(lateral_disp);

  int distance = laser.readRangeContinuousMillimeters();
  float K = 1.5; // 1.5 is fine
  distance -= laser_margin;
  if (laser.timeoutOccurred()) Serial.println("Timeout");
  target_distance = constrain(target_distance, min_distance, max_distance);
  int U = K * (target_distance - distance);
  if (U > 90) U = 90;
  else if (U < -90) U = -90;
  servos[lateral_disp.pin].write(90 - U);
  if (abs(distance - target_distance) <= disp_margin) {
    servos[lateral_disp.pin].write(90);
    detachServo(lateral_disp);
    return true;
  }
  else return false;
}

void openGripper(int percentage_right, int percentage_left) {
  int right = constrain(percentage_right, 0, 100);
  int left = constrain(percentage_left, 0, 100);
  right = map(right, 0, 100, 60, 145);
  left = map(left, 0, 100, 145, 37);
  writeServo(grp_d_inf, right);
  writeServo(grp_i_inf, left);
  delay(500);
}
```

```
// 4.4. COMMUNICATION FUNCTION DECLARATION

void Check_comms() {
 // Radio received
 char last = 'a';
 String in = "";
 String ax1 = "", ax2 = "", ay1 = "", az1 = "", gx1 = "", gy1 = "", gz1 = "";
 if (HC12.available()) {
  in = HC12.readStringUntil('w');
  int index1 = in.indexOf(",");
  ax1 = in.substring(0, index1);
  int index2 = in.indexOf(",", index1 + 1);
  ay1 = in.substring(index1 + 1, index2);
  index1 = in.indexOf(",", index2 + 1);
  az1 = in.substring(index2 + 1, index1);
  index2 = in.indexOf(",", index1 + 1);
  gx1 = in.substring(index1 + 1, index2);
  index1 = in.indexOf(",", index2 + 1);
  gy1 = in.substring(index2 + 1, index1);
  gz1 = in.substring(index1 + 1);
  ax = ax1.toInt();
  ay = ay1.toInt();
  az = az1.toInt();
  gx = gx1.toInt();
  gy = gy1.toInt();
  gz = gz1.toInt();
  received = true;
  communication_mode = 1;
```

```
  } else {
    ax = 0;
    ay = 0;
    az = 0;
    gx = 0;
    gy = 0;
    gz = 0;
    received = false;
    communication_mode = 0;
  }
  if (Serial2.available()) {
    hello = Serial2.readString();
    communication_mode = 2;
    return 0;
  }
}


// 4.5. COORDINATED MOVEMENTS AND KINEMATIC MODELS FUNCTIONS DECLARATION


void TEST_EFFECTOR() {
  ACTIVATE_EFFECTOR;
  openGripper(100, 100);
  openGripper(50, 50);
  openGripper(0, 0);
  openGripper(100, 100);
  openGripper(0, 0);
  openGripper(100, 100);
  DEACTIVATE_EFFECTOR;
}
```

```
void computeCubicTrajectory(const double q0[GJOINTS], const double qT[GJOINTS],
const double T, double *ai, double *bi, double *ci, double *di) {
  for (int i = 0; i < GJOINTS; i++) {
    ai[i] = -2 * ((qT[i] - q0[i]) / (pow(T, 3)));
    bi[i] = 3 * (qT[i] - q0[i]) / (pow(T, 2));
    ci[i] = 0;
    di[i] = q0[i];
  }
}

void evalCubicTrajectory(const double ai[GJOINTS], const double bi[GJOINTS], const
double ci[GJOINTS], const double di[GJOINTS], double t, double *qtraj) {
  for (int i = 0; i < GJOINTS; i++) {
    qtraj[i] = ai[i] * pow(t, 3) + bi[i] * pow(t, 2) + ci[i] * t + di[i];
  }
}

void moveAbsJ(const double q0[GJOINTS], const double qT[GJOINTS], const double
T) {
  // No displacement needed when no change in position, smoother operation
  if ((q0[0] != qT[0]) || laser.readRangeContinuousMillimeters() - 20 != q0[0]) while
(!displaceRobot(qT[0]));
  double ai[GJOINTS], bi[GJOINTS], ci[GJOINTS], di[GJOINTS];
  double qtraj[GJOINTS];
  for (int x = 0; x < GJOINTS; x++) {
    qtraj[x] = q0[x];
  }
  computeCubicTrajectory(q0, qT, T, ai, bi, ci, di);
  double t = 0;
```

```
for (int j = 0; j < N; j++) {
  long init1 = millis();
  evalCubicTrajectory(ai, bi, ci, di, t, qtraj);
  moveServos;
  if (q0[GJOINTS - 1] != qT[GJOINTS - 1]) {
    if (!rotWrist(qT[GJOINTS - 1]));
  }
  if (millis() - init1 < dT) delay(init1 + dT - millis());
  t = t + dT;
}
// No wrist movement needed when no change in position, smoother operation
if (q0[GJOINTS - 1] != qT[GJOINTS - 1]) while (!rotWrist(qT[GJOINTS - 1]));
for (int x = 0; x < GJOINTS; x++) {
  qtraj[x] = qT[x];
}
for (int x = 0; x < GJOINTS; x++) {
  currentQ[x] = qT[x];
}
Glob_displacement = qT[0];
Glob_rot = qT[1];
Glob_horizontal = qT[2];
Glob_vertical = qT[3];
Glob_wrist = qT[4];
moveServos;
delay(20);
}

void moveAbsJ0(const double qT[GJOINTS], const double T) { // Moves axis coordinatedly from the Global position, not from a specific configuration to another
 const double q0[GJOINTS] = {Glob_displacement, Glob_rot, Glob_horizontal,
```

```
Glob_vertical, Glob_wrist};
 // No displacement needed when no change in position, smoother operation
 if ((q0[0] != qT[0]) || laser.readRangeContinuousMillimeters() - 20 != q0[0]) while (!displaceRobot(qT[0]));
 double ai[GJOINTS], bi[GJOINTS], ci[GJOINTS], di[GJOINTS];
 double qtraj[GJOINTS];
 Serial.println("INI");
 for (int x = 0; x < GJOINTS; x++) {
   qtraj[x] = q0[x];
   Serial.println(q0[x]);
 }
 Glob_displacement = qT[0];
 Glob_rot = qT[1];
 Glob_horizontal = qT[2];
 Glob_vertical = qT[3];
 Glob_wrist = qT[4];
 computeCubicTrajectory(q0, qT, T, ai, bi, ci, di);
 double t = 0;
 for (int j = 0; j < N; j++) {
   long init1 = millis();
   evalCubicTrajectory(ai, bi, ci, di, t, qtraj);
   moveServos;
   if (q0[GJOINTS - 1] != qT[GJOINTS - 1]) {
     if (!rotWrist(qT[GJOINTS - 1]));
   }
   if (millis() - init1 < dT) delay(init1 + dT - millis());
   t = t + dT;
 }
 // No wrist movement needed when no change in position, smoother operation
 if (q0[GJOINTS - 1] != qT[GJOINTS - 1]) while (!rotWrist(qT[GJOINTS - 1]));
```

```
  for (int x = 0; x < GJOINTS; x++) {
    qtraj[x] = qT[x];
  }
  moveServos;
  delay(20);
}


void goToPoint(float x, float y, float z, float alpha, float beta) { // 3-coordinates point,
rotation angle, wrist angle
  // Check that the robot does not self-destruct
  if (x > 0 && ((x > 15 && y >= 25) || y < 25)) {
    float alpha1 = 135 - alpha;
    float r = x / cos(radians(alpha1));
    float theta1 = z - x * tan(radians(alpha1));
    float dx = lw * sin(radians(180 - beta));
    float dy = lw * cos(radians(180 - beta));
    float px = r - dx;
    float py = y + dy;
    float a = sqrt(pow(px, 2) + pow(py - h1, 2));
    float A = acos((pow(a, 2) - pow(d1, 2) - pow(d2, 2)) / (-2 * d1 * d2));
    float C = acos((pow(d2, 2) - pow(a, 2) - pow(d1, 2)) / (-2 * d1 * a));
    float theta2 = alpha;
    float gamma = acos(px / a);
    float theta3 = gamma + C;
    float hpA = h1 - d1 * sin(theta3);
    float theta4 = asin((py - hpA) / d2);
    float theta5 = beta + (degrees(theta4));
    //theta1 = (max_distance - min_distance) - (theta1 * 10);
    theta1 = (78 - theta1) * 10;
    theta3 = degrees(theta3);
```

```
  theta4 = 90 - degrees(theta4);
  const double qT[GJOINTS] = {theta1, theta2, theta3, theta4, theta5};
  moveAbsJ(currentQ, qT, T);
} else BT.println("Position out of boundaries");}
```

Table 31. *Function declarations*

Source: Own

```
#define HOME {moveAbsJ(currentQ, HOME_POSITION, T);currentQ[0] = HOME_POSITION[0];currentQ[1] = HOME_POSITION[1];currentQ[2] = HOME_POSITION[2];currentQ[3] = HOME_POSITION[3];currentQ[4] = HOME_POSITION[4];}

#define START_POSITION {NOTIFICATION;writeServo(vertical, Glob_vertical);delay(20);delay(1000);writeAdvServo(Glob_horizontal);delay(20);delay(1000);writeServo(rot, Glob_rot);delay(20);delay(1000);while (!displaceRobot(midway)); while (!rotWrist(90)); TEST_EFFECTOR(); NOTIFICATION;}

#define moveServos {writeServo(rot, qtraj[1]); writeAdvServo(qtraj[2]); writeServo(vertical, qtraj[3]);}


void START_ROBOT() {
 // 1. MOVEMENT INITIALIZATIONS
 pinMode(Movement_relay, OUTPUT);
 pinMode(Effector_relay, OUTPUT);
 DEACTIVATE_MOTORS;
 DEACTIVATE_EFFECTOR;
 // 2. COMMS INITIALIZATIONS
 Serial.begin(9600);
 Serial.setTimeout(10);
 Serial.println("Initializing systems");
```

```
  HC12.begin(9600);
  BT.begin(9600);
  USER.begin(9600);
  Wire.begin();
  if (laser.init()) {
    Serial.println("Laser initialized correctly");
  }
  laser.setTimeout(500);
  laser.startContinuous();
  // 3. OTHER PIN DECLARATIONS
  pinMode(Lights, OUTPUT); // Lights placed on robot's gripper
  ACTIVATE_MOTORS;
  START_POSITION;
  delay(2000);
}
```

Table 32. *Initialization sequence of the robot*

Source: Own

```
// SHORT PROGRAMMER GUIDE SUMMARY:
// Setup function must ALWAYS contain the START_ROBOT() sequence.
// Function Check_comms() is automatic and shall check receception
//  channels to seek for new user input.
//  This is the way the robot optimizes user input processing to avoid
//  unnecessary readings.
// Robot behaviour according to modes MUST be programmed in
//  ROBOT_PROGRAM();
```

```
// Robot MUST return to initial position before turning off
void setup() {
  START_ROBOT();
}


void loop() {
  Check_comms();
  ROBOT_PROGRAM();
}
```

Table 33. *Ideal program functions of the device*

Source: Own

A3.3. EXTERNAL DEVICE EXAMPLE CODE

Below, an example code is presented for an external device transmission system responsible for sending motion information of a human member for the robot to replicate.

```
#include <SoftwareSerial.h>
SoftwareSerial Radio(8, 5); // RX, TX
#include "I2Cdev.h"
#include "MPU6050.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif
#define OUTPUT_READABLE_ACCELGYRO
MPU6050 accelgyro;
```

```
int16_t ax, ay, az;
int16_t gx, gy, gz;
String R_out = "";

void setup() {
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
  Wire.begin();
#elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
  Fastwire::setup(400, true);
#endif
  Radio.begin(9600);
  accelgyro.initialize();
}

void loop() {
  accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  ax = map(ax, -17000, 17000, -99, 99);
  ax = constrain(ax, -99, 99);
  ay = map(ay, -17000, 17000, -99, 99);
  ay = constrain(ay, -99, 99);
  az = map(az, -17000, 17000, -99, 99);
  az = constrain(az, -99, 99);
  gx = map(gx, -17000, 17000, -99, 99);
  gx = constrain(gx, -99, 99);
  gy = map(gy, -17000, 17000, -99, 99);
  gy = constrain(gy, -99, 99);
  gz = map(gz, -17000, 17000, -99, 99);
  gz = constrain(gz, -99, 99);
```

```
  String ax1, ay1, az1, gx1, gy1, gz1;
  ax1 = String(ax);
  ay1 = String(ay);
  az1 = String(az);
  gx1 = String(gx);
  gy1 = String(gy);
  gz1 = String(gz);

  R_out = "S" + ax1 + "," + ay1 + "," + az1 + "," + gx1 + "," + gy1 + "," + gz1 + "w";
  Radio.println(R_out);
  delay(50);
  R_out = "";
}
```

Table 34. *External device example code (Radio)*

Source: Own

Annex IV: User manual

A4.1. INTRODUCTION

This section consists of a series of instructions that shall guarantee the correct and most profitable use of the device, regarding safety, coding and handling of the robot.

This is not an industrial robot. However, it has powerful actuators and electrical connections, so the user must always be aware of this and operate/modify the device carefully.

The robot is started once its three plugs are connected to the electrical grid and its main motors can start working when the switch on the top of its electronics box is turned on.

A4.2. SAFETY PRECAUTIONS

These safety guidelines must be followed in order to reduce risk of personal or material damage.

Regarding mechanical safety:

1. Stay away from robot's operation area at any time while it is active.

2. Never introduce no body member or object inside moving parts while the device is active. This can cause severe personal or material damage.

3. When modifying device's structure if developing new exchangeable pieces to be used, the attachment of such pieces must always be performed when the robot is inactive, that is, when it is not plugged and not being able to move by itself.

4. Always remove unnecessary or undesired tools or objects from the operation area since this can cause damage to such objects or to the robot.

5. When new software is introduced to the robot, it is always recommended to stay close to device's power switch in order to stop it. This must always be done staying away from underneath the box where the electronics are and keeping an eye on every movement the robot does. This can avoid robot's self-damage or any other damage done to property or human beings.

6. When new software is introduced to the robot, be careful not to make the robot perform any movements out of its design boundaries regarding operation or weight carrying since this can cause damage to itself or to the objects involved.

Regarding electrical safety:

1. This an electrical device, always be aware of this on whatever situation when operating or modifying the device in order to avoid electrical shocks and derived problems from this.

2. Device's inner electronics modifications shall be done at own risk since the device is not prepared for this kind of alterations.

3. Do not touch any electrical outlet directly with no body member and always avoid modifying these parts, since they are not intended to be modified.

4. When plugging the device to the power source, always mind that this can be a dangerous process and avoid unnecessary risks.

A4.3. PROGRAMMER'S GUIDE & DEVICE HANDLING

The robot shall be programmed by means of the Arduino IDE or similar or compatible Integrated Development Environments with the Arduino Mega 2560 (ATmega2560) board.

When developing any software that shall be later uploaded to the main controller board, it is necessary to firstly include all the built-in functions and parameter variables for the device on new files in order to ease the coding process and assure no damage to the robot is made by means of some direct coding mistake (wrong pin declarations, incorrect joint boundaries definition, etc.).

Once this code has been introduced in the program, the initialization of all the systems, which is a necessary process, shall be made by means of incorporating the function **START_ROBOT()** in the *void setup()* function of the Arduino new file structure.

Then, when developing the proper program for the robot, the function **Check_comms()**, if declared on the *void loop()* of the Arduino file, shall check the incoming information received by the different communication ports and shall assign this information to a *String* type variable named **hello**, with another complementary variable called **communication_mode** of type *int*, which states the communication port the last registered information has been read from being value 0 for computer board serial 0 communication port, value 1 for HC-12 (Radio) incoming information, value 2 for Bluetooth or BLE information and value 3 for external communications on the VGA port, with a base communication speed of 9600 bds.

The rest of the program must be included in *void loop()* by means of coding directly this function or creating an external one that shall be later included. There are some internal functions that the proper built-in functions use for several calculations which shall not be developed in this section.

The functions that can be used regarding basic operations and hardware capabilities are:

- **void SPEAK(String output)** -> This function accepts a *String* type variable which shall be directly outputted out of the Bluetooth communication port and which, with the app, can be converted into voice by a Text-to-Voice Engine and sound through smartphone speakers.

- **void attachServo(TS1_Servo &servo)** -> This function can attach a specified servomotor to their determined pin. It can be used when desiring to control a specific motor or set of motors.

- **void detachServo(TS1_Servo &servo)** -> This function can detach a specified servomotor from their determined pin. This can be useful when wishing to stop sending a specific motor a signal.

- **void writeServo(TS1_Servo &servo, int angle1)** -> This function allows to send a specific POSITION servomotor to a certain angle, restraining this value to each servomotor's specific limits. It can only be applied to global rotation servomotor, vertical servomotor and gripper servomotors.

- **void writeAdvServo(int angle)** -> This function is specifically designed to send both shoulder servomotors to their corresponding position at the same time and complementarily to ensure coordination.

- **boolean rotWrist(int angle)** -> This function allows to send the wrist to its final position when used with a *while* loop and returns *true* when such angle has been accomplished.

- **boolean displaceRobot(int target_distance)** -> This function can make the robot reach a certain displacement position when used in a *while* loop and returns true when the final displacement has been reached.

- **void openGripper(int percentage_right, int percentage_left)** -> This function asks for two *int* type arguments indicating which percent (0-100) it is wanted that the gripper opens, one percent on each side, being able to have a certain aperture and change its position by moving the overall displacement from one side to the other.

- **void TEST_EFFECTOR()** -> This function performs basic gripper automated movements to test that it works.

NOTE: Joint 1 motor is referred to in the program as *lateral_disp*, joint 2 motor is referred to as *rot*, joint 3 motors are referred to as *advance_right* and *advance_left*, joint 4 motor is referred to as *vertical*, joint 5 motor is referred to as *wrist* and end-effector motors are referred to as *grp_d_inf* and *grp_i_inf*. These are the names with which the servomotors shall be called when inserting them on functions that require a *TS1_Servo* type of argument.

The following functions enable more advanced movements for the robot:

- **void moveAbsJ(const double q0[GJOINTS], const double qT[GJOINTS], const double T)** -> This function allows to, by entering an initial (q0) and a final (qT) set of angles and displacements, one for each joint, make the robot move smoothly between those positions interpolating a series of points in order to make a movement cubic curve. It reaches the final position on time T elapsed.

- **void moveAbsJ0(const double qT[GJOINTS], const double T)** -> This function enables to enter a final position (qT) so that the robot moves directly to this final set of angles and displacements from its current position.

- **void goToPoint(float x, float y, float z, float alpha, float beta)** -> This function employs the Inverse Kinematics model of the robot in order to determine, from a given point in space and a given set of orientations in Y and Z for the end-effector, the set of angles and displacements needed to reach that position and orientations and, finally, moves the robot there.

NOTE: In order to input a set of angles and displacements, a 1D array of doubles is inputted with the following elements: {Joint 1 displacement, Joint 2 angle, Joint 3 main angle, Joint 4 angle, Joint 5 angle}.

NOTE: After performing a specific movement of one joint alone, the global variable for that joint must be updated with the new value. Joint 1 uses *Glob_displacement*, joint 2 uses *Glob_rot*, joint 3 uses *Glob_horizontal*, joint 4 uses *Glob_vertical* and joint 5 uses *Glob_wrist*.

Other parameters that can be varied in order to handle several other subsystems are:

- **ACTIVATE_MOTORS/DEACTIVATE_MOTORS:** This instruction activates or deactivates the relay that controls the 6V motors, giving the program the chance of activating or deactivating them when needed (reduce power consumption, security self-disconnect, etc.).

- **ACTIVATE_EFFECTOR/DEACTIVATE_EFFECTOR:** This instruction enables or disables the end-effector tool relay. This allows to program connection or disconnection of such subsystem in order to save energy when it is not needed.

- **Wrist_rotation:** This instruction retrieves wrist's angle of rotation at current time.

- **lights_on/lights_off:** This instruction allows to turn on and or off robot's front light.

- **NOTIFICATION:** This instruction activates a lighting sequence on the front light of the robot that can be used to notify something directly through the robot, with no external device need.

- **boolean Speak_Now:** This is a variable intended to activate or deactivate the transmission of spoken info by the robot which is normally sent to the app and passed through the Text-to-Voice Engine.

- **HOME:** This instruction returns the device to initial (home) position from wherever it is located.

Any port can be written by typing COM.println() (Computer), HC12.println() (Radio module), BT.println() (Bluetooth HC-05 module) or USER.println() (External VGA comms).

A4.4. DEVELOPING NEW FUNCTIONALITIES

The device can be modified to fit specific necessities using its hardware and electronics capabilities as it is pleased always inside the safety boundaries established by the safety guidelines on section A4.2. The standard use of the robot involves either handling with a computer, with a smartphone app, with an external radio IMU device or with an external wired device of some kind (usually a µC).

Regarding communication ports

Possible ampliations that can be made using communications in a different war are:

1. Using Arduino's serial interface with a **computer**: The development of a new program that handles the robot by itself opens this platform to many new possibilities, including computer vision, automated operation based on an IoT instruction, etc.

2. Using **Bluetooth/BLE** interface: The development of app-based software that can communicate wirelessly with smart devices via Bluetooth or BLE like smartphones, smartwatches, smartbands, home automation devices, etc.

3. Using **radio HC-12** module interface: The development of alternative radio control devices to handle the robot or the possibility of connecting several Arduino systems between them wirelessly, etc.

4. Using the VGA **serial port** interface: The development of external systems with sensors that allow to sense the environment and indicate the robot where to go or where not to go, with the possibility of integrating device-based computer vision neural networks (with Raspberry for instance) giving instructions to the arm and even the coordination of several of these arms.

Regarding mechanisms

Some of the ampliations that can be done involve:

1. Creation of new end-effector for different tasks, being able to experiment with modern types of actuators like soft actuators or even origami actuators.

2. Creation of new sensing attachments for sensors to be used on the robot and processed by an external device.

2. PLANS & DRAWINGS

2.1. Introduction

This section aims at presenting the several design diagrams in detail mentioned on section 5. ("Detailed design"), with specific dimensioning that shall allow a better understanding of this device's working.

Plans to be added in the following pages shall comprise designed pieces. Other standard or bought pieces are both referenced on their corresponding subsection on section 5. and shall not be presented on this section (for example wheels, screws, bolts, nuts, washers, bearings…).

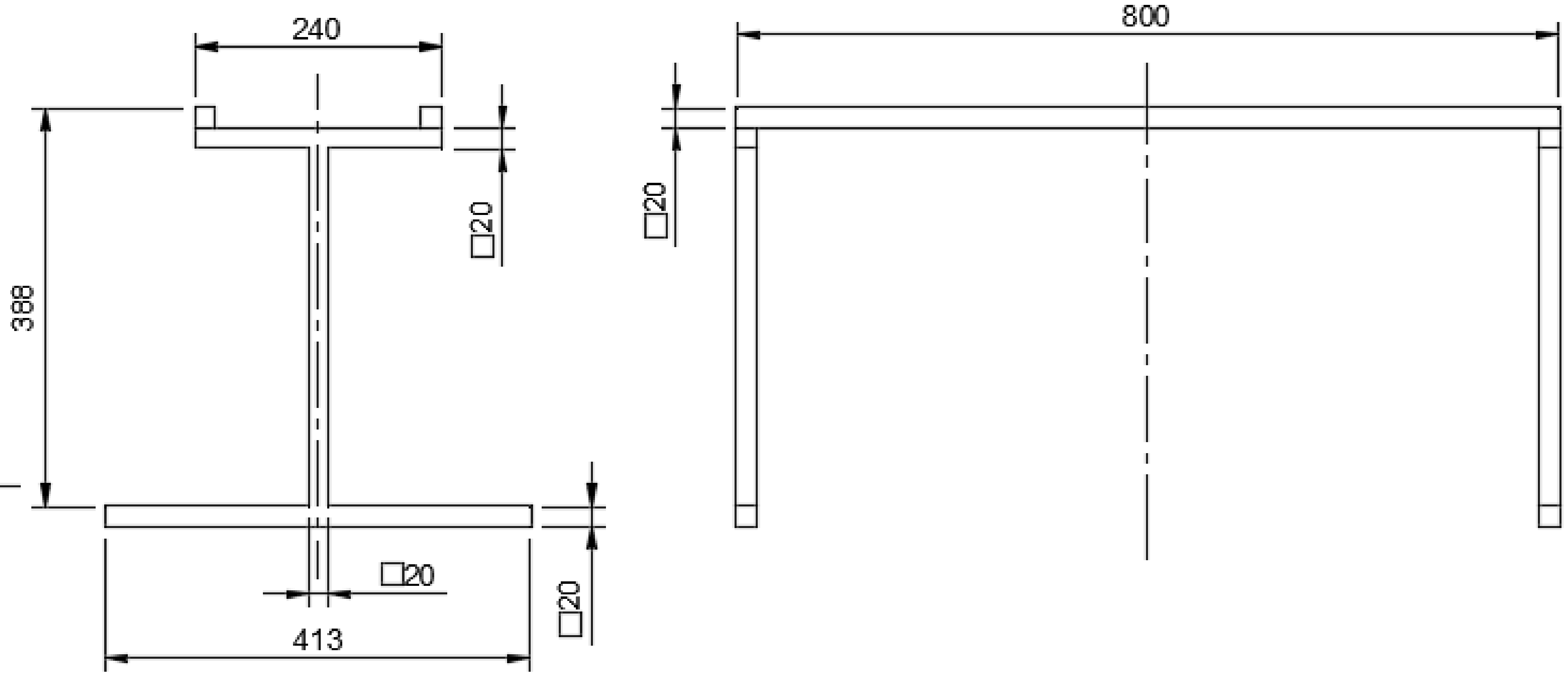
Assembly plans are not presented on this section, since all necessary explanations on component fitting have been made on their corresponding subsection on section 5. with some assembly figures already presented.

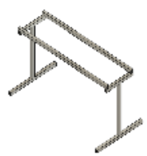
Some of the views have no hidden lines shown in order to increase clarity, since the use of hidden views could mean a difficult understanding of the model. In those cases, if necessary, section cuts have been made to show hidden features.

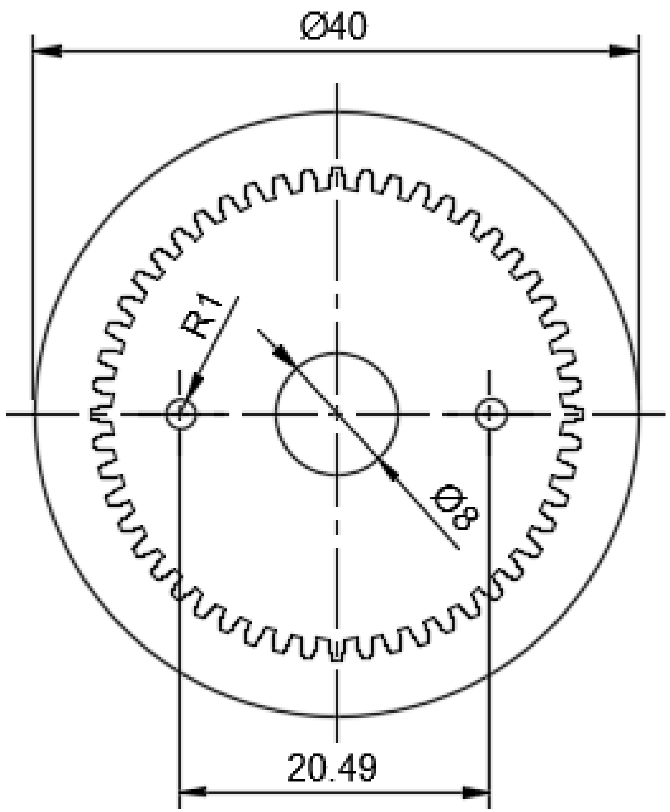
The scale of each plan stands for the scale at which the different perspectives have been represented, not the value of the dimensions.

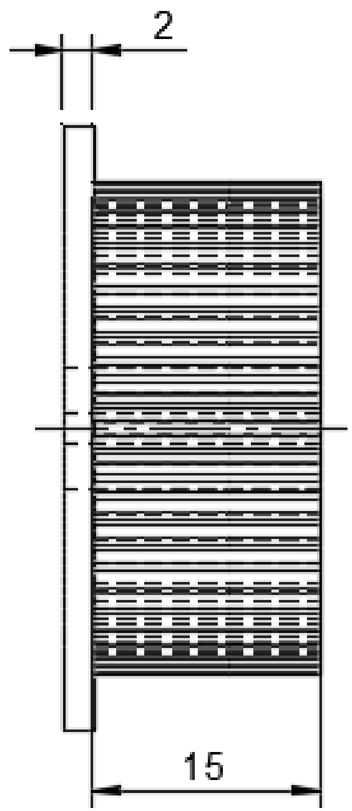
All plans have been made on A4 format except for *plan 15*, *plan 18*, *plan 19*, *plan 22*, *plan 29*, *plan 33*, *plan 34*, *plan 37*, *plan 39*, *plan 40*, *plan 41*, *plan 44*, *plan 45*, *plan 46* and *plan 48*, which, due to limitations of the dimension text in the software used (*Autodesk Fusion 360*), could not be done smaller and had to be developed in A3 format to fit all the necessary information.

On *plan 49*, modules HC-05, HC-12 and servomotors have been added as means of labels, since they are not among the libraries of the program. They are represented by their pins.



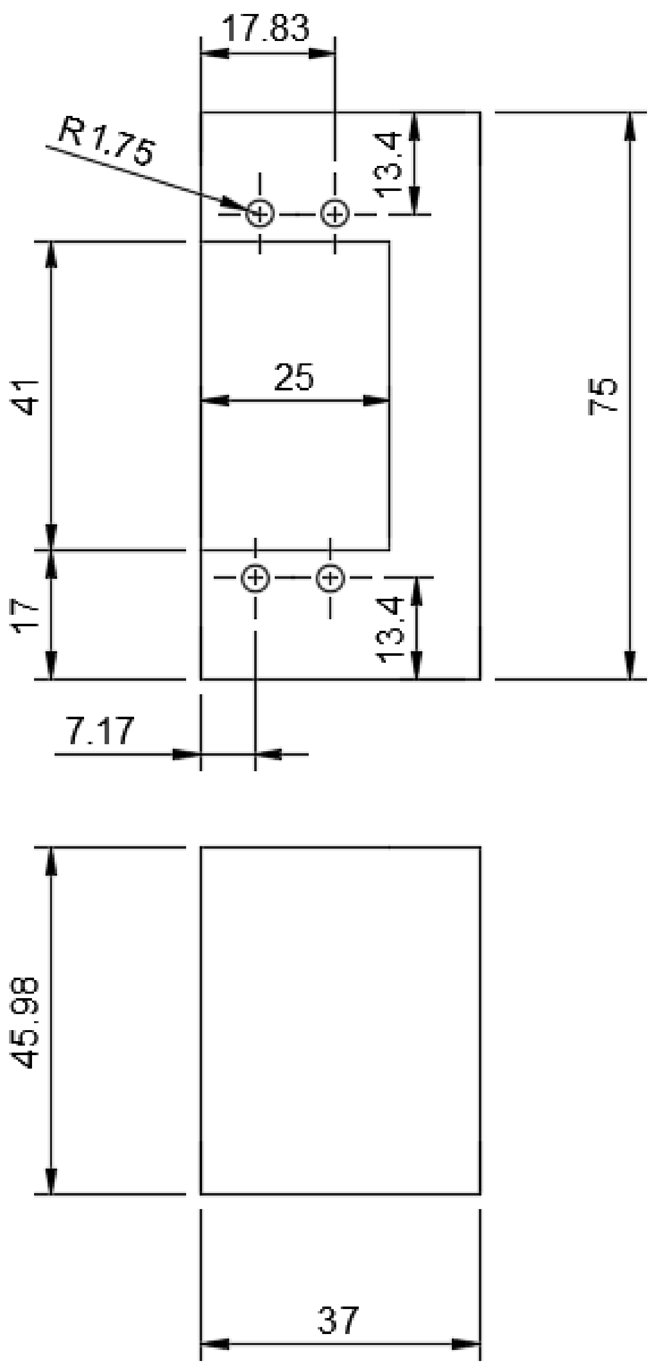
Date: 10/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:6
	Plan name: Aluminum rail structure		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 1	POLYTECHNIC UNI-VERSITY OF VALENCIA	

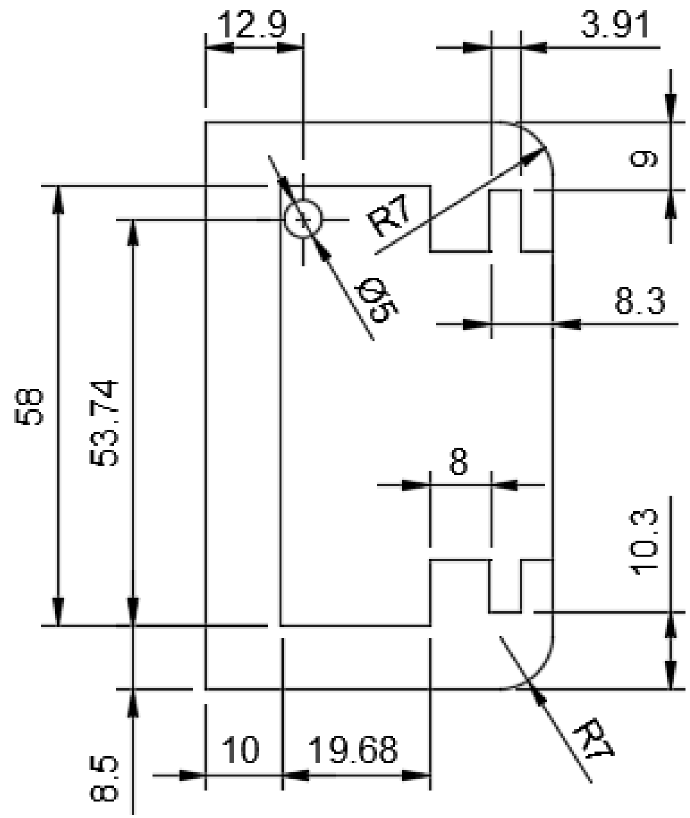


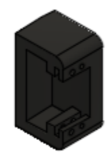


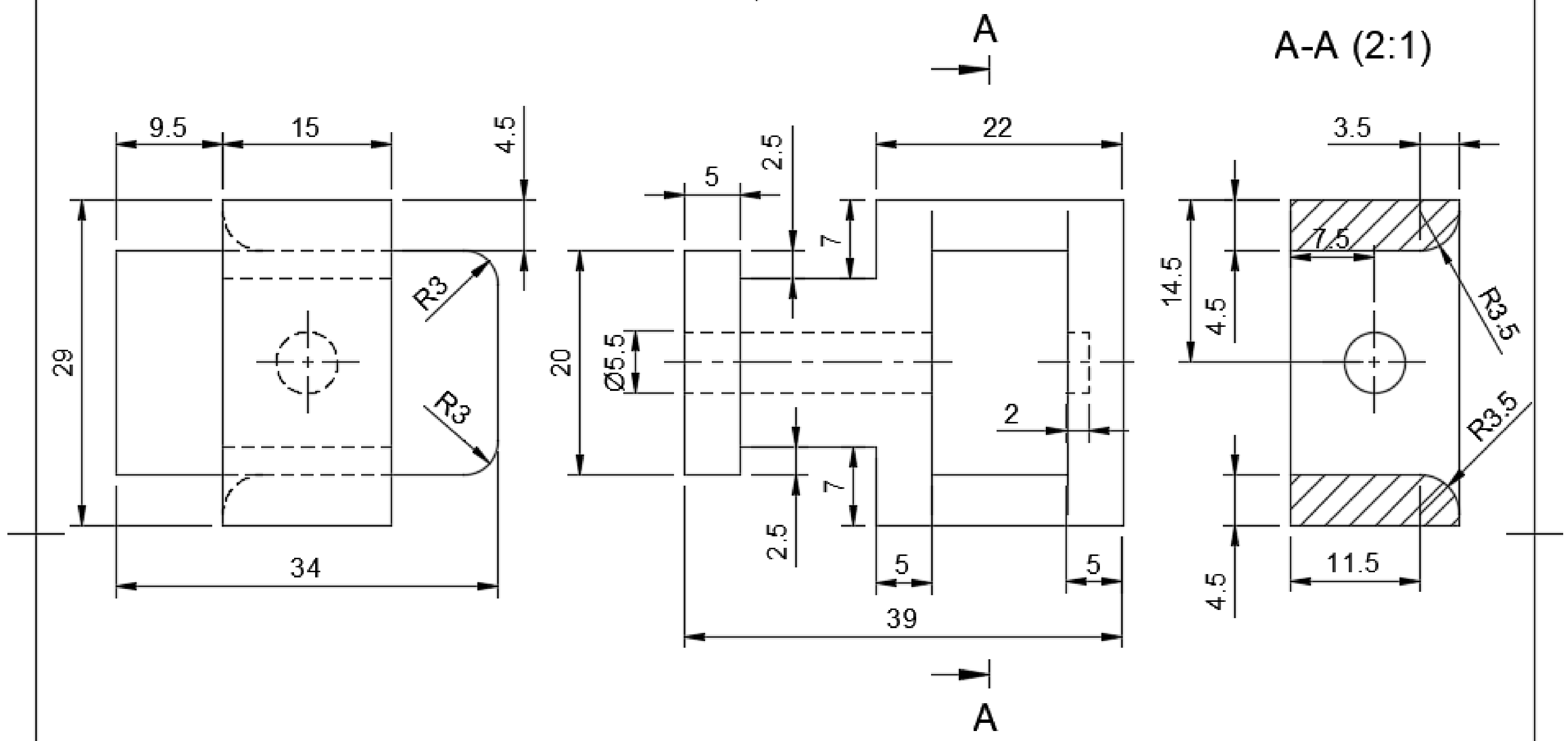
Gear description	
Pitch	2 mm (from belt)
Number of teeth	52
Pressure angle	14.5
Teeth height	1 mm

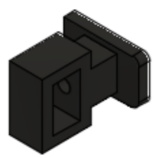
Date: 10/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Belt driving gear for joint 1		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 2	POLYTECHNIC UNIVERSITY OF VALENCIA	

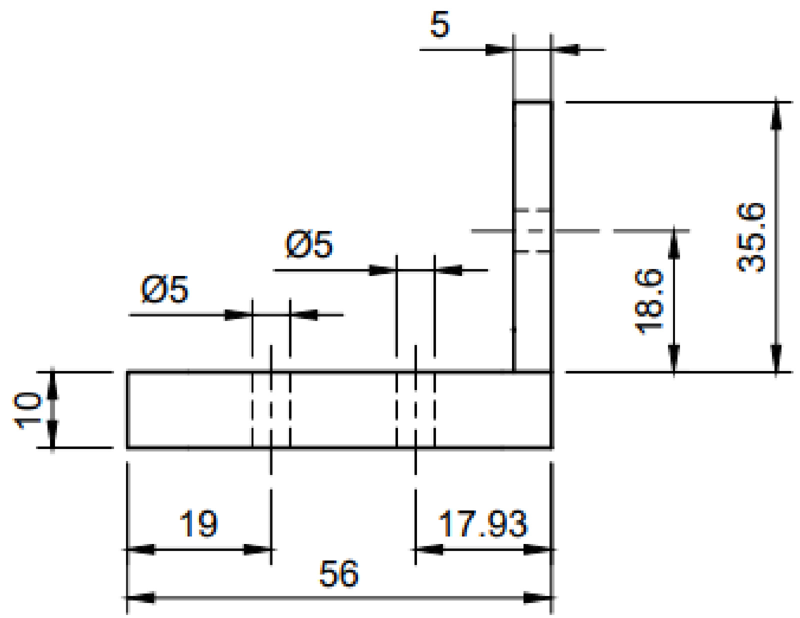


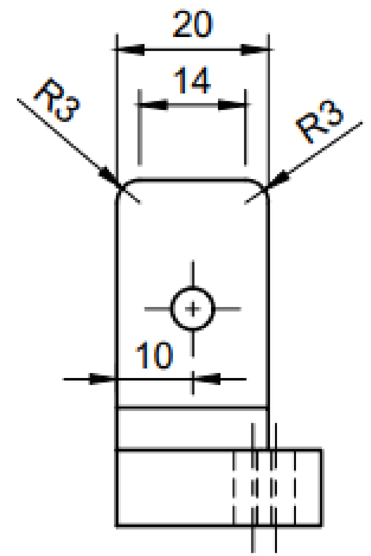


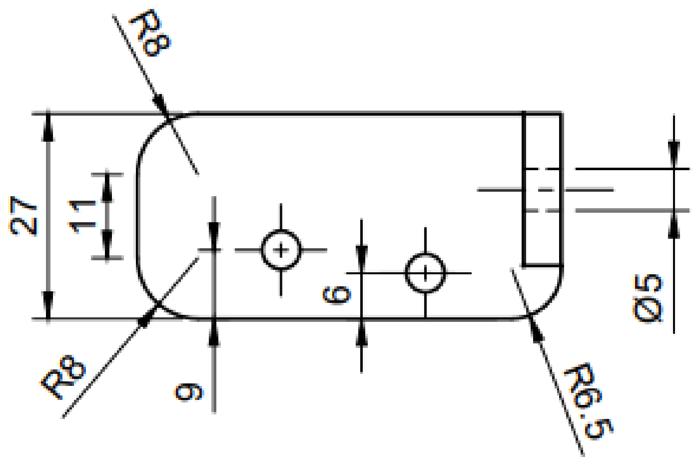
Date: 10/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 1:1
	Plan name: Motor bracing for joint 1	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 3	POLYTECHNIC UNIVERSITY OF VALENCIA

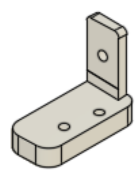


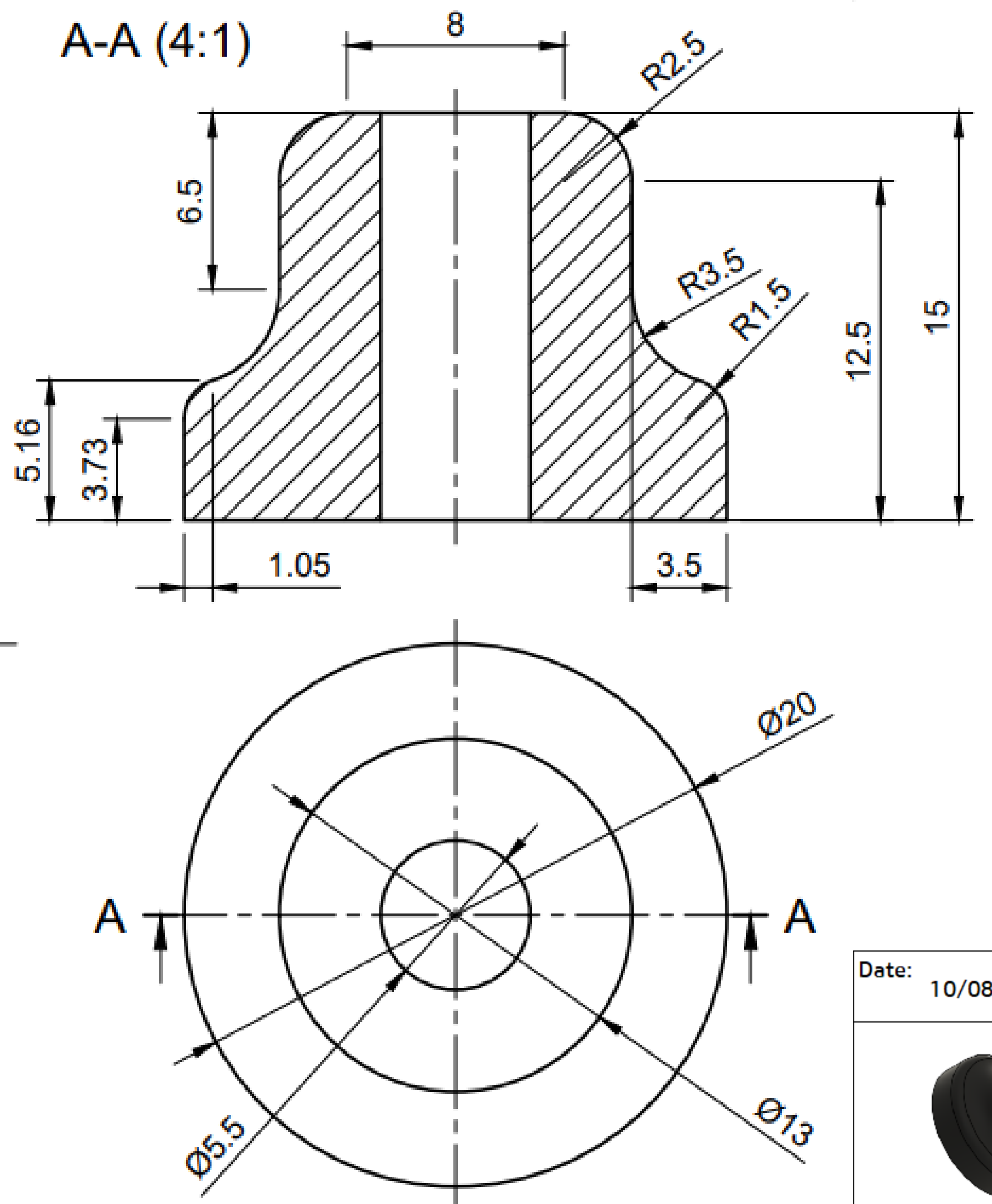
Date: 10/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Belt pulley casing for joint 1		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 4	POLYTECHNIC UNIVERSITY OF VALENCIA	



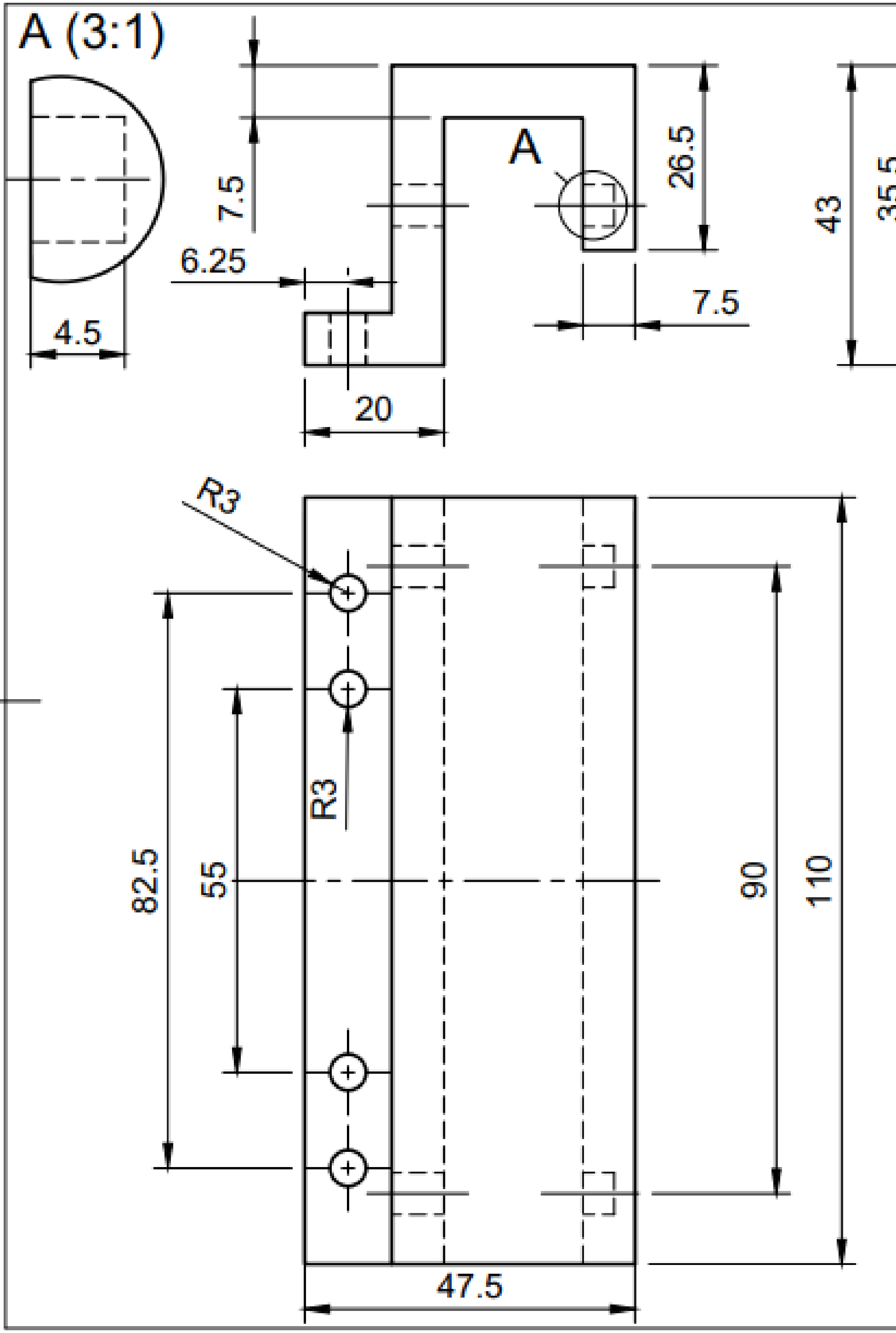


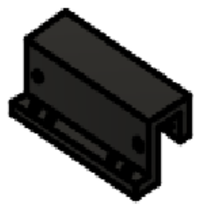


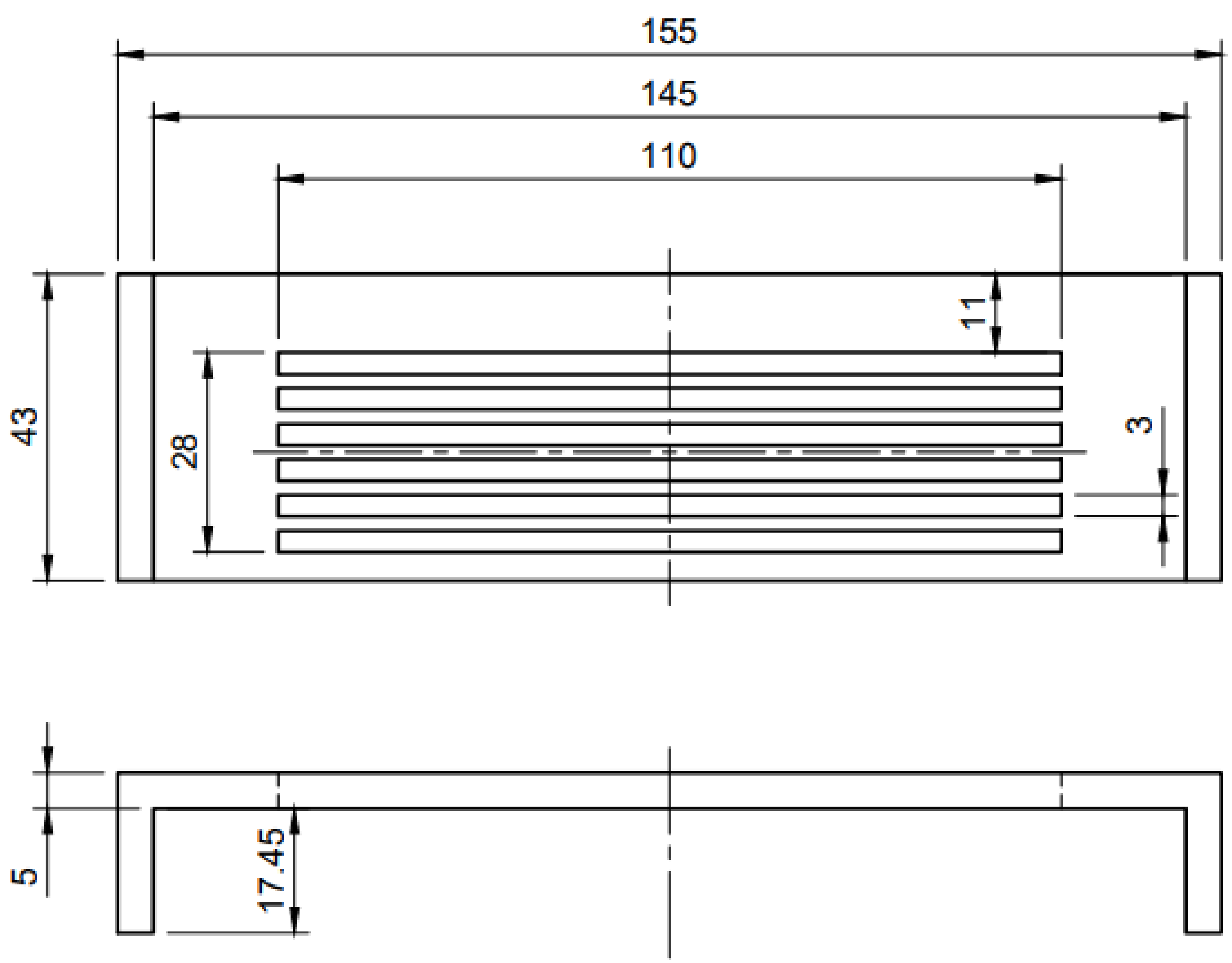
Date: 10/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Belt pulley attachment for joint 1		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 5	POLYTECHNIC UNIVERSITY OF VALENCIA	

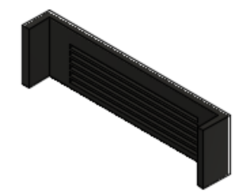


Date: 10/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 4:1
	Plan name: Shaft-stopping piece for joint 1		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 6	POLYTECHNIC UNIVERSITY OF VALENCIA	

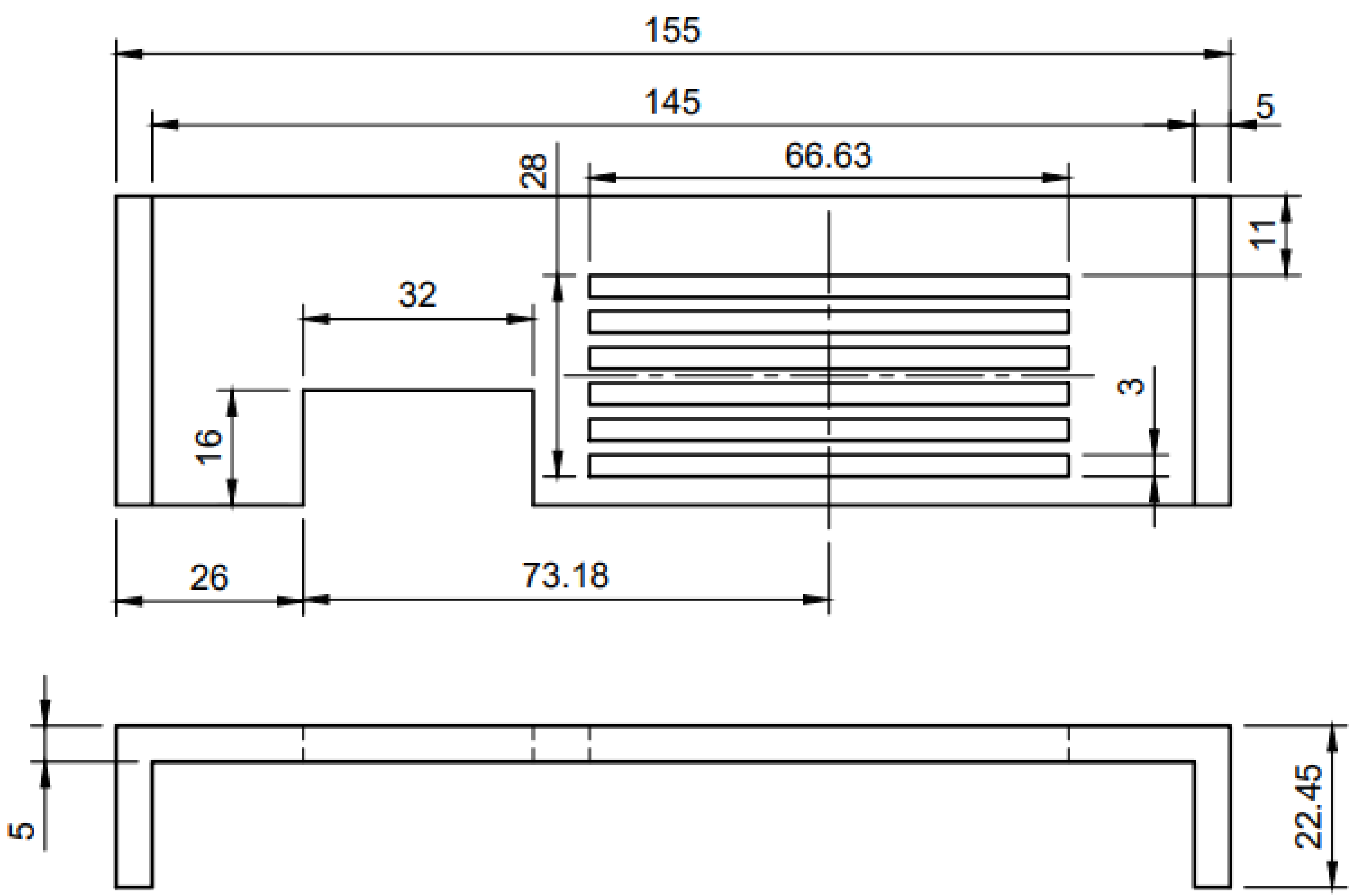


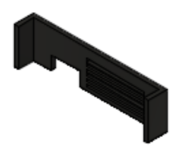
Date: 05/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Shaft fixing and support piece for the mobile base		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 7	POLYTECHNIC UNIVERSITY OF VALENCIA	

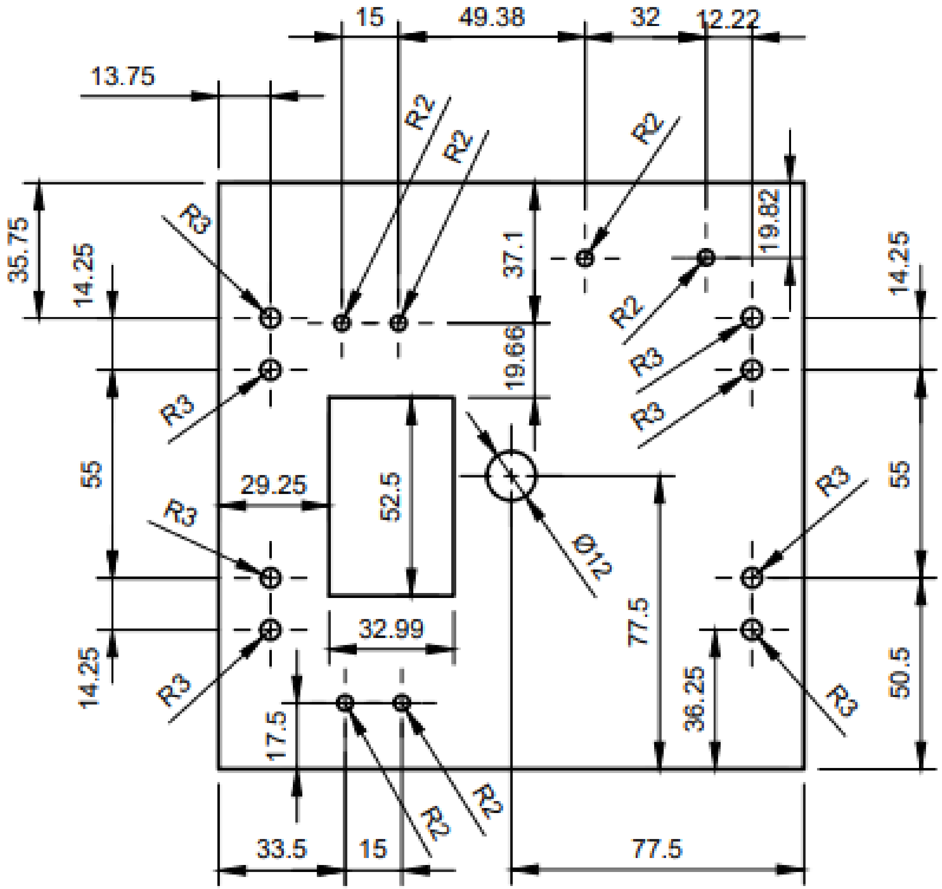


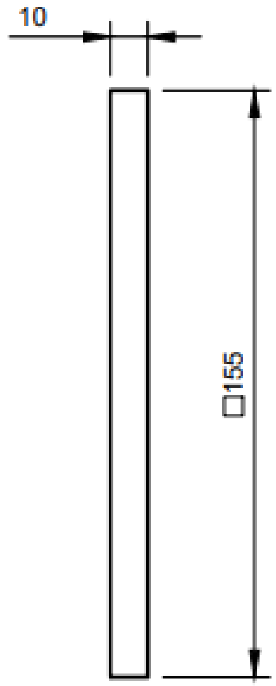


Date: 04/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Left lateral panel for mobile base		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 8	POLYTECHNIC UNIVERSITY OF VALENCIA	

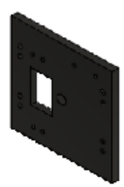


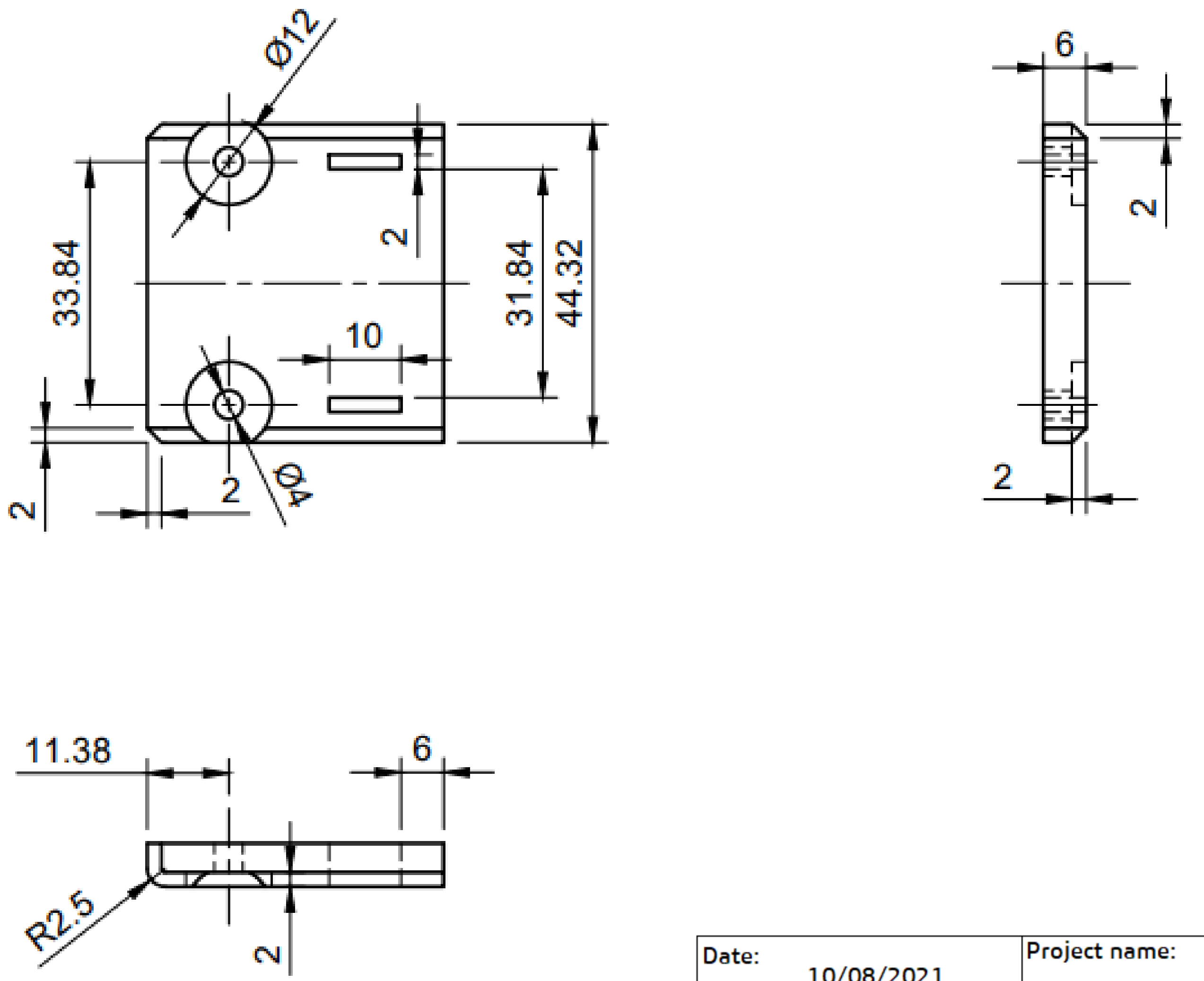
Date: 04/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Right lateral panel for mobile base		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 9	POLYTECHNIC UNIVERSITY OF VALENCIA	



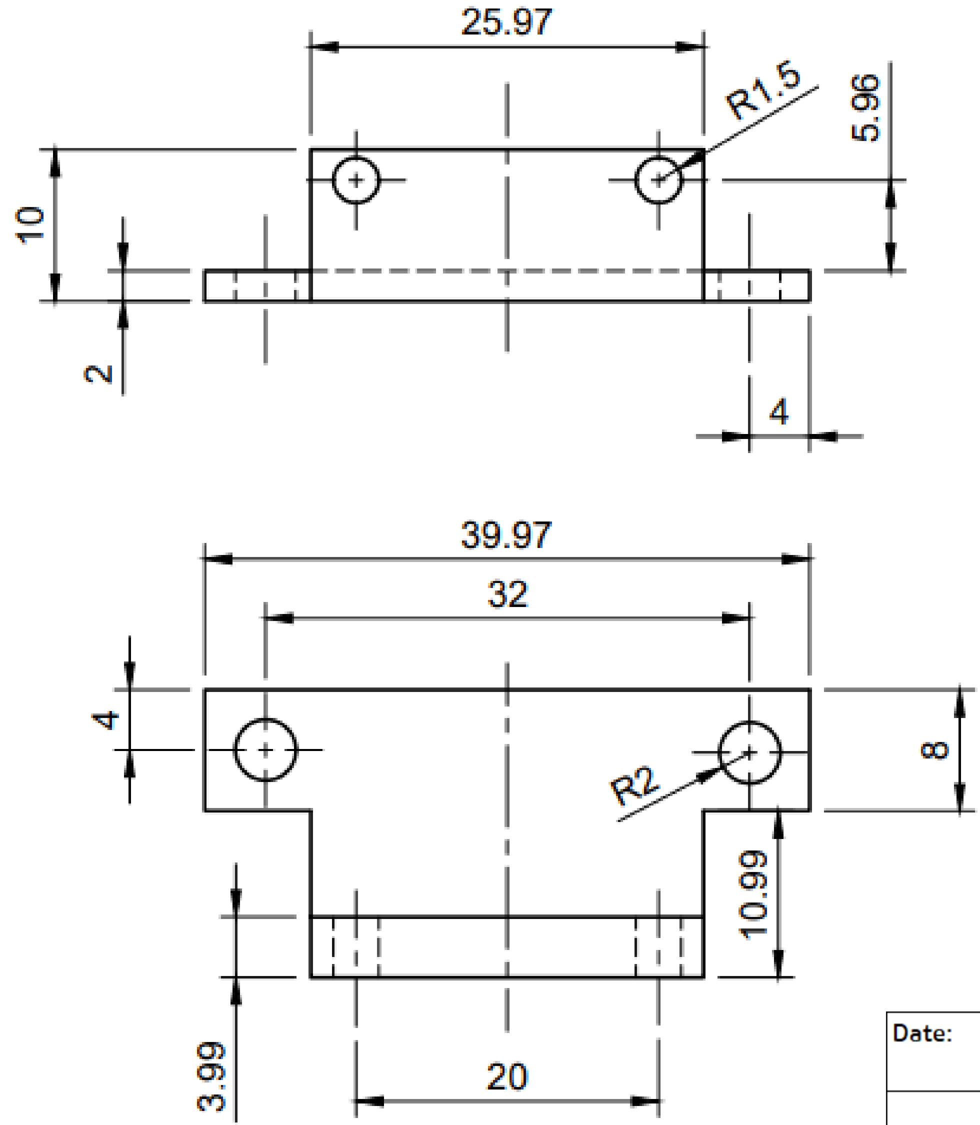


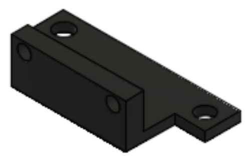
All holes and cavities are passing.

Date: 05/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 1:2
	Plan name: Central base panel for mobile base	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 10	POLYTECHNIC UNIVERSITY OF VALENCIA

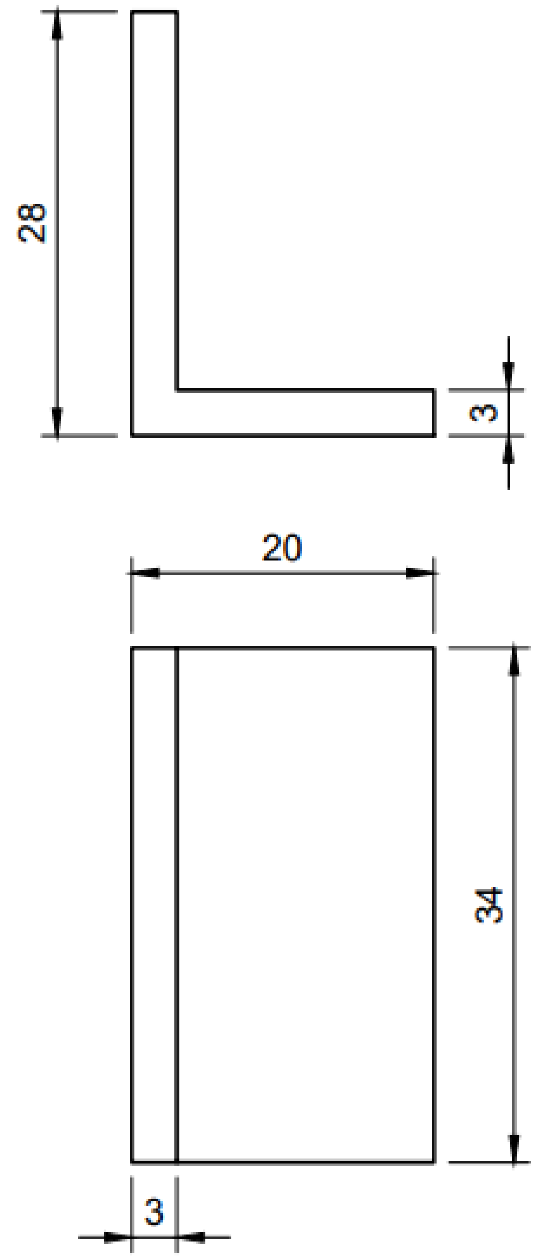


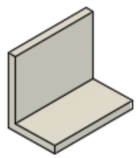
Date: 10/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Belt attachment piece for joint 1		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 11	POLYTECHNIC UNIVERSITY OF VALENCIA	

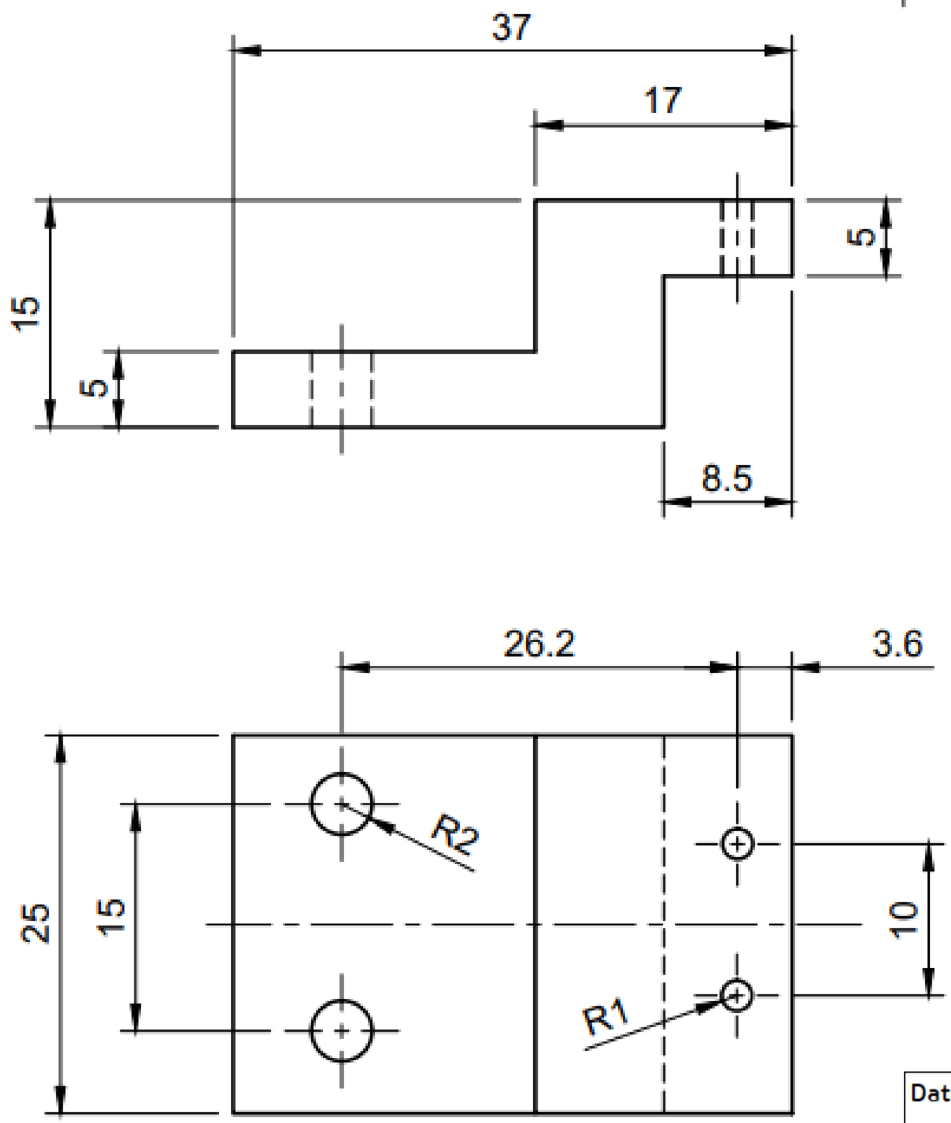


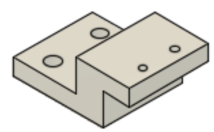


Date: 05/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Laser mounting piece for mobile base		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 12	POLYTECHNIC UNIVERSITY OF VALENCIA	

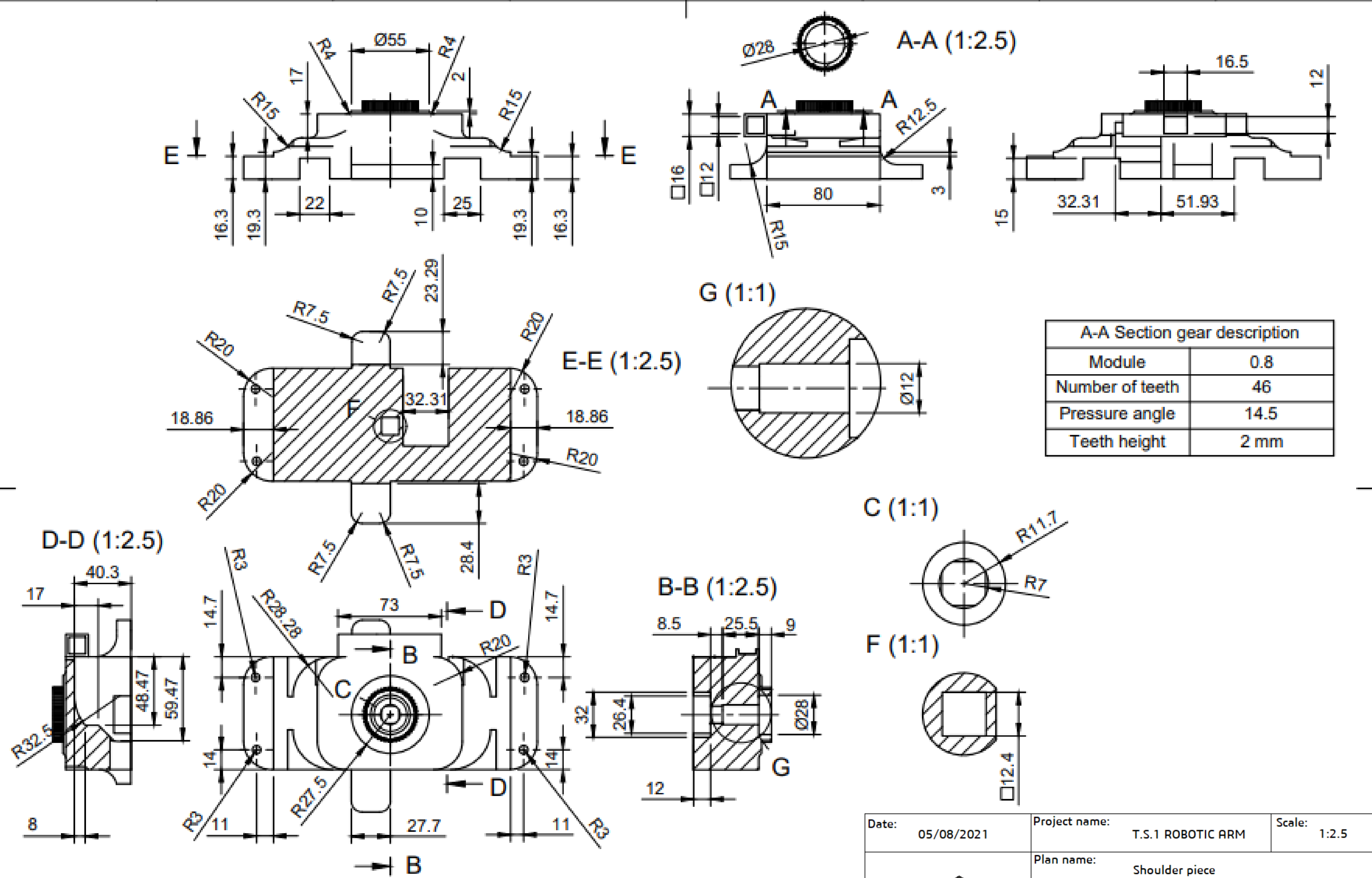


Date: 05/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Laser detection surface		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 13	POLYTECHNIC UNIVERSITY OF VALENCIA	



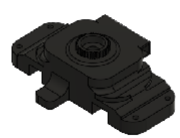


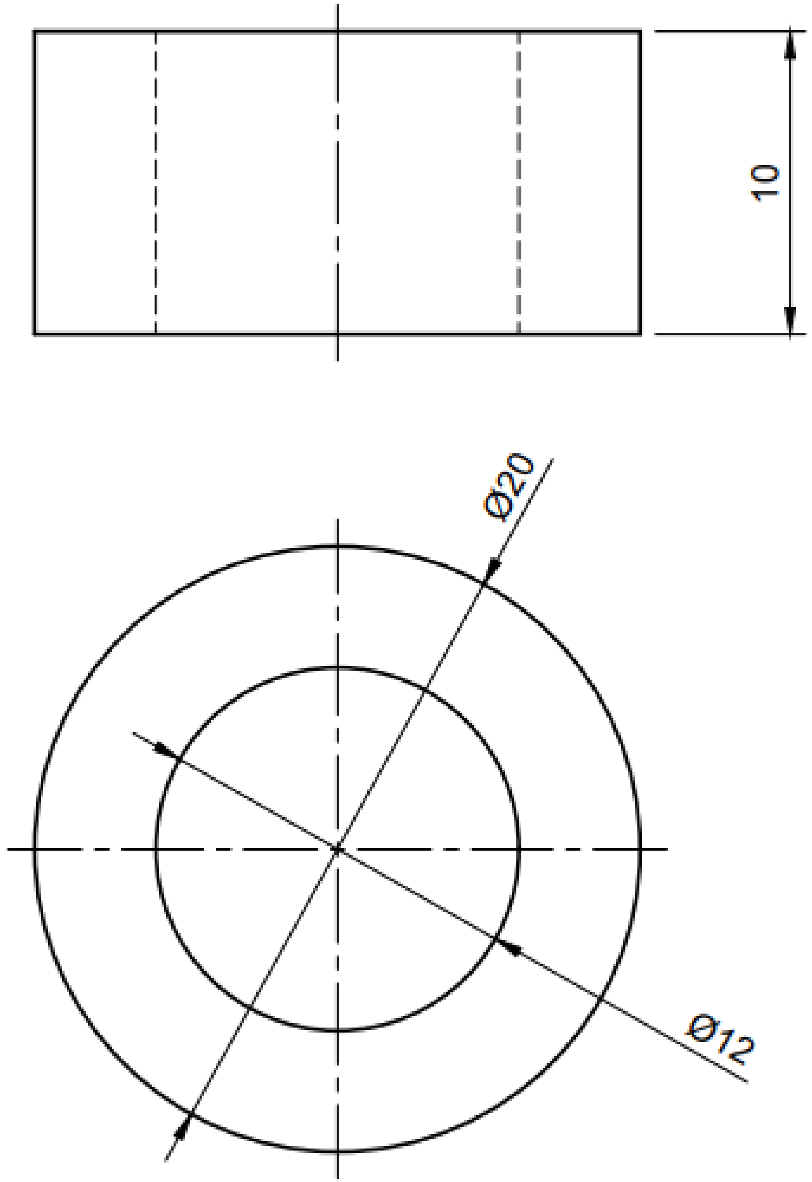
Date: 04/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Rotation motor holding		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 14	POLYTECHNIC UNIVERSITY OF VALENCIA	

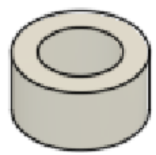


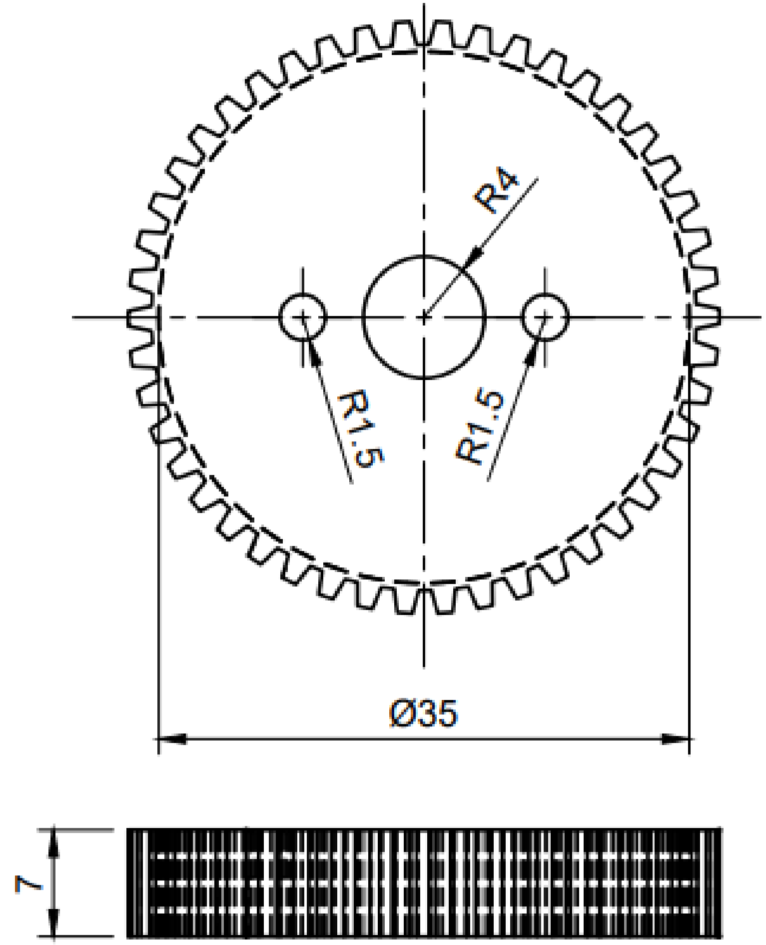
A-A Section gear description	
Module	0.8
Number of teeth	46
Pressure angle	14.5
Teeth height	2 mm

Hidden edges have been omitted on these views to ease comprehension. Sections have been used instead

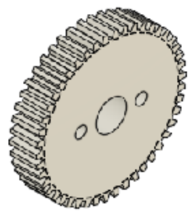
Date: 05/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 1:2.5
	Plan name: Shoulder piece	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 15	POLYTECHNIC UNIVERSITY OF VALENCIA

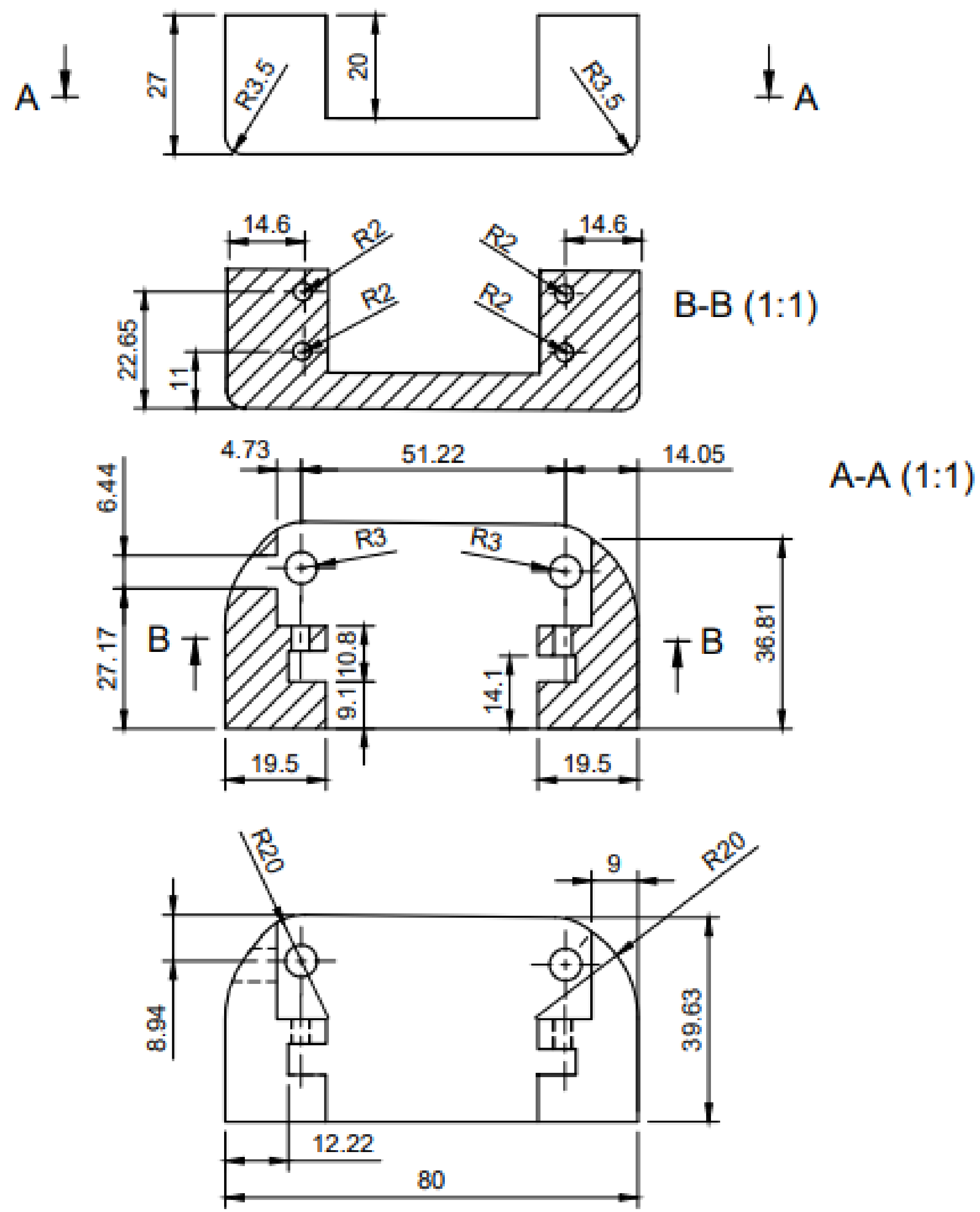


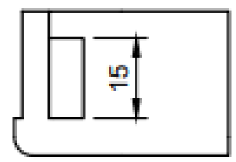
Date: 04/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 4:1
	Plan name: Cylindrical plastic separator		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 16	POLYTECHNIC UNI-VERSITY OF VALENCIA	



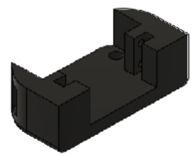
Gear description	
Module	0.8 mm
Number of teeth	46
Pressure angle	14.5 degrees
Teeth height	2 mm

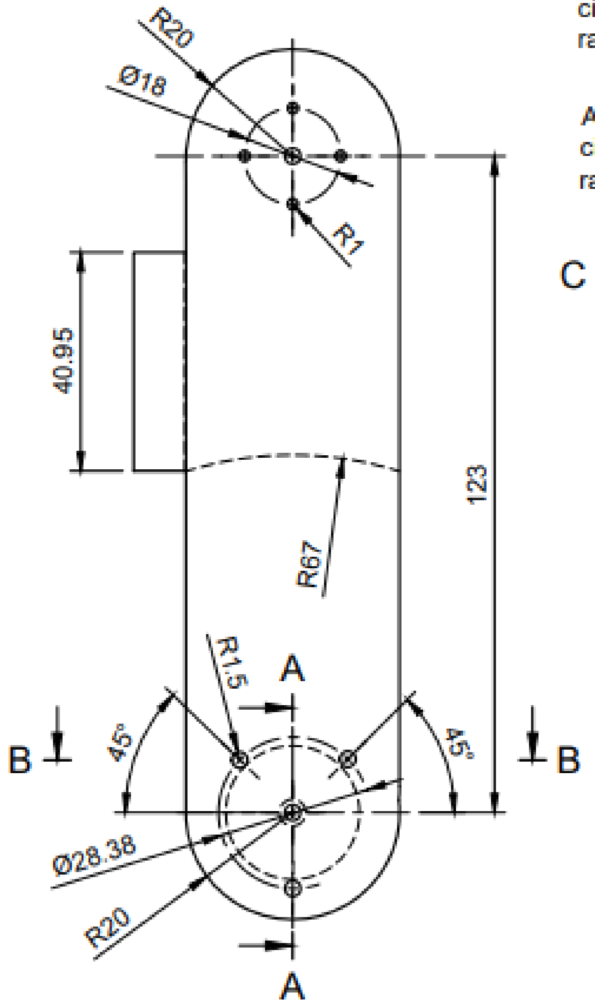
Date: 05/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Rotation motor spur gear		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 17	POLYTECHNIC UNIVERSITY OF VALENCIA	





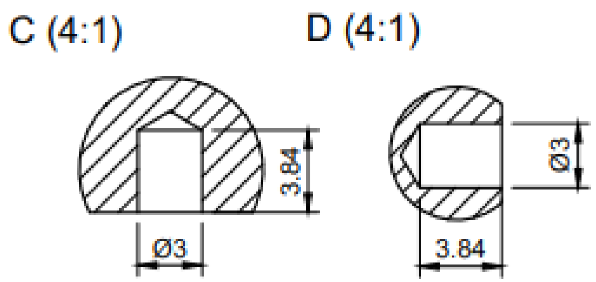
This drawing here presented corresponds to the left motor mounting piece on the shoulder. The piece used for the right motor has exactly the same measures but is a specular reflection of this piece respect to its front face

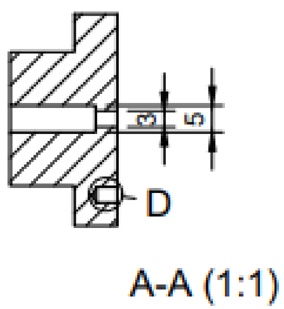
Date: 06/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 1:1
	Plan name: Shoulder motors mounting piece	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 18	POLYTECHNIC UNIVERSITY OF VALENCIA

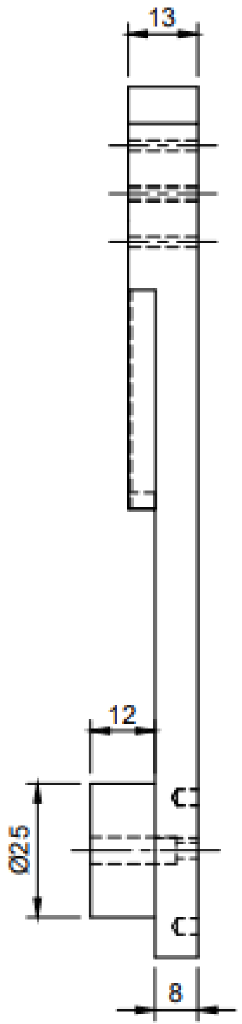


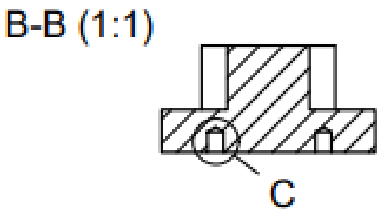
All four holes on the 18 mm diameter circumference locus have the same radius (R1)

All three holes on the 28.38 diameter circumference locus have the same radius (R1.5)

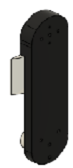


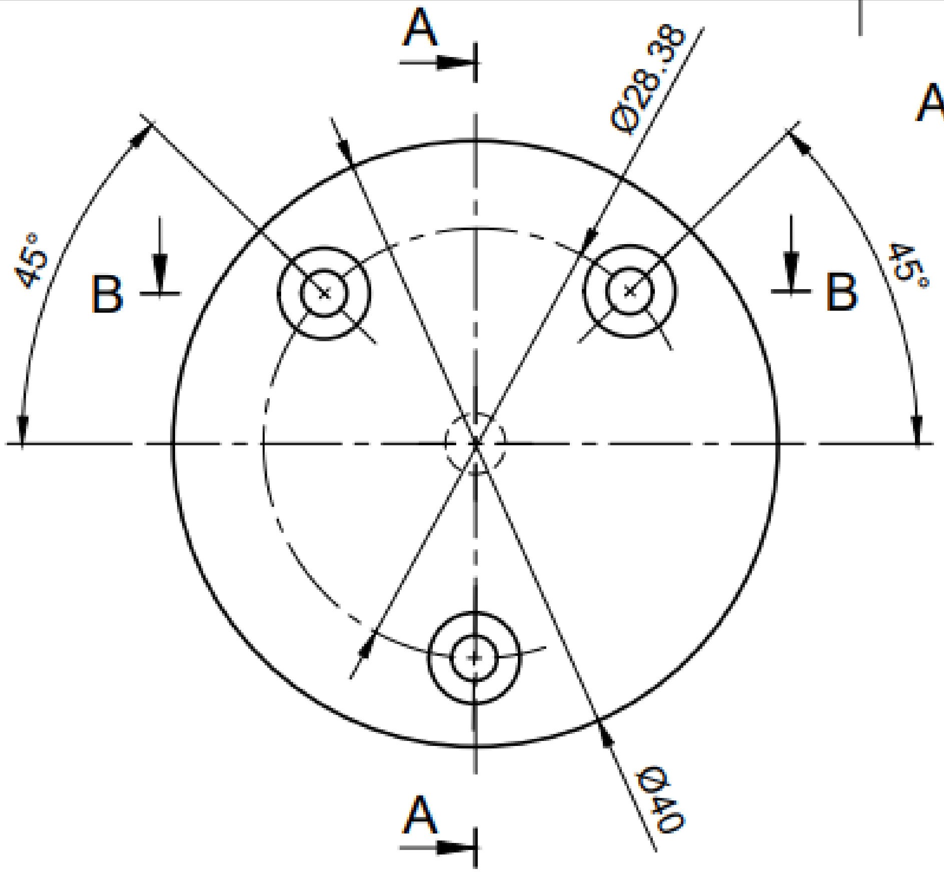




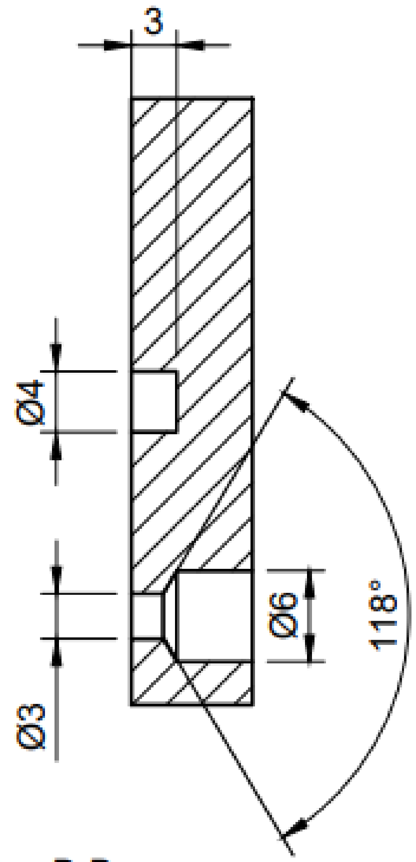


Both holes seen on section B-B have the same characteristics

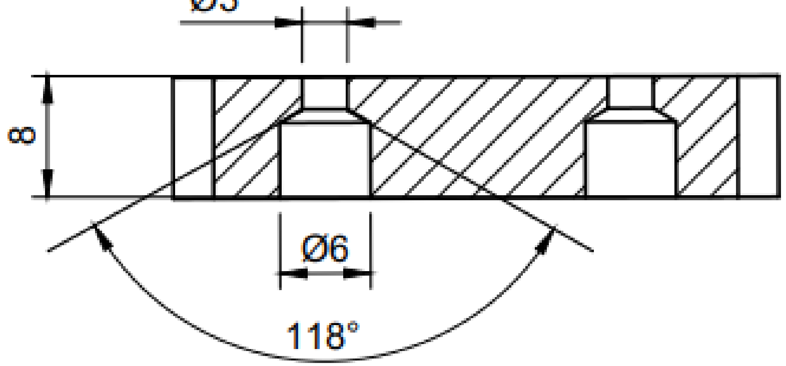
Date: 06/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Shoulder to elbow link		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 19	POLYTECHNIC UNIVERSITY OF VALENCIA	



A-A (2:1)

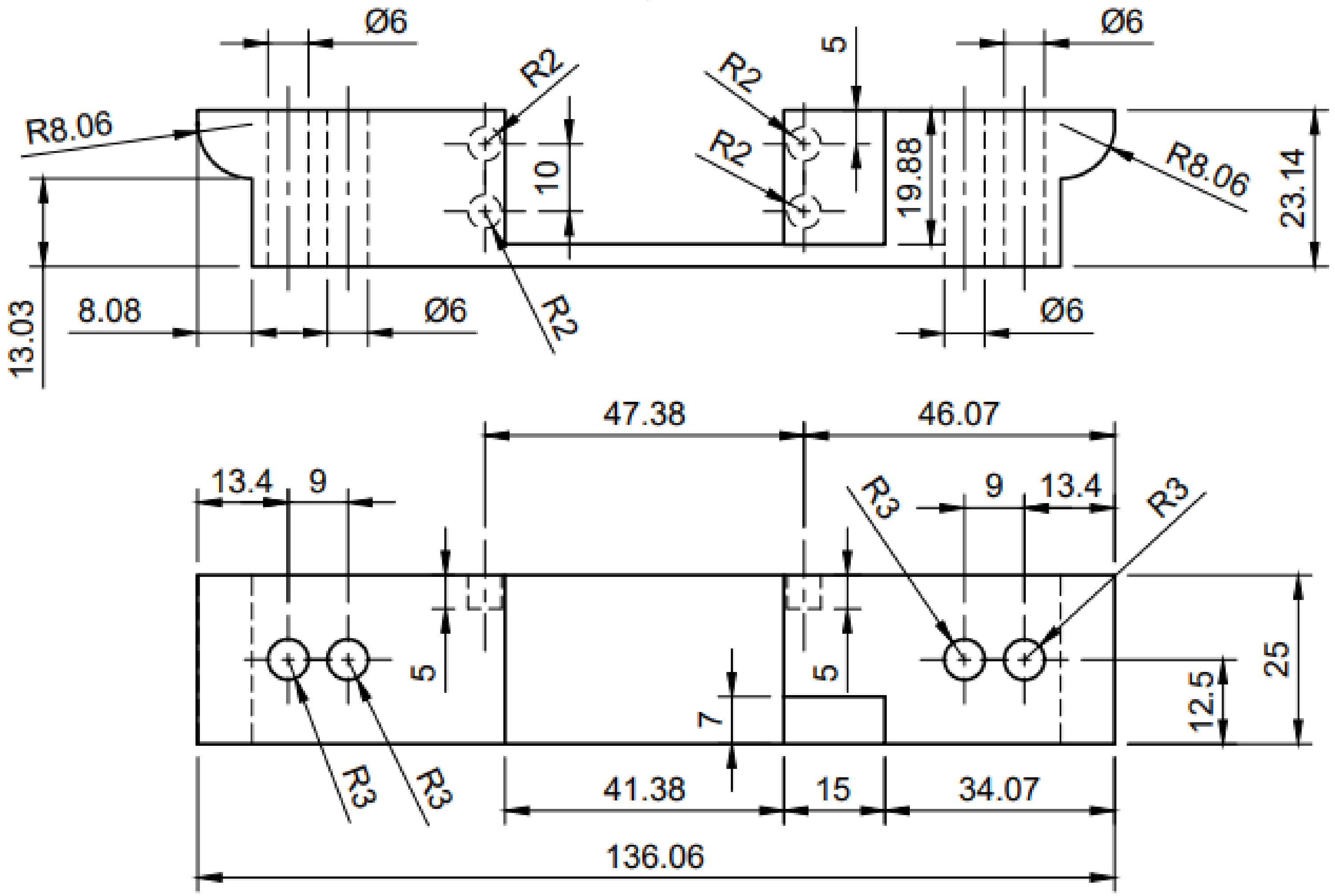


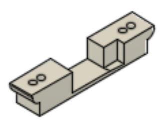
B-B (2:1)

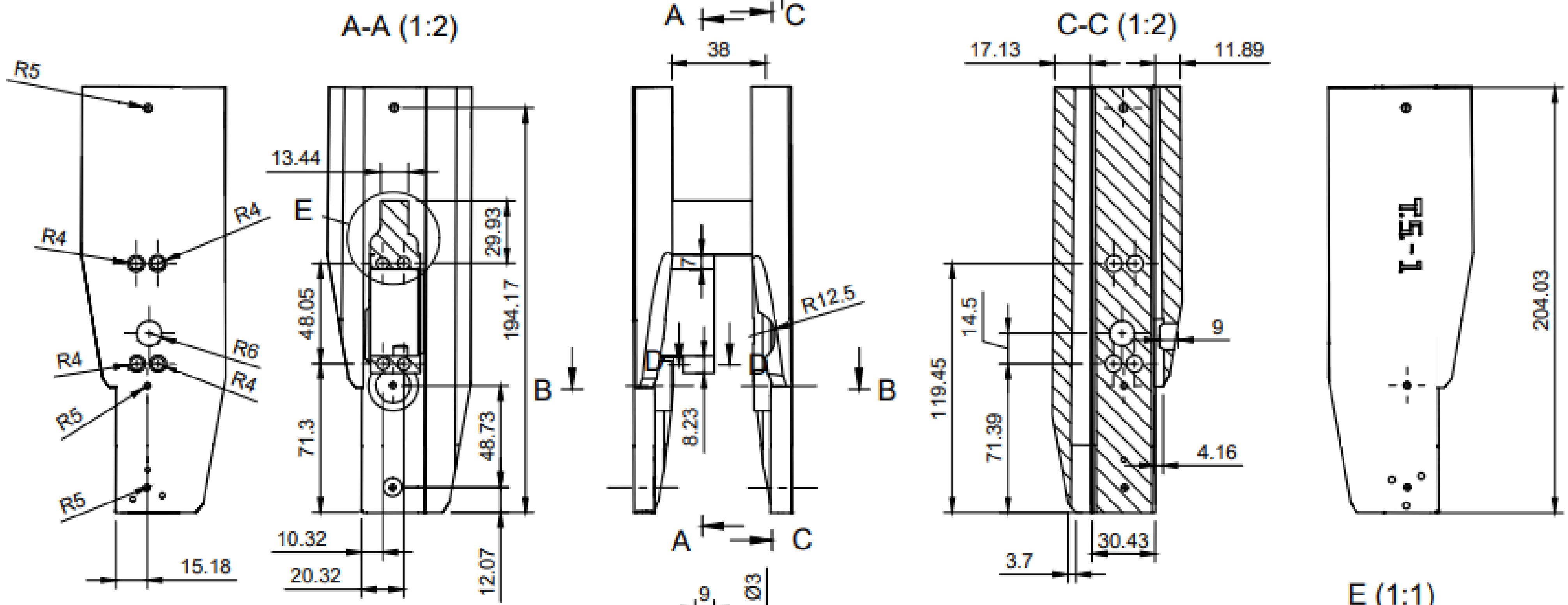


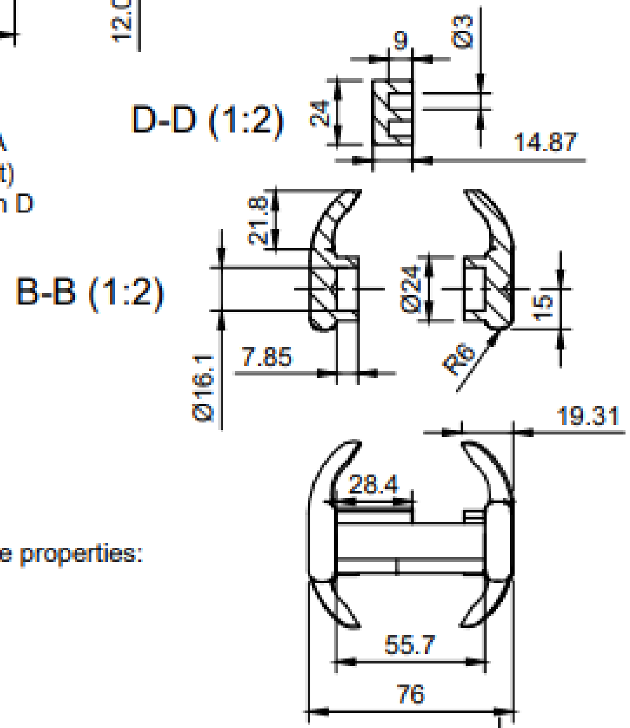
Both holes present on section B-B
have the same features

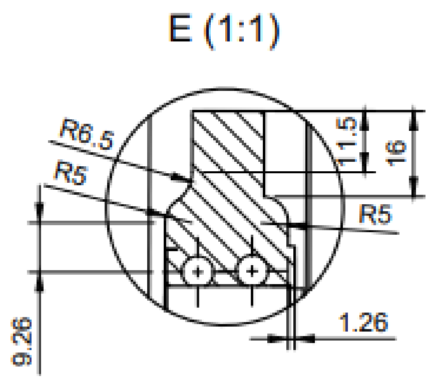
Date: 06/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Elbow shaft tap		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 20	POLYTECHNIC UNIVERSITY OF VALENCIA	



Date: 06/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 1:1
	Plan name: Elbow motor mounting	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 21	POLYTECHNIC UNIVERSITY OF VALENCIA





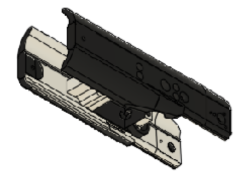


The four motor attaching holes on section A (being in groups of two and 48.05 mm apart) are exactly equal to those shown on section D

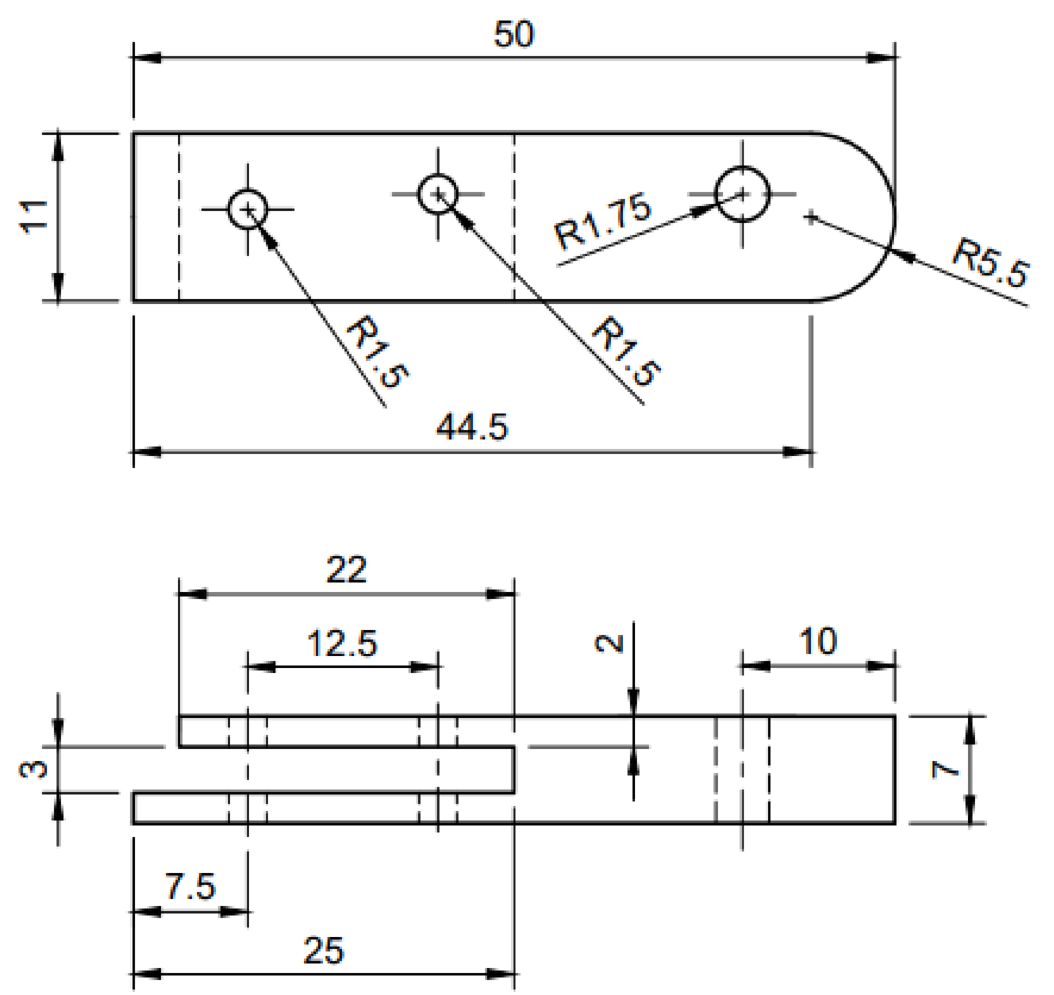
Both holes on section D have the same properties

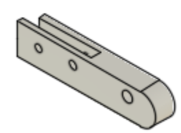
Both sides on section B are the same and opposite

Four fins on this view have exacty the same properties: an outer radius of 42.5 mm and 36.5 mm

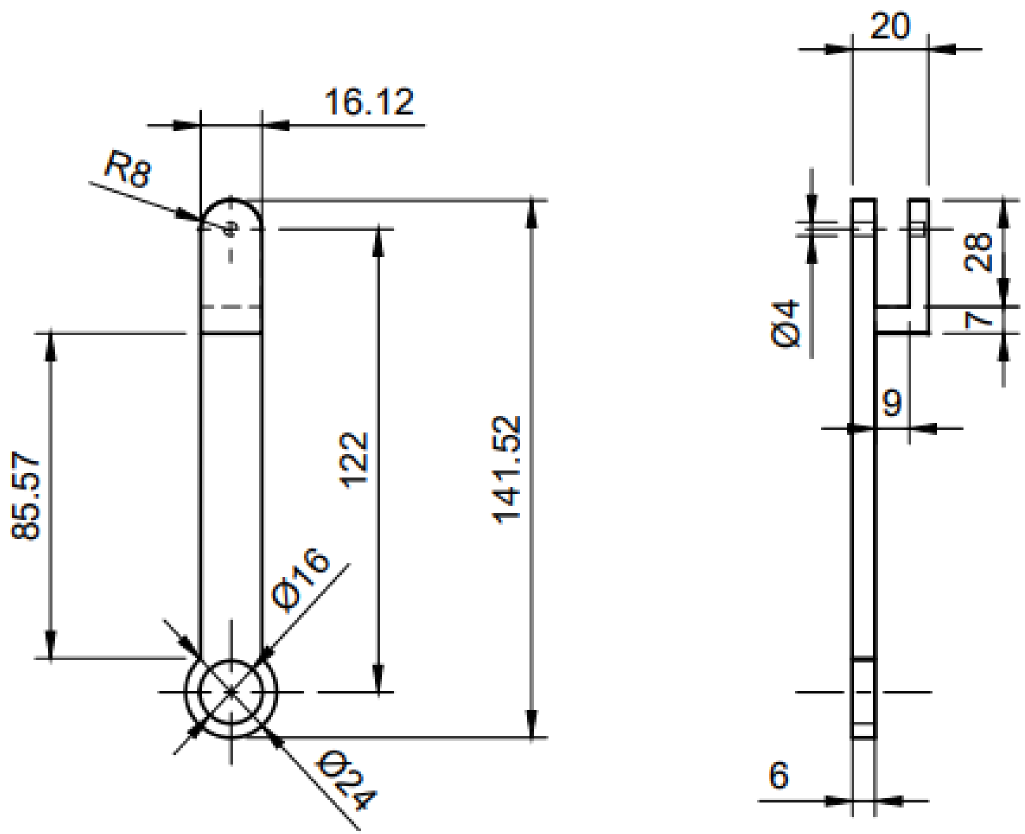


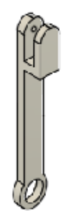
Date: 08/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:2
	Plan name: Forearm piece		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 22	POLYTECHNIC UNIVERSITY OF VALENCIA	

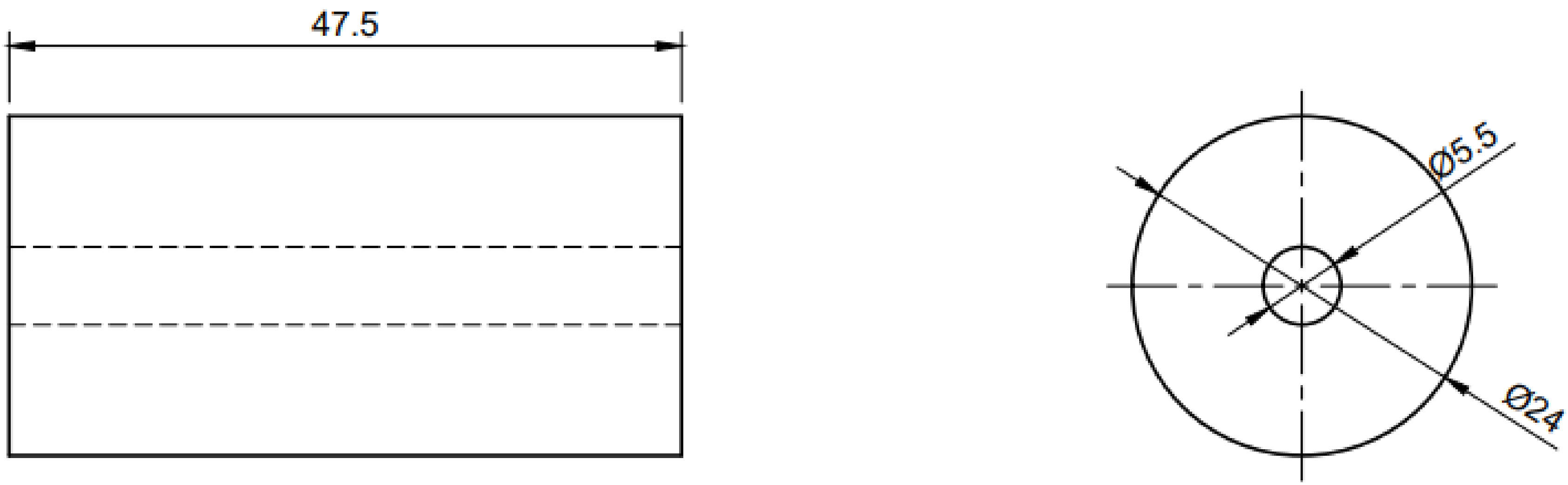


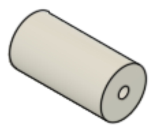


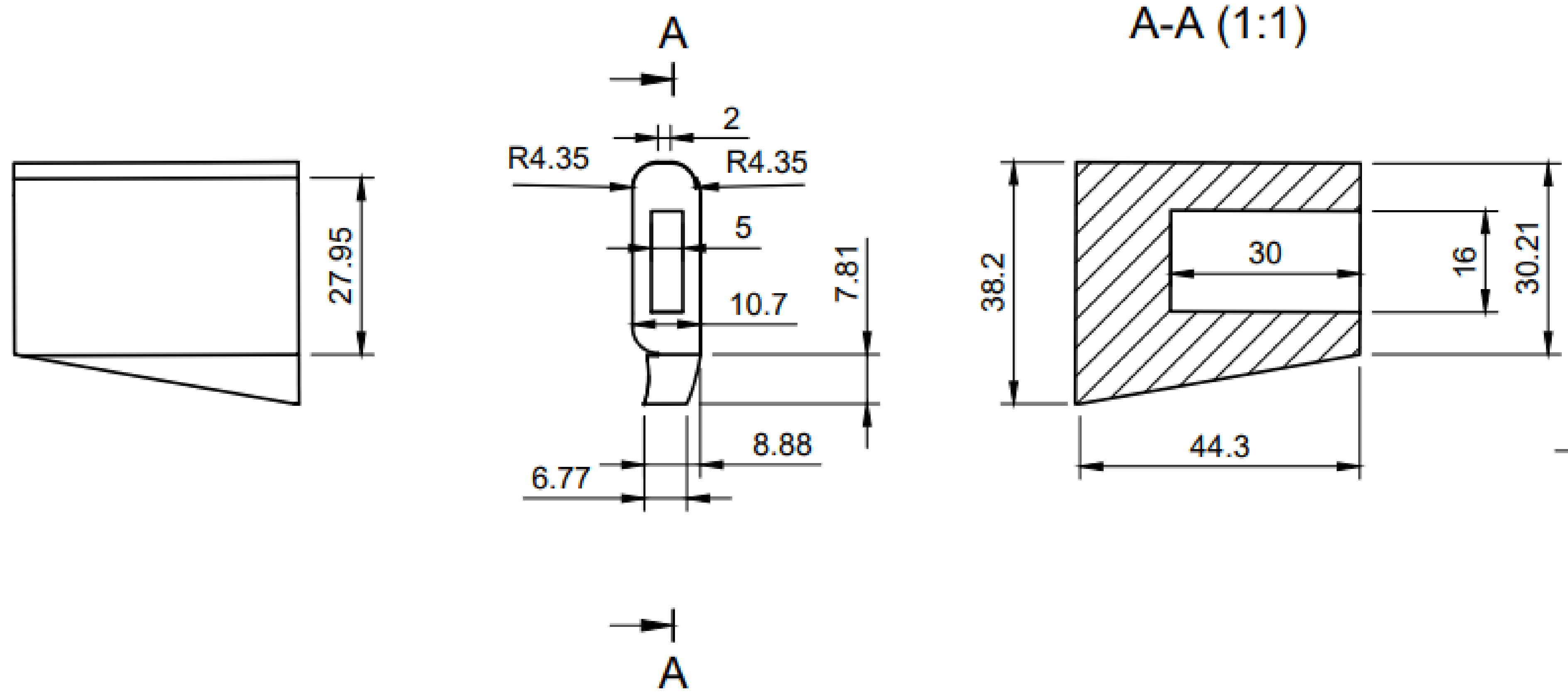
Date: 06/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 2:1
	Plan name: Elbow motor link 1 of 4-link mechanism	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 23	POLYTECHNIC UNIVERSITY OF VALENCIA



Date: 06/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:2
	Plan name: Elbow link 2 of 4-link mechanism		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 24	POLYTECHNIC UNIVERSITY OF VALENCIA	

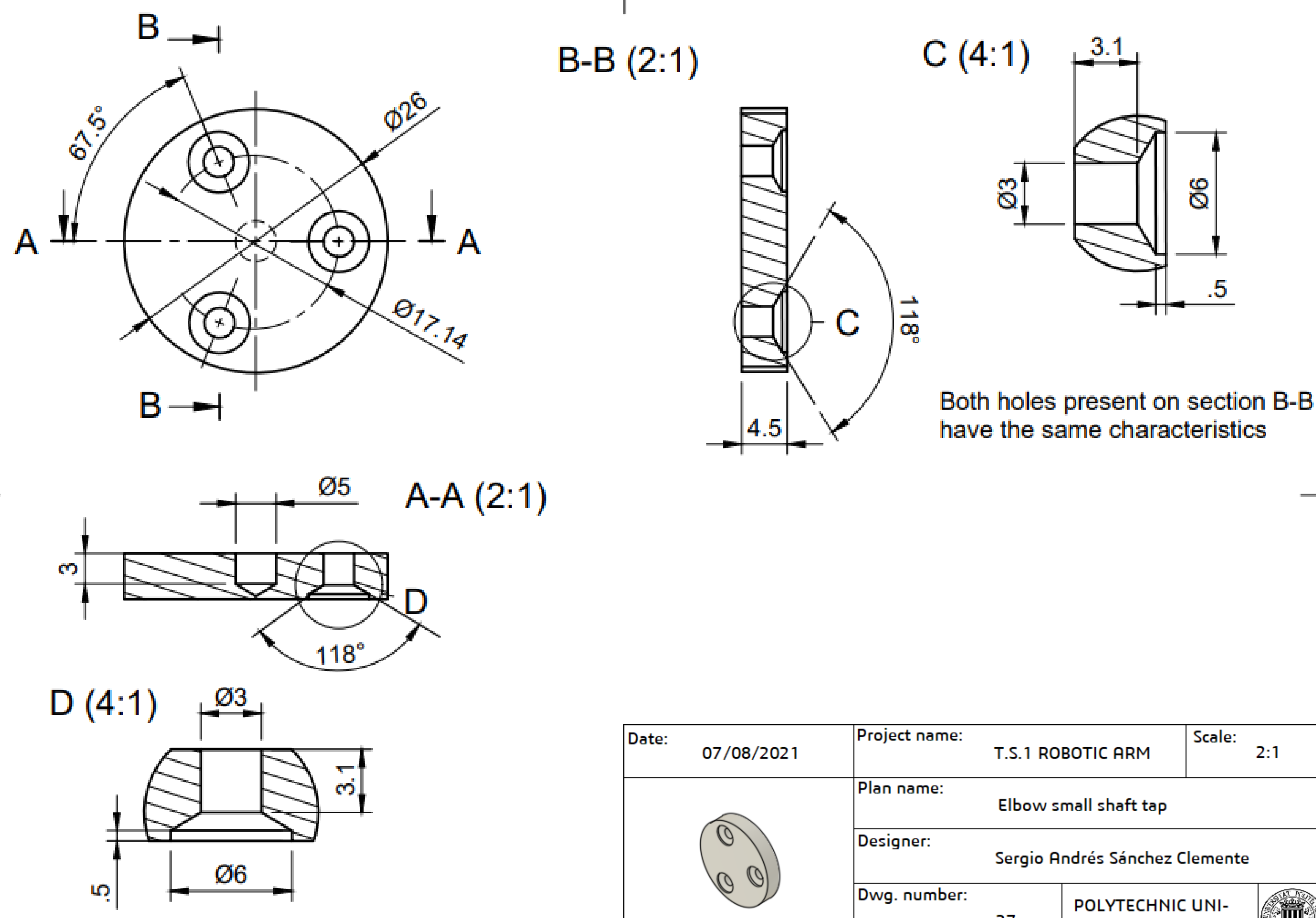


Date: 06/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 2:1
	Plan name: Wire rolling cylinder	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 25	POLYTECHNIC UNIVERSITY OF VALENCIA

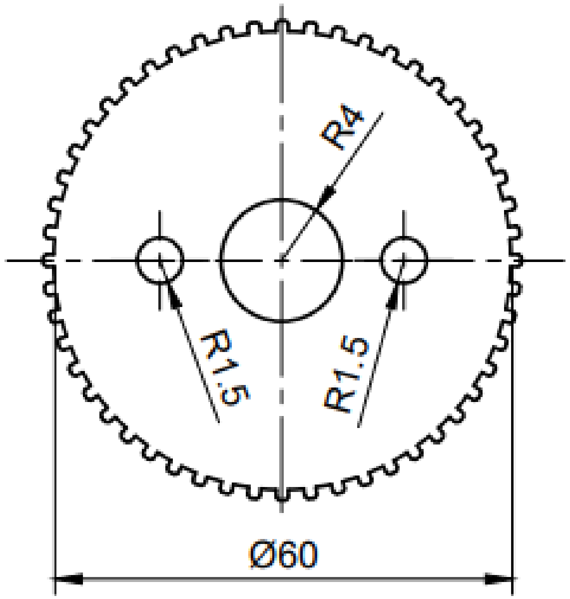


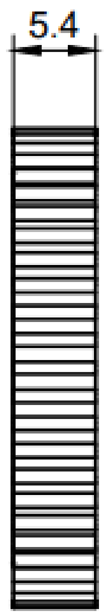
This piece has a mirror one, each one used at one of each side of the robot

Date: 08/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Forearm's cosmetic attachment		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 26	POLYTECHNIC UNIVERSITY OF VALENCIA	

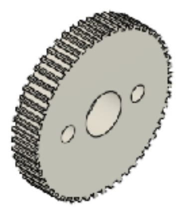


Date: 07/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Elbow small shaft tap		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 27	POLYTECHNIC UNIVERSITY OF VALENCIA	

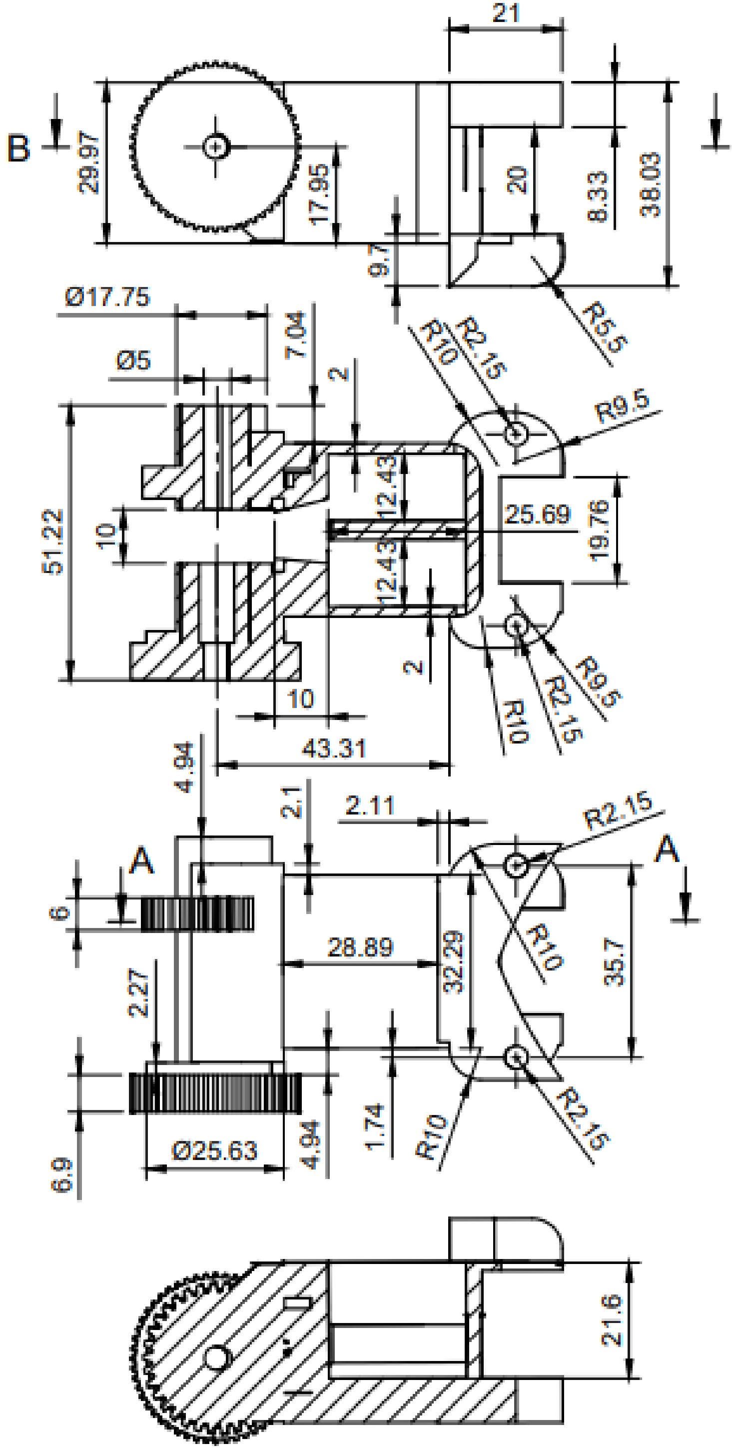




Gear description	
Pitch	2 mm (from belt)
Number of teeth	52
Pressure angle	14.5 degrees
Teeth height	1 mm

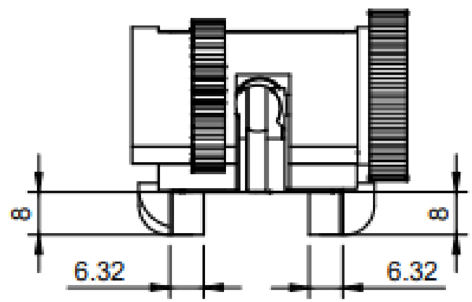


Date: 08/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 2:1
	Plan name: Wrist motor belt gear	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 28	POLYTECHNIC UNIVERSITY OF VALENCIA



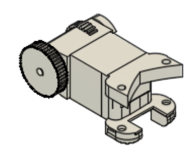
B-B (1:1)

A-A (1:1)

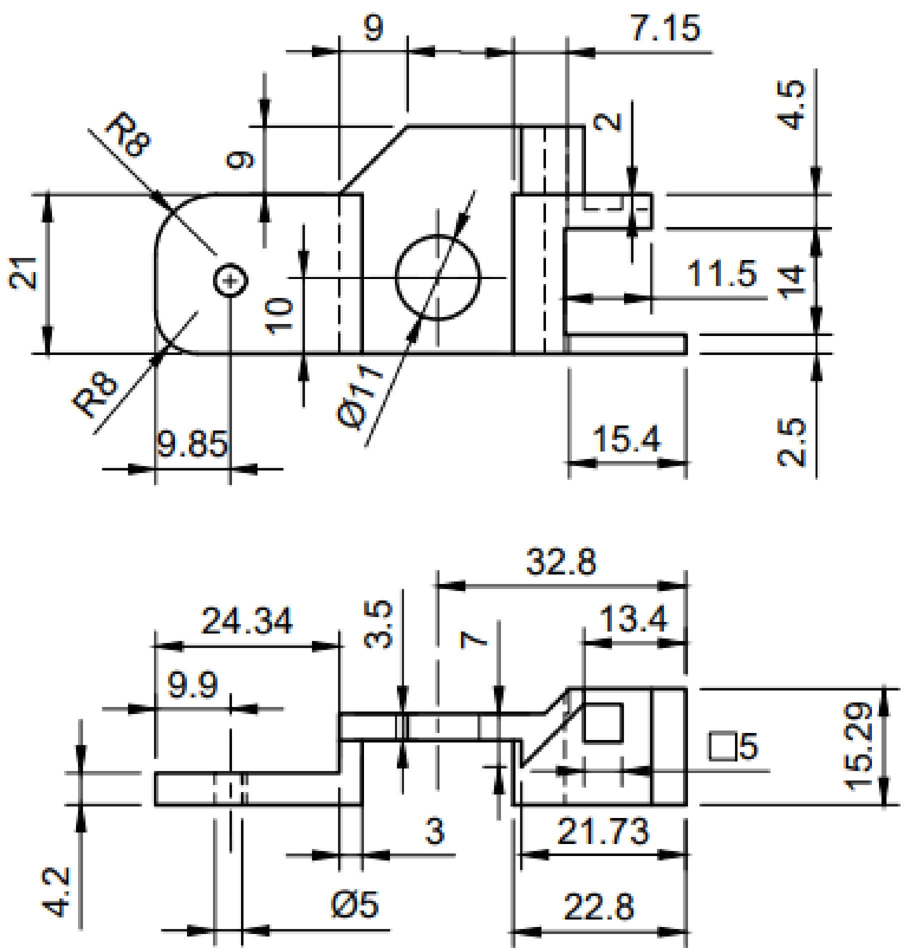


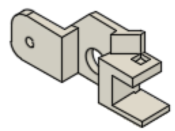
Belt gear description	
Pitch	2 mm (belt)
Number of teeth	52
Pressure angle	14.5 degrees
Teeth height	1 mm

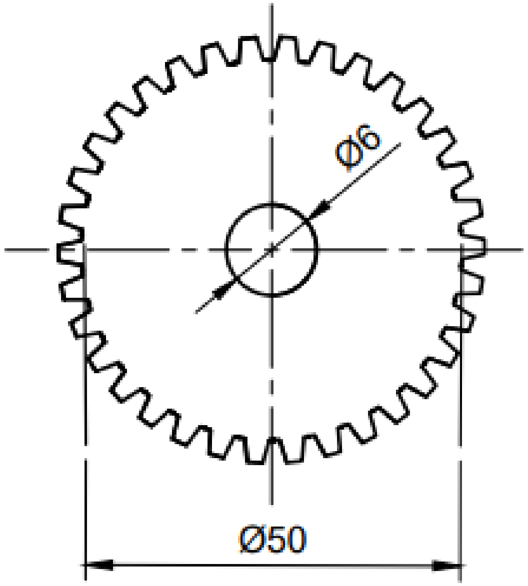
Wrist sensor gear description (section A)	
Module	0.7 mm
Number of teeth	34
Pressure angle	14.5 degrees
Teeth height	2 mm

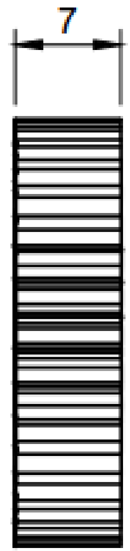


Date: 09/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 1:1
	Plan name: Wrist piece	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 29	POLYTECHNIC UNIVERSITY OF VALENCIA



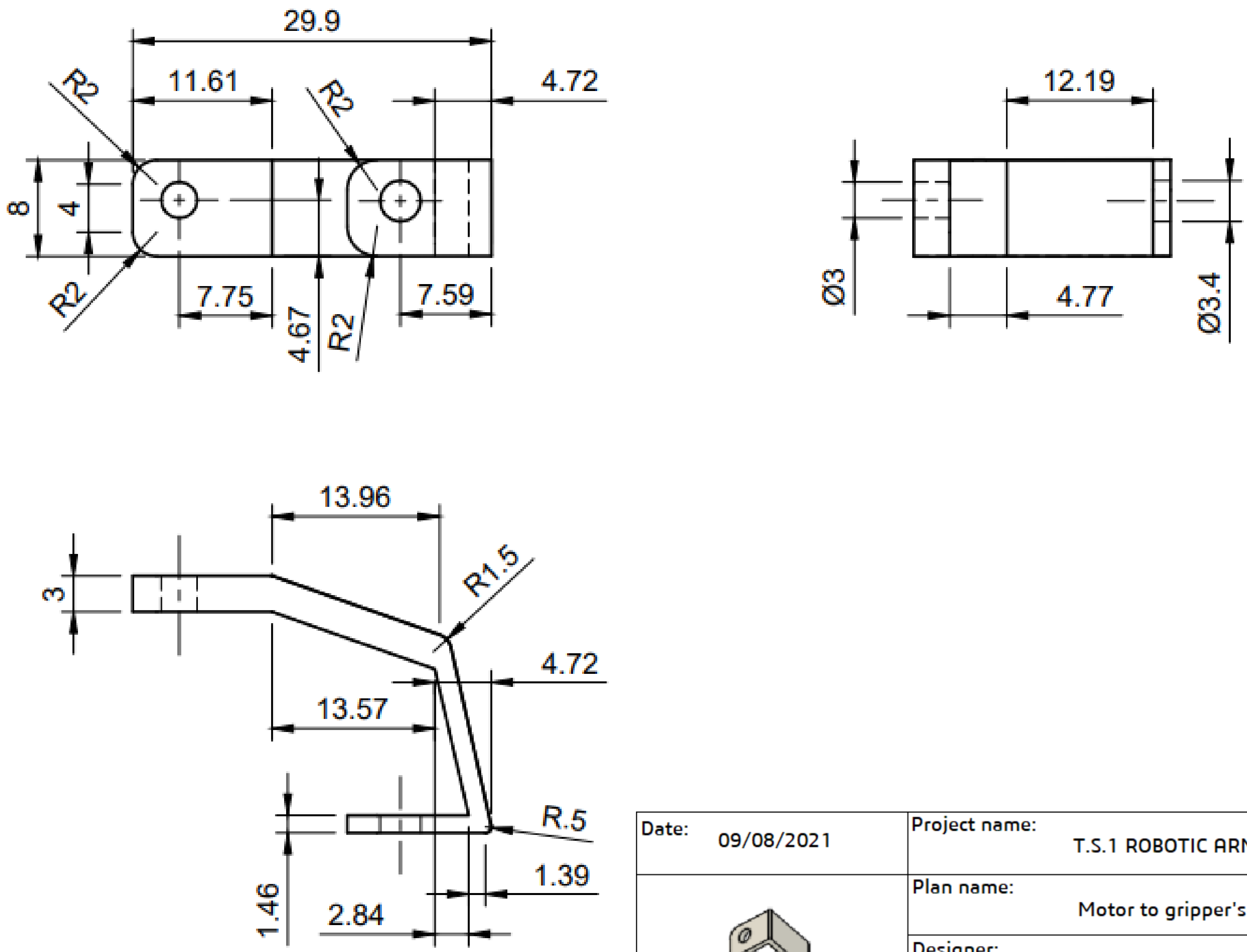
Date: 08/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Wrist potentiometer support		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 30	POLYTECHNIC UNIVERSITY OF VALENCIA	

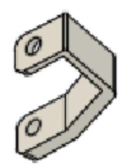


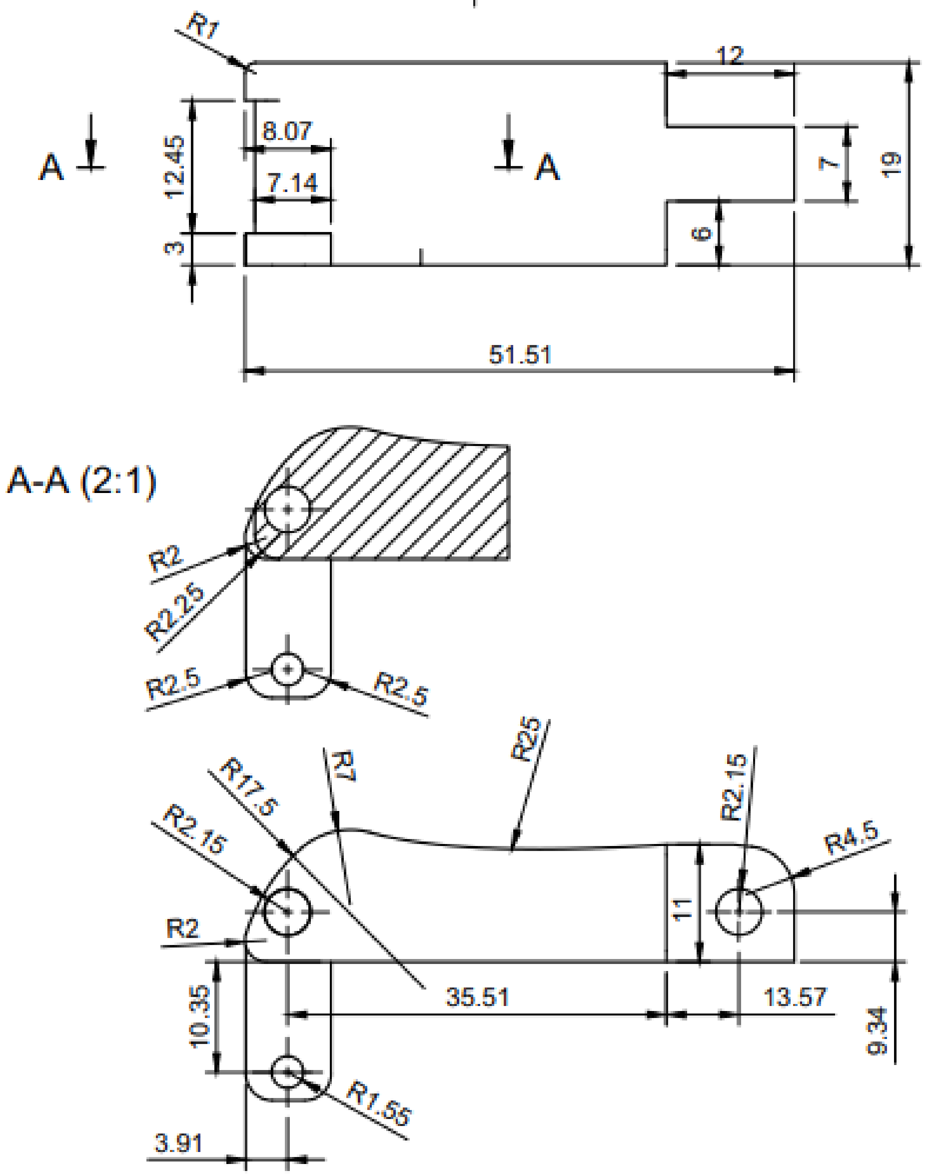


Gear description	
Module	0.7 mm (from belt)
Number of teeth	34
Pressure angle	14.5 degrees
Teeth height	2 mm

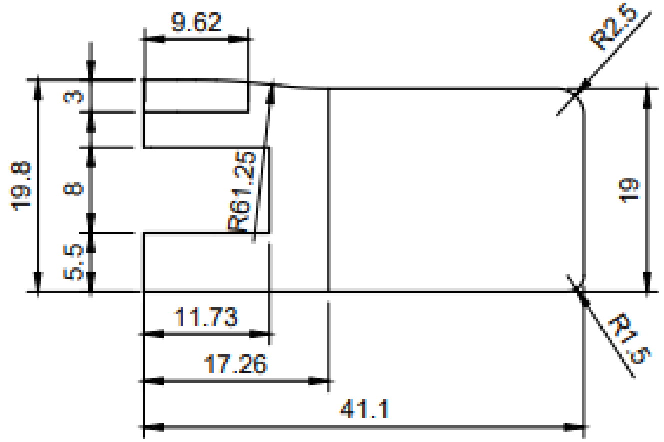
Date: 08/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 2:1
	Plan name: Wrist position sensor gear	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 31	POLYTECHNIC UNIVERSITY OF VALENCIA

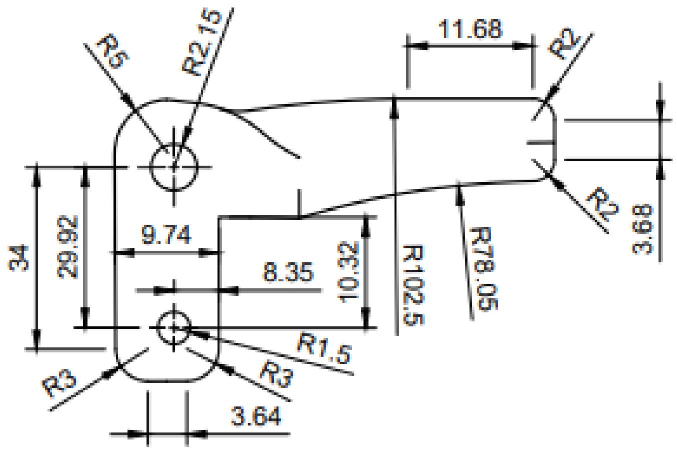


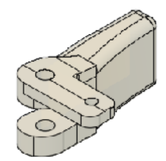
Date: 09/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Motor to gripper's link		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 32	POLYTECHNIC UNIVERSITY OF VALENCIA	

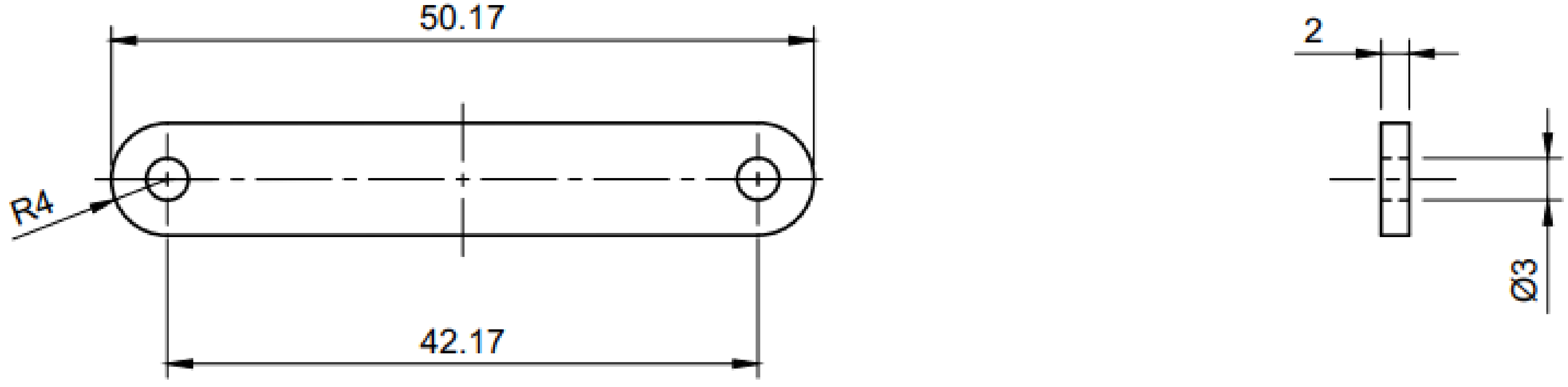


Date:	Project name:	Scale:
10/08/2021	T.S.1 ROBOTIC ARM	2:1
	Plan name: Gripper's upper link	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 33	POLYTECHNIC UNIVERSITY OF VALENCIA

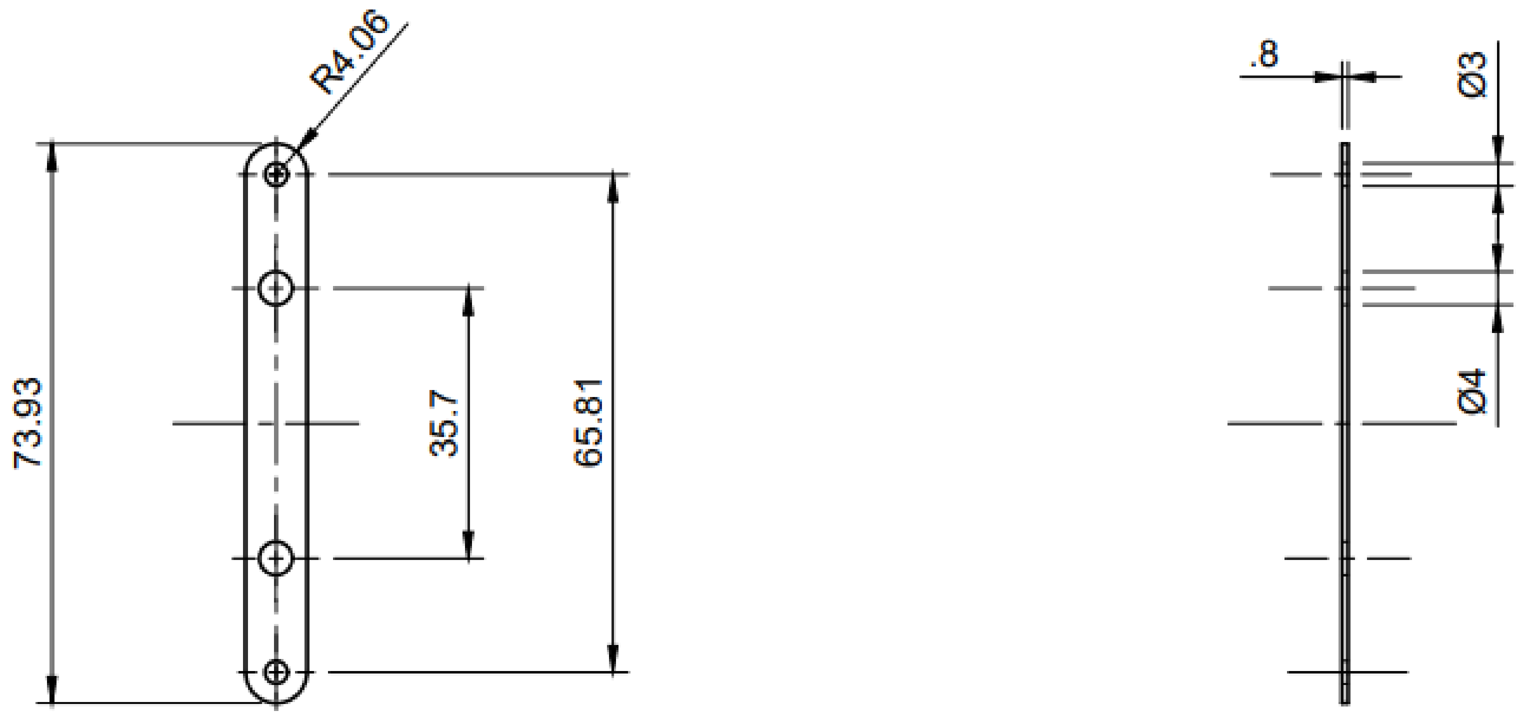




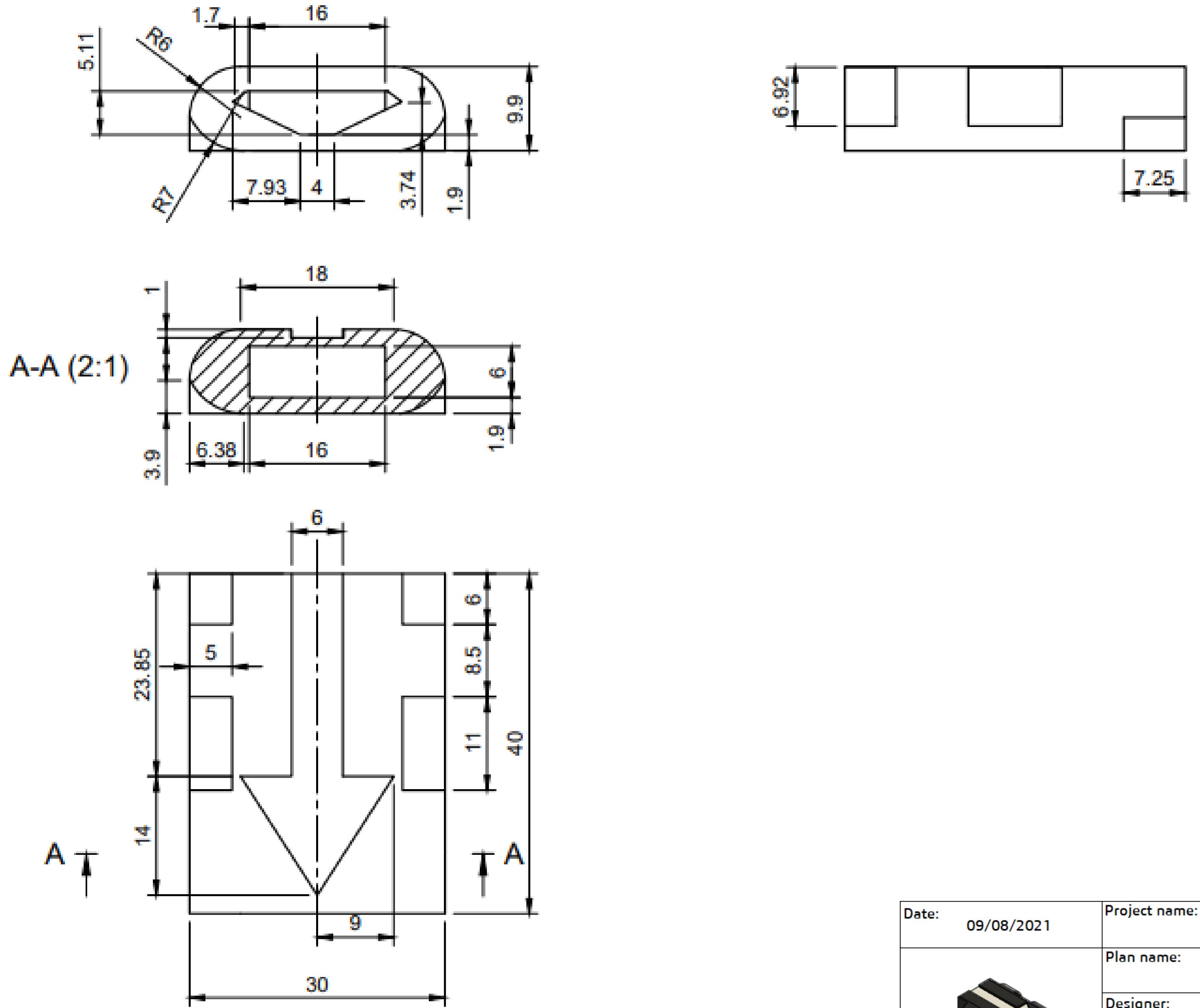
Date: 10/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 2:1
	Plan name: Gripper's lower link	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 34	POLYTECHNIC UNIVERSITY OF VALENCIA

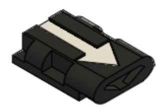


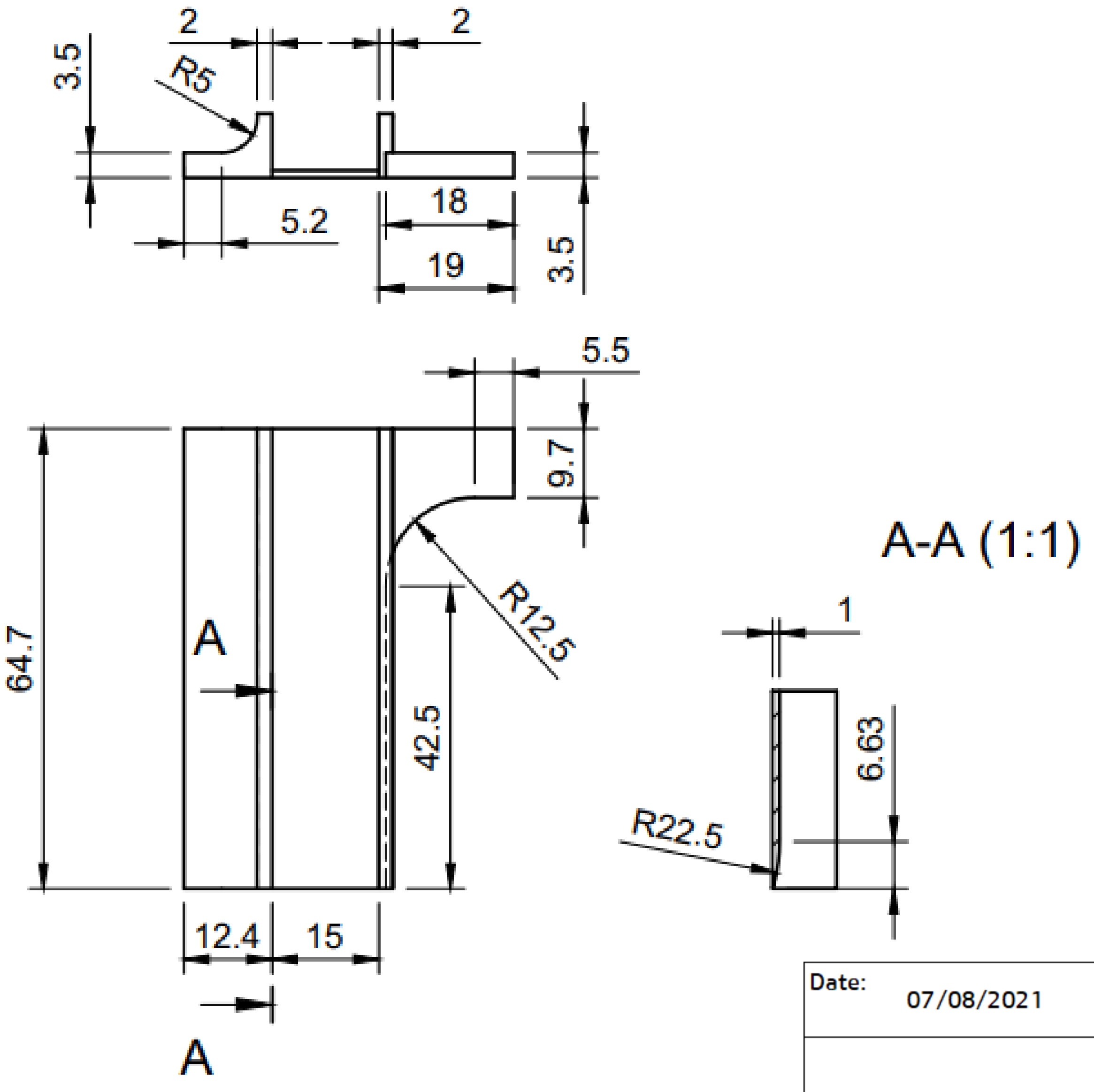
Date: 09/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Lateral link connector		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 35	POLYTECHNIC UNIVERSITY OF VALENCIA	



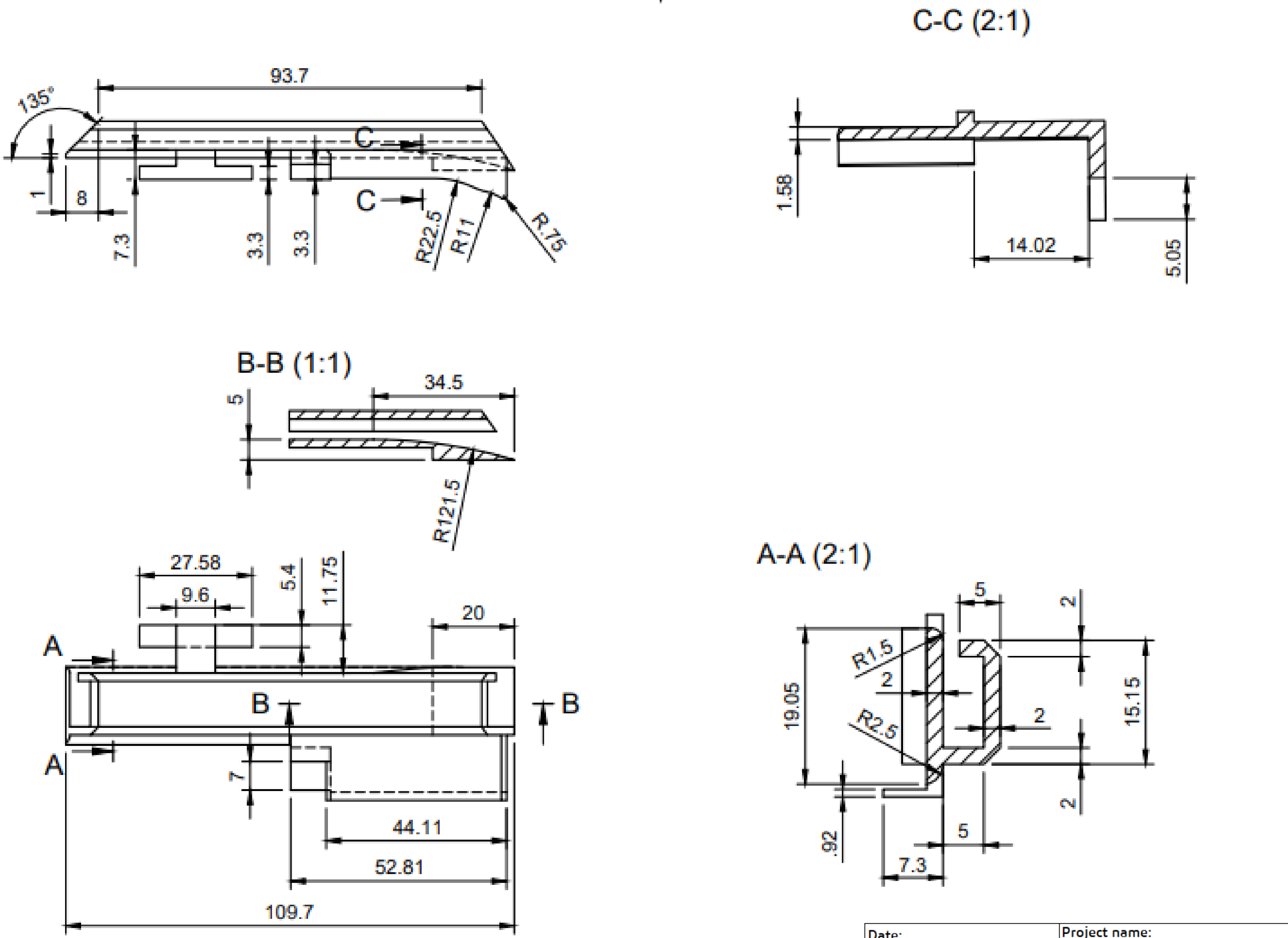
Date: 09/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Central gripper connector		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 36	POLYTECHNIC UNIVERSITY OF VALENCIA	

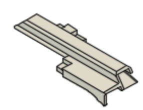


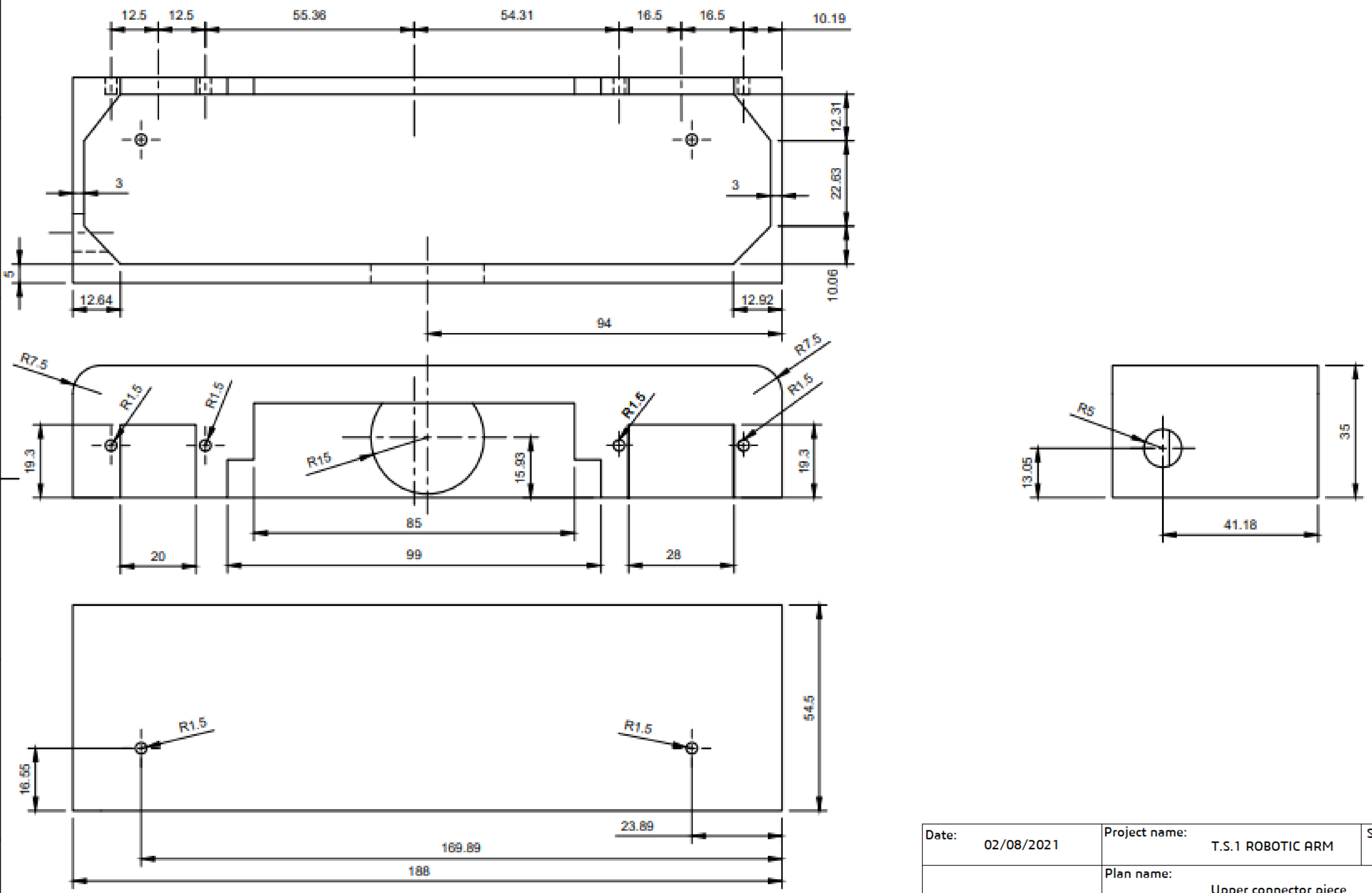
Date:	Project name:	Scale:
09/08/2021	T.S.1 ROBOTIC ARM	2:1
	Plan name: Frontal light case	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 37	POLYTECHNIC UNI-VERSITY OF VALENCIA



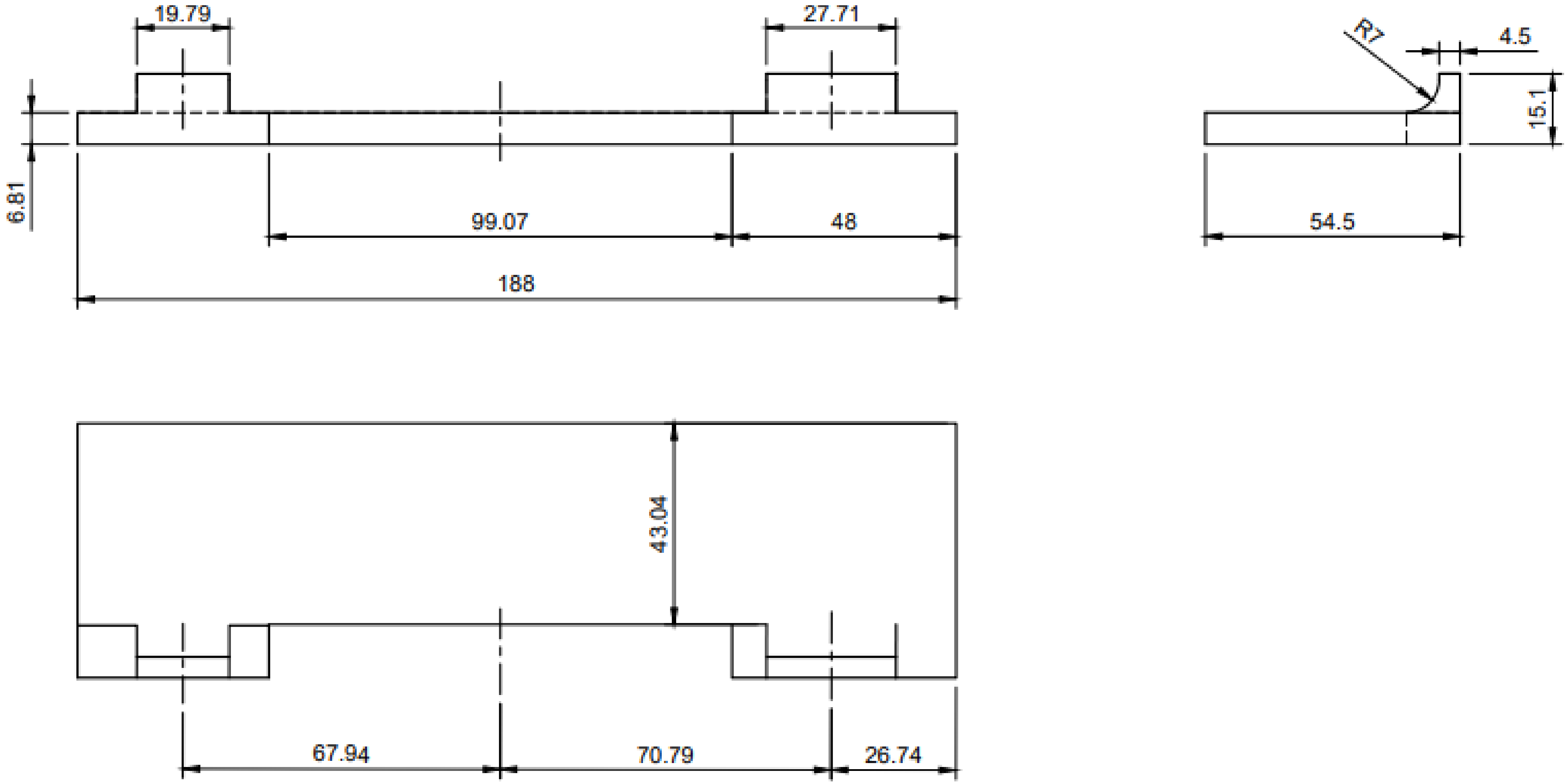
Date: 07/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Lower wire cover		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 38	POLYTECHNIC UNIVERSITY OF VALENCIA	



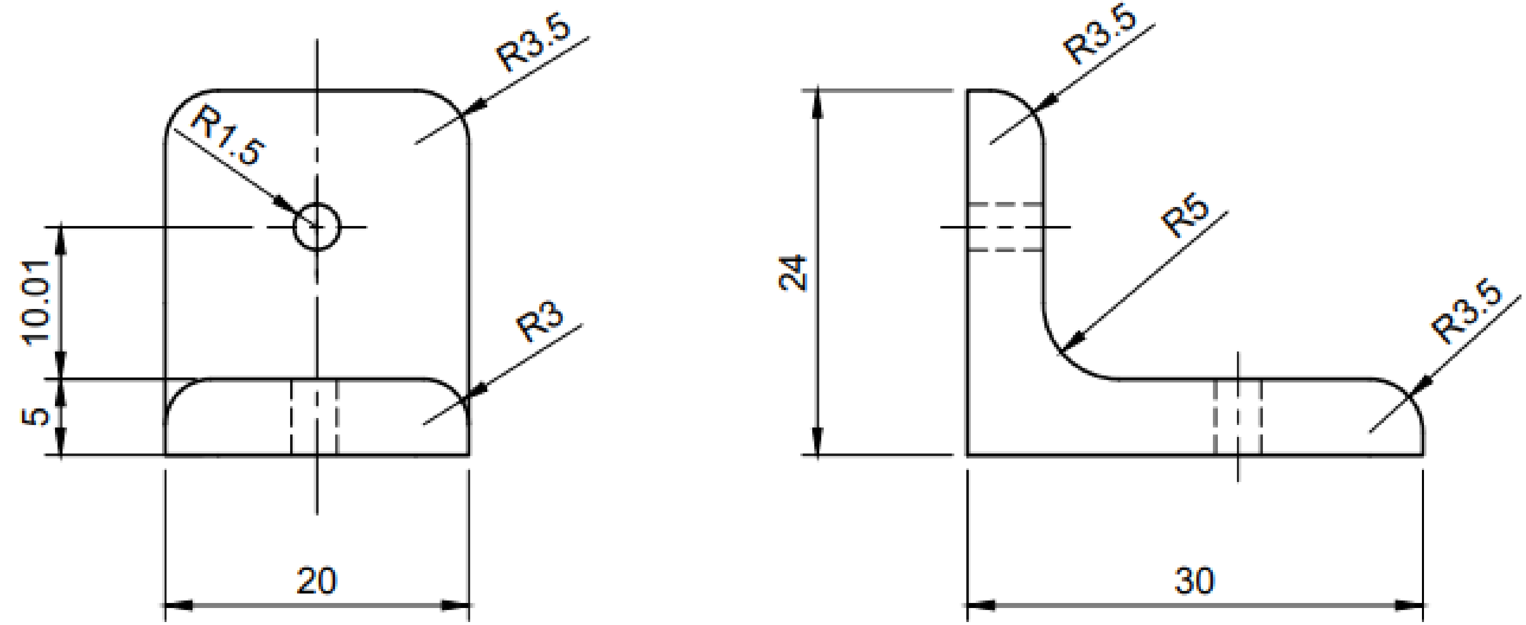
Date: 07/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 1:1
	Plan name: Upper wire cover	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 39	POLYTECHNIC UNIVERSITY OF VALENCIA

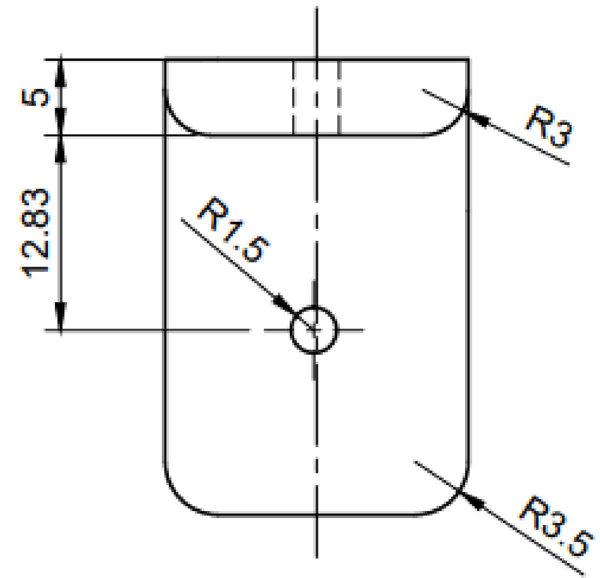


Date: 02/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 1:1
	Plan name: Upper connector piece	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 40	POLYTECHNIC UNIVERSITY OF VALENCIA

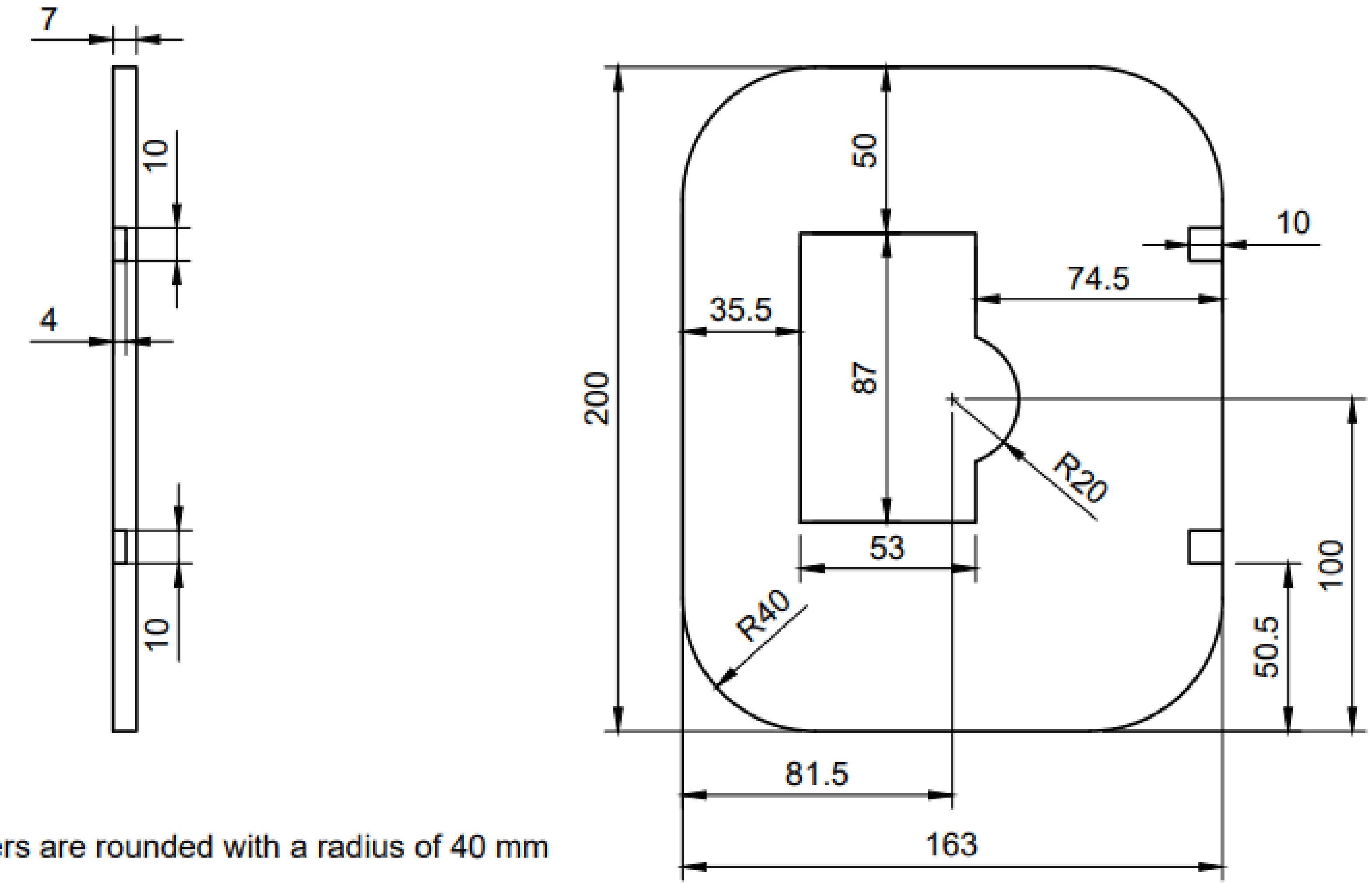


Date: 02/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Lower connector piece		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 41	POLYTECHNIC UNIVERSITY OF VALENCIA	

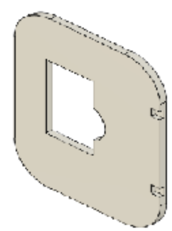




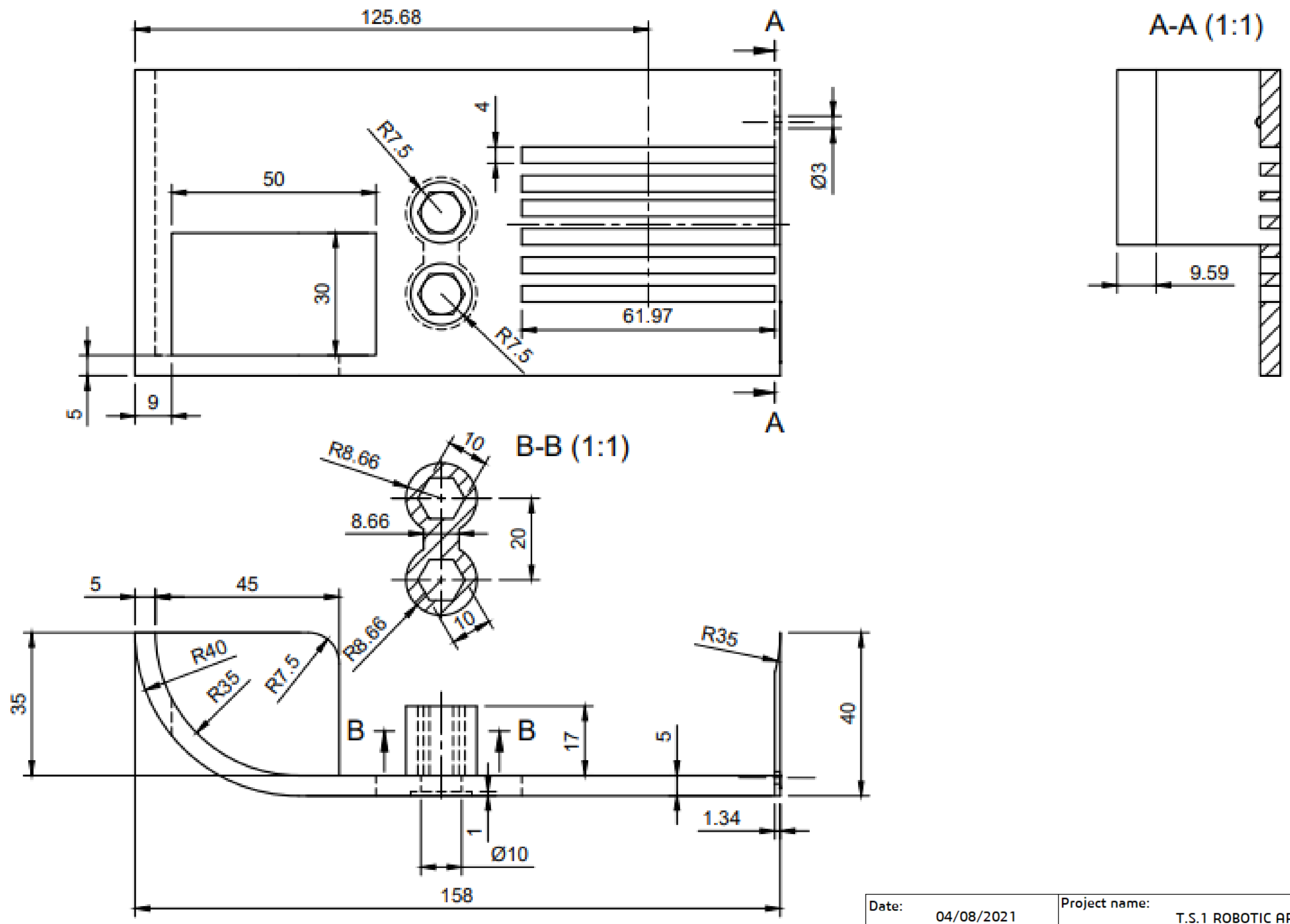
Date: 02/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 2:1
	Plan name: Connector attachment		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 42	POLYTECHNIC UNIVERSITY OF VALENCIA	

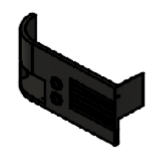


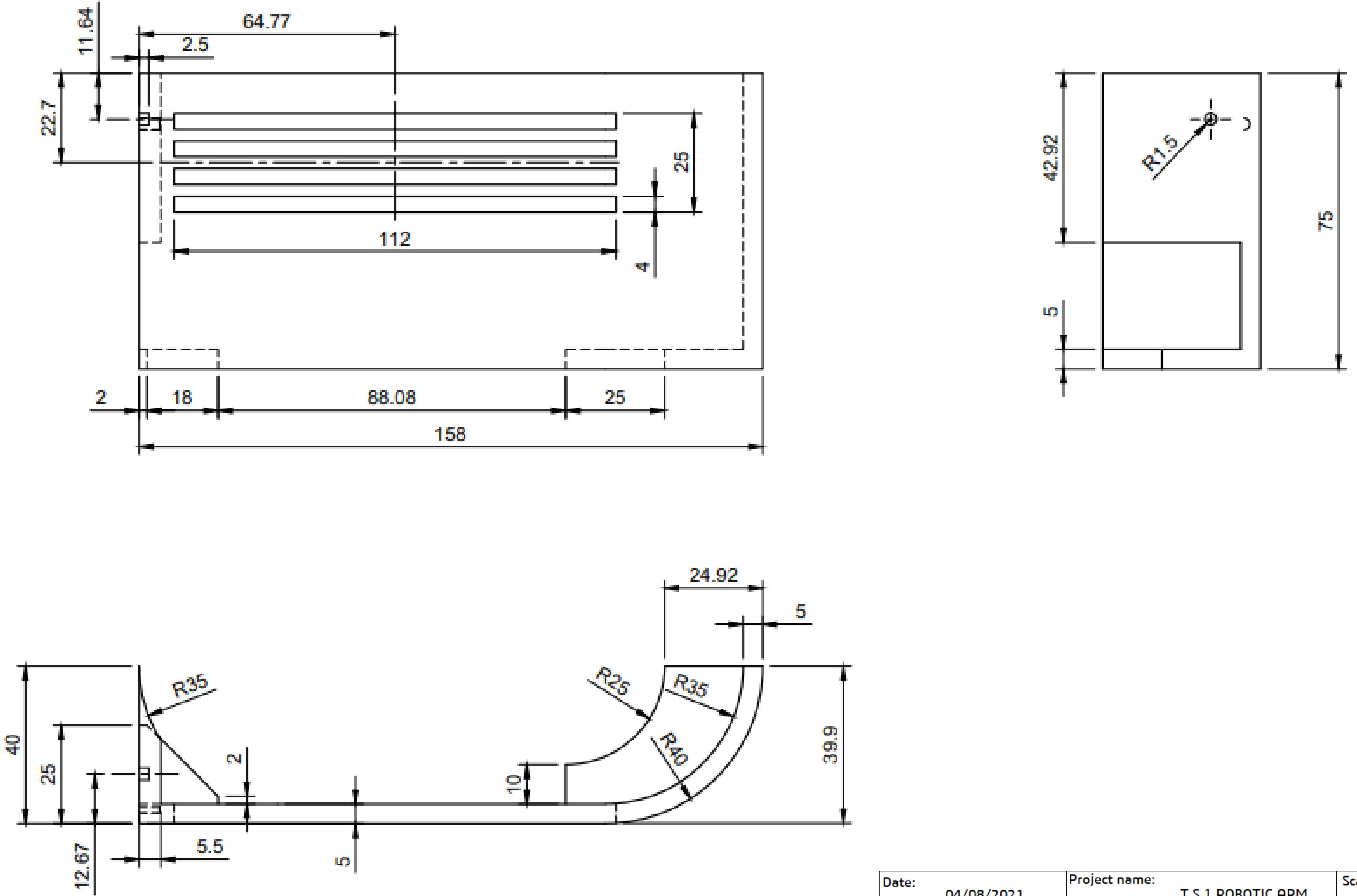
All exterior corners are rounded with a radius of 40 mm



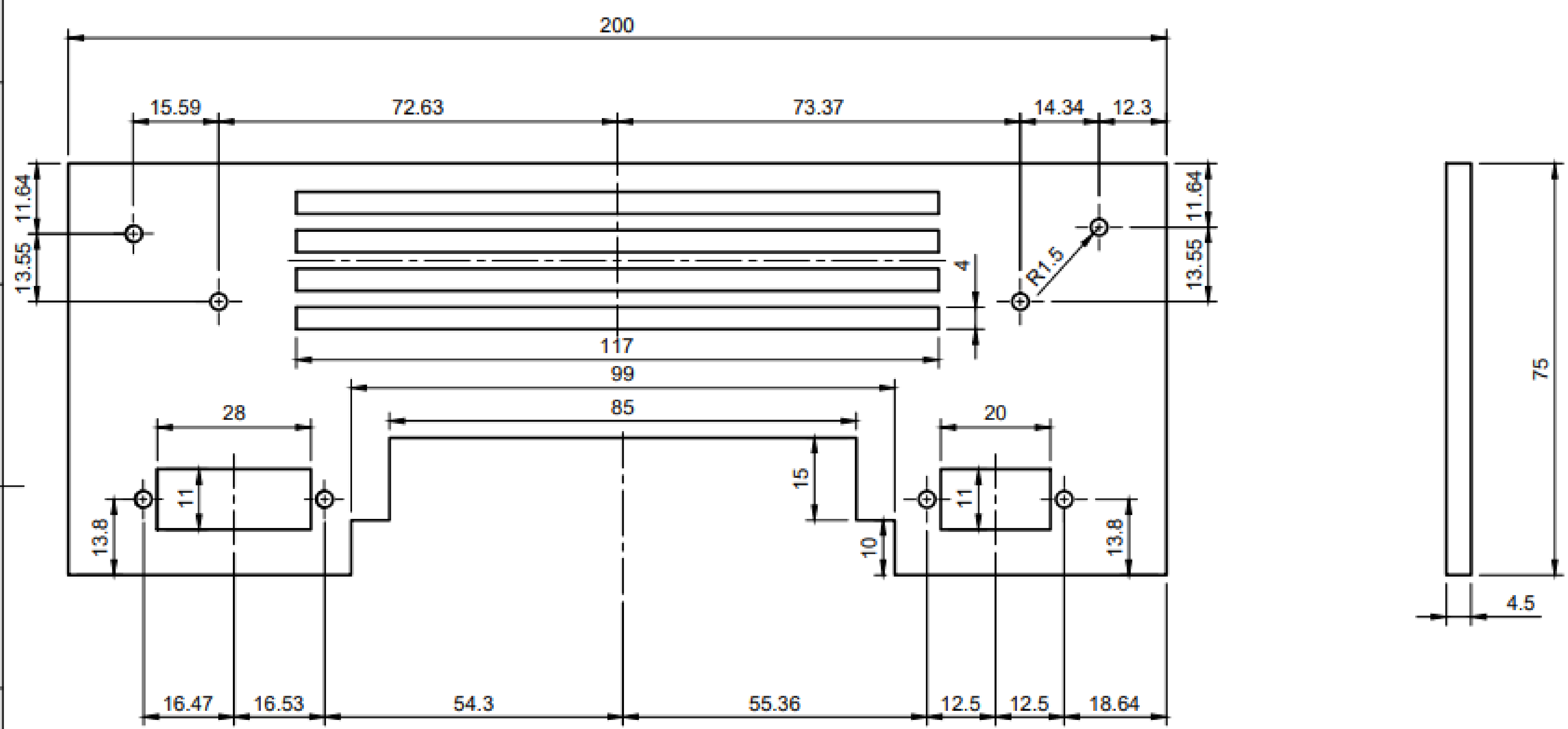
Date: 09/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:2
	Plan name: Electronics box base		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 43	POLYTECHNIC UNI-VERSITY OF VALENCIA	



Date: 04/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Electronics box left lateral panel		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 44	POLYTECHNIC UNIVERSITY OF VALENCIA	

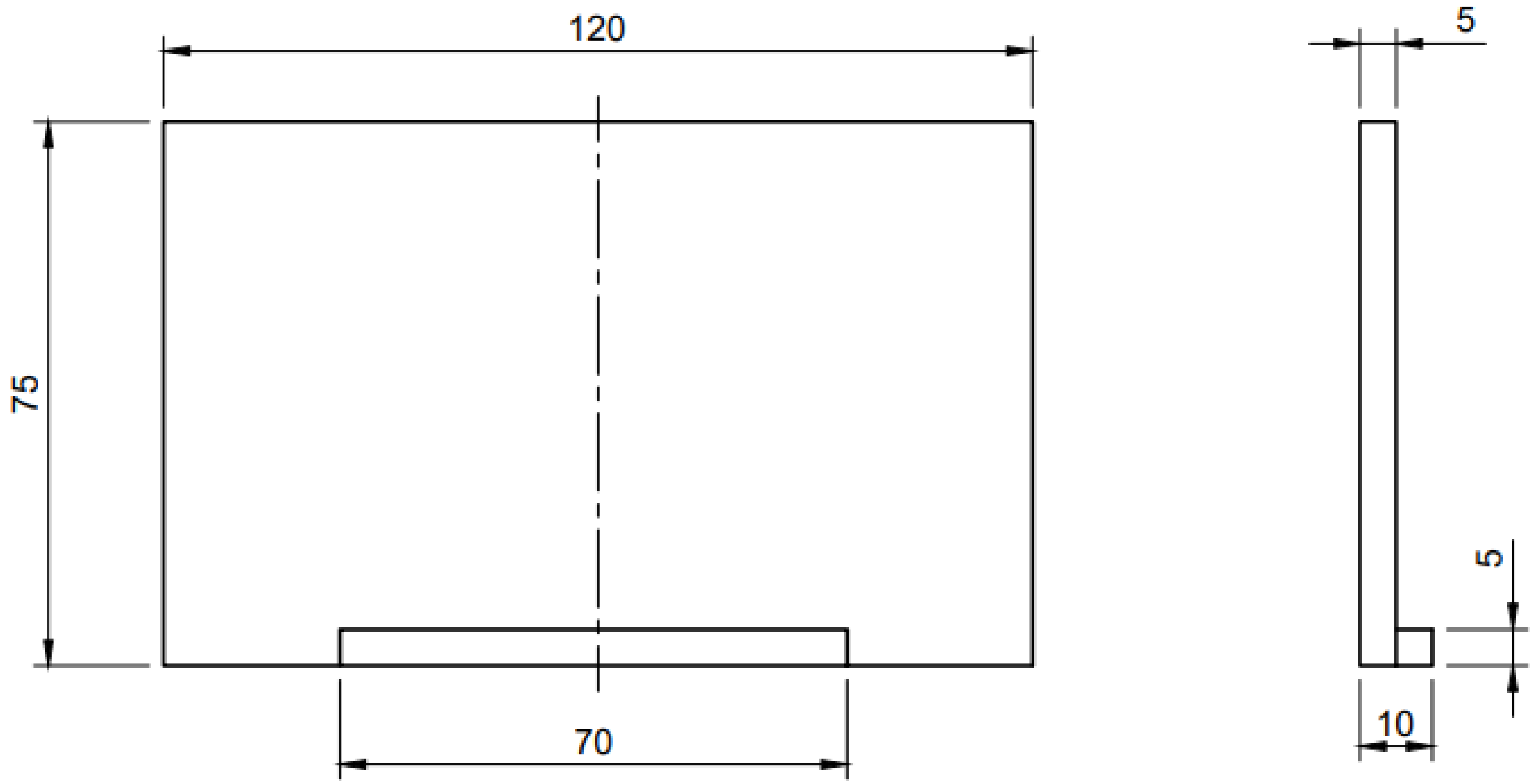


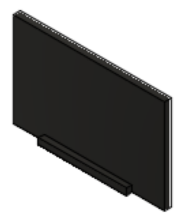
Date: 04/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Electronics box right lateral panel		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 45	POLYTECHNIC UNIVERSITY OF VALENCIA	



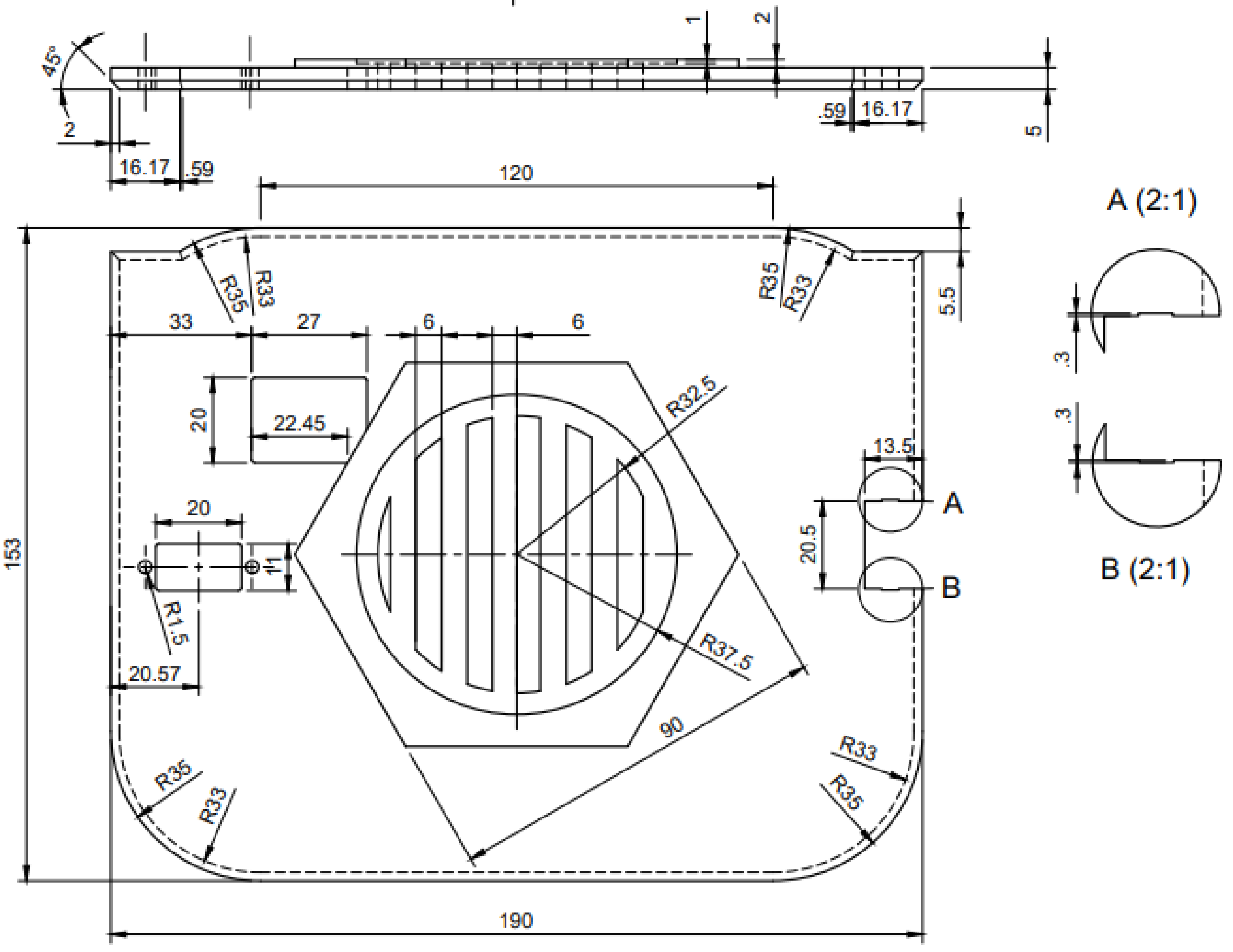
Only visible edges are displayed
All cavities and holes are passing
All circular holes on these views have a 1.5 mm radius

Date: 02/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Electronics box rear panel		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 46	POLYTECHNIC UNIVERSITY OF VALENCIA	

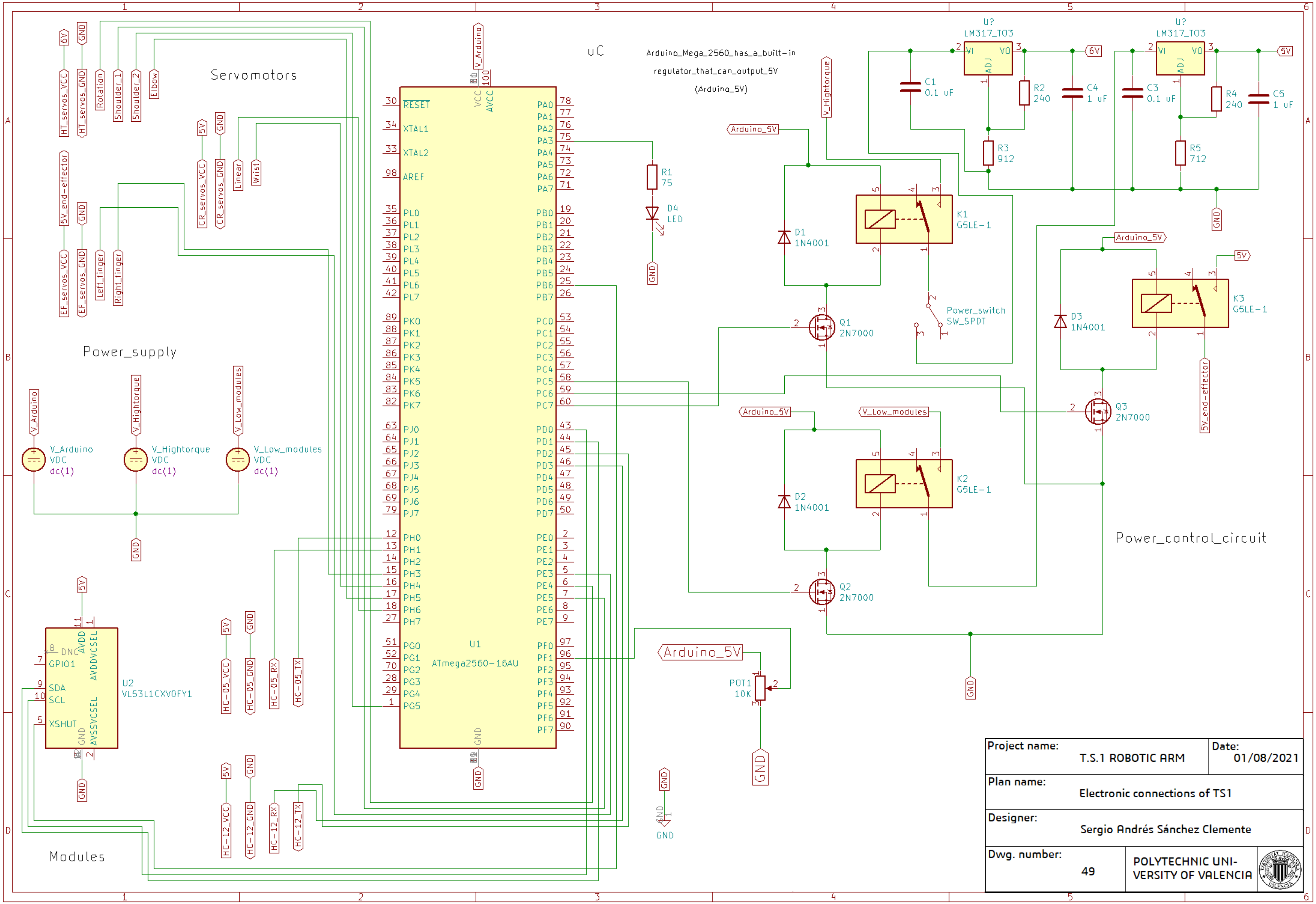




Date: 04/08/2021	Project name: T.S.1 ROBOTIC ARM		Scale: 1:1
	Plan name: Electronics box frontal panel		
	Designer: Sergio Andrés Sánchez Clemente		
	Dwg. number: 47	POLYTECHNIC UNIVERSITY OF VALENCIA	



Date: 04/08/2021	Project name: T.S.1 ROBOTIC ARM	Scale: 1:1
	Plan name: Electronics box cover panel	
	Designer: Sergio Andrés Sánchez Clemente	
	Dwg. number: 48	POLYTECHNIC UNIVERSITY OF VALENCIA


Servomotors
HT_servos_VCC
HT_servos_GND
6V
GND
Rotation
Shoulder_1
Shoulder_2
Elbow
CR_servos_VCC
CR_servos_GND
5V
Linear
Wrist
EF_servos_VCC
EF_servos_GND
5V_end-effector
Left_finger
Right_finger
Power_supply
V_Arduino
VDC
dc(1)
V_Hightorque
V_Low_modules
uC
Arduino_Mega_2560_has_a_built-in
regulator_that_can_output_5V
(Arduino_5V)
U1
ATmega2560-16AU
RESET
XTAL1
XTAL2
AREF
VCC
AVCC
R1
75
D4
LED
Modules
U2
VL53L1CXV0FY1
AVDD
AVDDVCSEL
DNC
GPIO1
SDA
SCL
XSHUT
AVSSVCSEL
HC-05_VCC
HC-05_GND
HC-05_RX
HC-05_TX
HC-12_VCC
HC-12_GND
HC-12_RX
HC-12_TX
POT1
10K
Arduino_5V
U?
LM317_TO3
VI
VO
ADJ
C1
0.1 uF
R2
240
C4
1 uF
R3
912
C3
0.1 uF
R4
240
C5
1 uF
R5
712
K1
G5LE-1
K2
G5LE-1
K3
G5LE-1
D1
1N4001
D2
1N4001
D3
1N4001
Q1
2N7000
Q2
2N7000
Q3
2N7000
Power_switch
SW_SPDT
Power_control_circuit
Project name:
T.S.1 ROBOTIC ARM
Date:
01/08/2021
Plan name:
Electronic connections of TS1
Designer:
Sergio Andrés Sánchez Clemente
Dwg. number:
49
POLYTECHNIC UNIVERSITY OF VALENCIA

3. WRITTEN SPECIFICATIONS

ROBOTIC PLATFORM SPECIFICATION

1. Scope

This technical specification refers to the device presented on the memory report (section 1. ("MEMORY REPORT")). It explains in detail what the materials to be used need to accomplish, what the general execution conditions are, what the final characteristics of the device shall be and what the tests to be conducted have to be before commissioning.

This technical specification does not develop any aspect related to the process of fabrication of the device, only to the conditions in which it must be performed.

2. Materials

2.1. Mechanical components

2.1.1. Structural components

Aluminum profiles shall be 20 x 20 mm in section (exterior dimensions).

Aluminum profiles shall have a wall thickness of 1.5 mm.

Aluminum profiles shall have a total length of 400 mm.

Aluminum square tubes shall have a section of 20 x 20 mm (exterior dimensions).

Ball bearings shall have an inner hole diameter of 5 mm.

Black plastic taps shall be 20 x 20 mm in section.

Brass square shall be 50 mm long, 50 mm wide and 15 mm tall.

PLA plastic filament shall have a diameter of 1.75 mm.

The plywood planche shall be 10 mm thick.

2.1.2. Screws, bolts, washers and nuts

M12 carriage bolts shall be cut at a length of 100 mm.

M6 carriage bolts shall be cut only at lengths of 40 mm and 50 mm.

M6 hexagon nuts shall be standard DIN934.

M6 flat washers shall be standard DIN9021.

M6 hexagon bolts with full thread shall be standard DIN933.

M6 hexagon bolts with full thread shall be cut at a length of 40 mm.

M6 Pozidriv bolts shall be 30 mm in length.

M4 truss head self-tapping screws type AB shall have a length of 12 mm.

M3 truss head self-tapping screws type AB shall only be 8 mm and 12 mm long.

M3 chipboard screws shall have a length of 10 mm.

M3 socket head shoulder screws shall be standard ISO7379.

M3 socket head shoulder screws shall have a diameter of 4 mm.

M3 flat washers shall be standard DIN9021.

M3 hexagon nuts shall be standard DIN9034.

M3 countersunk machine screws shall have a length of 10 mm.

Cylindrical shafts shall only be 4 mm and 5 mm in diameter.

2.1.3. Acrylic paint

Acrylic colors used to be used shall be black and yellow.

2.1.4. Mechanisms

Axial ball bearings shall have an inner hole diameter of 12 mm.

Belt pulley shall be model 5M GT2.

2.2. Electronic components

2.2.1. Connectors

Connector slots shall be types DE-9, DE15HD and DA-15.

All three connector types shall be needed in male and female format.

2.2.2. Power control circuit

DC relays shall be model G5LE-1 DC5 from *Omron Electronics*.

Diodes shall be model 1N4001 from *Diodes Incorporated*.

MOSFETS shall be model 2N7000 from *On Semiconductor.*

2.2.3. Power supply & µC

Each transformer shall be able to output 3 A of current at 12 V (36 W).

The power switch shall be model RA1113112R.

Voltage regulators shall have linear behavior.

Voltage regulators shall be model LM317.

The µC board to be used shall be an Arduino Mega 2560.

2.2.4. Distance measurement

The distance measurement shall be made using a VL53L0X module.

2.2.4. Wires

Highest-rated wire used shall have dual polarity.

Highest-rated wire used shall have a section of 0.5 mm^2 on each polarity.

Heat shrink sleeve to be used shall have 6.4 mm inner diameter.

Wire spiral cover shall be made of polypropylene.

Wire spiral cover shall have an inner diameter of 34 mm.

2.2.5. Wireless communications

The Bluetooth module to be used shall be model HC-05.

The radio module to be used shall be model HC-12.

2.2.6. Potentiometers

The potentiometer to be used shall have a linear behavior.

The potentiometer to be used shall have a top resistivity value of 10 KΩ.

2.2.7. Resistors

The only resistor to be used has a resistivity of 75 Ω.

The only resistor to be used has a power rate of ¼ W.

The only resistor to be used shall have a maximum tolerance of 10 %.

2.2.8. LED

The LED to be used shall be white when turned on.

The LED to be used shall have a voltage drop of 3.5 V.

2.2.9. Motors

High-torque position servomotors shall be model DS3225.

High-torque position servomotors shall have a rated torque at 7.4 V of at least 25 kg·cm.

Low-torque position servomotors shall be model SG90.

Continuous rotation servomotors shall be model DS04-NFC.

Continuous rotation servomotors shall have a rated torque at 5 V of at least 5 kg·cm.

3. General execution conditions

3.1. Mechanical assembly execution conditions

3.1.1. Mechanical component manufacturing

Each part of the robot that has not been bought and has been specifically designed shall be manufactured in PLA filament by means of a 3D printer model Creality Ender 3.

The only exception to previous statement is the part seen on *plan 10*, which shall be made of 10 mm thick plywood planche.

When welding metallic components between them, gas welding shall be the technique to be used.

Holes shall be made or ampliated by means of a drilling machine with different drill sizes to fit each necessity.

3.1.2. Mechanical component assembly

When fastening attachment pieces (bolts, etc.), manual screwdrivers shall be used.

When applying pressure to fit one piece inside another one with very similar dimensions, the most damaging tool allowed to be used shall be a rubber mallet.

3.1.3. Quality controls

Any manufactured part shall follow their corresponding dimensions from plans with a maximum tolerance of a ± 0.2 mm.

No looseness can be observed on any joint part of any assembly.

3.2. Electronics execution conditions

3.2.1. Electronic components assembly

Every end of soldered electrical connections shall be covered by an insulating heat shrink sleeve.

When soldering a wire to a header pin or socket (pin on their bottom side), the physical connection between those two components shall be covered by a heat shrink sleeve.

Every connection shall be made bearing in mind that only the strictly necessary amount of wire shall be used.

All discovered connections that a technician or a user may interfere with shall be covered by heat shrink sleeve.

No connection ends can be accessible for a user.

3.2.2. Quality controls

No positive polarity can be measured between any visible part of any wire or mechanical part and system's ground.

System's tension levels must be the ones from design with a ± 1 % deviation.

4. Finished device features

Finished device shall be able to lift up to 0.33 kg.

Finished device shall have an accuracy of ± 2 mm of maximum deviation.

Finished device shall have a repeatability with a ± 2 mm maximum deviation.

Overall weight of the finished device assembly (excluding the rail structure) shall not be greater than 4.3 kg.

5. Testing prior to commissioning

When loading the end-effector of the device with half the maximum mass it can hold (0.165 kg) at an elbow position of 0° respect to ground plane, a shoulder position of 90° respect to ground plane and an end-effector position of 0° respect to ground plane, temperature increase at the surface of any of the motors involved must never be superior to 2 degrees after 30 minutes.

Loading the device in the same way as explained in the last paragraph but with the maximum load for 2 minutes must result in a temperature increase never higher than 2 degrees.

A test sequence that shall alternate between two different positions over a millimetric paper plane must show a maximum accuracy deviation of ± 2 mm and a maximum repeatability deviation of ± 2 mm when reproduced 100 consecutive times.

The device shall be weighted on a weighing machine and show a weight of 4.3 kg with a 0.1 % maximum deviation.

4. BUDGET

4.1. Introduction

This section presents the budget that has been handled for the project according to raw materials' price, piece fabrication and treatment, and the different assemblies.

Each section piece shall be numbered with the first letter being representative of the joint to which each piece belongs ("A" corresponding to joint 1, "B" corresponding to joint 2, etc.), with electronics also having a specific letter. Letter "S" is reserved for sub-assemblies and letter "M" corresponds to assemblies".

4.2. Valuation tables

4.2.1. ELEMENTARY PRICES PRESENTATION

This subsection presents the table containing the price of the components used on the device, as well as the price of the different sections regarding workers' salary, cost of automatic manufacturing, etc. Components named starting with an "m" represent materials, whereas those starting with an "s" represent the cost of a procedure.

Ref.	Unit	Description	Price (€)
m1	**m**	20x20 mm aluminum profile	**4,35**
m2	**U.**	Black plastic tap for 20 mm profile	**0,86**
m3	**U.**	50x50x15 mm brass square	**0,80**
m4	**U.**	M6x50mm carriage bolts	**0,10**
m5	**U.**	M6 hexagon nuts (DIN934)	**0,07**
m6	**U.**	M6 flat washers (DIN9021)	**0,05**
m7	**U.**	M6x40 mm hexagon bolt full thread (DIN933)	**0,10**
m8	**U.**	M6x40 mm carriage bolts	**0,10**
m9	**U.**	M12x100 mm carriage bolts	**0,75**
m10	**U.**	M3x8 mm truss head self-tapping screw type AB	**0,08**
m11	**U.**	M3x12 mm truss head self-tapping screw type AB	**0,08**
m12	**U.**	M4x12 mm truss head self-tapping screw type AB	**0,08**
m13	**U.**	M3x10 mm chipboard screws	**0,02**
m14	**U.**	M3xΦ4 mm socket head shoulder screw (ISO7379)	**0,07**
m15	**U.**	M3 flat washers (DIN9021)	**0,05**

m16	**U.**	M3 hexagon nuts (DIN9034)	**0,07**
m17	**U.**	M3x10 mm countersunk machine screws	**0,09**
m18	**m**	Φ5 cylindrical shaft	**6,00**
m19	**U.**	M6x30 mm Pozidriv bolt	**0,12**
m20	**U.**	M12 hexagon nuts	**0,50**
m21	**U.**	Bosch rexroth T-slot M6 nuts	**0,79**
m22	**Kg**	PLA plastic	**17,99**
m23	**cm3**	Pattex Extreme Pro	**0,41**
m24	**U.**	Acrylic paint tube	**1,04**
m25	**U.**	DE-9 connector (male and female)	**1,60**
m26	**U.**	DE15HD connector (male and female)	**1,70**
m27	**U.**	DA-15 connector (male and female)	**1,80**
m28	**U.**	36W (12 V 3A) transformer	**5,63**
m29	**U.**	G5LE-1 DC5 relay from Omron Electronics	**1,32**
m30	**U.**	1N4001 diode from Diodes Incorporated	**0,18**
m31	**U.**	2N7000 MOSFET from On Semiconductor	**0,34**
m32	**U.**	DC jack female connector	**0,55**
m33	**U.**	RA1113112R power switch	**0,56**
m34	**U.**	LM317 linear voltage regulator	**3,21**
m35	**m**	2 x 0.5 mm2 plane electric wires	**0,45**
m36	**m**	TUOFENG 1007 22 AWG wires	**0,37**
m37	**U.**	HC-05 Bluetooth module	**4,99**
m38	**U.**	HC-12 Radio module	**10,86**
m39	**U.**	10 kohm linear potentiometer	**1,92**
m40	**U.**	VL53L0X laser distance sensor	**5,89**
m41	**U.**	Arduino Mega 2560	**9,74**
m42	**m**	6.4 mm BPX0064 heat shrink sleeve	**1,02**
m43	**U.**	Header pins	**0,08**
m44	**U.**	Header sockets	**0,08**
m45	**U.**	Solder boards	**3,04**
m46	**U.**	High-brightness white LED	**0,57**
m47	**U.**	SG90 position servomotors	**2,96**
m48	**U.**	Male and female pluggable 9-terminal blocks from Uxcell	**4,00**
m49	**m**	Polypropylene spiral 30-34 mm cable protector from Hellermann Tyton	**15,94**
m50	**m**	Soldering tin	**6,71**
m51	**m**	5M GT2 belt	**1,60**
m52	**U.**	5M GT2 belt pulley (20 teeth)	**1,82**
m53	**m**	20x20 mm aluminum square tube with 1,5 mm thick walls	**6,99**
m54	**U.**	DS3225 position servomotor	**20,99**
m55	**U.**	DS04-NFC continuous rotation servomotor	**5,99**
m56	**U.**	3D printer wheels	**1,20**

m57	**U.**	Axial ball bearing	**2,00**
m58	**U.**	Ball bearing	**2,00**
m59	**m2**	10 mm thick plywood planche	**93,00**
s1	**h**	3D printing with CREALITY ENDER 3	**0,04**
s2	**h**	Inspection & sandpaper application	**8,00**
s3	**h**	Piece cutting	**9,00**
s4	**h**	Piece drilling	**9,00**
s5	**h**	Piece welding	**9,00**
s6	**h**	Piece soldering	**8,50**
s7	**h**	Painting	**8,00**
s8	**h**	Assemble	**8,50**

Table 35. *Elementary prices table*

Source: Own

Table 35 represents a set of references that shall be used for the calculation of the price it costs to develop each of the parts of the robot and their several assemblies.

4.2.2. SECTION DEVELOPMENT PRESENTATION

The first step when calculating the price of each of the parts in the development of the robot is obtaining the price of manufacturing each of the pieces. This price has been calculated on a *Microsoft Excel* sheet by calculating the amount of material used and the working time employed in modifying it. The result for each of the pieces has been summarized in the following table:

Ref.	Unit	Description	Price (€)
A1	**U.**	**Aluminum structure fabrication based on aluminum profiles**	**27,73**
A2	**U.**	**Belt drive gear wheel manufacturing**	**1,29**
A3	**U.**	**Belt drive motor's bracing manufacturing**	**1,67**
A4	**U.**	**Pulley case manufacturing**	**1,02**
A5	**U.**	**Pulley adapting piece manufacturing**	**0,79**
A6	**U.**	**Shaft's stop piece manufacturing**	**0,50**
A7	**U.**	**5 mm wide cylindrical shaft for belt pulley rotation preparation**	**0,31**
A8	**U.**	**Robot's mobile base's wheel subjection piece manufacturing**	**4,00**
A9	**U.**	**Robot's mobile base's external panel (no sensor) manufacturing**	**2,71**
A10	**U.**	**Robot's mobile base's external panel (sensor) manufacturing**	**2,53**

A11	U.	Robot's mobile base's bottom piece manufacturing	7,22
A12	U.	Belt attachment piece manufacturing	0,83
A13	U.	Laser holding piece manufacturing	0,56
A14	U.	Laser reflecting piece manufacturing	0,24
A15	U.	5 mm wide cylindrical shaft for base's wheels preparation	1,22
B1	U.	Rotation motor holding piece manufacturing	0,23
B2	U.	Robot's shoulder piece manufacturing	13,92
B3	U.	Cylindrical plastic separator piece manufacturing	0,79
B4	U.	Rotation toothed-wheel manufacturing	0,43
C1	U.	Shoulder motors' casing piece manufacturing	3,58
C2	U.	Shoulder to elbow linkage piece manufacturing	4,22
C3	U.	Elbow stop piece manufacturing	0,24
C4	U.	5 mm wide cylindrical elbow's shaft preparation	0,72
D1	U.	Elbow motor's casing piece manufacturing	1,55
D2	U.	Forearm piece manufacturing	12,63
D3	U.	4-linkage mechanism's shaft-stopping piece manufacturing	0,24
D4	U.	5 mm wide cylindrical 4-linkage mechanism's shaft preparation	0,44
D5	U.	Adapting piece for elbow's servomotor manufacturing	0,18
D6	U.	Motor to rear 4-linkage shaft transmission link manufacturing	1,44
D7	U.	Cylinder roll for wires manufacturing	0,97
D8	U.	Forearm's cosmetic piece manufacturing	0,61
E1	U.	Belt drive gear wheel for joint 5 manufacturing	0,52
E2	U.	End-effector base piece manufacturing	1,51
E3	U.	Position sensor attachment manufacturing	0,42
E4	U.	Position sensor gear wheel manufacturing	0,39
E5	U.	Motor to gripper's upper link connecting piece manufacturing	0,16
E6	U.	Gripper's upper link piece manufacturing	0,26
E7	U.	Lateral link connector manufacturing	0,12
E8	U.	Gripper's lower link piece manufacturing	0,26
E9	U.	Central end-effector bonding piece manufacturing	0,06
E10	U.	Frontal light casing manufacturing	1,95
E11	U.	Lower wire covering piece manufacturing	0,24
E12	U.	Upper wire covering piece manufacturing	0,33
E13	U.	5 mm wide cylindrical wrist's shaft preparation	0,72
E14	U.	Wrist mechanism's shaft-stopping piece manufacturing	0,24
E15	U.	5 mm wide cylindrical lower finger's shaft preparation	0,20

E16	**U.**	**5 mm wide cylindrical upper finger's shaft preparation**	**0,13**
F1	**U.**	**Electronics box base manufacturing**	**2,95**
F2	**U.**	**Left lateral panel of the electronics box manufacturing**	**2,90**
F3	**U.**	**Right lateral panel of the electronics box manufacturing**	**3,26**
F4	**U.**	**Back panel of the electronics box manufacturing**	**2,83**
F5	**U.**	**Frontal panel of the electronics box manufacturing**	**1,81**
F6	**U.**	**Upper panel of the electronics box manufacturing**	**3,33**
F7	**U.**	**Base piece of the wire plug manufacturing**	**2,80**
F8	**U.**	**Upper piece of the wire plug manufacturing**	**5,45**
F9	**U.**	**Attachment of the wire plug manufacturing**	**0,96**

Table 36. *Each of the manufactured/prepared pieces*

Source: Own

With the materials presented on section 4.2.1. ("Section development presentation") and with the prepared parts from *table 36*, the following subassemblies have been made (considering preparation times of a batch of 100 units):

Ref.	Unit	Description	Price (€)	Quantity	Total (€)
S1	**U.**	**Aluminum structure, rails and belt transmission mechanism subassembly**			
		MATERIALS			
A1	**U.**	**Aluminum structure fabrication based on aluminum profiles**	**27,73**	1	27,73
A2	**U.**	**Belt drive gear wheel manufacturing**	**1,29**	1	1,29
A3	**U.**	**Belt drive motor's bracing manufacturing**	**1,67**	1	1,67
A4	**U.**	**Pulley case manufacturing**	**1,02**	1	1,02
A5	**U.**	**Pulley adapting piece manufacturing**	**0,79**	1	0,79
A6	**U.**	**Shaft's stop piece manufacturing**	**0,50**	1	0,50
A7	**U.**	**5 mm wide cylindrical shaft for belt pulley rotation preparation**	**0,31**	1	0,31
A14	**U.**	**Laser reflecting piece manufacturing**	**0,24**	1	0,24
m55	**U.**	DS04-NFC continuous rotation servomotor	**5,99**	1	5,99
m1	**m**	20x20 mm aluminum profile	**4,35**	1,60	6,96
m2	**U.**	Black plastic tap for 20 mm profile	**0,86**	12	10,28
m3	**U.**	50x50x15 mm brass square	**0,80**	2	1,60

m19	**U.**	M6x30 mm Pozidriv bolt	**0,12**	8	0,98
m6	**U.**	M6 flat washers (DIN9021)	**0,05**	8	0,40
m21	**U.**	Bosch rexroth T-slot M6 nuts	**0,79**	8	6,29
m10	**U.**	M3x8 mm truss head self-tapping screw type AB	**0,08**	10	0,80
m12	**U.**	M4x12 mm truss head self-tapping screw type AB	**0,08**	4	0,32
m5	**U.**	M6 hexagon nuts (DIN934)	**0,07**	2	0,14
m6	**U.**	M6 flat washers (DIN9021)	**0,05**	2	0,10
m7	**U.**	M6x40 mm hexagon bolt full thread (DIN933)	**0,10**	2	0,20
m51	**m**	5M GT2 belt	**1,60**	1,8	2,88
m52	**U.**	5M GT2 belt pulley (20 teeth)	**1,82**	1	1,82
SECTIONS					
s8	**h**	Assemble	**8,50**	1,20	10,20
				Total	**82,52**

S2	**U.**	**Robot's mobile base subassembly**			
MATERIALS					
A8	**U.**	**Robot's mobile base's wheel subjection piece manufacturing**	**4,00**	2	7,99
A9	**U.**	**Robot's mobile base's external panel (no sensor) manufacturing**	**2,71**	1	2,71
A10	**U.**	**Robot's mobile base's external panel (sensor) manufacturing**	**2,53**	1	2,53
A11	**U.**	**Robot's mobile base's bottom piece manufacturing**	**7,22**	1	7,22
A12	**U.**	**Belt attachment piece manufacturing**	**0,83**	1	0,83
A13	**U.**	**Laser holding piece manufacturing**	**0,56**	1	0,56
A15	**U.**	**5 mm wide cylindrical shaft for base's wheels preparation**	**1,22**	2	2,45
B1	**U.**	**Rotation motor holding piece manufacturing**	**0,23**	2	0,46
B4	**U.**	**Rotation toothed-wheel (the one attached to the motor) manufacturing**	**0,43**	1	0,43
m56	**U.**	3D printer wheels	**1,20**	4	4,80
m40	**U.**	VL53L0X laser distance sensor	**5,89**	1	5,89
m23	**cm3**	Pattex Extreme Pro	**0,41**	2,5	1,02
m8	**U.**	M6x40 mm carriage bolts	**0,10**	4	0,40

m5	**U.**	M6 hexagon nuts (DIN934)	**0,07**	4	0,28
m6	**U.**	M6 flat washers (DIN9021)	**0,05**	4	0,20
m12	**U.**	M4x12 mm truss head self-tapping screw type AB	**0,08**	8	0,64
m54	**U.**	DS3225 position servomotor	**20,99**	1	20,99
m13	**U.**	M3x10 mm chipboard screws	**0,02**	2	0,04
SECTIONS					
s8	**h**	Assemble	**8,50**	0,40	3,40
				Total	**62,84**

S3	**U.**	**Robot's forearm and arm subassembly**			
MATERIALS					
B2	**U.**	**Robot's shoulder piece manufacturing**	**13,92**	1	13,92
C1	**U.**	**Shoulder motors' casing piece manufacturing**	**3,58**	2	7,17
C2	**U.**	**Shoulder to elbow linkage piece manufacturing**	**4,22**	2	8,44
C3	**U.**	**Elbow stop piece manufacturing**	**0,24**	2	0,47
C4	**U.**	**5 mm wide cylindrical elbow's shaft preparation**	**0,72**	1	0,72
D1	**U.**	**Elbow motor's casing piece manufacturing**	**1,55**	1	1,55
D2	**U.**	**Forearm piece manufacturing**	**12,63**	1	12,63
D3	**U.**	**4-linkage mechanism's shaft-stopping piece manufacturing**	**0,24**	2	0,47
D4	**U.**	**5 mm wide cylindrical 4-linkage mechanism's shaft preparation**	**0,44**	1	0,44
D5	**U.**	**Adapting piece for elbow's servomotor manufacturing**	**0,18**	1	0,18
D6	**U.**	**Motor to rear 4-linkage mechanism's shaft transmission link manufacturing**	**1,44**	1	1,44
D7	**U.**	**Cylinder roll for wires manufacturing**	**0,97**	1	0,97
D8	**U.**	**Forearm's cosmetic piece manufacturing**	**0,61**	2	1,21
m9	**U.**	M12x100 mm carriage bolts	**0,75**	1	0,75
m39	**U.**	10 kohm linear potentiometer	**1,92**	1	1,92
m14	**U.**	M3xΦ4 mm socket head shoulder screw (ISO7379)	**0,07**	1	0,07
m58	**U.**	Ball bearing	**2,00**	3	6,00
m54	**U.**	DS3225 position servomotor	**20,99**	3	62,97
m23	**cm3**	Pattex Extreme Pro	**0,41**	3	1,23

m13	**U.**	M3x10 mm chipboard screws	**0,02**	28	0,56
m4	**U.**	M6x50mm carriage bolts	**0,10**	4	0,40
m5	**U.**	M6 hexagon nuts (DIN934)	**0,07**	6	0,42
m6	**U.**	M6 flat washers (DIN9021)	**0,05**	14	0,70
m8	**U.**	M6x40 mm carriage bolts	**0,10**	2	0,56
m16	**U.**	M3 hexagon nuts (DIN9034)	**0,07**	2	0,40
m17	**U.**	M3x10 mm countersunk machine screws	**0,09**	2	0,42
SECTIONS					
s8	**h**	Assemble	**8,50**	1,50	12,75
				Total	**138,78**

S4	**U.**	**Robot's end-effector section subassembly**			
MATERIALS					
E1	**U.**	**Belt drive gear wheel for joint 5 manufacturing**	**0,52**	1	0,52
E2	**U.**	**End-effector base piece manufacturing**	**1,51**	1	1,51
E3	**U.**	**Position sensor attachment manufacturing**	**0,42**	1	0,42
E4	**U.**	**Position sensor gear wheel manufacturing**	**0,39**	1	0,39
E5	**U.**	**Motor to gripper's upper link connecting piece manufacturing**	**0,16**	2	0,32
E6	**U.**	**Gripper's upper link piece manufacturing**	**0,26**	2	0,53
E7	**U.**	**Lateral link connector manufacturing**	**0,12**	2	0,24
E8	**U.**	**Gripper's lower link piece manufacturing**	**0,26**	2	0,53
E9	**U.**	**Central end-effector bonding piece manufacturing**	**0,06**	1	0,06
E10	**U.**	**Frontal light casing manufacturing**	**1,95**	1	1,95
E11	**U.**	**Lower wire covering piece manufacturing**	**0,24**	1	0,24
E12	**U.**	**Upper wire covering piece manufacturing**	**0,33**	1	0,33
E13	**U.**	**5 mm wide cylindrical wrist's shaft preparation**	**0,72**	1	0,72
E14	**U.**	**Wrist mechanism's shaft-stopping piece manufacturing**	**0,24**	2	0,47
E15	**U.**	**5 mm wide cylindrical lower finger's shaft preparation**	**0,20**	2	0,41
E16	**U.**	**5 mm wide cylindrical upper finger's shaft preparation**	**0,13**	2	0,26
m47	**U.**	SG90 position servomotors	**2,96**	2	5,92
m55	**U.**	DS04-NFC continuous rotation servomotor	**5,99**	1	5,99
m46	**U.**	High-brightness white LED	**0,57**	1	0,57
m51	**m**	5M GT2 belt	**1,60**	0,33	0,53
m23	**cm3**	Pattex Extreme Pro	**0,41**	3	1,23
m13	**U.**	M3x10 mm chipboard screws	**0,02**	14	0,28

m15	**U.**	M3 flat washers (DIN9021)	**0,05**	6	0,30
m16	**U.**	M3 hexagon nuts (DIN9034)	**0,07**	6	0,42
m17	**U.**	M3x10 mm countersunk machine screws	**0,09**	6	0,54
		SECTIONS			
s8	**h**	Assemble	**8,50**	0,75	6,38
				Total	**31,05**

S5	**U.**	**Electronics subassembly**			
		MATERIALS			
m25	**U.**	DE-9 connector (male and female)	**1,60**	1	1,60
m26	**U.**	DE15HD connector (male and female)	**1,70**	1	1,70
m27	**U.**	DA-15 connector (male and female)	**1,80**	1	1,80
m28	**U.**	36W (12 V 3A) transformer	**5,63**	3	16,89
m29	**U.**	G5LE-1 DC5 relay from Omron Electronics	**1,32**	3	3,96
m30	**U.**	1N4001 diode from Diodes Incorporated	**0,18**	3	0,54
m31	**U.**	2N7000 MOSFET from On Semiconductor	**0,34**	3	1,02
m32	**U.**	DC jack female connector	**0,55**	2	1,10
m33	**U.**	RA1113112R power switch	**0,56**	1	0,56
m34	**U.**	LM317 linear voltage regulator	**3,21**	2	6,42
m35	**m**	2 x 0.5 mm2 plane electric wires	**0,45**	20	9,00
m36	**m**	TUOFENG 1007 22 AWG wires	**0,37**	10	3,70
m37	**U.**	HC-05 Bluetooth module	**4,99**	**1**	4,99
m38	**U.**	HC-12 Radio module	**10,86**	1	10,86
m41	**U.**	Arduino Mega 2560	**9,74**	1	9,74
m42	**m**	6.4 mm BPX0064 heat shrink sleeve	**1,02**	3	3,07
m43	**U.**	Header pins	**0,08**	25	1,95
m44	**U.**	Header sockets	**0,08**	4	0,32
m45	**U.**	Solder boards	**3,04**	3	9,12
m46	**U.**	High-brightness white LED	**0,57**	1	0,57
m47	**U.**	SG90 position servomotors	**2,96**	2	5,92
m48	**U.**	Male and female pluggable 9-terminal blocks from Uxcell	**4,00**	5	20,00
m49	**m**	Polypropylene spiral 30-34 mm cable protector from Hellermann Tyton	**15,94**	1,2	19,13
m50	**m**	Soldering tin	**6,71**	0,5	3,36
		SECTIONS			
s6	**h**	Piece soldering	**8,50**	5	42,50
s8	**h**	Assemble	**8,50**	0,17	1,42
				Total	**181,23**

S6	**U.**	**Electronics box subassembly**			
		MATERIALS			
F1	**U.**	**Electronics box base manufacturing**	**2,95**	1	2,95
F2	**U.**	**Left lateral panel of the electronics box manufacturing**	**2,90**	1	2,90
F3	**U.**	**Right lateral panel of the electronics box manufacturing**	**3,26**	1	3,26
F4	**U.**	**Back panel of the electronics box manufacturing**	**2,83**	1	2,83
F5	**U.**	**Frontal panel of the electronics box manufacturing**	**1,81**	1	1,81
F6	**U.**	**Upper panel of the electronics box manufacturing**	**3,33**	1	3,33
F7	**U.**	**Base piece of the wire plug manufacturing**	**2,80**	1	2,80
F8	**U.**	**Upper piece of the wire plug manufacturing**	**5,45**	1	5,45
F9	**U.**	**Attachment of the wire plug manufacturing**	**0,96**	2	1,92
m23	**cm3**	Pattex Extreme Pro	**0,41**	1	0,41
m10	**U.**	M3x8 mm truss head self-tapping screw type AB	**0,08**	4	0,32
m11	**U.**	M3x12 mm truss head self-tapping screw type AB	**0,08**	2	0,16
m13	**U.**	M3x10 mm chipboard screws	**0,02**	10	0,20
		SECTIONS			
s8	**h**	Assemble	**8,50**	0,40	3,40
				Total	**31,73**

Table 37. *Section subassemblies*

Source: Own

Subassemblies presented on *table 37* are made in order (S1 to S6) and are interdependent. With these subassemblies and with the original materials and sections from *table 36*, greater assemblies are made. They can be seen on the following table:

Ref.	Unit	Description	Price (€)	Quantity	Total (€)

M1	U.	Robot arm with no rails or electronics assembly			
		MATERIALS			
S2	U.	Robot's mobile base subassembly	62,84	1	62,84
S3	U.	Robot's forearm and arm subassembly	138,78	1	138,78
S4	U.	Robot's end-effector section subassembly	31,05	1	31,05
S6	U.	Electronics box subassembly	31,73	1	31,73
B3	U.	Cylindrical plastic separator piece manufacturing	0,29	1	0,29
m20	U.	M12 hexagon nuts	0,50	2	1,00
m23	cm3	Pattex Extreme Pro	0,41	1	0,41
m57	U.	Axial ball bearing	2,00	1	2,00
m13	U.	M3x10 mm chipboard screws	0,02	5	0,10
		SECTIONS			
s8	h	Assemble	8,50	0,17	1,42
				Total	269,62

M2	U.	Robot arm with no rails assembly			
		MATERIALS			
M1	U.	Robot arm with no rails or electronics assembly	269,62	1	269,62
S5	U.	Electronics subassembly	181,23	1	181,23
m15	U.	M3 flat washers (DIN9021)	0,05	10	0,50
m16	U.	M3 hexagon nuts (DIN9034)	0,07	10	0,70
m17	U.	M3x10 mm countersunk machine screws	0,09	10	0,90
m13	U.	M3x10 mm chipboard screws	0,02	5	0,10
		SECTIONS			
s8	h	Assemble	8,50	0,25	2,13
				Total	455,17

M3	U.	Robot arm final assembly			
		MATERIALS			
M2	U.	Robot arm with no rails assembly	455,17	1	455,17
S1	U.	Aluminum structure, rails and belt transmission mechanism subassembly	82,52	1	82,52

m13	**U.**	M3x10 mm chipboard screws	**0,02**	2	0,04
SECTIONS					
s8	**h**	Assemble	**8,50**	0,25	2,13
				Total	**539,86**

Table 38. *Section assemblies*

Source: Own

These assemblies are, like the subassemblies, interdependent, and performed in order from M1 to M3, being this M3 the final price:

Ref.	**Unit**	**Description**	**Price (€)**	**Quantity**
M3	U.	Robot arm final assembly	539,86	1

Table 39. *Final prototype price (no VAT included)*

Source: Own

Ref.	**Unit**	**Description**	**Price (€)**	**Quantity**
M3	U.	Robot arm final assembly	**653,23**	1

Table 40. *Final prototype price (VAT included)*

Source: Own

The VAT has been included on *table 40* in order to state what the price of developing one of these devices would be such that, when taxes are paid, no benefit or loss is perceived when sold.

4.3. Other costs

4.3.1. INTRODUCTION

Besides the physical cost of manufacturing the device, there are a series of other costs implicit in developing and selling such. This section aims at presenting these costs that enable a correct economic study of the project that allows to observe its profitability.

Fixed costs of the project represent a given amount that has been spent once and that need to be returned to make the project profitable before starting to retrieve benefits. In this case, this number shall include engineering costs, software licenses, indirect costs and experimentation costs.

Variable costs depend directly on the number of products that are sold. In this project's scope, this shall include manufacturing costs (machine utilization since operators' salaries have been taken into account when developing the budget for the proper device and therefore are already included) and distribution costs.

4.3.2. DETERMINING FIXED COSTS

Four factors have been taken into account when calculating the value of the fixed costs of the project: engineering costs (that is, the value of the total expenses of having engineers designing, developing and testing the first prototype and having them attending possible problems that could appear once the product is being commercialized), software licenses (the price license of the CAD programs used during development of the prototype), indirect costs (considered here to be organization costs, being this specifically the salary of the administrative side, electricity payments, etc. and marketing costs) and experimentation costs (when developing the first prototype).

Engineering costs

In order to determine the total expense of having engineers designing a prototype of this kind and testing it (including experimentation in between the designing process), the average salary for an electromechanical engineer in Spain shall be taken into account: **2650 €** each month (according to *salaryexplorer.com*, considering a common 8h a day with 5 workdays a week, which results in a salary of **16.56 €/h**.

It shall be supposed that only 1 engineer has worked on this project until its completion.

Software licenses

To be able to determine the price for this section, the programs that have been used need to be defined, being those the Autodesk Fusion 360, the Microsoft Excel and the Microsoft Word.

Autodesk Fusion 360 costs 503 € per year. Given a workday of 8 h and 5 days per week, that makes a total of 1920 h a year (discounting vacation). Then, the price per hour in which this software can be useful is **0.26 €/h**.

Microsoft 365 Enterprise Premium (which includes Microsoft Word, Microsoft Excel and others among them) costs 16.90 € per user and per month. As there has only been one user working on this project, that shall be the price each month. The price per hour, given the previous labor conditions, is **0.11 €/h**.

The total licenses in price per hour sum an estimate of **0.37 €/h**.

Indirect costs

As this is an academic exercise, no organization costs are reported (no administration expenses). However, the use of electricity when performing the designing part must be taken into account.

The price of electricity in Spain is **0.10 €/Kwh** (30-08-2021, on *globalpetrolprices.com*). Given the use of a computer with a consumption of 220 W in the designing process, the resulting price is **0.02 €/h**.

Experimentation costs

This section refers to the price of manufacturing during the design process to test how different subsystems behave before considering them finished. It is considered to be a fixed price of **25 €**, considering failed 3D prints or discarded component options.

Summary

The total fixed costs of developing the project are presented on *table 41*, based on the number of hours worked and the prices expressed on previous paragraphs.

Unit	Description	Price (€)	Quantity	Total (€)
Fixed costs				
h	Engineering costs	16,56	800	13248
h	Software licenses	0,37	500	185
U.	Indirect costs	0,02	500	10
	Experimentation costs	25		25
				13468

Table 41. *Fixed costs of the project*

Source: Own

4.3.3. DETERMINING VARIABLE COSTS

These costs shall depend on the volume of sales and are given below as their price per unit manufactured and sold. They are formed by machine utilization price (regarding machine maintenance, discarding the price of 3D printing, which has already been included on the value of manufacturing each device) and distribution costs (each sale's distribution cost).

Machine utilization

Machines and their components are susceptible to breakdown, for which some money per batch shall be spent necessarily. This project has required little number of machines (4 in total), having small reparation expenses. For instance, a radial saw or a barrel can break almost at any time after use.

This price has been estimated considering 1 radial saw replacement every 50 units (**9.66 €**, **0.19 €/unit**), 1 solder machine change every 10000 units (**23 €**, **0.0023 €/unit**), 1 drill end replacement every 200 units (**1 €**, **0.005 €/unit**) and 1 gas replacement for the welding machine every 60 units (**15 €**, **0.25 €/unit**).

This all makes a total price per unit of approximately **0.45 €**.

Distribution costs

Distribution has not been done on the project. However, in the case that distribution had to be done, the price has been calculated according to how much it costs to send a package with *Correos*. Adding the total weight of robot (4.3 kg) to the weight of the structure (1.5 kg), the selected package is the 6 kg one being sent across Spain, which costs **11.67 €**.

Unit	Description	Price (€)
Variable costs		
U.	Machine utilization	**0,45**
U.	Distribution costs*	**11,67**
		12,12

Table 42. *Variable costs of the project*

Source: Own

NOTE: (*) This is the worst-case scenario.

4.4. Conclusions

The device costs, no VAT included, 539.86 €, which accomplishes limitations established inside 3-A1 on section 1.3.2., set at 550 €. However, it is not low enough to comply with the limitation established for the price with VAT included (650 €), being 3.23 € above the limit (653.23 €).

The overall price that the device costs to manufacture and assemble can be reduced by adopting certain techniques like different ways of manufacturing (molding for instance, although this would be practical for larger numbers, since it requires an initial investment that should be amortized later), choosing better manufacturers (higher quality-cost ratio), dealing with suppliers (for better prices and services), buying wholesale, etc.

Fixed and variable costs have been calculated and used on Annex II. ("Economic study") and a profitability analysis has been performed on that section.

This section has focused on the analysis of the device presented on the memory report, commercialization hypotheses have been made on Annex II. ("Economic study").

BIBLIOGRAPHY

- ✓ Cortés, F. R. (2021). *Robotica: Control de Robots Manipuladores*. Marcombo.
- ✓ Alcides Gutiérrez, J. (2010). *Transmisión por Engranajes - 358-maquinas*. 358-máquinas. https://sites.google.com/site/358maquinas/transmision-por-engranajes
- ✓ Elgendi, M., Picon, F., Magnenat-Thalmann, N., & Abbot, D. (2014). *Biomedical Engineering online*. Biomedical Engineering Online. https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-13-88
- ✓ Lynch, D. K., Marchuk, N., & Elwin, M. (2015). *Embedded Computing and Mechatronics with the Pic32 Microcontroller*. Newnes.
- ✓ Christ, R. D., & Wernli, R. L. (2018). *The ROV Manual*. Elsevier Gezondheidszorg.
- ✓ Khalil, W., & Dombre, E. (2004). *Modeling, Identification & Control of Robots* (Illustrated ed.). Elsevier Butterworth Heinemann.
- ✓ Asada, H., & Youcef-Toumi, K. (1987). *Direct-drive Robots*. Amsterdam University Press.
- ✓ Lab, M. (2018, 28 agosto). *How to use GPIO pins of SMT32F4 discovery board*. Microcontrollers Lab. https://microcontrollerslab.com/gpio-pins-of-smt32f4-discovery-board/
- ✓ *Módulo Bluetooth HC05*. (2021). Naylamp Mechatronics - Perú. https://naylampmechatronics.com/inalambrico/43-modulo-bluetooth-hc05.html
- ✓ *Módulo Bluetooth 4.0 BLE HM-10*. (2021). Naylamp Mechatronics - Perú. https://naylampmechatronics.com/inalambrico/133-modulo-bluetooth-40-ble-hm-10.html

- ✓ TRADING ECONOMICS. (2021). *Steel | 2016–2021 Data | 2022–2023 Forecast | Price | Quote | Chart | Historical*. https://tradingeconomics.com/commodity/steel
- ✓ Plastics, R. (2020, 10 junio). *How Much Does Plastic Injection Molding Cost*. Rex Plastics. https://rexplastics.com/plastic-injection-molds/how-much-do-plastic-injection-molds-cost
- ✓ Heim, H. P. (2015). *Specialized Injection Molding Techniques*. Elsevier Gezondheidszorg.
- ✓ *Standard Bolts & Nuts Sdn. Bhd. – We are committed to delivering excellence*. (2021). Standardbolts.Net. https://standardbolts.net/